



Version 8

For Windows: Vista, 7, 8, 10 and 11

<https://www.xailer.com>

User Guide

Xailer: User Guide

Copyright Xailer 2003-2023

All rights reserved. No parts of this work may be reproduced in any form or by any means - graphic, electronic, or mechanical, including photocopying, recording, taping, or information storage and retrieval systems - without the written permission of the publisher.

Products that are referred to in this document may be either trademarks and/or registered trademarks of the respective owners. The publisher and the author make no claim to these trademarks.

While every precaution has been taken in the preparation of this document, the publisher and the author assume no responsibility for errors or omissions, or for damages resulting from the use of information contained in this document or from the use of programs and source code that may accompany it. In no event shall the publisher and the author be liable for any loss of profit or any other commercial damage caused or alleged to have been caused directly or indirectly by this document.

Printed: septiembre 2023 in (wherever you are located)

Publisher

Xailer

Managing Editor

Ignacio Ortiz de Zúñiga

Technical Editors

Ignacio Ortiz de Zúñiga

José Giménez Salazar

José Lalín Ferreiro

Cover Designer

María Eugenia Parra

Team Coordinator

Ignacio Ortiz de Zúñiga

Production

Ignacio Ortiz de Zúñiga

José Lutrillo

Special thanks to:

Specially to José Lalín and Pedro Gil for their great collaboration on Xailer developing and its documentation, including the translation to Galician done by José Lalín.

Particularly to Francesc Selles for the French traduction, Timm Sodtabers for the German traduction, Rafael Carmona for the Catalan traduction, Aguiar Costa for the Portuguese traduction and Enrico Maria Giordano por the Italian traduction.

At last we would like thanks to all the people that have contributed to this document.

Table of Contents

Capítulo I Xailer Guide	2
1 Userlicense	2
2 Firsts Steps	6
Introduction	6
Running Xailer	7
Xailer Configuration	7
Creating a new project	9
The project manager	11
Creating a form	12
The forms editor	14
The objects inspector	15
The properties	15
The events	15
The source code editor	17
Compiling and running the project	18
The resource manager	19
A more complex sample	20
3 Classes Hierarchy	25
Object Oriented Programming	26
Previous knowledge	26
The classes structure	26
The properties	26
The components	29
The events	29
The forms	30
Symbology	31
TComponent	32
TComponent.Properties	32
TComponent:Cargo	32
TComponent:ICreated	32
TComponent:oParent	33
TComponent.Methods	33
TComponent:New	33
TComponent:Create	34
TComponent:Destroy	34
TComponent:End	35
TComponent:EventAssigned	35
TComponent:Free	35
TWinObject	36
TWinObject.Properties	36
TWinObject.Handle	36
TControl	37
TControl.Properties	37
TControl:HelpID	38
TControl:cBalloon	38
TControl:cText	39
TControl:cTooltip	39

TControl:cTooltipTitle.....	40
TControl:IAlignKeepSpace.....	40
TControl:IAlignShiftOnRotate.....	40
TControl:IEnabled.....	41
TControl:IFocused.....	41
TControl:IParentFont.....	41
TControl:IRedraw.....	42
TControl:IVisible.....	42
TControl:nAlign.....	42
TControl:nAlignMarginBottom.....	43
TControl:nAlignMarginLeft.....	43
TControl:nAlignMarginRight.....	43
TControl:nAlignMarginTop.....	44
TControl:nAlignWeight.....	44
TControl:nAlignWeightSibling.....	45
TControl:nAnchors.....	46
TControl:nBorderSides.....	47
TControl:nClientHeight.....	48
TControl:nClientLeft.....	48
TControl:nClientTop.....	48
TControl:nClientWidth.....	48
TControl:nClrPane.....	49
TControl:nClrText.....	49
TControl:nHeight.....	49
TControl:nLeft.....	49
TControl:nTooltipIcon.....	50
TControl:nTop.....	50
TControl:nWidth.....	50
TControl:oBrush.....	51
TControl:oCursor.....	51
TControl:oFont.....	51
TControl:oForm.....	51
TControl:oParent.....	52
TControl.Methods.....	52
TControl:BeginPaint.....	53
TControl:Create.....	53
TControl:Destroy.....	54
TControl:Disable.....	54
TControl:Enable.....	55
TControl:EndPaint.....	55
TControl:GetSiblingControl.....	56
TControl:GoNextControl.....	56
TControl:GoPrevControl.....	56
TControl:Hide.....	57
TControl:HideBalloon.....	57
TControl:HideTooltip.....	57
TControl:New.....	57
TControl:PostMsg.....	58
TControl:Redraw.....	58
TControl:Refresh.....	59
TControl:SendMsg.....	59
TControl:SetBounds.....	60
TControl:SetFocus.....	60
TControl:Show.....	60

TControl:ShowBalloon	61
TControl:ShowPopupMenu	61
TControl:ShowTooltip	62
TControl:Update	62
TControl.Events	63
TControl:OnBeforeCreate	63
TControl:OnBeginDrag	63
TControl:OnClick	64
TControl:OnContextMenu	64
TControl:OnCreate	65
TControl:OnDestroy	65
TControl:OnDestroyed	65
TControl:OnDispTooltip	66
TControl:OnDragOver	66
TControl:OnEndDrag	67
TControl:OnHide	67
TControl:OnLButtonDown	67
TControl:OnLButtonUp	68
TControl:OnMouseMove	68
TControl:OnPaint	69
TControl:OnPainted	69
TControl:OnRButtonDown	70
TControl:OnRButtonUp	70
TControl:OnShow	70
TStdControl	71
TStdControl:Properties	71
TStdControl:aDropFiles	71
TStdControl:cMessage	71
TStdControl:lBorder	72
TStdControl:lDragAcceptFiles	72
TStdControl:lTabStop	72
TStdControl:Methods	72
TStdControl:RequestState	73
TStdControl:Events	73
TStdControl:OnChar	73
TStdControl:OnCheckState	74
TStdControl:OnEnter	74
TStdControl:OnExit	74
TStdControl:OnDropFiles	75
TStdControl:OnKeyDown	75
TStdControl:OnKeyUp	76
TWinControl	76
TWinControl:Properties	77
TWinControl:aControls	77
TWinControl:nClrPaneEnd	77
TWinControl:nGradient	77
TWinControl:nBkGndMarginX	78
TWinControl:nBkGndMarginY	78
TWinControl:nBkGndMode	78
TWinControl:oActiveControl	78
TWinControl:oBkGnd	79
TWinControl:oMenu	79
TWinControl:Methods	79
TWinControl:GoFirstControl	79

TWinControl:GoNextControl	80
TWinControl:GoPrevControl	80
TWinControl:InsertControl	80
TWinControl:Redraw	80
TWinControl:RemoveControl	81
TWinControl:RequestState	81
TScrollingWinControl	81
TScrollingWinControl:Properties	81
TScrollingWinControl:IAutoScroll	82
TScrollingWinControl:IHideScrollBars	82
TScrollingWinControl:nScrollIncrement	82
TScrollingWinControl:nScrollShowDelay	82
TScrollingWinControl:nClientLeft	83
TScrollingWinControl:nClientTop	83
TScrollingWinControl:nVirtualHeight	83
TScrollingWinControl:nVirtualWidth	83
TScrollingControl:Events	84
TScrollingWinControl:OnScrollDown	84
TScrollingWinControl:OnScrollLeft	84
TScrollingWinControl:OnScrollPageDown	85
TScrollingWinControl:OnScrollPageLeft	85
TScrollingWinControl:OnScrollPageRight	85
TScrollingWinControl:OnScrollPageUp	86
TScrollingWinControl:OnScrollRight	86
TScrollingWinControl:OnScrollUp	86
TScrollingWinControl:OnThumbPosH	87
TScrollingWinControl:OnThumbPosV	87
TScrollingWinControl:OnThumbTrackH	87
TScrollingWinControl:OnThumbTrackV	87
TForm	88
TForm:Properties	88
TForm:aComponents	89
TForm:IActive	89
TForm:IAppWindow	90
TForm:ICentered	90
TForm:ICloseBox	90
TForm:IDirectShortCuts	90
TForm:IHelpIcon	91
TForm:IHideOnClose	91
TForm:IMaximizeBox	91
TForm:IMinimizeBox	92
TForm:IModal	92
TForm:IMsgAuto	92
TForm:IParentIcon	92
TForm:ISysMenu	93
TForm:nBorderStyle	93
TForm:nFormType	94
TForm:nMaxHeight	94
TForm:nMaxPosX	94
TForm:nMaxPosY	94
TForm:nMaxWidth	95
TForm:nMinHeight	95
TForm:nMinWidth	95
TForm:nModalResult	95

TForm:nShowMode	96
TForm:oActiveControl.....	96
TForm:oCancelButton	96
TForm:oDefaultButton	96
TForm:olcon	97
TForm:oMDIMenu.....	97
TForm:oMDIClient	97
TForm:oMsgBar.....	98
TForm:oPopup.....	98
TForm:oPrevWindow	98
TForm:XHtmlHelp	98
TForm:Methods.....	99
TForm:Close.....	99
TForm:Maximize.....	100
TForm:Minimize.....	100
TForm:Restore.....	100
TForm:RestoreState.....	100
TForm:SaveState.....	101
TForm:SetFocus.....	101
TForm:Show	101
TForm:ShowModal	101
TForm:Events.....	102
TForm:OnActivate.....	102
TForm:OnChangeFocus	102
TForm:OnClose.....	103
TForm:OnCopyData.....	103
TForm:OnCreate.....	103
TForm:OnDeactivate.....	104
TForm:OnHelp	104
TForm:OnHelpClick.....	104
TForm:OnInitialize	105
TForm:OnMenuSelect	105
TForm:OnMove.....	106
TForm:OnSize	106
4 Public Objects	107
TApplication	107
TApplication.Properties.....	108
TApplication:aArguments.....	109
TApplication:aForms	109
TApplication:cComments.....	109
TApplication:cCompany.....	110
TApplication:cCopyright.....	110
TApplication:cDescription.....	110
TApplication:cDirectory	110
TApplication:cFileName.....	111
TApplication.cTitle.....	111
TApplication:cTradeMarks.....	111
TApplication:cVersion.....	111
TApplication:IAutoScale	111
TApplication:IBusy	112
TApplication:IDpiAware.....	112
TApplication:ISingleInstance	112
TApplication:IUseReturn	113
TApplication:nBuild.....	113

TApplication:nRelease	113
TApplication:nScale	114
TApplication:nVersion	114
TApplication:nWinPlatform	114
TApplication:nWinVer	114
TApplication:oActiveControl	115
TApplication:oActiveForm	115
TApplication.oFont	115
TApplication:oHelp	115
TApplication.olcon	116
TApplication:oMainForm	116
TApplication.Methods	116
TApplication:CurrentDir	116
TApplication:CurrentDrive	117
TApplication:Initialize	117
TApplication:DecimalSeparator	117
TApplication:Run	117
TApplication:SetOverlayIcon	118
TApplication:SetProgress	118
TApplication:MessageBox	118
TApplication:Terminate	119
TApplication:Events	119
TApplication:OnHelp	120
TApplication:OnQueryEndSession	120
TAppData	121
TAppData.Methods	121
TAppData:AddData	121
TAppData:Destroy	122
TAppData:Initialize	122
TAppDataEvents	122
TAppData:OnClose	122
TPrinter	123
TPrinter:Properties	123
TPrinter:aPrinterNames	124
TPrinter:cJobTitle	124
TPrinter:cPort	124
TPrinter:cPrinterName	124
TPrinter:hDC	125
TPrinter:IAborted	125
TPrinter:ICollate	125
TPrinter:IPreview	125
TPrinter:IPreviewModal	125
TPrinter:IPrintWithNext	126
TPrinter:nColor	126
TPrinter:nCopies	126
TPrinter:nDefaultBin	126
TPrinter:nDuplex	127
TPrinter:nOrientation	127
TPrinter:nPageNumber	127
TPrinter:nPaperLength	128
TPrinter:nPaperSizeType	128
TPrinter:nPaperWidth	130
TPrinter:nPDFPrinter	130
TPrinter:nPrinterIndex	131

TPrinter:nPrintQuality.....	131
TPrinter:nPreviewShowMode.....	131
TPrinter:nStatus.....	131
TPrinter:oCanvas.....	132
TPrinter:oExportInfo.....	132
TPrinter:oPageSetupDlg.....	134
TPrinter:oPreviewDC.....	134
TPrinter:oPrnDlg.....	134
TPrinter:oWndOwner.....	134
TPrinter:Methods.....	135
TPrinter:AbortDoc.....	136
TPrinter:Attributes.....	136
TPrinter:CanPrintToPdf.....	136
TPrinter:Comment.....	136
TPrinter:CountBinSources.....	137
TPrinter:CountPapers.....	137
TPrinter:Create.....	137
TPrinter:Destroy.....	137
TPrinter:DocumentSetup.....	138
TPrinter:DriverName.....	138
TPrinter:EndDoc.....	138
TPrinter:EndPage.....	138
TPrinter:GetBinSourceNames.....	139
TPrinter:GetBinSourceTypes.....	139
TPrinter:GetPaperNames.....	139
TPrinter:GetPaperSizes.....	139
TPrinter:GetPaperTypes.....	140
TPrinter:GetPrinterDefault.....	140
TPrinter:GetPrinterFonts.....	140
TPrinter:GetPrinterNames.....	140
TPrinter:GetResolutions.....	141
TPrinter:IsDuplex.....	141
TPrinter:IsPrinting.....	141
TPrinter:Location.....	141
TPrinter:PageSetup.....	142
TPrinter:PaperPhysicalSize.....	142
TPrinter:PaperRes.....	142
TPrinter:PaperSize.....	142
TPrinter:PhysicalOffset.....	143
TPrinter:Port.....	143
TPrinter:Preview.....	143
TPrinter:PreviewControl.....	143
TPrinter:PrintProcessor.....	144
TPrinter:Resolution.....	144
TPrinter:Run.....	144
TPrinter:SepFile.....	145
TPrinter:ServerName.....	145
TPrinter:Setup.....	145
TPrinter:ShareName.....	145
TPrinter:StartDoc.....	146
TPrinter:StartDocPdf.....	146
TPrinter:StartPage.....	146
TPrinter:WriteData.....	147
TPrinter:Events.....	147

TPrinter:OnPreview.....	147
TScreen	147
TScreen.Properties	148
TScreen:aMonitors	148
TScreen:Handle.....	149
TScreen:hDC.....	149
TScreen:ITouch	149
TScreen:nBitsPerPixel.....	149
TScreen:nCaptionHeight.....	150
TScreen:nClientHeight.....	150
TScreen:nClientLeft.....	150
TScreen:nClientTop.....	150
TScreen:nClientWidth.....	151
TScreen:nColorPlanes.....	151
TScreen:nHeight.....	151
TScreen:nMonitors	151
TScreen:nPixelsPerInch.....	151
TScreen:nWidth.....	152
TScreen:oCursorArrow	152
TScreen:oCursorDropNo.....	152
TScreen:oCursorDropYes.....	152
TScreen.Methods.....	153
TScreen:CreateDC	153
TScreen>DeleteDC.....	153
TScreen:Initialize	153
TScreen:PaperPhysicalSize.....	154
TScreen:PaperRes	154
TScreen:Resolution.....	154
5 Windows Objects	154
TBalloon	154
TBalloon:Methods	155
TBalloon:AddTool	155
TBalloon:SetToolInfo.....	155
TBitmap	156
TBitmap:Properties	156
TBitmap:cName.....	156
TBitmap:nHeight.....	156
TBitmap:nWidth.....	156
TBitmap:Methods.....	157
TBitmap:CreateFromHandle	157
TBitmap:CreateNew.....	157
TBitmap:Destroy.....	158
TBitmap:LoadFromFile.....	158
TBitmap:LoadFromResource.....	159
TBitmap:New	159
TBitmap:Paint.....	159
TBitmap:Save	160
TBrush	160
TBrush:Properties	160
TBrush:hBitmap.....	160
TBrush:nColor.....	161
TBrush:nStyle.....	161
TBrush:Methods.....	161
TBrush:Clone.....	161

TBrush:Create	162
TBrush:Destroy	162
TCanvas	162
TCanvas:Properties	163
TCanvas:aPenPos	163
TCanvas:hDC	163
TCanvas:ITransparent	164
TCanvas:nClrPane	164
TCanvas:nMapMode	164
TCanvas:nTextAlignment	165
TCanvas:nTextVAlignment	165
TCanvas:oDevice	165
TCanvas:oFont	166
TCanvas:oPen	166
TCanvas:Methods	166
TCanvas:AngleArc	167
TCanvas:Arc	168
TCanvas:ArcTo	168
TCanvas:Chord	168
TCanvas:Create	169
TCanvas:DrawPicture	169
TCanvas:DrawRichEdit	170
TCanvas:Ellipse	170
TCanvas:ExtFloodFill	170
TCanvas:FillRect	171
TCanvas:LineTo	171
TCanvas:MoveTo	172
TCanvas:NewFromDC	172
TCanvas:Pie	172
TCanvas:PixelsToMapMode	173
TCanvas:PixelToTwips	173
TCanvas:PointToPixels	173
TCanvas:Polygon	174
TCanvas:Rectangle	174
TCanvas:RectToPixels	174
TCanvas:RoundRect	175
TCanvas>SelectFont	175
TCanvas>SelectPen	176
TCanvas:TextExtent	176
TCanvas:TextHeight	176
TCanvas:TextLines	177
TCanvas:TextOut	177
TCanvas:TextRect	178
TCanvas:TextWidth	178
TClipboard	179
TClipboardMethods	179
TClipboard:Close	179
TClipboard:CountFormats	180
TClipboard:Empty	180
TClipboard:EnumFormats	180
TClipboard:EnumFormatNames	181
TClipboard:GetData	181
TClipboard:GetDropFiles	182
TClipboard:GetFormatName	182

TClipboard:GetOwner	183
TClipboard:GetText	183
TClipboard:GetWindow	183
TClipboard:HasData	184
TClipboard:IsEmpty	184
TClipboard:IsPrivateFormat	184
TClipboard:IsSystemFormat	184
TClipboard:Open	185
TClipboard:RegisterFormat	185
TClipboard:SetData	186
TClipboard:SetDropFiles	186
TClipboard:SetText	187
TCursor	187
TCursor:Properties	187
TCursor:cName	187
TCursor:Methods	188
TCursor:Clone	188
TCursor:Create	189
TCursor:Destroy	189
TCursor:LoadFromFile	189
TCursor:LoadFromResource	189
TEnhMetafile	190
TEnhMetafile:Properties	190
TEnhMetafile:cDescription	190
TEnhMetafile:cName	191
TEnhMetafile:hNewDC	191
TEnhMetafile:nHeight	191
TEnhMetafile:nWidth	191
TEnhMetafile:Methods	191
TEnhMetafile:Close	192
TEnhMetafile:Copy	192
TEnhMetafile:Create	192
TEnhMetafile:Destroy	193
TEnhMetafile:Load	193
TEnhMetafile:Paint	193
TEnhMetafile:Play	194
TFont	194
TFont:Properties	194
TFont:cName	195
TFont:IBold	195
TFont:Italic	195
TFont:IStrikeOut	195
TFont:IUnderline	195
TFont:nCharSet	196
TFont:nHeight	196
TFont:nOrientation	196
TFont:nSize	197
TFont:nStyle	197
TFont:nWeight	197
TFont:nWidth	198
TFont:oDevice	198
TFont:Methods	198
TFont:Clone	198
TFont:Compare	199

TFont:Create.....	199
TFont:Destroy.....	200
TFont:GetTextHeight.....	200
TFont:GetTextSize.....	200
TFont:GetTextWidth.....	200
TFont:RestoreFromText.....	201
TFont:SaveToText.....	201
THelp.....	201
XHTML Help editor.....	202
THelp:Properties.....	204
THelp:cFilename.....	204
THelp:IModal.....	204
THelp:IShowTopics.....	205
THelp:nType.....	205
THelp:Methods.....	205
THelp:Close.....	206
THelp:Create.....	206
THelp:New.....	206
THelp:PopUp.....	206
THelp:SetFile.....	207
THelp:ShowContext.....	207
THelp:ShowFinder.....	207
THelp:ShowIndex.....	207
THelp:ShowKeyword.....	208
THelp:ShowSearch.....	208
THelp:ShowTOC.....	208
THelp:ShowTopic.....	208
TIcon.....	209
TIcon:Properties.....	209
TIcon:cName.....	209
TIcon:nHeight.....	209
TIcon:nWidth.....	209
TIcon:Methods.....	210
TIcon:Clone.....	210
TIcon:Create.....	210
TIcon:Destroy.....	211
TIcon:Load.....	211
TIcon:LoadFromFile.....	211
TIcon:LoadFromResource.....	211
TIcon:Paint.....	212
TIni.....	212
TIni:Properties.....	212
TIni:cFilename.....	212
TIni:nBuffer.....	213
TIni:Methods.....	213
TIni:AddSection.....	213
TIni:Commit.....	214
TIni:Create.....	214
TIni>DeleteEntry.....	214
TIni>DeleteSection.....	215
TIni>DeleteSections.....	215
TIni:EnumSectionEntries.....	215
TIni:EnumSectionValues.....	215
TIni:GetEntry.....	216

TIni:GetArray	216
TIni:GetDate	216
TIni:GetLogical	217
TIni:GetNumeric	217
TIni:GetSection	217
TIni:GetSectionEx	217
TIni:GetSections	218
TIni:GetSectionsEx	218
TIni:RenameSection	218
TIni:SetEntry	219
TIni:SetSection	219
TIni:SortSection	219
TIni:SystemChanged	219
TMutex	220
TMutex:Properties	220
TMutex:cName	220
TMutex:nTimeOut	220
TMutex:Methods	220
TMutex:Create	221
TMutex:Destroy	221
TMutex:Eventos	221
TMutex:OnCreate	221
TMutex:OnFail	222
TPen	222
TPen:Properties	222
TPen:nStyle	222
TPen:nColor	223
TPen:nWidth	223
TPen:oDevice	223
TPen:Methods	223
TPen:Clone	224
TPen:Create	224
TPen:Destroy	224
TPicture	224
TPicture:Properties	225
TPicture:aTypes	225
TPicture:cName	226
TPicture:hPalette	226
TPicture:IScalable	226
TPicture:ITransparent	226
TPicture:nHeight	226
TPicture:nImageCount	227
TPicture:nImageIndex	227
TPicture:nImageType	227
TPicture:nLogicalHeight	227
TPicture:nLogicalWidth	228
TPicture:nTimeFrame	228
TPicture:nType	228
TPicture:nWidth	228
TPicture:nZoomH	228
TPicture:nZoomV	229
TPicture:Methods	229
TPicture:Clone	229
TPicture:Create	230

TPicture:CreateIndirect.....	230
TPicture:Destroy.....	230
TPicture:GetCaptureDateTime.....	231
TPicture:GetPictureHandle.....	231
TPicture:GetThumbnailSize.....	231
TPicture:Load.....	231
TPicture:LoadFromFile.....	232
TPicture:LoadFromResource.....	232
TPicture:LoadFromStream.....	233
TPicture:IsBitmap.....	233
TPicture:IsEnhMetafile.....	233
TPicture:IsIcon.....	233
TPicture:IsMetafile.....	234
TPicture:IsPicture.....	234
TPicture:MakeThumbNail.....	234
TPicture:Paint.....	236
TPicture:Save.....	237
TPicture:SaveToStream.....	237
TPicture:SetTypes.....	238
TSysVer.....	238
TSysVer:Properties.....	238
TSysVer:cCSDVersion.....	239
TSysVer:cSysName.....	239
TSysVer:cSysVer.....	239
TSysVer:cWinNTPProduct.....	240
TSysVer:llsWin10.....	240
TSysVer:llsWin7.....	240
TSysVer:llsWin98.....	240
TSysVer:llsWinMe.....	241
TSysVer:llsWin2000.....	241
TSysVer:llsWinXP.....	241
TSysVer:llsWinXP64.....	241
TSysVer:llsWinServer2003.....	241
TSysVer:llsWinServer2016.....	242
TSysVer:llsWinServer2019.....	242
TSysVer:llsWinVista.....	242
TSysVer:llsWinHomeServer.....	242
TSysVer:llsWow64.....	243
TSysVer:nBuildNumber.....	243
TSysVer:nMajorVersion.....	243
TSysVer:nMemoryRAM.....	244
TSysVer:nMinorVersion.....	244
TSysVer:nPlatformID.....	244
TSysVer:nProcessorArch.....	245
TSysVer:nProductType.....	245
TSysVer:nReleaseID.....	245
TSysVer:nServicePackMajor.....	246
TSysVer:nServicePackMinor.....	246
TSysVer:nSuiteMask.....	246
TSysVer:Methods.....	247
TSysVer:Create.....	247
TToolTip.....	247
TToolTip:Methods.....	247
TToolTip:AddTool.....	248

TToolTip:Create	248
TToolTip:DelTool	248
TToolTip:SetToolInfo	249
6 Controls	249
Standard	249
TButton	249
TButton:Properties	249
TButton:ICancel	249
TButton:IDefault	250
TButton:IMultiLine	250
TButton:IPushed	250
TButton:nAlignment	250
TButton:nHeight	251
TButton:nModalResult	251
TButton:nVAlignment	251
TButton:nWidth	252
TButton:Methods	252
TButton:Click	252
TButton:Events	252
TButton:OnClick	252
TCheckBox	253
TCheckBox:Properties	253
TCheckBox:IChecked	253
TCheckBox:IMultiLine	253
TCheckBox:IPushLike	254
TCheckBox:ITransparent	254
TCheckBox:nAlignment	254
TCheckBox:nHeight	254
TCheckBox:nVAlignment	254
TCheckBox:nWidth	255
TCheckBox:Methods	255
TCheckBox:Click	255
TCheckBox:Toggle	255
TCheckBox:Events	256
TCheckBox:OnChange	256
TCheckBox:OnClick	256
TComboBox	256
TComboBox:Properties	256
TComboBox:altems	257
TComboBox:IAutoHScroll	257
TComboBox:IDropped	258
TComboBox:IExtendedUI	258
TComboBox:ISort	258
TComboBox:nClrPane	258
TComboBox:nDroppedHeight	258
TComboBox:nDroppedWidth	259
TComboBox:nEditSelEnd	259
TComboBox:nEditSelStart	259
TComboBox:nHeight	259
TComboBox:nHorzExtent	260
TComboBox:nIndex	260
TComboBox:nMaxLength	260
TComboBox:nStyle	260
TComboBox:nTopIndex	261

TComboBox:nWidth	261
TComboBox:Methods	261
TComboBox:AddItem	261
TComboBox:DeleteItem	262
TComboBox:InsertItem	262
TComboBox:ModifyItem	262
TComboBox>SelectString	262
TComboBox:Events	263
TComboBox:OnChange	263
TComboBox:OnCloseUp	263
TComboBox:OnDbClick	264
TComboBox:OnDropDown	264
TComboBox:OnEditChange	264
TComboBox:OnEditUpdate	264
TEdit	265
TEdit:Properties	265
TEdit:cPasswordChar	265
TEdit:IAutoSelect	266
TEdit:IBorder	266
TEdit:IHScroll	266
TEdit:IModify	266
TEdit:INumeric	267
TEdit:IPassword	267
TEdit:IReadOnly	267
TEdit:IShowSystemBalloon	267
TEdit:nAlignment	268
TEdit:nCharCase	268
TEdit:nClrPaneDisabled	268
TEdit:nClrPaneFocus	268
TEdit:nClrTextDisabled	269
TEdit:nClrTextFocus	269
TEdit:nHeight	269
TEdit:nMaxLength	269
TEdit:nWidth	270
TEdit:Value	270
TEdit:Methods	270
TEdit:CanCopy	270
TEdit:CanCut	271
TEdit:CanPaste	271
TEdit:CanUndo	271
TEdit:EmptyUndoBuffer	271
TEdit:GetSel	272
TEdit:GetSelText	272
TEdit:nPos	272
TEdit>SelectAll	272
TEdit>SelectNone	273
TEdit:SetSel	273
TEdit:Events	273
TEdit:OnChange	273
TGroupBox	274
TGroupBox:Properties	274
TGroupBox:nAlignment	274
TGroupBox:nHeight	274
TGroupBox:nWidth	275

TLabel	275
TLabel:Properties	275
TLabel:IAutosize.....	275
TLabel:IMultiLine.....	276
TLabel:IPathStyle	276
TLabel:ITransparent.....	276
TLabel:nAlignment.....	276
TLabel:nBorderStyle.....	276
TLabel:nVAlignment	277
TLabel:Methods.....	277
TLabel:Adjust.....	277
TLine	277
TLine:Properties	278
TLine:IVertical.....	278
TLine:nPenColor	278
TLine:nPenSize	278
TLine:nPenStyle.....	278
TListBox.....	279
TListBox:Properties.....	279
TListBox:altems	280
TListBox:cText	280
TListBox:IAutoSort	280
TListBox:IBorder.....	280
TListBox:IHScroll.....	280
TListBox:IntegralHeight.....	281
TListBox:IMultiColumn.....	281
TListBox:IMultipleSel.....	281
TListBox:IScrollBarAlways.....	281
TListBox:IUseTabStops.....	282
TListBox:nClrPane.....	282
TListBox:nColsWidth.....	282
TListBox:nHeight.....	282
TListBox:nHorzExtent	282
TListBox:nIndex.....	283
TListBox:nTopIndex.....	283
TListBox:nWidth.....	283
TListBox:Methods	283
TListBox:AddItem	284
TListBox:DeleteItem	284
TListBox:DeleteItems	284
TListBox:GetCurSel.....	284
TListBox:GetHorzExtent	285
TListBox:GetSelItems	285
TListBox:InsertItem	285
TListBox:ModifyItem.....	285
TListBox>SelectString.....	286
TListBox:SetCurSel.....	286
TListBox:SetHorzExtent.....	286
TListBox:SetSel	287
TListBox:SetSelItems.....	287
TListBox:Events	287
TListBox:OnChange	287
TListBox:OnDbClick	288
TMemo.....	288

TMemo:Properties	288
TMemo:IAutoHScroll	289
TMemo:IAutoVScroll	289
TMemo:IHScroll	289
TMemo:IVScroll	289
TMemo:IWantReturn	289
TMemo:IWantTab	290
TMemo:nColumn	290
TMemo:nHeight	290
TMemo:nLine	290
TMemo:nLineCount	291
TMemo:nWidth	291
TMemo:Methods	291
TMemo:Append	291
TMemo:Clear	291
TMemo:Replace	292
TMenu	292
TMenu:Properties	292
TMenu:altems	293
TMenu:nStyle	293
TMenu:cMessage	293
TMenu:olmageList	293
TMenu:oParent	294
TMenu:oFont	294
TMenu:Methods	294
TMenu:AddItem	295
TMenu:AddSeparator	295
TMenu:Create	296
TMenu:Destroy	296
TMenu:DoDefault	296
TMenu:Refresh	296
TMenu:SetMenu	297
TMenuItem	297
TMenuItem:Properties	297
TMenuItem:cMessage	297
TMenuItem:cText	298
TMenuItem:IChecked	298
TMenuItem:IDefault	298
TMenuItem:IEnabled	298
TMenuItem:IHilited	299
TMenuItem:IRight	299
TMenuItem:ISeparator	299
TMenuItem:nImage	299
TMenuItem:oSubMenu	300
TMenuItem:Methods	300
TMenuItem:AddMenu	300
TMenuItem:Create	300
TMenuItem:Destroy	301
TMenuItem:GetMainMenu	301
TMenuItem:olmageList	301
TMenuItem:Events	301
TMenuItem:OnCheckState	301
TMenuItem:OnClick	302
TRadio	302

TRadio:Properties.....	302
TRadio:IChecked.....	303
TRadio:IGroup.....	303
TRadio:IMultiLine.....	303
TRadio:IPushLike.....	303
TRadio:ITransparent.....	303
TRadio:nAlignment.....	304
TRadio:nHeight.....	304
TRadio:nVAlignment.....	304
TRadio:nWidth.....	304
TRadio:Methods.....	305
TRadio:Click.....	305
TRadio:Events.....	305
TRadio:OnChange.....	305
TRadio:OnClick.....	305
TScrollBar.....	306
TScrollBar:Properties.....	306
TScrollBar:IDisableNoScroll.....	306
TScrollBar:nClrPane.....	306
TScrollBar:nHeight.....	307
TScrollBar:nMax.....	307
TScrollBar:nMin.....	307
TScrollBar:nOrientation.....	307
TScrollBar:nPage.....	308
TScrollBar:nPos.....	308
TScrollBar:nWidth.....	308
TScrollBar:Methods.....	308
TScrollBar:Dec.....	308
TScrollBar:Inc.....	309
TScrollBar:Events.....	309
TScrollBar:OnChange.....	309
TScrollBar:OnDown.....	309
TScrollBar:OnPageDown.....	310
TScrollBar:OnPageUp.....	310
TScrollBar:OnThumbPos.....	310
TScrollBar:OnThumbTrack.....	311
TScrollBar:OnUp.....	311
Additional.....	311
TAnimatedGif.....	311
TAnimatedGif:Properties.....	311
TAnimatedGif:IPlay.....	312
TAnimatedGif:ITransparent.....	312
TAnimatedGif:nBorderStyle.....	312
TAnimatedGif:oPicture.....	313
TBevel.....	313
TBevel:Properties.....	313
TBevel:IOwnBorder.....	313
TBevel:ITransparent.....	314
TBevel:nBorderStyle.....	314
TBevel:nHeight.....	314
TBevel:nOpacity.....	314
TBevel:nPenColor.....	315
TBevel:nPenSize.....	315
TBevel:nPenStyle.....	315

TBevel:nWidth	315
TBtnBmp	316
TBtnBmp:Properties	316
TBtnBmp:IFlat	316
TBtnBmp:IXPLook	316
TBtnBmp:nBmpHeight	317
TBtnBmp:nBmpWidth	317
TBtnBmp:nFlatStyle	317
TBtnBmp:nMargin	317
TBtnBmp:nOrientation	318
TBtnBmp:nWidth	318
TBtnBmp:oBitmaps	318
TColorComboBox	319
TColorComboBox:Properties	319
TColorComboBox:aColors	319
TColorComboBox:IAddColor	320
TColorComboBox:IShowText	320
TColorComboBox:nColorBoxSize	320
TColorComboBox:nColor	320
TColorComboBox:nLineHeight	320
TDateEdit	321
TDateEdit:Propiedades	321
TDateEdit:nWidth	321
TDateEdit:Value	321
TEditBtn	322
TEditBtn:Properties	322
TEditBtn:cPrompt	322
TEditBtn:nStyle	322
TEditBtn:oBitmap	322
TEditBtn:Events	323
TEditBtn:OnBtnClick	323
TExplorerBar	323
TExplorerBar:Properties	324
TExplorerBar:altems	324
TExplorerBar:ITransparent	324
TExplorerBar:nAlign	324
TExplorerBar:nClrPane	325
TExplorerBar:nOpacity	325
TExplorerBar:nSpacing	325
TExplorerBar:nWidth	326
TExplorerBar:oPicture	326
TExplorerBar:Methods	326
TExplorerBar:AddItem	326
TExplorerBar:HitTest	327
TExplorerBar:Eventos	327
TExplorerBar:OnItemExpand	327
TExplorerGroup	327
TExplorerGroup:Properties	328
TExplorerGroup:cText	328
TExplorerGroup:IExpanded	328
TExplorerGroup:ISpecial	328
TExplorerGroup:IVisible	329
TExplorerGroup:nItem	329
TExplorerGroup:oControl	329

TExplorerGroup:olcon.....	329
TExplorerGroup:oParent.....	329
TExplorerGroup:Methods.....	330
TExplorerGroup:Create.....	330
TExplorerGroup:Refresh.....	330
TFontComboBox.....	330
TFontComboBox:Properties.....	331
TFontComboBox:cSampleText.....	331
TFontComboBox:lComboFont.....	331
TFontComboBox:lEditBmp.....	331
TFontComboBox:lFixedPitch.....	332
TFontComboBox:nFontHeight.....	332
TFontComboBox:nCharSet.....	332
TFontComboBox:nFontType.....	332
TGlassBtn.....	333
TGlassBtn:Properties.....	333
TGlassBtn:nClrGlow.....	333
TGlassBtn:nClrInnerBorder.....	334
TGlassBtn:nClrOuterBorder.....	334
TGlassBtn:nClrPane.....	334
TGlassBtn:nClrShine.....	334
TGlassBtn:nClrText.....	335
TGlassBtn:nHeight.....	335
TGlassBtn:nWidth.....	335
THyperLink.....	335
THyperLink:Properties.....	336
THyperLink:cURL.....	336
THyperLink:nClrOver.....	336
THyperLink:nClrText.....	336
THyperLink:lUnderline.....	336
THyperLink:Events.....	337
THyperLink:OnClick.....	337
THyperLink:OnRClick.....	337
TImage.....	338
TImage:Properties.....	338
TImage:lAutoFit.....	339
TImage:lSmooth.....	339
TImage:lTabStop.....	339
TImage:lTransparent.....	339
TImage:nBorderStyle.....	340
TImage:nClrPane.....	340
TImage:nHeight.....	340
TImage:nWidth.....	340
TImage:nZoom.....	341
TImage:oPicture.....	341
TImageEditor.....	341
TImageEditor:Properties.....	341
TImageEditor:lShowHandlers.....	342
TImageEditor:nBrightness.....	342
TImageEditor:nClrPane.....	342
TImageEditor:nContrast.....	342
TImageEditor:nGamma.....	343
TImageEditor:nSaturation.....	343
TImageEditor:nTransparency.....	343

TImageEditor:Methods	343
TImageEditor:ApplyLevels	344
TImageEditor:BlackWhite.....	344
TImageEditor:Crop.....	344
TImageEditor:FlipH.....	344
TImageEditor:FlipV.....	345
TImageEditor:GrayScale.....	345
TImageEditor:Negative.....	345
TImageEditor:Polaroid.....	345
TImageEditor:RGBToBGR.....	346
TImageEditor:Rotate.....	346
TImageEditor:Save.....	346
TImageEditor:SaveToStream.....	346
TImageEditor:Sepia.....	347
TImageEditor:SetColorMatrix.....	347
TImageEditor:Events.....	347
TImageEditor:OnChangeRect.....	347
TLabelBuddy.....	348
TLabelBuddy:Properties.....	348
TLabelBuddy:nBuddyAlign.....	348
TLabelBuddy:nBuddyOffset.....	349
TLabelBuddy:nVAlignment.....	349
TLabelBuddy:oBuddy.....	349
TLabelEx.....	349
TLabelEx:Properties.....	350
TLabelEx:lAutoSize.....	350
TLabelEx:lMultiLine.....	351
TLabelEx:lTransparent.....	351
TLabelEx:nAlignment.....	351
TLabelEx:nBorderStyle.....	351
TLabelEx:nClrLink.....	352
TLabelEx:nLineSpacing.....	352
TLabelEx:nMargins.....	352
TLabelEx:nOpacity.....	352
TLabelEx:nScrollIncrement.....	352
TLabelEx:nVAlignment.....	353
TLabelEx:Events.....	353
TLabelEx:OnLinkClick.....	353
TLabelEx:OnScrollDown.....	353
TLabelEx:OnScrollPageDown.....	354
TLabelEx:OnScrollPageUp.....	354
TLabelEx:OnScrollUp.....	354
TLabelEx:OnThumbPosV.....	355
TLabelEx:OnThumbTrackV.....	355
TMaskEdit.....	355
TMaskEdit:Propiedades.....	356
TMaskEdit:cPicture.....	356
TMaskEdit:lTrimSpaces.....	357
TMaskEdit:lZeroFill.....	357
TNoEdit.....	357
TNoEdit:Properties.....	358
TNoEdit:cPicture.....	358
TNoEdit:nClrPane.....	358
TNoEdit:nClrText.....	358

TNoEdit:nHeight	359
TNoEdit:nWidth	359
TNoEdit:Value	359
TOptionItem	359
TOptionItem:Properties	360
TOptionItem:alters	360
TOptionItem:cText	360
TOptionItem:lChecked	360
TOptionItem:lCollapsable	361
TOptionItem:lEnabled	361
TOptionItem:lExpanded	361
TOptionItem:lMultiLine	361
TOptionItem:nImage	362
TOptionItem:nImageDisabled	362
TOptionItem:nType	362
TOptionItem:oOptionList	362
TOptionItem:Methods	363
TOptionItem:AddItem	363
TOptionItem:AddSeparator	364
TOptionItem:Create	364
TOptionItem>Delete	365
TOptionItem:InsertItem	365
TOptionItem:Events	366
TOptionItem:OnClick	366
TOptionList	366
TOptionList:Properties	366
TOptionList:alters	367
TOptionList:lAutoSize	367
TOptionList:lTabStop	367
TOptionList:lTransparent	368
TOptionList:nBorderStyle	368
TOptionList:nClrPane	368
TOptionList:nClrText	368
TOptionList:nClrTextHot	369
TOptionList:nHeight	369
TOptionList:nHotStyle	369
TOptionList:nItemHeight	369
TOptionList:nLeftMargin	370
TOptionList:nMargin	370
TOptionList:nScrollIncrement	370
TOptionList:nTopMargin	370
TOptionList:nWidth	371
TOptionList:oImageList	371
TOptionList:Methods	371
TOptionList:AddItem	371
TOptionList:AddSeparator	372
TOptionList:GetItemByPos	372
TOptionList:GetItemCount	373
TOptionList:GetItemPos	373
TOptionList:HitTest	373
TOptionList:InsertItem	373
TOptionList:Events	374
TOptionList:OnScrollDown	374
TOptionList:OnScrollPageDown	375

TOptionList:OnScrollPageUp	375
TOptionList:OnScrollUp.....	375
TOptionList:OnThumbPosV.....	376
TOptionList:OnThumbTrackV	376
TPage.....	376
TPage:Properties.....	377
TPage:ITransparent	377
TPage:nItem	377
TPage:Methods	378
TPage>Delete.....	378
TPage>Select.....	378
TPages.....	378
TPages:Properties.....	379
TPages:altems.....	379
TPages:ITabStop.....	380
TPages:ITransparent	380
TPages:nBorderStyle	380
TPages:nHeight.....	380
TPages:nIndex.....	380
TPages:nWidth.....	381
TPages:Methods	381
TPages:Additem.....	381
TPages:Deleteltem.....	381
TPages:Deleteltems	382
TPages:InsertItem	382
TPages:Next.....	382
TPages:Previous.....	382
TPanel.....	383
TPanel:Properties.....	383
TPanel:IXPLook	383
TPanel:nHeight	383
TPanel:nBevelOuter.....	384
TPanel:nBevellInner.....	384
TPanel:nWidth.....	384
TPreviewControl.....	385
TPreviewControl:Properties	385
TPreviewControl:nPage.....	386
TPreviewControl:nZoomIndex	386
TPreviewPage	387
TPreviewPage:Properties	387
TPreviewPage:nFitInPage.....	387
TPreviewPage:nZoomFactor.....	388
TPreviewPage:oEmf.....	388
TPreviewPage:oPrinter	388
TPreviewPage:Methods.....	388
TPreviewPage:Create New	389
TPreviewPage:Events	389
TPreviewPage:OnFirstPage.....	389
TPreviewPage:OnLastPage.....	389
TPreviewPage:OnNextPage	390
TPreviewPage:OnPreviousPage	390
TPreviewPage:OnZoomChanged	390
TPreviewPage:OnZoomIn	390
TPreviewPage:OnZoomOut.....	391

TRadioMenu.....	391
TRadio:Properties.....	391
TRadioMenu:altems.....	392
TRadioMenu:ITabStop.....	392
TRadioMenu:ITransparent.....	392
TRadioMenu:nBorderStyle.....	392
TRadioMenu:nClrPane.....	392
TRadioMenu:nColumns.....	393
TRadioMenu:nHeight.....	393
TRadioMenu:nIndex.....	393
TRadioMenu:nMargin.....	393
TRadioMenu:nOrientation.....	394
TRadioMenu:nSpacing.....	394
TRadioMenu:nWidth.....	394
TRadio:Events.....	394
TRadio:OnChange.....	394
TRadio:OnClick.....	395
TShiftPanel.....	395
TShiftPanel:Properties.....	395
TShiftPanel:IOpened.....	396
TShiftPanel:nCaptionSize.....	396
TShiftPanel:nClrTitleEnd.....	396
TShiftPanel:nClrTitleStart.....	396
TShiftPanel:nClrTitleText.....	397
TShiftPanel:nIconSize.....	397
TShiftPanel:nOrientation.....	397
TShiftPanel:nVAlignment.....	397
TShiftPanel:oFontTitle.....	398
TShiftPanel:Events.....	398
TShiftPanel:OnClose.....	398
TShiftPanel:OnClosed.....	398
TShiftPanel:OnOpen.....	398
TShiftPanel:OnOpened.....	399
TScrollBar.....	399
TImage:Properties.....	399
TScrollBar:ITransparent.....	399
TScrollBar:nBorderStyle.....	400
TScrollBar:nHeight.....	400
TScrollBar:nWidth.....	400
TSearchComboBox.....	400
TSearchComboBoxEx:Properties.....	400
TSearchComboBox:IEExact.....	401
TSearchComboBox:INoCase.....	401
TSplitButton.....	401
TSplitButton:Properties.....	401
TSplitButton:IMenuOnRight.....	401
TSplitButton:oMenu.....	402
TSplitter.....	402
TSplitter:Properties.....	402
TSplitter:IGripper.....	403
TSplitter:nBorderStyle.....	403
TSplitter:nClrPane.....	403
TSplitter:nHeight.....	403
TSplitter:nMaxPos.....	404

TSplitter:nMinPos	404
TSplitter:nMinSize	404
TSplitter:nWidth	404
TSplitter:Events	405
TSplitter:OnMove	405
Browsea	405
TBrowse	405
TBrowse:Properties	405
TBrowse:aCols	407
TBrowse:aSelected	407
TBrowse:IAllowAppend	407
TBrowse:IAllowColHiding	407
TBrowse:IAllowColSizing	407
TBrowse:IAllowRowSizing	408
TBrowse:IAllowColSwapping	408
TBrowse:IAllowDelete	408
TBrowse:IAllowEdit	408
TBrowse:IAllowInsert	408
TBrowse:IAutoSave	409
TBrowse:IColDividerComplete	409
TBrowse:IEditMode	409
TBrowse:IFastEdit	409
TBrowse:IFilterBar	409
TBrowse:IFooter	410
TBrowse:IHScroll	410
TBrowse:IHeader	410
TBrowse:IRecordSelector	410
TBrowse:ITabStop	410
TBrowse:IVScroll	411
TBrowse:nBorderStyle	411
TBrowse:nClrAltPane	411
TBrowse:nClrPane	411
TBrowse:nClrSelFocusPane	412
TBrowse:nClrSelFocusText	412
TBrowse:nClrSelPane	412
TBrowse:nClrSelText	412
TBrowse:nClrText	412
TBrowse:nColDividerStyle	413
TBrowse:nColSel	413
TBrowse:nDataLines	413
TBrowse:nDataRows	413
TBrowse:nEditMode	413
TBrowse:nFooterHeight	414
TBrowse:nFreeze	414
TBrowse:nHeaderHeight	414
TBrowse:nHeight	414
TBrowse:nKeyNo	414
TBrowse:nLen	415
TBrowse:nMarqueeStyle	415
TBrowse:nRowDividerStyle	415
TBrowse:nRowHeight	415
TBrowse:nRowSel	416
TBrowse:nWidth	416
TBrowse:oHeader	416

TBrowse:oFooter	416
TBrowse:oImageList	416
TBrowse:oRecSel	417
TBrowse:oSeek	417
TBrowse:Methods	417
TBrowse:AddCol	418
TBrowse:AddImage	419
TBrowse:Adjust	419
TBrowse:AdjustCols	419
TBrowse:Append	420
TBrowse:Bookmark	420
TBrowse:CancelEdit	420
TBrowse:ClearAllFilters	420
TBrowse:ColAtIndex	421
TBrowse:ColAtPos	421
TBrowse:ColPos	421
TBrowse:ColWithHeader	421
TBrowse:DelCol	422
TBrowse>Delete	422
TBrowse>Edit	422
TBrowse:GoBottom	422
TBrowse:GoDown	422
TBrowse:GoHPos	423
TBrowse:GoLeft	423
TBrowse:GoLeftMost	423
TBrowse:GoPageDown	423
TBrowse:GoPageUp	424
TBrowse:GoRight	424
TBrowse:GoRightMost	424
TBrowse:Goto	424
TBrowse:GoTop	425
TBrowse:GoUp	425
TBrowse:InsCol	425
TBrowse:Insert	425
TBrowse:IsBof	426
TBrowse:IsEditable	426
TBrowse:IsEof	426
TBrowse:IsMultipleEdit	426
TBrowse:KeyCount	426
TBrowse:KeyNo	427
TBrowse:MouseColPos	427
TBrowse:MouseRowPos	427
TBrowse:MoveCol	427
TBrowse:Refill	428
TBrowse:Refresh	428
TBrowse:RefreshCurrent	428
TBrowse:Reset	428
TBrowse:RestoreState	428
TBrowse:SaveState	429
TBrowse:Seek	429
TBrowse>Select	429
TBrowse>SelectAll	430
TBrowse>SelectCellFromPoint	430
TBrowse>SelectCol	430

TBrowse:SelectNone.....	430
TBrowse:SelectedCol.....	430
TBrowse:StretchCols.....	431
TBrowse:SwapCols.....	431
TBrowse:ToExcel.....	431
TBrowse:Events.....	432
TBrowse:OnAppend.....	432
TBrowse:OnBof.....	432
TBrowse:OnBookMark.....	433
TBrowse:OnChange.....	433
TBrowse:OnChangeHeight.....	433
TBrowse:OnChangeSelection.....	434
TBrowse:OnClick.....	434
TBrowse:OnDelete.....	434
TBrowse:OnDblClick.....	435
TBrowse:OnDrawRow.....	435
TBrowse:OnEdit.....	435
TBrowse:OnEof.....	436
TBrowse:OnFilterChange.....	436
TBrowse:OnGoBottom.....	436
TBrowse:OnGoTop.....	437
TBrowse:OnInsert.....	437
TBrowse:OnKeyCount.....	437
TBrowse:OnKeyNo.....	437
TBrowse:OnNewItem.....	438
TBrowse:OnPastEof.....	438
TBrowse:OnPostEdit.....	438
TBrowse:OnRClick.....	439
TBrowse:OnSeek.....	439
TBrowse:OnSkip.....	440
TArrayBrowse.....	440
TArrayBrowse:Properties.....	440
TArrayBrowse:aArrayData.....	441
TArrayBrowse:IAutoOrder.....	441
TArrayBrowse:nArrayAt.....	441
TArrayBrowse:Methods.....	441
TArrayBrowse:AddRow.....	442
TArrayBrowse:DelRow.....	442
TArrayBrowse:GetRow.....	442
TArrayBrowse:GetValue.....	443
TArrayBrowse:InsRow.....	443
TArrayBrowse:IsFiltered.....	443
TArrayBrowse:nArrayOrgAt.....	443
TArrayBrowse:nArrayPos.....	444
TArrayBrowse:SetArray.....	444
TArrayBrowse:SetRow.....	444
TArrayBrowse:SetValue.....	445
TArrayBrowse:Zap.....	445
TArrayBrowse:Events.....	445
TArrayBrowse:OnSort.....	445
TDbfBrowse.....	446
TDbfBrowse:Properties.....	446
TDbfBrowse:cAlias.....	446
TDbfBrowse:cMsgNoFilter.....	447

TDbfBrowse:Methods	447
TDbfBrowse:SetDbf	447
TBrwColumn	447
TBrwColumn:Properties.....	448
TBrwColumn:aEditListBound.....	449
TBrwColumn:aEditListText	449
TBrwColumn:cFieldName	450
TBrwColumn:cFilter	450
TBrwColumn:cFooter.....	450
TBrwColumn:cHeader.....	451
TBrwColumn:cNullValue.....	451
TBrwColumn:cPicture.....	451
TBrwColumn:cSortEval.....	451
TBrwColumn:cTooltip.....	452
TBrwColumn:FilterEval	452
TBrwColumn:ICanFilter.....	453
TBrwColumn:IParentFont.....	453
TBrwColumn:IVisible	454
TBrwColumn:nAlignment	454
TBrwColumn:nBmpAlignment.....	454
TBrwColumn:nBmpMargin.....	455
TBrwColumn:nClrPane	455
TBrwColumn:nClrText.....	455
TBrwColumn:nEditLength.....	455
TBrwColumn:nEditType.....	456
TBrwColumn:nFooterImage	457
TBrwColumn:nHeaderImage	457
TBrwColumn:nHeight.....	457
TBrwColumn:nIndex.....	458
TBrwColumn:nPos.....	458
TBrwColumn:nSort.....	458
TBrwColumn:nWidth.....	459
TBrwColumn:oDataField	459
TBrwColumn:oGridEdit	459
TBrwColumn:oHeader	460
TBrwColumn:oFooter	460
TBrwColumn:oFont.....	461
TBrwColumn:oParent.....	461
TBrwColumn:uDefaultValue.....	461
TBrwColumnMethods	461
TBrwColumn:Edit.....	462
TBrwColumn>Select.....	462
TBrwColumn:SetAlignment.....	462
TBrwColumn:Value	463
TBrwColumn:Events	463
TBrwColumn:OnCreate.....	464
TBrwColumn:OnBtnClick.....	464
TBrwColumn:OnChangeWidth.....	464
TBrwColumn:OnDataClick.....	465
TBrwColumn:OnDatadBLClick.....	465
TBrwColumn:OnDataRClick	466
TBrwColumn:OnDrawCell	466
TBrwColumn:OnEdit.....	467
TBrwColumn:OnExit.....	468

TBrwColumn:OnFooterClick	468
TBrwColumn:OnFooterDbClick.....	468
TBrwColumn:OnFooterRClick.....	469
TBrwColumn:OnGetData.....	469
TBrwColumn:OnGetValue.....	470
TBrwColumn:OnGridEditCreate.....	470
TBrwColumn:OnHeaderClick.....	471
TBrwColumn:OnHeaderDbClick.....	471
TBrwColumn:OnHeaderRClick.....	471
TBrwColumn:OnPostEdit.....	472
TBrwRecSel.....	473
TBrwRecSel:Properties	473
TBrwRecSel:IParentFont	474
TBrwRecSel:IRecordPointer	474
TBrwRecSel:nAlignment.....	474
TBrwRecSel:nBmpAlignment	474
TBrwRecSel:nBmpMargin.....	475
TBrwRecSel:nClrPane	475
TBrwRecSel:nClrText	475
TBrwRecSel:nHeight	475
TBrwRecSel:nWidth	476
TBrwRecSel:oFont.....	476
TBrwRecSel:oParent.....	476
TBrwRecSel:Events.....	477
TBrwRecSel:OnClick.....	477
TBrwRecSel:OnDbClick.....	477
TBrwRecSel:OnRClick	478
TBrwRecSel:OnGetData.....	478
TGridEdit	479
TGridEdit:Properties	479
TGridEdit:alters.....	479
TGridEdit:altersBound	480
TGridEdit:IAutoSelect.....	480
TGridEdit:IEditable.....	480
TGridEdit:nEditMode	480
TGridEdit:oDataField	481
TGridEdit:oDataSet	481
Win32	481
TComboBoxEx.....	481
TComboBoxEx:Propiedades	482
TComboBoxEx:almages.....	482
TComboBoxEx:alters	482
TComboBoxEx:nIndentWidth	483
TComboBoxEx:olmageList.....	483
TDatePicker	483
TDatePicker:Properties	483
TDatePicker:cCustomFormat	484
TDatePicker:dMaxDate	485
TDatePicker:dMinDate	485
TDatePicker:dValue.....	485
TDatePicker:ICheckBox.....	485
TDatePicker:IUpDown.....	486
TDatePicker:nAlignment.....	486
TDatePicker:nClrMonthPane.....	486

TDatePicker:nClrText	486
TDatePicker:nClrTitlePane	487
TDatePicker:nClrTitleText.....	487
TDatePicker:nClrTrailingText.....	487
TDatePicker:nFormatType.....	487
TDatePicker:nHeight	488
TDatePicker:nWidth	488
TDatePicker:Events.....	488
TDatePicker:OnChange.....	488
TDatePicker:OnCloseUp	488
TDatePicker:OnDropDown.....	489
TFolder	489
TTabCtrl:Properties.....	489
TTabCtrl:alters	489
TFolderPage.....	490
TFolderPage:Properties.....	490
TFolderPage:ITransparent.....	490
TFolderPage:cText	491
TFolderPage:nClrText.....	491
TFolderPage:nImage	491
TFolderPage:Methods	491
TFolderPage:Delete.....	491
TFolderPage:Disable	492
TFolderPage:Enable.....	492
TFolderPage:Select.....	492
THeader.....	492
THeader:Properties.....	493
THeader:alters	493
THeader:IBorder.....	493
THeader:IButtons.....	494
THeader:ICancelDrag.....	494
THeader:ICancelTrack.....	494
THeader:IDragDrop	494
THeader:IFilterBar.....	495
THeader:IFlat	495
THeader:IFullDrag.....	495
THeader:IHotTrack.....	495
THeader:IResize	495
THeader:ITabStop.....	496
THeader:IWordBreak	496
THeader:nAlign.....	496
THeader:nHeight.....	497
THeader:nWidth.....	497
THeader:oEditFilter.....	497
THeader:olmageList.....	497
THeader:Methods	498
THeader:AddItem.....	498
THeader:DeleteItem.....	498
THeader:DeleteItems	499
THeader>EditFilter.....	499
THeader:HitTest.....	499
THeader:InsertItem	499
THeader:Events	500
THeader:OnBeginDrag.....	500

THeader:OnBeginTrack	501
THeader:OnClick	501
THeader:OnDbClick	501
THeader:OnDividerDbClick	502
THeader:OnEndDrag	502
THeader:OnEndTrack	502
THeader:OnFilterBtnClick	502
THeader:OnFilterChange	503
THeader:OnFilterEdit	503
THeader:OnItemChanged	503
THeader:OnItemChanging	504
THeader:OnRClick	504
THeader:OnTrack	504
THeaderItem	504
THeaderItem:Properties	505
THeaderItem:cFilter	505
THeaderItem:cText	505
THeaderItem:cToolTip	505
THeaderItem:nAlignment	506
THeaderItem:nImage	506
THeaderItem:nIndex	506
THeaderItem:nOrder	506
THeaderItem:nSort	507
THeaderItem:nWidth	507
THeaderItem:Events	507
THeaderItem:OnClick	507
THeaderItem:OnDbClick	508
THeaderItem:OnDividerDbClick	508
THeaderItem:OnRClick	508
THotKey	509
THotKey:Properties	509
THotKey:IAlt	509
THotKey:IBorder	509
THotKey:ICtrl	510
THotKey:IShift	510
THotKey:IWinKey	510
THotKey:nClrPane	510
THotKey:nHeight	511
THotKey:nKey	511
THotKey:nWidth	511
THotKey:Events	511
THotKey:OnChange	511
TListView	512
TListView:Properties	512
TListView:aColumns	513
TListView:aGroups	513
TListView:alItems	513
TListView:IBorder	514
TListView:IAlignLeft	514
TListView:IAutoArrange	514
TListView:ICheckBoxes	514
TListView:IDoubleBuffer	515
TListView:IEditLabels	515
TListView:IFullRowSelect	515

TListView:IGridLines.....	515
TListView:IGroupView.....	515
TListView:IHeaderDragDrop.....	516
TListView:IHotTrackSelect.....	516
TListView:ILabelTip.....	516
TListView:ILabelWrap.....	516
TListView:INoColumnHeader.....	517
TListView:INoSortHeader.....	517
TListView:IShowSelAlways.....	517
TListView:ISingleSel.....	517
TListView:ITooltips.....	517
TListView:nClrPane.....	518
TListView:nClrText.....	518
TListView:nHeight.....	518
TListView:nView.....	518
TListView:nWidth.....	519
TListView:olmageListHeader.....	519
TListView:olmageListLarge.....	519
TListView:olmageListSmall.....	519
TListView:Methods.....	520
TListView:AddColumn.....	520
TListView:AddGroup.....	521
TListView:AddItem.....	521
TListView:Arrange.....	522
TListView:CancelEdit.....	522
TListView:DelColumn.....	522
TListView>DeleteAllItems.....	522
TListView:DelGroup.....	523
TListView:GetFocused.....	523
TListView:GetSelection.....	523
TListView:GetVisibleCount.....	523
TListView:HitTest.....	524
TListView:InsertItem.....	524
TListView:IsEmpty.....	525
TListView:SetBkImage.....	525
TListView:Sort.....	525
TListView:Events.....	525
TListView:OnBeginLabelEdit.....	526
TListView:OnCheckStateChanged.....	526
TListView:OnClick.....	526
TListView:OnColumnClick.....	527
TListView:OnDeleteItem.....	527
TListView:OnDrawCell.....	527
TListView:OnItemChanged.....	528
TListView:OnEndLabelEdit.....	528
TListView:OnKeyDown.....	528
TListView:OnRClick.....	529
TListViewColumn.....	529
TListViewColumn:Properties.....	529
TListViewColumn:cText.....	529
TListViewColumn:nAlignment.....	530
TListViewColumn:nImage.....	530
TListViewColumn:nIndex.....	530
TListViewColumn:nWidth.....	530

TListViewColumn:oParent.....	531
TListViewColumn:Methods.....	531
TListViewColumn:Create.....	531
TListViewGroup.....	531
TListViewGroupProperties.....	531
TListViewGroup:cFooter.....	532
TListViewGroup:cHeader.....	532
TListViewGroup:nFooterAlign.....	532
TListViewGroup:nHeaderAlign.....	532
TListViewGroup:nIndex.....	533
TListViewGroup:nState.....	533
TListViewGroup:oParent.....	533
TListViewGroup:Methods.....	533
TListViewGroup:Create.....	533
TListViewItem.....	534
TListViewItem:Properties.....	534
TListViewItem:aData.....	534
TListViewItem:cText.....	534
TListViewItem:cTooltip.....	535
TListViewItem:IChecked.....	535
TListViewItem:nGroup.....	535
TListViewItem:nImage.....	535
TListViewItem:nIndent.....	536
TListViewItem:nIndex.....	536
TListViewItem:oParent.....	536
TListViewItem:Methods.....	536
TListViewItem:Create.....	536
TListViewItem>Delete.....	537
TListViewItem>Edit.....	537
TListViewItem:EnsureVisible.....	537
TListViewItem>Select.....	537
TMonthCalendar.....	538
TMonthCalendar:Properties.....	538
TMonthCalendar:aHolidays.....	539
TMonthCalendar:dMaxDate.....	539
TMonthCalendar:dMinDate.....	539
TMonthCalendar:dToday.....	539
TMonthCalendar:dValue.....	539
TMonthCalendar:IBorder.....	540
TMonthCalendar:IMultiSelect.....	540
TMonthCalendar:INoToday.....	540
TMonthCalendar:INoTodayCircle.....	540
TMonthCalendar:IWeekNumbers.....	541
TMonthCalendar:nClrMonthPane.....	541
TMonthCalendar:nClrPane.....	541
TMonthCalendar:nClrText.....	541
TMonthCalendar:nClrTitlePane.....	542
TMonthCalendar:nClrTitleText.....	542
TMonthCalendar:nClrTrailingText.....	542
TMonthCalendar:nHeight.....	542
TMonthCalendar:nMaxSelCount.....	543
TMonthCalendar:nWidth.....	543
TMonthCalendar:Events.....	543
TMonthCalendar:OnChange.....	543

TMonthCalendar:OnSelect	543
TPager.....	544
TPager:Properties.....	544
TPager:IAutoScroll.....	544
TPager:nHeight.....	545
TPager:nOrientation.....	545
TPager:nPosition.....	545
TPager:nScrollIncrement.....	545
TPager:nWidth.....	545
TProgressBar.....	546
TProgressBar:Properties.....	546
TProgressBar:IMarquee.....	546
TProgressBar:ISmooth.....	547
TProgressBar:nHeight.....	547
TProgressBar:nMax.....	547
TProgressBar:nMarqueeSpeed.....	547
TProgressBar:nMin.....	548
TProgressBar:nOrientation.....	548
TProgressBar:nStep.....	548
TProgressBar:nValue.....	548
TProgressBar:nWidth.....	549
TProgressBar:Methods.....	549
TProgressBar:SetBarColor.....	549
TProgressBar:SetBkColor.....	549
TProgressBar:StepIt.....	550
TProgressBar:Events.....	550
TProgressBar:OnChange.....	550
TRebar.....	550
TRebar:Properties.....	551
TRebar:altems.....	551
TRebar:IBandBorders.....	551
TRebar:IBorder.....	551
TRebar:IDbIClkToggle.....	552
TRebar:IFixedOrder.....	552
TRebar:nAlign.....	552
TRebar:IVarHeight.....	553
TRebar:nHeight.....	553
TRebar:nWidth.....	553
TRebar:olmageList.....	553
TRebar:Methods.....	554
TRebar:Additem.....	554
TRebarBand.....	555
TRebarBand:Properties.....	555
TRebarBand:cText.....	555
TRebarBand:IBreak.....	556
TRebarBand:IChildEdge.....	556
TRebarBand:IFixedBmp.....	556
TRebarBand:IFixedSize.....	556
TRebarBand:IGripper.....	556
TRebarBand:IVisible.....	557
TRebarBand:nImage.....	557
TRebarBand:nIndex.....	557
TRebarBand:oBkgnd.....	557
TRebarBand:oControl.....	558

TRebarBand:Methods.....	558
TRebarBand:Create.....	558
TRebarBand>Delete.....	559
TRichEdit.....	559
TRichEdit:Properties.....	559
TRichEdit:aMargins.....	561
TRichEdit:cFileName.....	561
TRichEdit:cFindText.....	561
TRichEdit:cFontName.....	562
TRichEdit:cReplace.....	562
TRichEdit:cText.....	562
TRichEdit:IBold.....	562
TRichEdit:IBorder.....	563
TRichEdit:IDisableNoScroll.....	563
TRichEdit:IFindCase.....	563
TRichEdit:IFindDown.....	563
TRichEdit:IFindWord.....	564
TRichEdit:IHScroll.....	564
TRichEdit:IHot.....	564
TRichEdit:IHTML.....	564
TRichEdit:Italic.....	564
TRichEdit:IPDF.....	565
TRichEdit:IProtected.....	565
TRichEdit:IRE30.....	565
TRichEdit:IRE40.....	565
TRichEdit:IRE50.....	566
TRichEdit:IRE60.....	566
TRichEdit:IRE70.....	566
TRichEdit:IRE80.....	566
TRichEdit:IReadOnly.....	567
TRichEdit:IRule.....	567
TRichEdit:IShowMessages.....	567
TRichEdit:ISmallCaps.....	567
TRichEdit:IStrikeOut.....	568
TRichEdit:ISubScript.....	568
TRichEdit:ISupScript.....	568
TRichEdit:ITransparent.....	568
TRichEdit:IURLDetect.....	569
TRichEdit:IUnderline.....	569
TRichEdit:IVScroll.....	569
TRichEdit:IWord.....	570
TRichEdit:IWordEx.....	570
TRichEdit:IWordX.....	570
TRichEdit:IWrite.....	570
TRichEdit:IWordSel.....	571
TRichEdit:IWordWrap.....	571
TRichEdit:nAlignment.....	571
TRichEdit:nClrPane.....	571
TRichEdit:nClrText.....	572
TRichEdit:nConverter.....	572
TRichEdit:nFontColor.....	572
TRichEdit:nFontSize.....	572
TRichEdit:nFormat.....	573
TRichEdit:nHeight.....	573

TRichEdit:nHighLight	573
TRichEdit:nLineSpacing	573
TRichEdit:nMaxLength	574
TRichEdit:nMaxUndo	574
TRichEdit:nNumStyle	574
TRichEdit:nNumbering	574
TRichEdit:nPageWidth	575
TRichEdit:nUlineColor	575
TRichEdit:nUlineType	575
TRichEdit:nUnits	575
TRichEdit:nWidth	576
TRichEdit:oPopup	576
TRichEdit:Methods	576
TRichEdit:CanClear	578
TRichEdit:CanCopy	579
TRichEdit:CanCut	579
TRichEdit:CanPaste	579
TRichEdit:CanRedo	579
TRichEdit:CanUndo	580
TRichEdit:CharFromPos	580
TRichEdit:Clear	580
TRichEdit:ColorDlg	580
TRichEdit:Copy	581
TRichEdit:Cut	581
TRichEdit:EmptyUndoBuffer	581
TRichEdit:FindDlg	581
TRichEdit:FindText	582
TRichEdit:FontDlg	582
TRichEdit:GetBullet	582
TRichEdit:GetCol	582
TRichEdit:GetEventMask	583
TRichEdit:GetFirstVisibleLine	583
TRichEdit:GetLine	583
TRichEdit:GetLineCount	583
TRichEdit:GetLineFromChar	584
TRichEdit:GetLineIndex	584
TRichEdit:GetIndent	584
TRichEdit:GetModify	584
TRichEdit:GetNumbering	585
TRichEdit:GetOptions	585
TRichEdit:GetParagraphSpacing	585
TRichEdit:GetPos	585
TRichEdit:GetRedoName	586
TRichEdit:GetRow	586
TRichEdit:GetPrintHeight	586
TRichEdit:GetRTF	586
TRichEdit:GetRTFRange	587
TRichEdit:GetSel	587
TRichEdit:GetSelection	587
TRichEdit:GetSelectionType	588
TRichEdit:GetText	588
TRichEdit:GetTextMode	588
TRichEdit:GetTextRange	588
TRichEdit:GetTypographicOptions	589

TRichEdit:GetUndoName	589
TRichEdit:GetZoom	589
TRichEdit:GotoLine	589
TRichEdit:HasText	590
TRichEdit:HTMLToRTF	590
TRichEdit:InsertBitmap	590
TRichEdit:InsertFile	591
TRichEdit:InsertObject	591
TRichEdit:InsertObjectFromFile	591
TRichEdit:InsertPageBreak	591
TRichEdit:InsertPicture	592
TRichEdit:InsertRTF	592
TRichEdit:InsertTable	592
TRichEdit:InsertText	593
TRichEdit:IsRTF	593
TRichEdit:IsSelection	593
TRichEdit:Len	593
TRichEdit:LoadFile	594
TRichEdit:LoadRTF	594
TRichEdit:MovePos	594
TRichEdit:PageDlg	595
TRichEdit:Paste	595
TRichEdit:PasteSpecial	595
TRichEdit:PictureDlg	595
TRichEdit:PosFromChar	596
TRichEdit:Preview	596
TRichEdit:Print	596
TRichEdit:PrintBox	597
TRichEdit:PrintDlg	597
TRichEdit:PrintSetup	597
TRichEdit:Redo	598
TRichEdit:Replace	598
TRichEdit:ReplaceAll	598
TRichEdit:ReplaceDlg	599
TRichEdit:RTFToHTML	599
TRichEdit:RTFToPDF	599
TRichEdit:RTFToWord	599
TRichEdit:RTFToWordX	600
TRichEdit:RTFToWrite	600
TRichEdit:SaveFile	600
TRichEdit:SelectAll	600
TRichEdit:SelectNone	601
TRichEdit:SetBullet	601
TRichEdit:SetEventMask	601
TRichEdit:SetHideSelection	601
TRichEdit:SetHyperLinkTooltips	602
TRichEdit:SetIndent	602
TRichEdit:SetModify	602
TRichEdit:SetNumbering	603
TRichEdit:SetOptions	603
TRichEdit:SetParagraphSpacing	603
TRichEdit:SetPos	604
TRichEdit:SetText	604
TRichEdit:SetSel	604

TRichEdit:SetTextMode.....	604
TRichEdit:SetTypographyOptions	605
TRichEdit:SetZoom	605
TRichEdit:StopGroupTyping	605
TRichEdit:Undo.....	606
TRichEdit:WordToRtf.....	606
TRichEdit:WordXToRtf.....	606
TRichEdit:WriteToRtf.....	606
TRichEdit:Events	607
TRichEdit:OnChange.....	607
TRichEdit:OnContextMenu.....	607
TRichEdit:OnHScroll.....	608
TRichEdit:OnInsertPicture.....	608
TRichEdit:OnPrinting.....	608
TRichEdit:OnRequestResize.....	608
TStatusBar.....	609
TStatusBar:Properties.....	609
TStatusBar:altems.....	609
TStatusBar:cText	609
TStatusBar:IFullSize	610
TStatusBar:ISimple.....	610
TStatusBar:ISizeGrip.....	610
TStatusBar:nAlign	610
TStatusBar:nBorderStyle.....	611
TStatusBar:nClrPane.....	611
TStatusBar:nHeight.....	611
TStatusBar:nMinHeight	611
TStatusBar:olcon.....	612
TStatusBar:olmImageList.....	612
TStatusBar:Methods	612
TStatusBar:AddBitmap.....	612
TStatusBar:AddIcon	613
TStatusBar:AddItem	613
TStatusBar:DeleteItem	614
TStatusBar:DeleteItems	614
TStatusBar:Events	614
TStatusBar:OnClick.....	614
TStatusBar:OnDrawItem.....	615
TStatusBar:OnRClick	615
TStatusBar:OnSimpleMode.....	615
TStatusPanel	615
TStatusPanel:Properties	616
TStatusPanel:cText.....	616
TStatusPanel:cToolTip.....	616
TStatusPanel:IOwnerDraw.....	617
TStatusPanel:nAlignment.....	617
TStatusPanel:nBorderStyle.....	617
TStatusPanel:nImage	617
TStatusPanel:nMinWidth	618
TStatusPanel:nResizeStyle	618
TStatusPanel:nType	618
TStatusPanel:nWidth	618
TStatusPanel:oControl	619
TStatusPanel:Methods.....	619

TStatusPanel:Create	619
TStatusPanel:End	620
TStatusPanel:New	620
TStatusPanel:Update	620
TStatusPanel:Events	621
TStatusPanel:OnClick	621
TStatusPanel:OnRClick	621
TSysAnimate	622
TSysAnimate:Properties	622
TSysAnimate:nCommonAvi	622
TSysAnimate:cName	623
TSysAnimate:IActive	623
TSysAnimate:IAutoPlay	623
TSysAnimate:ICentered	623
TSysAnimate:ITransparent	624
TSysAnimate:IUseTimer	624
TSysAnimate:nFirstFrame	624
TSysAnimate:nLastFrame	624
TSysAnimate:nRepeat	624
TSysAnimate:Methods	625
TSysAnimate:Close	625
TSysAnimate:Create	625
TSysAnimate:Destroy	626
TSysAnimate:Load	626
TSysAnimate:LoadCommon	626
TSysAnimate:LoadFromFile	626
TSysAnimate:LoadFromResource	627
TSysAnimate:Play	627
TSysAnimate:Seek	627
TSysAnimate:Stop	627
TSysAnimate:Events	628
TSysAnimate:OnClose	628
TSysAnimate:OnOpen	628
TSysAnimate:OnStart	628
TSysAnimate:OnStop	628
TTabCtrl	629
TTabCtrl:Properties	629
TTabCtrl:altems	630
TTabCtrl:IButtonStyle	630
TTabCtrl:IFixedWidth	630
TTabCtrl:IFlatButtons	630
TTabCtrl:IForcelconLeft	631
TTabCtrl:IForceLabelLeft	631
TTabCtrl:IHotTrack	631
TTabCtrl:IMultiLine	631
TTabCtrl:IRightJustify	631
TTabCtrl:IScrollOpposite	632
TTabCtrl:ITabCloseButtons	632
TTabCtrl:ITabStop	632
TTabCtrl:ITransparent	632
TTabCtrl:ITransparentBody	633
TTabCtrl:nHeight	633
TTabCtrl:nIndex	633
TTabCtrl:nMinTabWidth	633

TTabCtrl:nOrientation.....	633
TTabCtrl:nWidth.....	634
TTabCtrl:oImageList.....	634
TTabCtrl:Methods.....	634
TTabCtrl:AddImage.....	634
TTabCtrl:AddItem.....	635
TTabCtrl:DeleteItem.....	635
TTabCtrl:DeleteItems.....	635
TTabCtrl:InsertItem.....	636
TTabCtrl:SetIndex.....	636
TTabCtrl:Events.....	637
TTabCtrl:OnChange.....	637
TTabCtrl:OnChanging.....	637
TTabCtrl:OnCloseTab.....	637
TTabCtrl:OnRClick.....	638
TTabItem.....	638
TTabItem:Properties.....	638
TTabItem:cText.....	639
TTabItem:cToolTip.....	639
TTabItem:Enabled.....	639
TTabItem:nCrIText.....	639
TTabItem:nCrITextHot.....	639
TTabItem:nImage.....	640
TTabItem:nItem.....	640
TTabItem:Methods.....	640
TTabItem:Create.....	640
TTabItem>Delete.....	641
TTabItem>Select.....	641
TTabItem:SetItem.....	641
TTimePicker.....	642
TTimePicker:Properties.....	642
TTimePicker:cCustomFormat.....	642
TTimePicker:cMaxTime.....	643
TTimePicker:cMinTime.....	643
TTimePicker:cTime.....	643
TTimePicker:nHeight.....	644
TTimePicker:nWidth.....	644
TTimePicker:Events.....	644
TTimePicker:OnChange.....	644
TToolBar.....	644
TToolBar:Properties.....	645
TToolBar:alItems.....	645
TToolBar:AlwaysTip.....	645
TToolBar:AutoSize.....	646
TToolBar:IBallons.....	646
TToolBar:IFlat.....	646
TToolBar:IList.....	646
TToolBar:ITransparent.....	647
TToolBar:nAlign.....	647
TToolBar:nBorderStyle.....	647
TToolBar:nBtnHeight.....	648
TToolBar:nBtnWidth.....	648
TToolBar:nHeight.....	648
TToolBar:nWidth.....	648

TToolBar:nIndent.....	648
TToolBar:olmageList.....	649
TToolBar:olmageListDis.....	649
TToolBar:olmageListHot.....	649
TToolBar:Methods.....	650
TToolBar:AddButton.....	650
TToolBar:AddItem.....	650
TToolBar:AddSeparator.....	651
TToolBar:DeleteItem.....	651
TToolBar:InsertItem.....	652
TToolBar:InsertSeparator.....	652
TToolBar:Events.....	653
TToolBar:OnDropDown.....	653
TToolButton.....	653
TToolButton:Properties.....	653
TToolButton:cText.....	654
TToolButton:cTooltip.....	654
TToolButton:lAutoSize.....	654
TToolButton:lChecked.....	654
TToolButton:lEnabled.....	655
TToolButton:lGroup.....	655
TToolButton:lSeparator.....	655
TToolButton:lVisible.....	655
TToolButton:nImage.....	655
TToolButton:nIndex.....	656
TToolButton:oMenu.....	656
TToolButton:Methods.....	656
TToolButton:Create.....	656
TToolButton>Delete.....	657
TToolButton:Click.....	657
TToolButton:Events.....	658
TToolButton:OnClick.....	658
TTrackBar.....	658
TTrackBar:Properties.....	658
TTrackBar:lAutoTicks.....	659
TTrackBar:lDownsLeft.....	659
TTrackBar:lFixedLength.....	659
TTrackBar:lNoThumb.....	659
TTrackBar:lNoTicks.....	660
TTrackBar:lReversed.....	660
TTrackBar:lSelRange.....	660
TTrackBar:lTooltips.....	660
TTrackBar:lTransparent.....	661
TTrackBar:nHeight.....	661
TTrackBar:nLineSize.....	661
TTrackBar:nMax.....	661
TTrackBar:nMin.....	661
TTrackBar:nOrientation.....	662
TTrackBar:nPageSize.....	662
TTrackBar:nSelEnd.....	662
TTrackBar:nSelStart.....	662
TTrackBar:nTickFreq.....	663
TTrackBar:nTickStyle.....	663
TTrackBar:nTipSide.....	663

TTrackBar:nValue.....	663
TTrackBar:nWidth.....	664
TTrackBar:Methods.....	664
TTrackBar:ClearTicks.....	664
TTrackBar:GetNumTicks.....	664
TTrackBar:GetTick.....	665
TTrackBar:GetTickPos.....	665
TTrackBar:GetTicks.....	665
TTrackBar:SetTick.....	665
TTrackBar:Events.....	666
TTrackBar:OnChange.....	666
TTreeView.....	666
TTreeView:Properties.....	666
TTreeView:alters.....	667
TTreeView:IAutoHScroll.....	667
TTreeView:IBorder.....	668
TTreeView:ICheckBoxes.....	668
TTreeView:IDoubleBuffer.....	668
TTreeView:IDragDroptem.....	668
TTreeView:IEditLabels.....	668
TTreeView:IFadeInOutExp.....	669
TTreeView:IFullRowSelect.....	669
TTreeView:IHasButtons.....	669
TTreeView:IHasLines.....	669
TTreeView:IHotTrack.....	670
TTreeView:ILinesAtRoot.....	670
TTreeView:IRichTooltip.....	670
TTreeView:IShowSelAlways.....	670
TTreeView:ISingleExpand.....	670
TTreeView:IToottips.....	671
TTreeView:nClrLine.....	671
TTreeView:nClrPane.....	671
TTreeView:nClrText.....	671
TTreeView:nClrSelItem.....	672
TTreeView:nClrSelfFocus.....	672
TTreeView:nHeight.....	672
TTreeView:nIndent.....	672
TTreeView:nWidth.....	673
TTreeView:olmImageList.....	673
TTreeView:Methods.....	673
TTreeView:AddImage.....	674
TTreeView:AddItem.....	674
TTreeView:CollapseAll.....	675
TTreeView:DeleteItems.....	675
TTreeView:EndEditLabelNow.....	675
TTreeView:ExpandAll.....	675
TTreeView:GetAllItems.....	676
TTreeView:GetCount.....	676
TTreeView:GetItemByName.....	676
TTreeView:GetItemByHandle.....	676
TTreeView:GetSelectedItem.....	677
TTreeView:GetVisibleCount.....	677
TTreeView:HitTest.....	677
TTreeView:InsertItem.....	677

TTreeView:IsEmpty.....	678
TTreeView:RestoreExpandState.....	678
TTreeView:SaveExpandState.....	679
TTreeView:Selectitem.....	679
TTreeView:SelectRoot.....	679
TTreeView:Events.....	679
TTreeView:OnBeginDragItem.....	680
TTreeView:OnBeginLabelEdit.....	680
TTreeView:OnCheckStateChanged.....	680
TTreeView:OnClick.....	681
TTreeView:OnDragOverItem.....	681
TTreeView:OnEndDragItem.....	681
TTreeView:OnEndLabelEdit.....	682
TTreeView:OnItemExpanded.....	682
TTreeView:OnItemExpanding.....	682
TTreeView:OnKeyDown.....	683
TTreeView:OnRClick.....	683
TTreeView:OnSelChanged.....	683
TTreeView:OnSelChanging.....	684
TTreeView:OnSingleExpand.....	684
TTreeViewItem.....	685
TTreeViewItem:Properties.....	685
TTreeViewItem:alters.....	685
TTreeViewItem:cText.....	685
TTreeViewItem:Handle.....	686
TTreeViewItem:IBold.....	686
TTreeViewItem:IChecked.....	686
TTreeViewItem:IParam.....	686
TTreeViewItem:nImage.....	687
TTreeViewItem:nIndex.....	687
TTreeViewItem:nSellImage.....	687
TTreeViewItem:oParent.....	687
TTreeViewItem:oTreeView.....	688
TTreeViewItem:Methods.....	688
TTreeViewItem:AddItem.....	688
TTreeViewItem:AltSortChildren.....	689
TTreeViewItem:Collapse.....	689
TTreeViewItem:Create.....	690
TTreeViewItem>Delete.....	690
TTreeViewItem>DeleteChilds.....	690
TTreeViewItem>Edit.....	690
TTreeViewItem:EnsureVisible.....	691
TTreeViewItem:Expand.....	691
TTreeViewItem:ExpandPartial.....	691
TTreeViewItem:GetNextSibling.....	691
TTreeViewItem:GetPrevSibling.....	692
TTreeViewItem:InsertItem.....	692
TTreeViewItem:IsExpanded.....	693
TTreeViewItem:MoveAfter.....	693
TTreeViewItem:MoveDown.....	693
TTreeViewItem:MoveInto.....	693
TTreeViewItem:MoveUp.....	694
TTreeViewItem>Select.....	694
TTreeViewItem:SetExpandable.....	694

TTreeViewItem:SortChildren	694
TUpDown.....	695
TUpDown:Properties.....	695
TUpDown:aAccels	695
TUpDown:IArrowKeys.....	696
TUpDown:IAutoBuddy.....	696
TUpDown:IHotTrack.....	696
TUpDown:INoThousands	696
TUpDown:ISyncBuddy.....	697
TUpDown:IWrap.....	697
TUpDown:nAccelTime	697
TUpDown:nBase.....	697
TUpDown:nBuddyAlign.....	697
TUpDown:nIncrement	698
TUpDown:nMax	698
TUpDown:nMin	698
TUpDown:nOrientation.....	698
TUpDown:nValue	699
TUpDown:oBuddy.....	699
TUpDown:Events	699
TUpDown:OnChange	699
Communications	699
TWebView.....	699
TWebView:Properties.....	700
TWebView:cAdditionalArgs.....	700
TWebView:cRequestedFilter.....	701
TWebView:cRuntimeDir.....	701
TWebView:cUserDataDir.....	701
TWebView:IAAllowDevTools	701
TWebView:IAAllowScripts	701
TWebView:ICleanUserData	702
TWebView:IContextMenu	702
TWebView:IStatusBar	702
TWebView:nBorderStyle	702
TWebView:nHeight.....	703
TWebView:nWidth.....	703
TWebView:nZoomFactor	703
TWebView:Methods	703
TWebView:CanGoBack.....	704
TWebView:CanGoForward.....	704
TWebView:CleanUserData.....	704
TWebView>DeleteAllCookies.....	704
TWebView:ExecuteScript.....	705
TWebView:GetTitle.....	705
TWebView:GoBack	705
TWebView:GoForward.....	705
TWebView:IsInstalled.....	706
TWebView:Navigate.....	706
TWebView:NavigateEx	706
TWebView:Reload	706
TWebView:SetContent	707
TWebView:SetVirtualHostNameToFolder.....	707
TWebView:Stop.....	707
TWebView:Events	707

TWebView:OnAcceleratorKeyPressed.....	708
TWebView:OnContentLoading.....	708
TWebView:OnDocumentTitleChanged.....	708
TWebView:OnExecuteScriptCompleted.....	709
TWebView:OnHistoryChanged.....	709
TWebView:OnNavigationCompleted.....	709
TWebView:OnNavigationStarting.....	709
TWebView:OnNewWindowRequested.....	710
TWebView:OnSourceChanged.....	710
TWebView:OnWebMessageReceived.....	710
TWebView:OnWebResourceRequested.....	710
TAsyncDownload.....	711
TAsyncDownload:Properties.....	711
TAsyncDownload:cBuffer.....	711
TAsyncDownload:cError.....	712
TAsyncDownload:cUrl.....	712
TAsyncDownload:nStatus.....	712
TAsyncDownload:IConnected.....	712
TAsyncDownload:ISuccess.....	712
TAsyncDownload:Methods.....	713
TAsyncDownload:Cancel.....	713
TAsyncDownload:Run.....	713
TAsyncDownload:Events.....	713
TAsyncDownload:OnConnect.....	713
TAsyncDownload:OnDisconnect.....	714
TAsyncDownload:OnError.....	714
TAsyncDownload:OnTotalDnLoadInfo.....	714
TAsyncDownload:OnTotalLengthInfo.....	714
TAsynFileDownload.....	715
TAsynFileDownload:Properties.....	715
TAsynFileDownload:cError.....	715
TAsynFileDownload:cLocalFile.....	715
TAsynFileDownload:cRemotelFile.....	716
TAsynFileDownload:IConnected.....	716
TAsynFileDownload:ISuccess.....	716
TAsynFileDownload:nStatus.....	716
TAsynFileDownload:Methods.....	716
TAsynFileDownload:Cancel Close.....	717
TAsynFileDownload:Run.....	717
TAsynFileDownload:Events.....	717
TAsynFileDownload:OnConnect.....	717
TAsynFileDownload:OnDisconnect.....	718
TAsynFileDownload:OnError.....	718
TAsynFileDownload:OnTotalDnLoadInfo.....	718
TAsynFileDownload:OnTotalLengthInfo.....	718
TBlatMail.....	719
TBlatMail:Properties.....	719
TBlatMail:aAttachments.....	720
TBlatMail:aBCC.....	720
TBlatMail:aCC.....	720
TBlatMail:aReceipts.....	721
TBlatMail:cAddress.....	721
TBlatMail:cBody.....	721
TBlatMail:cCharset.....	721

TBlatMail:cExtra.....	722
TBlatMail:cPassword.....	722
TBlatMail:cServer.....	722
TBlatMail:cSubject.....	722
TBlatMail:cUser.....	723
TBlatMail:lAuth.....	723
TBlatMail:lLog.....	723
TBlatMail:lUndisclosedRecipients.....	723
TBlatMail:lReceipt.....	724
TBlatMail:nAttachAs.....	724
TBlatMail:nAuth.....	724
TBlatMail:nEncoding.....	724
TBlatMail:nLastError.....	725
TBlatMail:nPort.....	725
TBlatMail:nPriority.....	726
TBlatMail:nTimeout.....	726
TBlatMail:nTries.....	726
TBlatMail:nType.....	726
TBlatMail:Methods.....	727
TBlatMail:AddBCC.....	727
TBlatMail:AddCC.....	727
TBlatMail:AddReceipt.....	727
TBlatMail:Create.....	728
TBlatMail:Destroy.....	728
TBlatMail:Send.....	728
TBlatMail:Events.....	728
TBlatMail:OnError.....	729
TCefBrowser.....	729
TCefBrowser:Properties.....	729
TCefBrowser:cCachePath.....	730
TCefBrowser:cLocale.....	730
TCefBrowser:nBorderStyle.....	730
TCefBrowser:nHeight.....	730
TCefBrowser:nWidth.....	731
TCefBrowser:Methods.....	731
TCefBrowser:CanGoBack.....	731
TCefBrowser:CanGoForward.....	731
TCefBrowser:ClearHistory.....	732
TCefBrowser:CloseDevTools.....	732
TCefBrowser:GetZoomLevel.....	732
TCefBrowser:GoBack.....	732
TCefBrowser:GoForward.....	733
TCefBrowser:HasDocument.....	733
TCefBrowser:HidePopup.....	733
TCefBrowser:IsPopupVisible.....	733
TCefBrowser:Navigate.....	733
TCefBrowser:Reload.....	734
TCefBrowser:SetText.....	734
TCefBrowser:SetZoomLevel.....	734
TCefBrowser:ShowDevTools.....	734
TCefBrowser:StopLoad.....	735
TCefBrowser:Events.....	735
TCefBrowser:OnAddressChange.....	735
TCefBrowser:OnAfterCreated.....	736

TCefBrowser:OnAuthCredentials.....	736
TCefBrowser:OnBeforePopup.....	736
TCefBrowser:OnBeforeResourceLoad.....	736
TCefBrowser:OnClose.....	737
TCefBrowser:OnDownloadComplete.....	737
TCefBrowser:OnDownloadData.....	737
TCefBrowser:OnDownloadFile.....	738
TCefBrowser:OnDownloadStart.....	738
TCefBrowser:OnLoadEnd.....	739
TCefBrowser:OnLoadError.....	739
TCefBrowser:OnLoadStart.....	739
TCefBrowser:OnNavStateChange.....	740
TCefBrowser:OnProtocolExecution.....	740
TCefBrowser:OnStatusMessage.....	740
TCefBrowser:OnTitleChange.....	741
TCdoMail.....	741
TCdoMail:Properties.....	741
TCdoMail:aAttachments.....	742
TCdoMail:cBCC.....	742
TCdoMail:cCC.....	742
TCdoMail:cFrom.....	742
TCdoMail:cMessage.....	742
TCdoMail:cPassword.....	743
TCdoMail:cServer.....	743
TCdoMail:cSubject.....	743
TCdoMail:cTO.....	743
TCdoMail:cUser.....	744
TCdoMail:IAuthenticate.....	744
TCdoMail:IGmailOptions.....	744
TCdoMail:IHtml.....	744
TCdoMail:IInstaled.....	744
TCdoMail:ISSL.....	745
TCdoMail:nPort.....	745
TCdoMail:Methods.....	745
TCdoMail:Send.....	745
TGMailSend.....	745
TGMailSend:Properties.....	746
TGMailSend:cBCC.....	747
TGMailSend:cCC.....	747
TGMailSend:cClientId.....	747
TGMailSend:cError.....	747
TGMailSend:cLanguage.....	748
TGMailSend:cMessageId.....	748
TGMailSend:cSecretId.....	748
TGMailSend:cSubject.....	748
TGMailSend:cText.....	748
TGMailSend:cTo.....	749
TGMailSend:IHtml.....	749
TGMailSend:cToken.....	749
TGMailSend:nError.....	749
TGMailSend:Methods.....	749
TGMailSend:Authorize.....	750
TGMailSend:Send.....	750
TPop3Client.....	750

TPop3Client:Properties.....	751
TPop3Client:aMails.....	751
TPop3Client:cLastMail.....	751
TPop3Client:cServer.....	752
TPop3Client:cPassword.....	752
TPop3Client:cUsername.....	752
TPop3Client:nPort.....	752
TPop3Client:nRcvBufSize.....	753
TPop3Client:nTimeOut.....	753
TPop3Client:lConnected.....	753
TPop3Client:lDelOnRead.....	753
TPop3Client:lDigest.....	753
TPop3Client:Methods.....	754
TPop3Client:Close.....	754
TPop3Client:Count.....	754
TPop3Client>Delete.....	755
TPop3Client:List.....	755
TPop3Client:MailFiles.....	755
TPop3Client:MailFrom.....	755
TPop3Client:MailHtml.....	756
TPop3Client:MailText.....	756
TPop3Client:MailTo.....	756
TPop3Client:MailSubject.....	756
TPop3Client:New.....	757
TPop3Client:Open.....	757
TPop3Client:Retrieve.....	757
TPop3Client:RetrieveAll.....	757
TPop3Client:Events.....	758
TPop3Client:OnNewMail.....	758
TPop3Client:OnTimeOut.....	758
TPop3Client:Functions.....	758
ReadPop3Asyn.....	758
TInternet.....	759
TInternet:Properties.....	759
TInternet:cAgent.....	759
TInternet:cPassword.....	760
TInternet:cUser.....	760
TInternet:hSession.....	760
TInternet:nLastError.....	760
TInternet:nPort.....	760
TInternet:nService.....	761
TInternet:nTimeOut.....	761
TInternet:Methods.....	761
TInternet:AddRequestHeader.....	762
TInternet:AttemptConnect.....	762
TInternet:CanonicalizeUrl.....	763
TInternet:CheckConnection.....	763
TInternet:Close.....	763
TInternet:CloseFile.....	763
TInternet:CloseRequest.....	764
TInternet:CloseURL.....	764
TInternet:ConfirmZoneCrossing.....	764
TInternet:Connect.....	764
TInternet:Destroy.....	765

TInternet:ErrorDialog.....	765
TInternet:GetConnectedState.....	765
TInternet:GetErrorDescription.....	766
TInternet:GetFileSize.....	766
TInternet:GetRequest.....	766
TInternet:GetVersion.....	766
TInternet:Open.....	767
TInternet:OpenRequest.....	767
TInternet:OpenURL.....	768
TInternet:QueryInfo.....	768
TInternet:PostRequest.....	768
TInternet:ReadFile.....	769
TInternet:SendRequest.....	769
TInternet:SetProxyData.....	769
TInternet:SetTimeOut.....	770
TInternet:WriteFile.....	770
TInternet:Events.....	770
TInternet:OnClose.....	770
TInternet:OnConnect.....	771
TInternet:OnGetData.....	771
TInternet:OnOpen.....	771
TFTP.....	771
TFTP:Properties.....	772
TFTP:IPassive.....	772
TFTP:nBuffer.....	772
TFTP:nTransferType.....	772
TFTP:Methods.....	772
TFTP:Create.....	773
TFTP:CreateDirectory.....	773
TFTP:CloseFile.....	773
TFTP:Command.....	774
TFTP>DeleteFile.....	774
TFTP:Directory.....	774
TFTP:DownloadFile.....	775
TFTP:GetCurrentDirectory.....	775
TFTP:GetFile.....	775
TFTP:GetFileSize.....	776
TFTP:OpenFile.....	776
TFTP:OpenFileRead.....	776
TFTP:OpenFileWrite.....	776
TFTP:PutFile.....	777
TFTP:RemoveDirectory.....	777
TFTP:RenameFile.....	777
TFTP:SetCurrentDirectory.....	777
TFTP:UploadFile.....	778
TFTP:Events.....	778
TFTP:OnCommand.....	778
TFTP:OnComplete.....	779
TFTP:OnDirectory.....	779
TFTP:OnError.....	779
TFTP:OnProgress.....	779
TFTP:OnStart.....	780
TMapi.....	780
TMapi:Properties.....	780

TMapi:aAttachments	781
TMapi:aReceipts	781
TMapi:cAccount	782
TMapi:cFromAddress	782
TMapi:cFromName	782
TMapi:cMessage	782
TMapi:cName	782
TMapi:cPassword	783
TMapi:cSubject	783
TMapi:cTime	783
TMapi:dDate	783
TMapi:lAskPassword	784
TMapi:lAskRecipients	784
TMapi:lLoggedOn	784
TMapi:lNewSession	784
TMapi:lReceipt	784
TMapi:lUseUI	785
TMapi:nLastError	785
TMapi:Methods	785
TMapi:Create	785
TMapi:Destroy	786
TMapi:GetAccountNames	786
TMapi:Installed	786
TMapi:Logoff	786
TMapi:Logon	787
TMapi:Send	787
TMailSlotClient	787
TMailSlotClientr:Properties	787
TMailSlotClient:cName	788
TMailSlotClient:cServer	788
TMailSlotClient:Methods	788
TMailSlotClient:Close	788
TMailSlotClient:IsOpen	788
TMailSlotClient:Open	789
TMailSlotClient:Write	789
TMailSlotClient:Events	789
TMailSlotClient:OnClose	789
TMailSlotClient:OnOpen	790
TMailSlotClient:OnWrite	790
TMailSlotServer	790
TMailSlotServer:Properties	790
TMailSlotServer:cName	791
TMailSlotServer:cServer	791
TMailSlotServer:Methods	791
TMailSlotServer:Close	791
TMailSlotServer:GetMessageCount	792
TMailSlotServer:HasMessages	792
TMailSlotServer:IsOpen	792
TMailSlotServer:Open	792
TMailSlotServer:Read	793
TMailSlotServer:SetReadTimeout	793
TMailSlotServer:Events	793
TMailSlotServer:OnClose	793
TMailSlotServer:OnOpen	793

TMailSlotServer:OnRead	794
TWebBrowser.....	794
TWebBrowser:Properties	794
TWebBrowser:Methods.....	795
TWebBrowser:Events	796
Modern	798
TButtonEx.....	798
TButtonEx:Properties	798
TButtonEx:alters.....	799
TButtonEx:cShortCut	799
TButtonEx:IDropped	799
TButtonEx:IMultiLine	800
TButtonEx:IPushed.....	800
TButtonEx:nAlignment.....	800
TButtonEx:nBmpHeight.....	800
TButtonEx:nBmpWidth.....	801
TButtonEx:nClrBorder.....	801
TButtonEx:nClrPane.....	801
TButtonEx:nClrPanelItems.....	801
TButtonEx:nEffects	802
TButtonEx:nHeight	802
TButtonEx:nMargin.....	802
TButtonEx:nOrientation.....	802
TButtonEx:nStyle.....	803
TButtonEx:nVAlignment.....	803
TButtonEx:nWidth.....	803
TButtonEx:oBitmaps.....	803
TButtonEx:oFontShortCut	804
TButtonEx:Events.....	804
TButtonEx:OnClick.....	804
TButtonEx:OnDrop	804
TButtonMod.....	805
TButtonMod:Properties.....	805
TButtonMod:ICancel	805
TButtonMod:IDefault.....	806
TButtonMod:ILegacyBehaviour.....	806
TButtonMod:IMultiLine.....	806
TButtonMod:ITransparent.....	806
TButtonMod:nBmpHeight	807
TButtonMod:nBmpMargin.....	807
TButtonMod:nBmpWidth	807
TButtonMod:nBorderRadius	807
TButtonMod:nLineSpacing.....	808
TButtonMod:nModalResult	808
TButtonMod:nOrientation.....	808
TButtonMod:nVAlignment.....	809
TButtonMod:oBitmaps.....	809
TButtonMod:oMenu	809
TButtonMod:Events.....	810
TButtonMod:OnCustomDraw.....	810
TButtonMod:OnMenuClick	810
TBrowseMod.....	810
TBrowseMod:Propiedades.....	811
TBrowseMod:aCols.....	812

TBrowseMod:aDisplayCols.....	813
TBrowseMod:aGridData.....	813
TBrowseMod:aSelected.....	813
TBrowseMod:cRecSelectorHeader.....	813
TBrowseMod:IAAllowAppend.....	814
TBrowseMod:IAAllowColHiding.....	814
TBrowseMod:IAAllowSizing.....	814
TBrowseMod:IAAllowColSwapping.....	814
TBrowseMod:IAAllowDelete.....	814
TBrowseMod:IAAllowEdit.....	815
TBrowseMod:IAAllowInsert.....	815
TBrowseMod:IAAllowSort.....	815
TBrowseMod:IAAlternateColor.....	815
TBrowseMod:IAutoSave.....	816
TBrowseMod:IColumnSeparator.....	816
TBrowseMod:IFastAppend.....	816
TBrowseMod:IFastEdit.....	816
TBrowseMod:IFilterBar.....	817
TBrowseMod:IFooter.....	817
TBrowseMod:IHeader.....	817
TBrowseMod:IHideScrollBars.....	817
TBrowseMod:IHotTrack.....	817
TBrowseMod:IMultipleSel.....	818
TBrowseMod:IRowHeightButtons.....	818
TBrowseMod:IShowCellSelected.....	818
TBrowseMod:ISupraHeader.....	818
TBrowseMod:ITooltips.....	819
TBrowseMod:ITransparent.....	819
TBrowseMod:ITransparentGrid.....	819
TBrowseMod:IUseReturn.....	819
TBrowseMod:nClrDivider.....	820
TBrowseMod:nClrHeaderDivider.....	820
TBrowseMod:nClrHeaderEditPane.....	820
TBrowseMod:nClrHeaderEditPaneFocus.....	820
TBrowseMod:nClrHeaderEditText.....	821
TBrowseMod:nClrHeaderEditTextFocus.....	821
TBrowseMod:nClrHeaderHotPane.....	821
TBrowseMod:nClrHeaderHotText.....	821
TBrowseMod:nClrHeaderPane.....	822
TBrowseMod:nClrHeaderSelPane.....	822
TBrowseMod:nClrHeaderText.....	822
TBrowseMod:nClrHotPane.....	822
TBrowseMod:nClrHotText.....	823
TBrowseMod:nClrPane.....	823
TBrowseMod:nClrSelPane.....	823
TBrowseMod:nClrSelText.....	823
TBrowseMod:nClrText.....	824
TBrowseMod:nEditMode.....	824
TBrowseMod:nFooterHeight.....	824
TBrowseMod:nHeaderHeight.....	824
TBrowseMod:nIndex.....	824
TBrowseMod:nMargin.....	825
TBrowseMod:nRecSelectorWidth.....	825
TBrowseMod:nRowHeight.....	825

TBrowseMod:nRowIncrement.....	825
TBrowseMod:nSelectCol.....	826
TBrowseMod:oDataGrid.....	826
TBrowseMod:oEditCol.....	826
TBrowseMod:oFooter.....	826
TBrowseMod:oGroup.....	826
TBrowseMod:oHeader.....	827
TBrowseMod:olmageList.....	827
TBrowseMod:oRecSelector.....	827
TBrowseMod:oSelectCol.....	827
TBrowseMod:oSupra.....	828
TBrowseMod:Methods.....	828
TBrowseMod:AddCol.....	829
TBrowseMod:AdjustCols.....	829
TBrowseMod:AddSupraCol.....	829
TBrowseMod:AdjustColWidth.....	830
TBrowseMod:Append.....	830
TBrowseMod:CancelEdit.....	830
TBrowseMod:ClearFilter.....	830
TBrowseMod:ColAtPos.....	831
TBrowseMod:ColWithHeader.....	831
TBrowseMod:DelCol.....	831
TBrowseMod>Delete.....	831
TBrowseMod>Edit.....	832
TBrowseMod:GetDataRow.....	832
TBrowseMod:GoBottom.....	832
TBrowseMod:GoDown.....	832
TBrowseMod:GoLeft.....	833
TBrowseMod:GoLeftMost.....	833
TBrowseMod:GoRight.....	833
TBrowseMod:GoRightMost.....	833
TBrowseMod:GoTop.....	833
TBrowseMod:GroupOn.....	834
TBrowseMod:HitTest.....	834
TBrowseMod:HitTestCol.....	834
TBrowseMod:InsCol.....	834
TBrowseMod:Insert.....	835
TBrowseMod:IsFilter.....	835
TBrowseMod:IsOnEdit.....	835
TBrowseMod:IsSingleEdit.....	835
TBrowseMod:IsSorted.....	836
TBrowseMod:MinimumRowHeight.....	836
TBrowseMod:RestoreState.....	836
TBrowseMod:SaveState.....	836
TBrowseMod:Sort.....	837
TBrowseMod:StretchCols.....	837
TBrowseMod:UnSort.....	837
TBrowseMod:ToExcel.....	837
TBrowseMod:Events.....	838
TBrowseMod:OnAppend.....	839
TBrowseMod:OnCancelEdit.....	839
TBrowseMod:OnChange.....	839
TBrowseMod:OnChangeSelected.....	839
TBrowseMod:OnClick.....	840

TBrowseMod:OnClickRecSelector	840
TBrowseMod:OnContextMenu	840
TBrowseMod:OnDbClick	841
TBrowseMod:OnDispTooltip	841
TBrowseMod:OnDelete	842
TBrowseMod:OnDrawActiveRow	842
TBrowseMod:OnDrawCell	842
TBrowseMod:OnDrawFooter	843
TBrowseMod:OnDrawGroupFooter	844
TBrowseMod:OnDrawGroupHeader	844
TBrowseMod:OnDrawHeader	845
TBrowseMod:OnDrawRecSelector	845
TBrowseMod:OnEdit	846
TBrowseMod:OnEditCol	846
TBrowseMod:OnEditColBtnClick	846
TBrowseMod:OnFilterEnd	847
TBrowseMod:OnFilterStart	847
TBrowseMod:OnFooterClick	847
TBrowseMod:OnFooterRClick	847
TBrowseMod:OnHeaderClick	848
TBrowseMod:OnHeaderRClick	848
TBrowseMod:OnInsert	848
TBrowseMod:OnPostSave	848
TBrowseMod:OnSave	849
TBrowseMod:OnSaveCol	849
TBrowseMod:OnUpdate	849
TBrwColMod	849
TBrwColMod:Properties	850
TBrwColMod:aEditListBound	851
TBrwColMod:aEditListText	851
TBrwColMod:cFilter	852
TBrwColMod:cFooter	852
TBrwColMod:cHeader	852
TBrwColMod:cPicture	852
TBrwColMod:cTooltip	853
TBrwColMod:FilterEval	853
TBrwColMod:ICanDrag	854
TBrwColMod:ICanEdit	855
TBrwColMod:ICanFilter	855
TBrwColMod:ICanHide	855
TBrwColMod:ICanHot	855
TBrwColMod:ICanResize	855
TBrwColMod:ICanSort	856
TBrwColMod:IDataMultiLine	856
TBrwColMod:LeftPos	856
TBrwColMod:IMultiLine	856
TBrwColMod:IVisible	857
TBrwColMod:ITransparent	857
TBrwColMod:nAlignment	857
TBrwColMod:nBmpAlignment	857
TBrwColMod:nClrPane	858
TBrwColMod:nDataAlignment	858
TBrwColMod:nDataBmpAlignment	858
TBrwColMod:nDataCol	858

TBrwColMod:nDataVAlignment	859
TBrwColMod:nDroppedHeight	859
TBrwColMod:nDroppedWidth	859
TBrwColMod:nEditLength	859
TBrwColMod:nEditType	860
TBrwColMod:nFilterEditStyle	861
TBrwColMod:nImage	861
TBrwColMod:nIndex	861
TBrwColMod:nVAlignment	861
TBrwColMod:oEditFilter	862
TBrwColMod:oFooterCol	862
TBrwColMod:oGridEdit	862
TBrwColMod:oHeaderCol	862
TBrwColMod:oSupraCol	862
TBrwColMod:RightPos	863
TBrwColMod:uDefaultValue	863
TBrwColMod:Methods	863
TBrwColMod:Value	863
TBrwModGrid	864
TEditForGrid	864
TBrwGroup	864
TBrwGroup:Properties	864
TBrwGroup:aGroupCols	865
TBrwGroup:ICollapsed	865
TBrwGroup:IFooter	865
TBrwGroup:ILine	865
TBrwGroup:Methods	865
TBrwGroup:SumTotal	866
TBrwGroupSt	866
TBrwGroupSt:Methods	866
TBrwGroupSt:Average	866
TBrwGroupSt:Header	867
TBrwGroupSt:Max	867
TBrwGroupSt:Min	867
TBrwGroupSt:Sum	867
TBtnPanelMod	868
TBtnPanelMod:Properties	868
TBtnPanelMod:IHotBorder	868
TBtnPanelMod:nClrPane	868
TBtnPanelMod:nHeight	869
TBtnPanelMod:nHotOpacity	869
TBtnPanelMod:nPushedOpacity	869
TBtnPanelMod:nWidth	869
TCalendarMod	869
TCalendarMod:Properties	870
TCalendarMod:aDays	870
TCalendarMod:aDaysSelected	871
TCalendarMod:aMonths	871
TCalendarMod:aSDays	871
TCalendarMod:aSMonths	871
TCalendarMod:dValue	872
TCalendarMod:IHighLiteToday	872
TCalendarMod:IShowDOW	872
TCalendarMod:IShowFirstOfGroup	872

TCalendarMod:IShowLines	872
TCalendarMod:ITransparent	873
TCalendarMod:nClrBorder	873
TCalendarMod:nClrDaysDisabled	873
TCalendarMod:nClrHot	873
TCalendarMod:nClrSelection	874
TCalendarMod:nClrTextHeader	874
TCalendarMod:nClrToday	874
TCalendarMod:nDisplayMode	874
TCalendarMod:nNumberOfWeeks	875
TCalendarMod:nSelectionMode	875
TCalendarMod:oFontDOW	875
TCalendarMod:oFontHeader	875
TCalendarMod:oHeader	876
TCalendarMod:oNext	876
TCalendarMod:oPrev	876
TCalendarMod:Events	876
TCalendarMod:OnChange	876
TCalendarMod:OnChangeView	877
TCalendarMod:OnCheckDate	877
TCalendarMod:OnDrawItem	877
TCalendarMod:OnSelect	878
TCardBody	878
TCardBody:Introduction	879
TCardBody:Properties	883
TCardBody:aData	883
TCardBody:altems	884
TCardBody:aSelected	884
TCardBody:lCardBorder	884
TCardBody:lCardShadow	884
TCardBody:lGapFirstCard	885
TCardBody:lHideScrollBars	885
TCardBody:lHotTrack	885
TCardBody:lMultipleSel	885
TCardBody:nCardClrHot	885
TCardBody:nCardClrPane	886
TCardBody:nCardClrAct	886
TCardBody:nCardClrLink	886
TCardBody:nCardClrSel	886
TCardBody:nCardGap	887
TCardBody:nCardInnerGap	887
TCardBody:nCardMargin	887
TCardBody:nCardSize	887
TCardBody:nClickItem	888
TCardBody:nClickPos	888
TCardBody:nHeight	888
TCardBody:nHotItem	888
TCardBody:nHotPos	889
TCardBody:nIndex	889
TCardBody:nWidth	889
TCardBody:olmageList	889
TCardBody:oLabelEx	890
TCardBody:Methods	890
TCardBody:AddItem	890

TCardBox:Deleteltem	891
TCardBox:Deleteltems	891
TCardBox:InsertItem	891
TCardBox:HitTest	891
TCardBox:RecCount	892
TCardBox:Events	892
TCardBox:OnCardChange	892
TCardBox:OnCardClick	892
TCardBox:OnCardDbClick	893
TCardBox:OnCardGapPaint	893
TCardBox:OnCardPaint	893
TCardBox:OnCardSelection	894
TCardItem	894
TCardItem:Properties	894
TCardItem:IButton	895
TCardItem:IHotTrack	895
TCardItem:IMultiLine	895
TCardItem:IParentFont	895
TCardItem:IStretch	896
TCardItem:IVisible	896
TCardItem:nAlign	896
TCardItem:nAlignWeight	897
TCardItem:nAlignment	897
TCardItem:nClrLink	897
TCardItem:nClrText	897
TCardItem:nColumn	898
TCardItem:nItem	898
TCardItem:nSiblingWeight	898
TCardItem:nSize	898
TCardItem:nTextPad	899
TCardItem:nType	899
TCardItem:nVAlignment	899
TCardItem:oCursor	900
TCardItem:oFont	900
TCardItem:oParent	900
TCardItem:Value	900
TCardItem:Methods	901
TCardItem>Delete	901
TCardItem:IsSelected	901
TCardItem:IsActive	901
TCheckBoxMod	901
TCheckBoxMod:Properties	902
TCheckBoxMod:IChecked	902
TCheckBoxMod:ITransparent	902
TCheckBoxMod:nClrColorization	902
TCheckBoxMod:nClrTextDisabled	903
TCheckBoxMod:nLineSpacing	903
TCheckBoxMod:nVAlignment	903
TCheckBoxMod:Events	903
TCheckBoxMod:OnChange	903
TCircularImage	904
TCircularImage:Properties	904
TCircularImage:ITransparent	904
TCircularImage:nClrPane	904

TCircularImage:nClrPen.....	905
TCircularImage:nPenSize.....	905
TCircularImage:Events.....	905
TCircularImage:OnClick.....	905
TComboBoxMod.....	905
TComboBoxMod:Properties.....	906
TComboBoxMod:altems.....	906
TComboBoxMod:cText.....	906
TComboBoxMod:IDropAnimation.....	907
TComboBoxMod:IDropped.....	907
TComboBoxMod:IFreeEdit.....	907
TComboBoxMod:nClrDropHotPane.....	907
TComboBoxMod:nClrDropHotText.....	907
TComboBoxMod:nClrDropPane.....	908
TComboBoxMod:nClrDropText.....	908
TComboBoxMod:nClrDropSelPane.....	908
TComboBoxMod:nClrDropSelText.....	908
TComboBoxMod:nDropItemHeight.....	909
TComboBoxMod:nDropMargin.....	909
TComboBoxMod:nDroppedHeight.....	909
TComboBoxMod:nDroppedWidth.....	909
TComboBoxMod:nIndex.....	910
TComboBoxMod:nStyle.....	910
TComboBoxMod:olmageList.....	910
TComboBoxMod:Events.....	910
TComboBoxMod:OnCloseUp.....	911
TComboBoxMod:OnDrawItem.....	911
TComboBoxMod:OnDropCreate.....	911
TComboBoxMod:OnDropDown.....	912
TDateEditMod.....	912
TDateEditMod:Properties.....	912
TDateEditMod:IDropAnimation.....	913
TDateEditMod:IDropShowDOW.....	913
TDateEditMod:IDropShowFirstOfGroup.....	913
TDateEditMod:IDropShowLines.....	913
TDateEditMod:nClrDropBorder.....	913
TDateEditMod:nClrDropDaysDisabled.....	914
TDateEditMod:nClrDropHot.....	914
TDateEditMod:nClrDropTextHeader.....	914
TDateEditMod:nClrDropToday.....	914
TDateEditMod:nDropNumerOfWeeks.....	915
TDateEditMod:nDroppedHeight.....	915
TDateEditMod:nDroppedWidth.....	915
TDateEditMod:Events.....	915
TDateEditMod:OnDrawItem.....	915
TDateEditMod:OnCloseUp.....	916
TDateEditMod:OnDropCreate.....	916
TDateEditMod:OnDropDown.....	916
TEditMod.....	917
TEditMod:Properties.....	917
TEditMod:cHint.....	917
TEditMod:nClrColorization.....	917
TEditMod:nStyle.....	918
TEditMod:oBitmaps.....	918

TEditMod:Events	918
TEditBtn:OnBtnClick	919
TEditMod:OnClearClick	919
TGroupBoxMod	919
TGroupBox:Properties	919
TGroupBoxMod:nAlignment	920
TGroupBoxMod:nFrameClrPane	920
TGroupBoxMod:nFrameClrPen	920
TGroupBoxMod:nFrameElipseX	920
TGroupBoxMod:nFrameElipseY	921
TGroupBoxMod:nFrameRect	921
TGroupBoxMod:nFrameSize	921
TGroupBoxMod:nLineGap	921
TGroupBoxMod:nHeight	922
TGroupBoxMod:nTextGap	922
TGroupBoxMod:nTextVAlignment	922
TGroupBoxMod:nWidth	922
THeaderMod	922
THeaderMod:Properties	923
THeaderMod:altems	923
THeaderMod:ICanDrag	924
THeaderMod:ICanHide	924
THeaderMod:IFilterBar	924
THeaderMod:IHotTrack	924
THeaderMod:IResize	925
THeaderMod:nBmpMargin	925
THeaderMod:nClrDivider	925
THeaderMod:nClrEditPane	925
THeaderMod:nClrHotPane	926
THeaderMod:nClrHotText	926
THeaderMod:nClrPane	926
THeaderMod:nClrSelPane	926
THeaderMod:nClrText	927
THeaderMod:nFilterTimeOut	927
THeaderMod:nMarginHor	927
THeaderMod:nMarginVer	927
THeaderMod:nSelect	927
THeaderMod:olmageList	928
THeaderMod:Methods	928
THeaderMod:AddItem	928
THeaderMod:ClearAllFilters	929
THeaderMod:ClearFilter	929
THeaderMod:DeleteItem	929
THeaderMod:DeleteItems	930
THeaderMod:InsertItem	930
THeaderMod:HitTest	930
THeaderMod:IsEmpty	931
THeaderMod:IsFilter	931
THeaderMod:RestoreState	931
THeaderMod:SaveState	931
THeaderMod:SwapItems	932
THeaderMod:Events	932
THeaderMod:OnDividerDbClick	932
THeaderMod:OnDrawItem	932

THeaderMod:OnFilterBtnClick.....	933
THeaderMod:OnFilterChange.....	933
THeaderMod:OnFilterEdit.....	934
THeaderMod:OnFilterTimeOut.....	934
THeaderMod:OnItemBeginDrag.....	934
THeaderMod:OnItemChanged.....	934
THeaderMod:OnItemChanging.....	935
THeaderMod:OnItemClick.....	935
THeaderMod:OnItemEndDrag.....	935
THeaderMod:OnItemRClick.....	936
THeaderMod:OnRebuild.....	936
THeaderItemMod.....	936
THeaderItemMod:Properties.....	936
THeaderItemMod:cFilter.....	937
THeaderItemMod:cHeader.....	937
THeaderItemMod:cTooltip.....	937
THeaderItemMod:ICanDrag.....	937
THeaderItemMod:ICanFilter.....	938
THeaderItemMod:ICanHide.....	938
THeaderItemMod:ICanHot.....	938
THeaderItemMod:ICanResize.....	938
THeaderItemMod:ICanSort.....	938
THeaderItemMod:IMultiline.....	938
THeaderItemMod:IVisible.....	939
THeaderItemMod:nAlignment.....	939
THeaderItemMod:nFilterEditStyle.....	939
THeaderItemMod:nImage.....	939
THeaderItemMod:nSort.....	940
THeaderItemMod:nVAlignment.....	940
THeaderItemMod:nWidth.....	940
THeaderItemMod:oEditFilter.....	940
THeaderItemMod:oGridCol.....	940
TListBoxMod.....	941
TListBoxMod:Properties.....	941
TListBoxMod:aChecked.....	942
TListBoxMod:altems.....	942
TListBoxMod:aSelected.....	942
TListBoxMod:cText.....	942
TListBoxMod:ICheckBoxes.....	942
TListBoxMod:IHideScrollbars.....	943
TListBoxMod:IHotTrack.....	943
TListBoxMod:IHtmlText.....	943
TListBoxMod:IMultiLine.....	943
TListBoxMod:IMultipleSel.....	944
TListBoxMod:IShortCuts.....	944
TListBoxMod:IShowActive.....	944
TListBoxMod:nClrHotPane.....	944
TListBoxMod:nClrHotText.....	945
TListBoxMod:nClrSelPane.....	945
TListBoxMod:nClrSelText.....	945
TListBoxMod:nIndex.....	945
TListBoxMod:nItemHeight.....	946
TListBoxMod:nMargin.....	946
TListBoxMod:nOrientation.....	946

TListBoxMod:nVAlignment.....	946
TListBoxMod:olmageList.....	947
TListBoxMod:Methods.....	947
TListBoxMod:AddImage.....	947
TListBoxMod:AddItem.....	948
TListBoxMod:CheckAll.....	948
TListBoxMod:DeleteItem.....	948
TListBoxMod:DeleteItems.....	948
TListBoxMod:FirstVisible.....	949
TListBoxMod:ForceItemVisible.....	949
TListBoxMod:InsertItem.....	949
TListBoxMod:IsItemVisible.....	949
TListBoxMod:ItemChecked.....	950
TListBoxMod:ItemHot.....	950
TListBoxMod:ItemPushed.....	950
TListBoxMod:ItemSelected.....	950
TListBoxMod>LastVisible.....	951
TListBoxMod:ModifyItem.....	951
TListBoxMod:RecCount.....	951
TListBoxMod:SetFirstVisible.....	951
TListBoxMod:UnCheckAll.....	952
TListBoxMod:Events.....	952
TListBoxMod:OnCalcVirtualSize.....	952
TListBoxMod:OnChange.....	952
TListBoxMod:OnChanged.....	953
TListBoxMod:OnChangeSelected.....	953
TListBoxMod:OnCheckStateChanged.....	953
TListBoxMod:OnClick.....	953
TListBoxMod:OnDbClick.....	954
TListBoxMod:OnDrawItem.....	954
TListBoxMod:OnSelect.....	955
TMemoMod.....	955
TMemoMod:Properties.....	956
TMemoMod:IModBorder.....	956
TMemoMod:nClrColorization.....	956
TProgressBarMod.....	956
TProgressBarMod:Properties.....	957
TProgressBarMod:IMarquee.....	957
TProgressBarMod:nHeight.....	957
TProgressBarMod:nMax.....	958
TProgressBarMod:nMarqueeSpeed.....	958
TProgressBarMod:nMin.....	958
TProgressBarMod:nOrientation.....	958
TProgressBarMod:nStep.....	959
TProgressBarMod:nValue.....	959
TProgressBarMod:nWidth.....	959
TProgressBarMod:Events.....	959
TProgressBarMod:OnChange.....	959
TProgressCircle.....	960
TProgressCircle:Properties.....	960
TProgressCircular:IMarquee.....	961
TProgressCircular:IShowPercent.....	961
TProgressCircular:nCircleGap.....	961
TProgressCircular:nCircleSize.....	961

TProgressCircular:nClrCircle	962
TProgressCircular:nHeight	962
TProgressCircularMod:nMax	962
TProgressCircular:nMarqueeSpeed	962
TProgressCircular:nMin	963
TProgressCircular:nValue	963
TProgressCircular:nWidth	963
TProgressCircular:Events	963
TProgressCircular:OnChange	963
TRadioMod	964
TRadioMod:Properties	964
TRadioMod:cGroup	964
TRadioMod:IChecked	964
TRadioMod:ITransparent	965
TRadioMod:nClrColorization	965
TRadioMod:nClrTextDisabled	965
TRadioMod:nLineSpacing	965
TRadioMod:nVAlignment	966
TRadioModMod:Events	966
TRadioMod:OnChange	966
TRadioMod:OnClick	966
TScreenKeyboard	967
TScreenKeyboard:Properties	967
TScreenKeyboard:aClrKeys	967
TScreenKeyboard:aKeys	968
TScreenKeyboard:IHotTrack	968
TScreenKeyboard:ITransparent	968
TScreenKeyboard:nBorderStyle	968
TScreenKeyboard:nClrKeys	969
TScreenKeyboard:nHeight	969
TScreenKeyboard:nLayout	969
TScreenKeyboard:nStyle	969
TScreenKeyboard:nWidth	970
TScreenKeyboard:oFontShortCut	970
TScreenKeyboard:Events	970
TScreenKeyboard:OnKeyClick	970
TShape	971
TShape:Properties	971
TShape:IFillShape	971
TShape:nClrLine	972
TShape:nLineWidth	972
TShape:nHeight	972
TShape:nOpacity	972
TShape:nRadius	973
TShape:nShapeType	973
TShape:nVertices	973
TShape:nWidth	973
TSummerNote	974
TSummerNote:Properties	974
TSummerNote:cHtml	974
TSummerNote:cLanguage	974
TSummerNote:cPlaceholder	975
TSummerNote:cToolBar	975
TSummerNote:Methods	975

TSummerNote:GetHtml	975
TSummerNote:SetHtml	976
TSwitch	976
TSwitch:Properties	976
TSwitch:IChecked	977
TSwitch:ITransparent	977
TSwitch:nClrColorization	977
TSwitch:nHeight	977
TSwitch:nWidth	977
TSwitch:Methods	978
TSwitch:Click	978
TSwitch:Toggle	978
TSwitch:Events	978
TSwitch:OnChange	978
TSwitch:OnClick	979
TTrackBarMod	979
TTrackBarMod:Properties	979
TTrackBarMod:nHeight	979
TTrackBarMod:nClrBarBack	980
TTrackBarMod:nClrBarFore	980
TTrackBarMod:nMax	980
TTrackBarMod:nMin	980
TTrackBarMod:nOrientation	981
TTrackBarMod:nValue	981
TTrackBarMod:nWidth	981
TTrackBarMod:Eventos	981
TTrackBarMod:OnChange	981
TTabs	982
TTabs:Properties	982
TTabs:altems	983
TTabs:IFixedWidth	983
TTabs:IPhantomTab	983
TTabs:ISameFrameSize	983
TTabs:IShowActive	984
TTabs:IShowTabsMenu	984
TTabs:ITabCloseButtons	984
TTabs:ITabStop	985
TTabs:ITabsCanMove	985
TTabs:ITransparent	985
TTabs:nFrame	985
TTabs:nHeight	986
TTabs:nIndex	986
TTabs:nItemPadding	986
TTabs:nOrientation	986
TTabs:nPhantomImage	987
TTabs:nTabsClrPane	987
TTabs:nTabsClrPaneHot	987
TTabs:nTabsClrPaneSel	987
TTabs:nTabsClrPen	988
TTabs:nTabsClrText	988
TTabs:nTabsClrTextHot	988
TTabs:nTabsClrTextSel	988
TTabs:nTabsMargin	989
TTabs:nTabsSize	989

TTabs:nWidth	989
TTabs:olmageList	989
TTabs:Methods.....	990
TTabs:AddItem	990
TTabs>DeleteItem	990
TTabs>DeleteItems.....	991
TTabs:InsertItem.....	991
TTabs:MoveTab	991
TTabs:RestoreState.....	992
TTabs:SaveState	992
TTabs:Events.....	992
TTabs:OnCalcRect	992
TTabs:OnChange.....	993
TTabs:OnChanging.....	993
TTabs:OnCloseTab.....	993
TTabs:OnClick.....	994
TTabs:OnDrawTab	994
TTabs:OnDrawTabsBackground.....	995
TTabs:OnNewTab	995
TTabs:OnRClick	995
TTabsItem	995
TTabsItem:Properties	996
TTabsItem:cText.....	996
TTabsItem:cToolTip.....	996
TTabsItem:IEnabled.....	996
TTabsItem:IVisible	997
TTabsItem:nAlignment.....	997
TTabsItem:nBmpAlignment	997
TTabsItem:nCrLfText	997
TTabsItem:nImage	998
TTabsItem:nItem.....	998
TTabsItem:nSize.....	998
TTabsItem:oParent	998
TTabsItem:Methods.....	998
TTabsItem:Create	999
TTabsItem>Delete	999
TTabsItem>Select	999
TTabsItem:Selected.....	1000
TTabsItem:Events.....	1000
TTabsItem:OnClick.....	1000
TTabsItem:OnClose.....	1000
TTabsItem:OnRClick.....	1000
TTabsFolder	1001
TTabsFolder:Properties	1001
TTabsFolder:altems.....	1001
TTabsFolderPage	1001
TTabsFolderPage:Properties	1002
TTabsFolderPage:IEnabled.....	1002
TTabsFolderPage:IVisible.....	1002
TTabsFolderPage:ITransparent.....	1003
TTabsFolderPage:cText.....	1003
TTabsFolderPage:cToolTip.....	1003
TTabsFolderPage:nAlignment	1003
TTabsFolderPage:nBmpAlignment.....	1004

TTabsFolderPage:nClrText	1004
TTabsFolderPage:nImage.....	1004
TTabsFolderPage:nItem.....	1004
TTabsFolderPage:nSize	1005
TTabsFolderPage:oParent.....	1005
TTabsFolderPage:Methods.....	1005
TTabsFolderPage:Create.....	1005
TTabsFolderPage>Delete.....	1006
TTabsFolderPage>Select.....	1006
TTabsFolderPage:Selected.....	1006
TTabsFolderPage:Events.....	1006
TTabsFolderPage:OnClick.....	1007
TTabsFolderPage:OnClose.....	1007
TTabsFolderPage:OnRClick	1007
TTreeViewMod.....	1008
TTreeViewMod:Properties	1008
TTreeViewMod:alters	1008
TTreeViewMod:IEditLabels.....	1009
TTreeViewMod:IHasLines	1009
TTreeViewMod:IMultiLine.....	1009
TTreeViewMod:ISingleExpand.....	1009
TTreeViewMod:ITooltips.....	1010
TTreeViewMod:nGap	1010
TTreeViewMod:nMargin	1010
TTreeViewMod:oEdit.....	1010
TTreeViewMod:Methods	1010
TTreeViewMod:AddItem.....	1011
TTreeViewMod:BeginPaint	1011
TTreeViewMod:CollapseAll.....	1012
TTreeViewMod:DeleteItems.....	1012
TTreeViewMod:EditLabel.....	1012
TTreeViewMod:EndEditLabelNow.....	1012
TTreeViewMod:EndPaint	1012
TTreeViewMod:ExpandAll.....	1013
TTreeViewMod:GetAllItems.....	1013
TTreeViewMod:GetCount.....	1013
TTreeViewMod:GetItemAtPos	1013
TTreeViewMod:GetItemByID	1014
TTreeViewMod:GetItemByName	1014
TTreeViewMod:GetItemPos.....	1014
TTreeViewMod:GetSelectedItem	1015
TTreeViewMod:InsertItem	1015
TTreeViewMod:IsEmpty.....	1015
TTreeViewMod>SelectItem.....	1015
TTreeViewMod>SelectRoot.....	1016
TTreeViewMod:Events.....	1016
TTreeViewMo:OnBeginLabelEdit.....	1016
TTreeViewMod:OnClick.....	1016
TTreeViewMod:OnDbClick.....	1017
TTreeViewMod:OnDrawItem	1017
TTreeViewMod:OnEndLabelEdit.....	1018
TTreeViewMod:OnGetTooltipText.....	1018
TTreeViewMod:OnItemExpanded	1019
TTreeViewMod:OnItemExpanding	1019

TTreeViewItemMod.....	1019
TTreeViewItemMod:Properties.....	1019
TTreeViewItemMod:alters.....	1020
TTreeViewItemMod:cText.....	1020
TTreeViewItemMod:cTooltip.....	1020
TTreeViewItemMod:IChecked.....	1020
TTreeViewItemMod:IExpanded.....	1021
TTreeViewItemMod:ISelected.....	1021
TTreeViewItemMod:nClrText.....	1021
TTreeViewItemMod:nID.....	1021
TTreeViewItemMod:nImage.....	1021
TTreeViewItemMod:nIndex.....	1022
TTreeViewItemMod:nSellImage.....	1022
TTreeViewItemMod:oParent.....	1022
TTreeViewItemMod:oTreeView.....	1022
TTreeViewItemMod:Methods.....	1022
TTreeViewItemMod:AddItem.....	1023
TTreeViewItemMod:Collapse.....	1023
TTreeViewItemMod>Delete.....	1023
TTreeViewItemMod>DeleteChilds.....	1024
TTreeViewItemMod:EnsureVisible.....	1024
TTreeViewItemMod:Expand.....	1024
TTreeViewItemMod:FirstChild.....	1024
TTreeViewItemMod:GetNextSibling.....	1024
TTreeViewItemMod:GetPrevSibling.....	1025
TTreeViewItemMod:HasChilds.....	1025
TTreeViewItemMod:InsertItem.....	1025
TTreeViewItemMod>LastChild.....	1025
TTreeViewItemMod:MoveAfter.....	1026
TTreeViewItemMod:MoveDown.....	1026
TTreeViewItemMod:MoveUp.....	1026
TTreeViewItemMod>Select.....	1026
TTreeViewItemMod:SortChildren.....	1026
System.....	1027
TImageList.....	1027
TImageList:Properties.....	1027
TImageList:nHeight.....	1027
TImageList:nWidth.....	1028
TImageList:oParent.....	1028
TImageList:Methods.....	1028
TImageList:Add.....	1029
TImageList:AddIcon.....	1029
TImageList:Create.....	1029
TImageList:Destroy.....	1030
TImageList:GetBitmap.....	1030
TImageList:GetIcon.....	1030
TImageList:GetIconSize.....	1031
TImageList:GetImageCount.....	1031
TImageList:Remove.....	1031
TImageList:RemoveAll.....	1031
TImageList:Replacelcon.....	1032
TImageList:SetIconSize.....	1032
TNotifyIcon.....	1032
TNotifyIcon:Properties.....	1032

TNotifyIcon:cToolTip	1033
TNotifyIcon:IVisible	1033
TNotifyIcon:nImage	1033
TNotifyIcon:oImageList	1033
TNotifyIcon:oMenu	1034
TNotifyIcon:oParent	1034
TNotifyIcon:Methods	1034
TNotifyIcon:AddIcon	1034
TNotifyIcon:Create	1035
TNotifyIcon:Destroy	1035
TNotifyIcon:Hide	1035
TNotifyIcon:HideBalloon	1035
TNotifyIcon:New	1036
TNotifyIcon:Refresh	1036
TNotifyIcon:SetIcon	1036
TNotifyIcon:SetMenu	1036
TNotifyIcon:SetToolTip	1037
TNotifyIcon:Show	1037
TNotifyIcon:ShowBalloon	1037
TNotifyIcon:Events	1038
TNotifyIcon:OnBalloonClick	1038
TNotifyIcon:OnBalloonHide	1038
TNotifyIcon:OnBalloonShow	1038
TNotifyIcon:OnBalloonTimeout	1039
TNotifyIcon:OnDbClick	1039
TNotifyIcon:OnLButtonDown	1039
TNotifyIcon:OnLButtonUp	1039
TNotifyIcon:OnMouseMove	1040
TNotifyIcon:OnRButtonUp	1040
TNotifyIcon:OnRButtonDbClick	1040
TNotifyIcon:OnRButtonDown	1041
TProcess	1041
TProcessProperties	1041
TProcess:cName	1041
TProcess:IWait	1041
TProcessMethods	1042
TProcess:IsRunning	1042
TProcess:Run	1042
TProcess:Stop	1042
TRegistry	1043
TRegistry:Properties	1043
TRegistry:Root	1043
TRegistry:hCurrentKey	1043
TRegistry:nLastError	1044
TRegistry:Methods	1044
TRegistry:CloseKey	1045
TRegistry:Connect	1045
TRegistry:CreateKey	1045
TRegistry>DeleteKey	1045
TRegistry>DeleteValue	1046
TRegistry:Destroy	1046
TRegistry:Disconnect	1046
TRegistry:EnumKeys	1046
TRegistry:EnumValues	1047

TRegistry:Flush	1047
TRegistry:GetBinary	1047
TRegistry:GetDate	1048
TRegistry:GetLogical	1048
TRegistry:GetNumeric	1048
TRegistry:GetString	1048
TRegistry:OpenKey	1049
TRegistry:SetDate	1049
TRegistry:SetLogical	1049
TRegistry:SetNumeric	1050
TRegistry:SetString	1050
TRegistry:SetValue	1050
TRunProcess	1051
TRunProcess:Properties	1051
TRunProcess:cCommandLine	1051
TRunProcess:cErrorMsg	1051
TRunProcess:cWorkingDir	1052
TRunProcess:hProcess	1052
TRunProcess:hThread	1052
TRunProcess:nShowMode	1052
TRunProcess:Methods	1053
TRunProcess:Run	1053
TRunProcess:End	1053
TRunProcess:IsRunning	1054
TRunProcess:Kill	1054
TRunProcess:Write	1054
TRunProcess:Events	1054
TRunProcess:OnEnd	1054
TRunProcess:OnRead	1055
TRunProcess:OnStart	1055
TSharedModule	1055
TSharedModule:Properties	1056
TSharedModule:cClassname	1056
TSharedModule:Events	1056
TSharedModule:OnCreate	1056
TSharedModule:OnInitialize	1056
TSysImageList	1057
TSysImageList:Properties	1057
TSysImageList:ILargeIcons	1057
TSysImageList:IUseFileAttributes	1057
TSysImageList:IOverlay	1058
TSysImageList:ISelected	1058
TSysImageList:IShellSize	1058
TSysImageList:Methods	1058
TSysImageList:Create	1059
TSysImageList:GetBitmap	1059
TSysImageList:GetIcon	1059
TSysImageList:GetIconIndex	1059
TSysImageList:GetIconName	1060
TSysImageList:GetIconType	1060
TTimer	1060
TTimer:Properties	1060
TTimer:Enabled	1061
TTimer:Interval	1061

TTimer:oParent	1061
TTimer:Methods	1061
TTimer:Create	1062
TTimer:Destroy	1062
TTimer:Disable	1062
TTimer:Enable	1062
TTimer:New	1063
TTimer:SetInterval	1063
TTimer:Stop	1063
TTimer:Events	1063
TTimer:OnTimer	1063
TWinService	1064
TWinService:Properties	1064
TWinService:cDescription	1065
TWinService:cDisplayName	1065
TWinService:cName	1065
TWinService:nStatus	1065
TWinService:Methods	1065
TWinService:Install	1066
TWinService:IsInstalled	1066
TWinService:IsRunning	1066
TWinService:Run	1066
TWinService:Start	1067
TWinService:Stop	1067
TWinService:Uninstall	1067
TWinService:Events	1067
TWinService:OnPowerEvent	1068
TWinService:OnRun	1068
TWinService:OnShutDown	1068
Dialogs	1069
TBrowseForFolder	1069
TBrowseForFolder:Properties	1069
TBrowseForFolderDlg:cOkBtnText	1070
TBrowseForFolderDlg:cText	1070
TBrowseForFolderDlg:cRoot	1070
TBrowseForFolderDlg:cStatusText	1071
TBrowseForFolderDlg:cTitle	1071
TBrowseForFolderDlg:IBrowseComputer	1071
TBrowseForFolderDlg:ICenter	1071
TBrowseForFolderDlg:IDontGoBelowDomain	1071
TBrowseForFolderDlg:IEditBox	1072
TBrowseForFolderDlg:IExtendedUI	1072
TBrowseForFolderDlg:IIncludeFiles	1072
TBrowseForFolderDlg:IIncludeURLS	1072
TBrowseForFolderDlg:IForceCreate	1073
TBrowseForFolderDlg:INewDlgStyle	1073
TBrowseForFolderDlg:INewFolderButton	1073
TBrowseForFolderDlg:IReturnFSAncestors	1073
TBrowseForFolderDlg:IReturnFSDirs	1073
TBrowseForFolderDlg:IShareable	1074
TBrowseForFolderDlg:ITooltips	1074
TBrowseForFolderDlg:IValidate	1074
TBrowseForFolderDlg:nIconIndex	1074
TBrowseForFolderDlg:nRoot	1075

TBrowseForFolderDlg:Methods	1075
TBrowseForFolderDlg:Run.....	1075
TBrowseForFolderDlg:BrowseComputers.....	1076
TBrowseForFolderDlg:BrowseFile.....	1076
TBrowseForFolderDlg:BrowseFolder.....	1076
TBrowseForFolderDlg:BrowsePrinters.....	1077
TBrowseForFolderDlg:Events.....	1077
TBrowseForFolderDlg:OnChange.....	1077
TBrowseForFolderDlg:OnInitialized.....	1077
TBrowseForFolderDlg:OnValidate.....	1078
TChooseColorDlg.....	1078
TChooseColorDlg:Properties.....	1079
TChooseColorDlg:aCustomColors.....	1079
TChooseColorDlg:cHtmlColor.....	1079
TChooseColorDlg:HelpID.....	1080
TChooseColorDlg:IAnyColor.....	1080
TChooseColorDlg:ICenter.....	1080
TChooseColorDlg:IFullOpen.....	1080
TChooseColorDlg:IPreventFullOpen.....	1080
TChooseColorDlg:IRGBInit.....	1081
TChooseColorDlg:IShowHelp.....	1081
TChooseColorDlg:ISolidColor.....	1081
TChooseColorDlg:nColor.....	1081
TChooseColorDlg:nLastError.....	1082
TChooseColorDlg:Methods.....	1082
TChooseColorDlg:Run.....	1082
TChooseColorDlg:Events.....	1082
TChooseColorDlg:OnClose.....	1082
TChooseColorDlg:OnHelp.....	1083
TChooseColorDlg:OnShow.....	1083
TChooseColorDlg:Functions.....	1083
ChooseColorDlg.....	1084
RGBInverse.....	1084
RGBToHTML.....	1084
TChooseFontDlg.....	1085
TChooseFontDlg:Properties.....	1085
TChooseFontDlg:cName.....	1086
TChooseFontDlg:HelpID.....	1086
TChooseFontDlg:IAllFonts.....	1087
TChooseFontDlg:IBold.....	1087
TChooseFontDlg:ICenter.....	1087
TChooseFontDlg:IEffects.....	1087
TChooseFontDlg:IFixedPitch.....	1087
TChooseFontDlg:IForceFontExist.....	1088
TChooseFontDlg:Italic.....	1088
TChooseFontDlg:INoFaceSel.....	1088
TChooseFontDlg:INoScript.....	1088
TChooseFontDlg:INoSimulations.....	1089
TChooseFontDlg:INoSizeSel.....	1089
TChooseFontDlg:INoStyleSel.....	1089
TChooseFontDlg:INoVectorFonts.....	1089
TChooseFontDlg:INoVertFonts.....	1089
TChooseFontDlg:IPrinterFonts.....	1090
TChooseFontDlg:IScalableOnly.....	1090

TChooseFontDlg:IScreenFonts.....	1090
TChooseFontDlg:IScriptsOnly	1090
TChooseFontDlg:IShowHelp.....	1091
TChooseFontDlg:IStrikeOut.....	1091
TChooseFontDlg:ITrueTypeOnly.....	1091
TChooseFontDlg:IUnderline.....	1091
TChooseFontDlg:IWysiwyg	1091
TChooseFontDlg:nCharSet.....	1092
TChooseFontDlg:nColor.....	1092
TChooseFontDlg:nHeight.....	1092
TChooseFontDlg:nLastError	1092
TChooseFontDlg:nSize.....	1093
TChooseFontDlg:nSizeMax.....	1093
TChooseFontDlg:nSizeMin.....	1093
TChooseFontDlg:nStyle.....	1093
TChooseFontDlg:nWeight.....	1093
TChooseFontDlg:nWidth	1094
TChooseFontDlg:Methods.....	1094
TChooseFontDlg:CreateFont.....	1094
TChooseFontDlg:Run	1094
TChooseFontDlg:SetFont	1095
TChooseFontDlg:Events.....	1095
TChooseFontDlg:OnClose.....	1095
TChooseFontDlg:OnHelp.....	1095
TChooseFontDlg:OnShow	1096
TChooseFontDlg:Functions.....	1096
ChooseFontDlg.....	1096
CreateFontDlg.....	1097
TFileOpenDlg.....	1098
TFileOpenDlg:Properties	1098
TFileOpenDlg:cDefaultExt.....	1099
TFileOpenDlg:cFilename	1099
TFileOpenDlg:cFilenameOnly.....	1100
TFileOpenDlg:cFilter.....	1100
TFileOpenDlg:cFilePath.....	1100
TFileOpenDlg:cFullFilename	1101
TFileOpenDlg:cInitialDir.....	1101
TFileOpenDlg:cTitle	1101
TFileOpenDlg:HelpID	1101
TFileOpenDlg:IAllowMultiSelect	1102
TFileOpenDlg:ICenter	1102
TFileOpenDlg:ICreatePrompt.....	1102
TFileOpenDlg:IDontAddToRecent.....	1102
TFileOpenDlg:IEnableSizing.....	1103
TFileOpenDlg:IFileMustExist.....	1103
TFileOpenDlg:IForceShowHidden.....	1103
TFileOpenDlg:IHideReadOnly.....	1103
TFileOpenDlg:INoChangeDir.....	1104
TFileOpenDlg:INoDereferenceLinks.....	1104
TFileOpenDlg:INoNetworkButton	1104
TFileOpenDlg:INoValidate.....	1104
TFileOpenDlg:IPathMustExist	1105
TFileOpenDlg:IReadOnly	1105
TFileOpenDlg:IShareAware.....	1105

TFileOpenDlg:nFilterIndex	1105
TFileOpenDlg:Methods	1106
TFileOpenDlg:Run	1106
TFileOpenDlg:Events	1106
TFileOpenDlg:OnClose	1106
TFileOpenDlg:OnHelp	1106
TFileOpenDlg:OnShow	1107
TFileOpenDlg:Functions	1107
FileOpenImageDlg	1107
FileOpenDlg	1108
TFileOperationDlg	1108
TFileOperationDlg:Properties	1108
TFileOperationDlg:aFrom	1109
TFileOperationDlg:aRenamed	1109
TFileOperationDlg:aTo	1109
TFileOperationDlg:cText	1110
TFileOperationDlg:IAborted	1110
TFileOperationDlg:IAAllowUndo	1110
TFileOperationDlg:IAutoRename	1110
TFileOperationDlg:IFilesOnly	1111
TFileOperationDlg:INoConfirmation	1111
TFileOperationDlg:INoConfirmMkDir	1111
TFileOperationDlg:INoCopySecurityAttrib	1111
TFileOperationDlg:INoErrorUI	1112
TFileOperationDlg:ISilent	1112
TFileOperationDlg:ISimpleProgress	1112
TFileOperationDlg:IWantMapping	1112
TFileOperationDlg:nAction	1113
TFileOperationDlg:Methods	1113
TFileOperationDlg:Run	1113
TFileSaveDlg	1114
TFileSaveDlg:Properties	1114
TFileSaveDlg:cDefaultExt	1115
TFileSaveDlg:cTitle	1115
TFileSaveDlg:INoTestFileCreate	1115
TFileSaveDlg:IOverWritePrompt	1115
TFileSaveDlg:Functions	1116
FileSaveImageDlg	1116
FileSaveDlg	1116
TFindDlg	1117
TFindDlg:Properties	1117
TFindDlg:cText	1117
TFindDlg:HelpID	1118
TFindDlg:ICenter	1118
TFindDlg:IDown	1118
TFindDlg:IMatchCase	1118
TFindDlg:INoCase	1119
TFindDlg:INoUpDown	1119
TFindDlg:INoWholeWord	1119
TFindDlg:IShowHelp	1119
TFindDlg:IWholeWord	1119
TFindDlg:Methods	1120
TFindDlg:Run	1120
TFindDlg:Events	1120

TFindDlg:OnFind.....	1120
TFindDlg:OnHelp.....	1121
TFindDlg:OnClose.....	1121
TFindFile.....	1121
TFindFile:Properties.....	1122
TFindFile:cCreationTime.....	1122
TFindFile:cFileMask.....	1123
TFindFile:cFileName.....	1123
TFindFile:cFullPath.....	1123
TFindFile:cLastAccessTime.....	1123
TFindFile:cLastWriteTime.....	1123
TFindFile:cShortFileName.....	1124
TFindFile:cShortPath.....	1124
TFindFile:dCreationDate.....	1124
TFindFile:dLastAccessDate.....	1124
TFindFile:dLastWriteDate.....	1125
TFindFile:IArchive.....	1125
TFindFile:ICompressed.....	1125
TFindFile:IDirectory.....	1125
TFindFile:IEncrypted.....	1125
TFindFile:IHidden.....	1126
TFindFile:INormal.....	1126
TFindFile:IOffline.....	1126
TFindFile:IReadOnly.....	1126
TFindFile:ISystem.....	1127
TFindFile:ITemporary.....	1127
TFindFile:nFileSize.....	1127
TFindFile:Methods.....	1127
TFindFile:Create.....	1127
TFindFile:Destroy.....	1128
TFindFile:Find.....	1128
TFindFile:FindNext.....	1128
TFindFile:Events.....	1128
TFindFile:OnStart.....	1129
TFindFile:OnFind.....	1129
TFindFile:OnStop.....	1129
TFormatDriveDlg.....	1130
TFormatDriveDlg:Properties.....	1130
TFormatDriveDlg:cDrive.....	1130
TFormatDriveDlg:cLabel.....	1131
TFormatDriveDlg:nType.....	1131
TFormatDriveDlg:nLastError.....	1131
TFormatDriveDlg:Methods.....	1131
TFormatDriveDlg:Run.....	1131
TMessageBox.....	1132
TMessageBox:Properties.....	1132
TMessageBox:HelpId.....	1133
TMessageBox:aButtons.....	1133
TMessageBox:cSoundFile.....	1133
TMessageBox:cText.....	1133
TMessageBox:cTitle.....	1134
TMessageBox:IAppModal.....	1134
TMessageBox:ICenter.....	1134
TMessageBox:IHelpButton.....	1134

TMessageBox:IRightText	1135
TMessageBox:ISetForeground	1135
TMessageBox:ITaskModal	1135
TMessageBox:ITopMost	1135
TMessageBox:nDefaultButton	1136
TMessageBox:nResult	1136
TMessageBox:nStyle	1136
TMessageBox:nTimeout	1136
TMessageBox:nTimeoutResult	1137
TMessageBox:olcon	1137
TMessageBox Methods	1137
TMessageBox:Beep	1137
TMessageBox:Run	1138
TMessageBox Events	1138
TMessageBox:OnShow	1138
TPageSetupDlg	1139
TPageSetupDlg:Properties	1139
TPageSetupDlg:aMargins	1140
TPageSetupDlg:aMinMargins	1140
TPageSetupDlg:aPaperSize	1140
TPageSetupDlg:HelpID	1141
TPageSetupDlg:ICenter	1141
TPageSetupDlg:IDefMinMargins	1141
TPageSetupDlg:INInches	1142
TPageSetupDlg:IMargins	1142
TPageSetupDlg:IMinMargins	1142
TPageSetupDlg:INoMargins	1142
TPageSetupDlg:INoNetwork	1143
TPageSetupDlg:INoOrientation	1143
TPageSetupDlg:INoPageDraw	1143
TPageSetupDlg:INoPaper	1143
TPageSetupDlg:INoPrinter	1144
TPageSetupDlg:INoWarning	1144
TPageSetupDlg:IShowHelp	1144
TPageSetupDlg:nLastError	1144
TPageSetupDlg:oParent	1145
TPageSetupDlg:Methods	1145
TPageSetupDlg:Run	1145
TPageSetupDlg:Events	1145
TPageSetupDlg:OnClose	1145
TPageSetupDlg:OnHelp	1146
TPageSetupDlg:OnShow	1146
TPickIconDlg	1147
TPickIconDlg:Properties	1147
TPickIconDlg:cFileName	1147
TPickIconDlg:nCount	1148
TPickIconDlg:nIndex	1148
TPickIconDlg:oParent	1148
TPickIconDlg:Methods	1148
TPickIconDlg:Run	1148
TPrintDlg	1149
TPrintDlg:Properties	1149
TPrintDlg:hDC	1150
TPrintDlg:HelpID	1150

TPrintDlg:IAIIPages	1150
TPrintDlg:ICenter	1151
TPrintDlg:ICollate.....	1151
TPrintDlg:IHidePrintToFile.....	1151
TPrintDlg:INoPageNums.....	1151
TPrintDlg:INoSelection.....	1152
TPrintDlg:INoWarning	1152
TPrintDlg:IPageNums	1152
TPrintDlg:IPrintSetup.....	1152
TPrintDlg:IPrintToFile.....	1152
TPrintDlg:IReturnDC	1153
TPrintDlg:IReturnIC	1153
TPrintDlg:ISelection.....	1153
TPrintDlg:IShowHelp.....	1153
TPrintDlg:nCopies.....	1154
TPrintDlg:nFromPage	1154
TPrintDlg:nLastError	1154
TPrintDlg:nMaxPage	1154
TPrintDlg:nMinPage.....	1154
TPrintDlg:nToPage.....	1155
TPrintDlg:oParent	1155
TPrintDlg:Methods.....	1155
TPrintDlg:Run	1155
TPrintDlg:Events.....	1156
TPrintDlg:OnClose.....	1156
TPrintDlg:OnHelp.....	1156
TPrintDlg:OnShow	1157
TProgressDlg	1157
TProgressDlg:Properties	1157
TProgressDlg:cMessage	1158
TProgressDlg:cMessageCancel	1158
TProgressDlg:cText.....	1158
TProgressDlg:cTitle	1158
TProgressDlg:IAutoTime	1159
TProgressDlg:ICenter	1159
TProgressDlg:ICompactPath	1159
TProgressDlg:IModal	1159
TProgressDlg:INoCancel	1159
TProgressDlg:INoMinimize	1160
TProgressDlg:INoTime	1160
TProgressDlg:INoProgress	1160
TProgressDlg:nCommonAVI.....	1160
TProgressDlg:nMax.....	1161
TProgressDlg:nResourceAVI.....	1161
TProgressDlg:nValue.....	1161
TProgressDlg:Methods.....	1161
TProgressDlg:Create	1162
TProgressDlg:Destroy	1162
TProgressDlg:Execute	1162
TProgressDlg:IsCancel	1162
TProgressDlg:New.....	1163
TProgressDlg:Reset	1163
TProgressDlg:Stop	1163
TProgressDlg:Events.....	1163

TProgressDlg:OnCancel.....	1164
TProgressDlg:OnClose.....	1164
TProgressDlg:OnReset.....	1164
TProgressDlg:OnShow.....	1164
TReplaceDlg.....	1165
TReplaceDlg:Properties.....	1165
TReplaceDlg:cReplace.....	1165
TReplaceDlg:cText.....	1166
TReplaceDlg:HelpID.....	1166
TReplaceDlg:ICenter.....	1166
TReplaceDlg:IMatchCase.....	1166
TReplaceDlg:INoCase.....	1166
TReplaceDlg:INoWholeWord.....	1167
TReplaceDlg:IShowHelp.....	1167
TReplaceDlg:IWholeWord.....	1167
TReplaceDlg:Methods.....	1167
TReplaceDlg:Run.....	1167
TReplaceDlg:Events.....	1168
TReplaceDlg:OnClose.....	1168
TReplaceDlg:OnHelp.....	1168
TReplaceDlg:OnFind.....	1169
TReplaceDlg:OnReplace.....	1169
TReplaceDlg:OnReplaceAll.....	1169
TRunFileDlg.....	1170
TRunFileDlg:Properties.....	1170
TRunFileDlg:cDirectory.....	1171
TRunFileDlg:cMessage.....	1171
TRunFileDlg:cTitle.....	1171
TRunFileDlg:INoBrowse.....	1171
TRunFileDlg:ICalcDirectory.....	1171
TRunFileDlg:INoDefault.....	1172
TRunFileDlg:INoLabel.....	1172
TRunFileDlg:INoSeparateMem.....	1172
TRunFileDlg:olcon.....	1172
TRunFileDlg:Methods.....	1173
TRunFileDlg:Run.....	1173
TRunFileDlg:Events.....	1173
TRunFileDlg:OnExecute.....	1173
Others.....	1174
TInplaceEdit.....	1174
TInplaceEdit:Methods.....	1174
TInplaceEdit:Create.....	1174
7 DataControls.....	1174
DataControls introduction.....	1174
Enhancements on Xailer 7.....	1179
Data Sources.....	1180
TDataSource.....	1180
TDataSource:Properties.....	1180
TDataSource:aDataSet.....	1181
TDataSource:cConnect.....	1181
TDataSource:IAbortOnErrors.....	1181
TDataSource:IConnected.....	1182
TDataSource:IDisplayErrors.....	1182
TDataSource:nTimeOut.....	1182

TDataSource:Methods	1182
TDataSource:AddDataSet	1183
TDataSource:BuildSQLSt.....	1183
TDataSource:CloseProcess	1184
TDataSource:Create	1184
TDataSource:DelDataSet	1184
TDataSource:DelErrors.....	1184
TDataSource:DelProcess	1185
TDataSource:GetErrors	1185
TDataSource:IsError	1185
TDataSource:LastError	1185
TDataSource:NewProcess.....	1186
TDataSource:OpenProcess.....	1186
TDataSource:QueryMemDataset.....	1186
TDataSource:RunProc.....	1187
TDataSource:RunProcDirect	1187
TDataSource:ShowErrorList	1187
TDataSource:SqlInsert.....	1187
TDataSource:SqlUpdate.....	1188
TDataSource:Events.....	1188
TDataSource:OnConnect.....	1188
TDataSource:OnConnected.....	1189
TDataSource:OnCreate	1189
TDataSource:OnDisconnect	1189
TDataSource:OnDisconnected.....	1189
TDataSource:OnError.....	1189
TDataSource:OnExecute.....	1190
TDbfDataSource	1190
TDbfDataSource:Properties	1191
TDbfDataSource:cConnect.....	1191
TDbfDataSource:cDriver.....	1191
TDbfDataSource:cPassword.....	1192
TDbfDataSource:IOptimize	1192
TDbfDataSource:nLockScheme	1192
TDbfDataSource:nTimeOut.....	1193
TDbfDataSource:nMemoSize	1193
TDbfDataSource:nMemoType.....	1193
TDbfDataSource:Methods	1193
TDbfDataSource:CommitAll.....	1194
TDbfDataSource:CreateTable	1194
TDbfDataSource:DefDataExtension.....	1194
TDbfDataSource:DefIdxExtension	1195
TDbfDataSource:DefMemoExtension	1195
TDbfDataSource:DelTable.....	1195
TDbfDataSource:NewDataSet.....	1195
TNtxDataSource.....	1196
TNtxDataSource:Properties.....	1196
TNtxDataSet:cDriver	1196
TNtxDataSource:Methods	1196
TNtxDataSource:DefIdxExtension	1197
TCdxDataSource.....	1197
TCdxDataSource:Properties	1197
TCdxDataSet:cDriver	1197
TCdxDataSource:IAutoOpen	1198

TCdxDataSource:nTableType	1198
TCdxDataSource:Methods	1198
TCdxDataSource:DefIdxExtension	1198
TAdsDataSource	1198
TAdsDataSource:Properties	1199
TAdsDataSource:cConnect	1200
TAdsDataSource:cDriver	1200
TAdsDataSource:cPassword	1200
TAdsDataSource:cUser	1201
TAdsDataSource:hConnect	1201
TAdsDataSource:IAdsLocking	1201
TAdsDataSource:IRightsCheck	1201
TAdsDataSource:IUseDictionary	1202
TAdsDataSource:nCharType	1202
TAdsDataSource:nConnectionFlags	1202
TAdsDataSource:nFileType	1203
TAdsDataSource:nServerType	1203
TAdsDataSource:Methods	1203
TAdsDataSource:BeginTrans	1204
TAdsDataSource:CommitTrans	1204
TAdsDataSource:DefDataExtension	1204
TAdsDataSource:DefIdxExtension	1205
TAdsDataSource:QueryArray	1205
TAdsDataSource:QueryRow	1205
TAdsDataSource:QueryValue	1206
TAdsDataSource:RollBackTrans	1206
TADODDataSource	1206
TADODDataSource:Properties	1207
TADODDataSource:cConnect	1207
TADODDataSource:cPassword	1207
TADODDataSource:cUser	1208
TADODDataSource:nAffectedRows	1208
TADODDataSource:IConnectAsync	1208
TADODDataSource:IPromptIfError	1208
TADODDataSource:nAttributes	1209
TADODDataSource:nConnTimeOut	1209
TADODDataSource:nCursorLocation	1209
TADODDataSource:nIsolationLevel	1210
TADODDataSource:nMode	1211
TADODDataSource:oConnection	1212
TADODDataSource:Methods	1212
TADODDataSource:BeginTrans	1212
TADODDataSource:Cancel	1213
TADODDataSource:CommitTrans	1213
TADODDataSource:Connect	1213
TAdoDataSource:ConnectTag	1213
TADODDataSource:DelTable	1214
TADODDataSource:Disconnect	1214
TADODDataSource:Execute	1214
TADODDataSource:File	1215
TADODDataSource:GetCatalogs	1215
TADODDataSource:GetTables	1215
TADODDataSource:Properties	1216
TADODDataSource:Query	1216

TADODataSource:QueryArray.....	1216
TADODataSource:QueryReport.....	1217
TADODataSource:QueryRow.....	1217
TADODataSource:QueryValue.....	1217
TADODataSource:RollBackTrans.....	1218
TADODataSource:State.....	1218
TADODataSource:Table.....	1218
TADODataSource:Version.....	1219
TMySQLDataSource.....	1219
TMySQLDataSource:Propiedades.....	1219
TMySQLDataSource:cDatabase.....	1220
TMySQLDataSource:cHost.....	1220
TMySQLDataSource:cPassword.....	1220
TMySQLDataSource:cUser.....	1220
TMySQLDataSource:lAutoReconnect.....	1220
TMySQLDataSource:nPort.....	1221
TMySQLDataSource:nTimeOut.....	1221
TMySQLDataSource:nAffectedRows.....	1221
TMySQLDataSource:Metodos.....	1221
TMySQLDataSource:BeginTrans.....	1222
TMySQLDataSource:CommitTrans.....	1222
TMySQLDataSource:Connect.....	1222
TMySQLDataSource:CreateTable.....	1223
TMySQLDataSource:DelTable.....	1223
TMySQLDataSource:Disconnect.....	1223
TMySQLDataSource:Execute.....	1224
TMySQLDataSource:File.....	1224
TMySQLDataSource:GetCatalogs.....	1224
TMySQLDataSource:GetTables.....	1224
TMySQLDataSource:Query.....	1225
TMySQLDataSource:QueryArray.....	1225
TMySQLDataSource:QueryArrayHash.....	1225
TMySQLDataSource:QueryReport.....	1226
TMySQLDataSource:QueryRow.....	1226
TMySQLDataSource:QueryRowHash.....	1226
TMySQLDataSource:QueryValue.....	1227
TMySQLDataSource:RollBackTrans.....	1227
TMySQLDataSource:SqlNextResult.....	1227
TMySQLDataSource:Table.....	1228
TMariaDBDataSource.....	1228
TMariaDBDataSource:Propiedades.....	1229
TMariaDBDataSource:cDatabase.....	1229
TMariaDBDataSource:cHost.....	1229
TMariaDBDataSource:cPassword.....	1229
TMariaDBDataSource:cUser.....	1230
TMariaDBDataSource:lAutoReconnect.....	1230
TMariaDBDataSource:nPort.....	1230
TMariaDBDataSource:nTimeOut.....	1230
TMariaDBDataSource:nAffectedRows.....	1230
TMariaDBDataSource:Metodos.....	1231
TMariaDBDataSource:BeginTrans.....	1231
TMariaDBDataSource:CommitTrans.....	1231
TMariaDBDataSource:Connect.....	1232
TMariaDBDataSource:CreateTable.....	1232

TMariaDBDataSource:DelTable.....	1232
TMariaDBDataSource:Disconnect.....	1233
TMariaDBDataSource:Execute.....	1233
TMariaDBDataSource:File.....	1233
TMariaDBDataSource:GetCatalogs.....	1234
TMariaDBDataSource:GetTables.....	1234
TMariaDBDataSource:Query.....	1234
TMariaDBDataSource:QueryArray.....	1234
TMariaDBDataSource:QueryArrayHash.....	1235
TMariaDBDataSource:QueryReport.....	1235
TMariaDBDataSource:QueryRow.....	1235
TMariaDBDataSource:QueryRowHash.....	1236
TMariaDBDataSource:QueryValue.....	1236
TMariaDBDataSource:RollBackTrans.....	1237
TMariaDBDataSource:SqlNextResult.....	1237
TMariaDBDataSource:Table.....	1237
TSQLiteDataSource.....	1237
TSQLiteDataSource:Properties.....	1239
TSQLiteDataSource:cConnect.....	1239
TSQLiteDataSource:cPassword.....	1239
TSQLiteDataSource:IDateAsString.....	1239
TSQLiteDataSource:IDoubleQuotes.....	1241
TSQLiteDataSource:IReadToCache.....	1241
TSQLiteDataSource:nAffectedRows.....	1241
TSQLiteDataSource:Methods.....	1241
TSQLiteDataSource:BeginTrans.....	1242
TSQLiteDataSource:CommitTrans.....	1242
TSQLiteDataSource:Connect.....	1242
TSQLiteDataSource:DelTable.....	1243
TSQLiteDataSource:Disconnect.....	1243
TSQLiteDataSource:Encrypt.....	1243
TSQLiteDataSource:Execute.....	1243
TSQLiteDataSource:File.....	1244
TSQLiteDataSource:GetTables.....	1244
TSQLiteDataSource:Query.....	1244
TSQLiteDataSource:QueryArray.....	1245
TSQLiteDataSource:QueryArrayHash.....	1245
TSQLiteDataSource:QueryReport.....	1245
TSQLiteDataSource:QueryRow.....	1246
TSQLiteDataSource:QueryRowHash.....	1246
TSQLiteDataSource:QueryValue.....	1246
TSQLiteDataSource:RollBackTrans.....	1247
TSQLiteDataSource:Table.....	1247
TWebDataSource.....	1247
TWebDataSource:Properties.....	1249
TWebDataSource:cConnect.....	1249
TWebDataSource:cCryptKey.....	1249
TWebDataSource:cDatabase.....	1250
TWebDataSource:cLastError.....	1250
TWebDataSource:cPhpModule.....	1250
TWebDataSource:ITinyIntAsBoolean.....	1250
TWebDataSource:nDBType.....	1251
TWebDataSource:nAffectedRows.....	1251
TWebDataSource:nInsertId.....	1251

TWebDataSource:nPort.....	1251
TWebDataSource:nSendTimeOut.....	1252
TWebDataSource:Methods.....	1252
TWebDataSource:BeginTrans.....	1253
TWebDataSource:BulkExecute.....	1253
TWebDataSource:CommitTrans.....	1253
TWebDataSource:Connect.....	1254
TWebDataSource:CurrentDir.....	1254
TWebDataSource:Disconnect.....	1254
TWebDataSource>DeleteFile.....	1254
TWebDataSource:Execute.....	1255
TWebDataSource:File.....	1255
TWebDataSource:ForceDir.....	1255
TWebDataSource:GetCatalogs.....	1256
TWebDataSource:GetTables.....	1256
TWebDataSource:Query.....	1256
TWebDataSource:IsDir.....	1256
TWebDataSource:QueryArray.....	1257
TWebDataSource:QueryArrayHash.....	1257
TWebDataSource:QueryReport.....	1257
TWebDataSource:QueryRow.....	1258
TWebDataSource:QueryRowHash.....	1258
TWebDataSource:QueryValue.....	1258
TWebDataSource:Mkdir.....	1259
TWebDataSource:MyIP.....	1259
TWebDataSource:MyLocation.....	1259
TWebDataSource:RollBackTrans.....	1259
TWebDataSource:SendMail.....	1260
TWebDataSource:Table.....	1260
TWebDataSource:UpdateFile.....	1260
TWebDataSource:Events.....	1261
TWebDataSource:OnValidate.....	1261
TOdbcDataSource.....	1261
TOdbcDataSource:Properties.....	1262
TOdbcDataSource:cConnect.....	1262
TOdbcDataSource:cDSN.....	1262
TOdbcDataSource:cPassword.....	1263
TOdbcDataSource:cUser.....	1263
TOdbcDataSource:lAutoCommit.....	1263
TOdbcDataSource:lDateAsString.....	1263
TOdbcDataSource:nAffectedRecords.....	1264
TOdbcDataSource:Methods.....	1264
TOdbcDataSource:CommitAll.....	1264
TOdbcDataSource:DelTable.....	1264
TOdbcDataSource:Execute.....	1265
TOdbcDataSource:File.....	1265
TOdbcDataSource:GetTables.....	1265
TOdbcDataSource:Query.....	1266
TOdbcDataSource:QueryArray.....	1266
TOdbcDataSource:QueryRow.....	1266
TOdbcDataSource:QueryValue.....	1267
TOdbcDataSource:RollBackAll.....	1267
Data Sets.....	1267
TDataField.....	1267

TDataFieldProperties	1268
TDataField:Buffer	1268
TDataField:cDisplayName	1268
TDataField:cFullName	1269
TDataField:cName	1269
TDataField:cType	1269
TDataField:cTable	1269
TDataField:DefValue	1270
TDataField:InitValue	1270
TDataField:AllowsNull	1270
TDataField:IEditable	1270
TDataField:IModified	1270
TDataField:IPrimaryKey	1271
TDataField:ISetNullIfEmpty	1271
TDataField:nDec	1271
TDataField:nIndex	1271
TDataField:nLen	1272
TDataField:nSQLDec	1272
TDataField:nSQLLen	1272
TDataField:nSQLType	1272
TDataField:oDataSet	1273
TDataField:Value	1273
TDataFieldMethods	1273
TDataField:Blank	1273
TDataField:FieldGet	1274
TDataField:FieldPut	1274
TDataField:HasChanged	1274
TDataField:Load	1274
TDataField:Valtype	1275
TDataField:VarGet	1275
TDataField:VarPut	1275
TDataField:Events	1276
TDataField:OnGetValue	1276
TDataField:OnSetValue	1276
TDataSet	1276
TDataSetProperties	1277
TDataSet:aFields	1277
TDataSet:aLinkedControls	1277
TDataSet:aUserFields	1278
TDataSet:cLastError	1278
TDataSet:cName	1278
TDataSet:cProcess	1279
TDataSet:IDisplayErrors	1279
TDataSet:IOpen	1279
TDataSet:IReadOnly	1280
TDataSet:IUpdLinked	1280
TDataSet:nCursorLocation	1280
TDataSet:nCursorType	1281
TDataSet:nLastError	1281
TDataSet:nLockType	1282
TDataSet:nMaxRecords	1282
TDataSet:oDataSource	1283
TDataSet:oRecords	1283
TDataSet:Methods	1283

TDataSet.AddCalcField	1285
TDataSet.AddLinkedControl	1286
TDataSet.AddNew	1286
TDataSet.AddUserVar	1287
TDataSet.Assign	1287
TDataSet.Blank	1287
TDataSet.Bof	1288
TDataSet.Cancel	1288
TDataSet.CanEdit	1288
TDataSet.ClearFilter	1289
TDataSet.Close	1289
TDataSet.Clone	1289
TDataSet.Commit	1290
TDataSet.Create	1290
TDataSet.DelError	1290
TDataSet.Delete	1291
TDataSet.DelLinkedControl	1291
TDataSet.Edit	1291
TDataSet.End	1292
TDataSet.Eof	1292
TDataSet.FastAddNew	1292
TDataSet.FastEdit	1293
TDataSet.FastUpdate	1294
TDataSet.FieldCount	1294
TDataSet.FieldDec	1295
TDataSet.FieldDisplayName	1295
TDataSet.FieldExist	1295
TDataSet.FieldGet	1295
TDataSet.FieldGetByName	1296
TDataSet.FieldLen	1296
TDataSet.FieldName	1296
TDataSet.FieldNames	1297
TDataSet.FieldPos	1297
TDataSet.FieldPut	1297
TDataSet.FieldSQLType	1297
TDataSet.FieldType	1298
TDataSet.Filter	1298
TDataSet.GetRecord	1299
TDataSet.GetRow	1299
TDataSet.GetRowHash	1300
TDataSet.GetRows	1300
TDataSet.GetRowsHash	1300
TDataSet.GoBottom	1301
TDataSet.Goto	1301
TDataSet.GoTop	1301
TDataSet.InitFrom	1302
TDataSet.IsReadOnly	1302
TDataSet.KeyCount	1302
TDataSet.KeyGoto	1303
TDataSet.KeyNo	1303
TDataSet.Load	1304
TDataSet.Locate	1304
TDataSet.IOnAppend	1304
TDataSet.IOnEdit	1305

TDataSet.Modified	1305
TDataSet.nSkip	1305
TDataSet.oFieldByName	1305
TDataSet.Open	1306
TDataSet.RecCount	1306
TDataSet.RecNo	1307
TDataSet.Refresh	1307
TDataSet.RefreshCurrent	1307
TDataSet.ReleaseState	1307
TDataSet.RestoreState	1308
TDataSet.SaveFrom	1308
TDataSet.SaveState	1308
TDataSet.SetFilter	1309
TDataSet.Skip	1309
TDataSet.Sort	1310
TDataSet.Update	1310
TDataSet.UpdateControls	1310
TDataSet.VarGet	1311
TDataSet.VarPut	1311
TDataSet.Events	1311
TDataSet.OnAddNew	1312
TDataSet.OnCancel	1312
TDataSet.OnClose	1312
TDataSet.OnCreate	1312
TDataSet.OnEdit	1313
TDataSet.OnOpen	1313
TDataSet.OnPostChange	1313
TDataSet.OnPostClose	1314
TDataSet.OnPostDelete	1315
TDataSet.OnPostOpen	1315
TDataSet.OnPostSave	1315
TDataSet.OnPreChange	1315
TDataSet.OnPreDelete	1316
TDataSet.OnPreSave	1317
TDataSet.OnUpdate	1317
TDbfDataSet	1317
TDbfDataSetProperties	1317
TDbfDataSet:aldxFiles	1318
TDbfDataSet:cAlias	1318
TDbfDataSet:cName	1318
TDbfDataSet:cPassword	1319
TDbfDataSet:IKKeyExact	1319
TDbfDataSet:INew	1319
TDbfDataSet:IReUseDel	1320
TDbfDataSet:IShared	1320
TDbfDataSet:Methods	1321
TDbfDataSet:AddIdxFile	1322
TDbfDataSet:AddTag	1322
TDbfDataSet:AddTempTag	1323
TDbfDataSet:Alias	1323
TDbfDataSet:Append	1324
TDbfDataSet:Area	1324
TDbfDataSet:ClearRelations	1324
TDbfDataSet:ClearScope	1325

TDbfDataSet.Clone	1325
TDbfDataSet.CompleteSeek	1325
TDbfDataSet.Continue	1326
TDbfDataSet.Delete	1326
TDbfDataSet.Deleted	1327
TDbfDataSet.DelTag	1327
TDbfDataSet.DelTempTag	1328
TDbfDataSet.FillLock	1328
TDbfDataSet.FillUnLock	1328
TDbfDataSet.ForcedSeek	1329
TDbfDataSet.Found	1329
TDbfDataSet.GetScope	1329
TDbfDataSet.GetTags	1330
TDbfDataSet.IsRecordLocked	1330
TDbfDataSet.Locate	1330
TDbfDataSet.OrdBagName	1331
TDbfDataSet.OrdCount	1331
TDbfDataSet.OrdFor	1331
TDbfDataSet.OrdKey	1332
TDbfDataSet.OrdKeyVal	1332
TDbfDataSet.OrdName	1332
TDbfDataSet.OrdNumber	1333
TDbfDataSet.OrdSetFocus	1333
TDbfDataSet.Pack	1333
TDbfDataSet.Recall	1334
TDbfDataSet.RecLock	1334
TDbfDataSet.RecUnLock	1335
TDbfDataSet.Reindex	1335
TDbfDataSet.ReleaseState	1336
TDbfDataSet.RestoreState	1336
TDbfDataSet.Save	1336
TDbfDataSet.SaveState	1336
TDbfDataSet.SaveToArray	1337
TDbfDataSet.Seek	1338
TDbfDataSet.Select	1338
TDbfDataSet.SetRelation	1339
TDbfDataSet.SetScope	1339
TDbfDataSet.Sort	1340
TDbfDataSet.Struct	1340
TDbfDataSet.Zap	1340
TDbfDataSet.Events	1341
TDbfDataSet.OnPostAppend	1341
TDbfDataSet.OnPreAppend	1341
TMemDataSet	1342
TMemDataSet.Properties	1342
TMemDataSet.aData	1342
TMemDataSet.aWork	1342
TMemDataSet.IAbortOnErrors	1343
TMemDataSet.Methods	1343
TMemDataSet.AddField	1343
TMemDataSet.Clone	1344
TMemDataSet.Filter	1345
TMemDataSet.GetRow	1345
TMemDataSet.GetStructFrom	1345

TMemDataSet:Open.....	1346
TMemDataSet:OrgRecno.....	1346
TMemDataSet:ReleaseState.....	1346
TMemDataSet:RestoreState.....	1347
TMemDataSet:SaveState.....	1347
TMemDataSet:Search.....	1348
TMemDataSet:Sort.....	1348
TMemDataSet:SqlInsert.....	1349
TMemDataSet:SqlUpdate.....	1349
TMemDataSet:Zap.....	1349
TObdcDataSet.....	1350
TObdcDataSetProperties.....	1350
TObdcDataSet:cSelect.....	1350
TObdcDataSet:IAsynFetch.....	1351
TObdcDataSet:INoCompare.....	1351
TObdcDataSet:Methods.....	1351
TObdcDataSet:CanEdit.....	1352
TObdcDataSet:Complete.....	1352
TObdcDataSet:FieldIsNull.....	1352
TObdcDataSet:GetPrimaryKeys.....	1353
TObdcDataSet:IsComplete.....	1353
TObdcDataSet:RecordHasChanged.....	1353
TObdcDataSet:SetPrimaryKeys.....	1353
TObdcDataSet:Events.....	1354
TObdcDataSet:OnComplete.....	1354
Stored procedures.....	1354
TSqlProc.....	1355
TSqlProc:Properties.....	1355
TSqlProc:aSqlParams.....	1355
TSqlProc:aResults.....	1355
TSqlProc:cName.....	1356
TSqlProc:IDisplayErrors.....	1356
TSqlProc:oDataSource.....	1356
TSqlProc:IOpen.....	1356
TSqlProc:Methods.....	1357
TSqlProc:AddParam.....	1357
TSqlProc:Close.....	1357
TSqlProc:Create.....	1358
TSqlProc:Open.....	1358
TSqlProc:OutputParams.....	1358
TSqlProc:ParamByName.....	1358
TSqlProc:ReturnValue.....	1359
TSqlProc:Events.....	1359
TSqlProc:OnCreate.....	1359
TSqlProc:OnClose.....	1359
TSqlProc:OnOpen.....	1359
TSqlProc:OnPostClose.....	1360
TSqlProc:OnPostOpen.....	1360
TSqlParam.....	1360
TSqlParam:Properties.....	1360
TSqlParam:cName.....	1360
TSqlParam:Initial.....	1361
TSqlParam:nDirection.....	1361
TSqlParam:nSize.....	1361

TSqlParam:nType	1361
TSqlParam:Value	1362
TSqlParam:oSqlProc	1362
TSqlParams:Methods	1363
TSqlParam:Create	1363
TSqlParam:IsInputParam	1363
TSqlParam:IsOutputParam	1363
TSqlProcs	1364
TSQLQuery	1365
TSQLQueryProperties	1366
TSQLQuery:aSQLParams	1366
TSQLQuery:cSelect	1366
TSQLTable	1366
TSQLTableProperties	1367
TSQLTable:cTableName	1367
Controls	1367
TDBBrowse	1367
TDBBrowse:Properties	1368
TDBBrowse:cMsgNoFilter	1368
TDBBrowse:IAutoOrder	1368
TDBBrowse:IAutoSeek	1368
TDBBrowse:oDataSet	1369
TDBBrowseMod	1369
TDBBrowseMod:Propiedades	1370
TDBBrowseMod:oDataSet	1370
TDBBrowseMod:Methods	1370
TdbBrowseMod:DataCol	1370
TDBBrwColMod	1370
TDBBrwColMod:Properties	1371
TDBBrwColMod:oDataField	1371
TDBBrwColMod:cDataField	1371
TDBCardBox	1371
TDBCardBox:Properties	1372
TDBCardBox:oDataSet	1372
TDBCheckBox	1373
TDBCheckBox:Properties	1373
TDBCheckBox:IAutoSave	1373
TDBCheckBox:IEditable	1374
TDBCheckBox:oDataField	1374
TDBCheckBox:oDataSet	1374
TDBCheckBox:Methods	1375
TDBCheckBox:Value	1375
TDBCheckBoxMod	1375
TDBCheckBoxMod:Properties	1375
TDBCheckBoxMod:IAutoSave	1375
TDBCheckBoxMod:IEditable	1376
TDBCheckBoxMod:oDataField	1376
TDBCheckBoxMod:oDataSet	1376
TDBCheckBoxMod:Methods	1377
TDBCheckBoxMod:Value	1377
TDBComboBox	1377
TDBComboBox:Properties	1377
TDBComboBox:altItemsBound	1378
TDBComboBox:IAutoSave	1378

TDBComboBox: IEditable.....	1378
TDBComboBox: nDataType.....	1378
TDBComboBox: oDataField.....	1379
TDBCombokBox: oDataSet.....	1379
TDBComboBox: Methods.....	1380
TDBCombokBox: Value.....	1380
TDBComboBoxMod.....	1380
TDBComboBox: Propiedades.....	1380
TDBComboBox: altemsBound.....	1381
TDBComboBox: cHint.....	1381
TDBComboBox: IAutoSave.....	1381
TDBComboBox: IEditable.....	1382
TDBComboBox: nDataType.....	1382
TDBComboBox: oDataField.....	1382
TDBComboBox: oDataSet.....	1383
TDBComboBox: Methods.....	1383
TDBComboBox: Value.....	1383
TDBDateEdit.....	1384
TDBDateEdit: Properties.....	1384
TDBDateEdit: IAutoSave.....	1384
TDBDateEdit: IEditable.....	1384
TDBDateEdit: oDataField.....	1385
TDBDateEdit: oDataSet.....	1385
TDBDateEditMod.....	1385
TDBDateEditMod: Properties.....	1386
TDBDateEditMod: IAutoSave.....	1386
TDBDateEditMod: IEditable.....	1386
TDBDateEditMod: oDataField.....	1386
TDBDateEditMod: oDataSet.....	1387
TDBDatePicker.....	1387
TDBDatePicker: Properties.....	1387
TDBDatePicker: IAutoSave.....	1388
TDBDatePicker: IEditable.....	1388
TDBDatePicker: oDataField.....	1388
TDBDatePicker: oDataSet.....	1389
TDBDatePicker: Methods.....	1389
TDBDataPicker: Value.....	1389
TDBEdit.....	1389
TDBEdit: Properties.....	1390
TDBEdit: IAutoSave.....	1390
TDBEdit: IEditable.....	1390
TDBEdit: oDataField.....	1390
TDBEdit: oDataSet.....	1391
TDBEditBtn.....	1391
TDBEditBtn: Properties.....	1391
TDBEditBtn: IAutoSave.....	1391
TDBEditBtn: IEditable.....	1392
TDBEditBtn: oDataField.....	1392
TDBEditBtn: oDataSet.....	1392
TDBEditMod.....	1393
TDBEditMod: Properties.....	1393
TDBEditMod: IAutoSave.....	1393
TDBEditMod: IEditable.....	1393
TDBEditMod: oDataField.....	1394

TDBEditMod:oDataSet	1394
TDBHyperLink	1394
TDBHyperLink:Properties	1395
TDBHyperLink:oDataField	1395
TDBHyperLink:oDataSet	1395
TDBHyperLink:Methods	1395
TDBHyperLink:Refresh	1396
TDBLabel	1396
TDBLabel:Properties	1396
TDBLabel:IAutoSize	1397
TDBLabel:oDataField	1397
TDBLabel:oDataSet	1397
TDBLabel:Methods	1397
TDBLabel:Refresh	1398
TDBListBox	1398
TDBListBox:Properties	1398
TDBListBox:altemsBound	1398
TDBListBox:IAutoSave	1399
TDBListBox:IEditable	1399
TDBListBox:nDataType	1399
TDBListBox:oDataField	1400
TDBListBox:oDataSet	1400
TDBListBox:Methods	1400
TDBListBox:Value	1401
TDBListBoxMod	1401
TDBListBoxMod:Properties	1401
TDBListBoxMod:altemsBound	1401
TDBListBoxMod:IAutoSave	1402
TDBListBoxMod:IEditable	1402
TDBListBoxMod:nDataType	1402
TDBListBoxMod:oDataField	1403
TDBListBoxMod:oDataSet	1403
TDBMaskEdit	1403
TDBMaskEdit:Properties	1404
TDBMaskEdit:IAutoSave	1404
TDBMaskEdit:IEditable	1404
TDBMaskEdit:oDataField	1404
TDBMaskEdit:oDataSet	1405
TDBMemo	1405
TDBMemo:Properties	1405
TDBMemo:IAutoSave	1406
TDBMemo:IEditable	1406
TDBMemo:oDataField	1406
TDBMemo:oDataSet	1407
TDBMemoMod	1407
TDBMemoMod:Properties	1407
TDBMemoMod:IAutoSave	1408
TDBMemoMod:IEditable	1408
TDBMemoMod:oDataField	1408
TDBMemoMod:oDataSet	1409
TDBNavigator	1409
TDBNavigator:Properties	1409
TDBNavigator:aTooltips	1410
TDBNavigator:IBtnCancel	1410

TDBNavigator:IBtnDelete	1410
TDBNavigator:IBtnEdit	1410
TDBNavigator:IBtnFirst	1411
TDBNavigator:IBtnInsert.....	1411
TDBNavigator:IBtnLast.....	1411
TDBNavigator:IBtnNext.....	1411
TDBNavigator:IBtnPrevious.....	1412
TDBNavigator:IBtnRefresh	1412
TDBNavigator:IBtnUpdate	1412
TDBNavigator:nAlign.....	1412
TDBNavigator:oDataSet.....	1413
TDBNavigator:Events.....	1413
TDBNavigator:OnClickCancel.....	1413
TDBNavigator:OnClickDelete.....	1414
TDBNavigator:OnClickEdit.....	1414
TDBNavigator:OnClickFirst.....	1414
TDBNavigator:OnClickInsert	1414
TDBNavigator:OnClickLast.....	1415
TDBNavigator:OnClickNext.....	1415
TDBNavigator:OnClickPrevious.....	1415
TDBNavigator:OnClickRefresh.....	1415
TDBNavigator:OnClickUpdate.....	1416
TDBRadioMenu.....	1416
TDBRadioMenu:Properties.....	1416
TDBRadioMenu:IAutoSave.....	1416
TDBRadioMenu:IEditable.....	1417
TDBRadioMenu:oDataField	1417
TDBRadioMenu:oDataSet.....	1417
TDBRadioMenu:Methods	1418
TDBRadioMenu:Value.....	1418
TDBTimePicker	1418
TDBTimePicker:Properties	1418
TDBTimePicker:IAutoSave.....	1418
TDBTimePicker:IEditable.....	1419
TDBTimePicker:oDataField	1419
TDBTimePicker:oDataSet.....	1419
TDBTimePicker:Methods	1420
TDBTimePicker:Value	1420
TDBSwitch	1420
TDBSwitch:Properties	1421
TDBSwitch:IAutoSave.....	1421
TDBSwitch:IEditable.....	1421
TDBSwitch:oDataField	1421
TDBSwitch:oDataSet.....	1422
TDBSwitch:Methods.....	1422
TDBSwitch:Value.....	1422
8 Other Classes and Objects	1422
TBarcode	1422
TBarcode:Properties	1423
TBarcode:cText	1423
TBarcode:nLastError	1423
TBarcode:nType	1423
TBarcode:Methods.....	1424
TBarcode:AsString	1424

TBarcode: GetErrorDescription	1425
TBarcode: GetPicture	1425
TBarcode: Print	1425
TExStruct	1425
TExStruct: Methods	1426
TExStruct: AddMember	1427
TExStruct: Clone	1427
TExStruct: GetDefaults	1427
TExStruct: GetKeys	1428
TExStruct: GetValue	1428
TExStruct: GetValues	1428
TExStruct: GetName	1428
TExStruct: GetMember	1429
TExStruct: Len	1429
TExStruct: MemberList	1429
TExStruct: Modified	1429
TExStruct: New	1430
TExStruct: Reset	1430
TExStruct: SetValue	1430
TExStruct: SetValues	1430
TExStruct: ToJson	1431
TFileName	1431
TFileName: Properties	1431
TFileName: cFileName	1432
TFileName: Methods	1432
TFileName: Drive	1432
TFileName: Extension	1433
TFileName: FullName	1433
TFileName: HasDrive	1433
TFileName: HasExtension	1433
TFileName: HasPath	1434
TFileName: New	1434
TFileName: Path	1434
TFileName: SetExtension	1434
TFileName: SetName	1435
TFileName: SetPath	1435
TFileName: ShortName	1435
TFileName: Unique	1435
TFileName: WithExtension	1436
TFileName: WithName	1436
TFileName: WithPath	1436
TRecentList	1437
TRecentList: Properties	1437
TRecentList: altems	1437
TRecentList: cIniFile	1437
TRecentList: cIniSection	1438
TRecentList: IAutoRemove	1438
TRecentList: IAutoSave	1438
TRecentList: nMaxItems	1438
TRecentList: Methods	1438
TRecentList: Add	1439
TRecentList: Delete	1439
TRecentList: DeleteItems	1439
TRecentList: Read	1440

TRecentList:Refresh.....	1440
TRecentList:SetIniFile.....	1440
TRecentList:SetItems.....	1440
TRecentList:Write.....	1441
TRecentListMenu	1441
TRecentListMenu:Properties.....	1441
TRecentListMenu:cMessage.....	1442
TRecentListMenu:IFullPath.....	1442
TRecentListMenu:INumerate.....	1442
TRecentListMenu:IPrefix.....	1442
TRecentListMenu:ISeparator.....	1442
TRecentListMenu:ISubMenu.....	1443
TRecentListMenu:IText.....	1443
TRecentListMenu:cTextEmpty.....	1443
TRecentListMenu:nMaxWidth.....	1443
TRecentListMenu:nMenu.....	1444
TRecentListMenu:nMenuPos.....	1444
TRecentListMenu:oMenu.....	1444
TRecentListMenu:Methods.....	1444
TRecentListMenu:Create.....	1444
TRecentListMenu:Refresh.....	1445
TRecentListMenu:Events.....	1445
TRecentListMenu:OnClick.....	1445
TUnzipFile	1445
TUnzipFile:Properties.....	1446
TUnzipFile:aFileMask.....	1446
TUnzipFile:aFiles.....	1447
TUnzipFile:cDirectory.....	1447
TUnzipFile:cFileName.....	1447
TUnzipFile:cLastError.....	1447
TUnzipFile:cPassword.....	1447
TUnzipFile:IIncludePath.....	1448
TUnzipFile:nBufferSize.....	1448
TUnzipFile:nCount.....	1448
TUnzipFile:nExtractBufferSize.....	1448
TUnzipFile:nLastError.....	1449
TUnzipFile:nReadBufferSize.....	1449
TUnzipFile:nWriteBufferSize.....	1449
TUnzipFile:Methods.....	1449
TUnzipFile:FilesCount.....	1450
TUnzipFile>DeleteFiles.....	1450
TUnzipFile:FilesInfo.....	1450
TUnzipFile:FilesSize.....	1451
TUnzipFile:HasPassword.....	1451
TUnzipFile:Run.....	1451
TUnzipFile:Events.....	1452
TUnzipFile:OnChangeDisk.....	1452
TUnzipFile:OnChangeFile.....	1452
TUnzipFile:OnProgress.....	1452
TWiaScan	1453
TWiaScan:Properties.....	1453
TWiaScan:cDeviceId.....	1453
TWiaScan:nImageMode.....	1454
TWiaScan:nResolution.....	1454

TWiaScan:Methods	1454
TWiaScan:DeviceInfo.....	1454
TWiaScan:Devices	1455
TWiaScan:ImageSize.....	1455
TWiaScan:Scan	1455
TWiaScan:Events.....	1456
TWiaScan:OnScan.....	1456
TZipFile	1456
TZipFile:Properties	1456
TZipFile:aFileMask	1457
TZipFile:aFiles	1457
TZipFile:cComment.....	1457
TZipFile:cFileName.....	1458
TZipFile:cLastError	1458
TZipFile:cPassword.....	1458
TZipFile:nBufferSize.....	1458
TZipFile:nCompressionLevel.....	1458
TZipFile:nCount	1459
TZipFile:nExtractBuferSize.....	1459
TZipFile:nLastError	1459
TZipFile:nReadBufferSize	1460
TZipFile:nSpanMode.....	1460
TZipFile:nWriteBufferSize.....	1460
TZipFile:lIncludePath	1460
TZipFile:lMasksComPath	1461
TZipFile:lOverwrite	1461
TZipFile:lRecurse.....	1461
TZipFile:lUnicode.....	1461
TZipFile:Methods.....	1462
TZipFile:AddFile.....	1462
TZipFile:FilesSize.....	1462
TZipFile:Run.....	1462
TZipFile:Events.....	1462
TZipFile:OnChangeDisk.....	1463
TZipFile:OnChangeFile.....	1463
TZipFile:OnProgress.....	1463
9 Printing	1464
Report Generator TReport	1464
Commands.....	1465
Classes.....	1471
TReport.....	1471
TReport:Properties.....	1471
TReport:aColumns.....	1472
TReport:aFonts.....	1472
TReport:aGroups	1473
TReport:bFor.....	1473
TReport:bSkip.....	1473
TReport:bWhile.....	1473
TReport:cGrandTotal.....	1473
TReport:cJobName	1474
TReport:cPageTotal.....	1474
TReport:cPdfFilename	1474
TReport:cXLSDData	1474
TReport:lAutoLand.....	1475

TReport:ICancel	1475
TReport:IPageTotal	1475
TReport:IPreview	1475
TReport:IPreviewModal	1475
TReport:IPrintWithNext	1476
TReport:ISummary	1476
TReport:ITotal	1476
TReport:IXLSNumeric	1476
TReport:nBottomMargin	1476
TReport:nClrLined	1477
TReport:nCounter	1477
TReport:nGridStyle	1477
TReport:nGroupLine	1477
TReport:nLeftMargin	1477
TReport:nLineHeight	1478
TReport:nPage	1478
TReport:nRow	1478
TReport:nSeparator	1478
TReport:nTitleDnLine	1478
TReport:nTitleUpLine	1479
TReport:nTopMargin	1479
TReport:nTotalLine	1479
TReport:oDataset	1479
TReport:oDevice	1480
TReport:oFooter	1480
TReport:oHeader	1480
TReport:oParent	1480
TReport:oPen	1480
TReport:oTitle	1481
TReport:Methods	1481
TReport:AddColumn	1481
TReport:AddGroup	1482
TReport:Adjust	1482
TReport:AreGroups	1482
TReport:BackLine	1482
TReport:DelColumn	1482
TReport:DelGroup	1483
TReport:EndLine	1483
TReport:Finished	1483
TReport:LinesLeft	1483
TReport:NeedNewPage	1484
TReport:NewLine	1484
TReport:NewPage	1484
TReport:PrintAtCol	1484
TReport:PreviewNow	1485
TReport:Run	1485
TReport:StartLine	1485
TReportToExcel	1485
TReport:Events	1486
TReport:OnChange	1486
TReport:OnChanged	1486
TReport:OnEnd	1486
TReport:OnEndGroup	1487
TReport:OnEndLine	1487

TRpt:OnInit	1487
TRpt:OnEndPage	1487
TRpt:OnPostEnd	1488
TRpt:OnPostGroup	1488
TRpt:OnPostPage	1488
TRpt:OnPreChange	1488
TRpt:OnStartGroup	1488
TRpt:OnStartLine	1489
TRpt:OnStartPage	1489
TRptColumn	1489
TRptColumn:Properties	1489
TRptColumn:aData	1490
TRptColumn:aPicture	1490
TRptColumn:aTitle	1490
TRptColumn:cTotalPict	1491
TRptColumn:fTotal	1491
TRptColumn:nAlignment	1491
TRptColumn:nCharSize	1491
TRptColumn:nClrPane	1492
TRptColumn:nClrText	1492
TRptColumn:nDataFont	1492
TRptColumn:nTitleAlignment	1492
TRptColumn:nTitleClrPane	1493
TRptColumn:nTitleClrText	1493
TRptColumn:nTitleFont	1493
TRptColumn:nTitleHeight	1493
TRptColumn:nTotal	1493
TRptColumn:nTotalFont	1494
TRptColumn:nWidth	1494
TRptColumn:oReport	1494
TRptColumn:Events	1494
TRptColumn:OnAddToTotal	1494
TRptColumn:OnPrintData	1495
TRptColumn:OnPrintTitle	1495
TRptColumn:OnPrintTotal	1496
TRptLine	1496
TRptLine:Properties	1496
TRptLine:aAlignment	1496
TRptLine:aFonts	1497
TRptLine:aLine	1497
TRptLine:nColor	1498
TRptLine:nHeight	1498
TRptLine:nWidth	1498
TRptLine:oReport	1498
TRptLine:Events	1498
TRptLine:OnPrintLine	1498
TRptGroup	1499
TRptGroup:Properties	1499
TRptGroup:aTotal	1500
TRptGroup:bFooter	1500
TRptGroup:bGroup	1500
TRptGroup:bHeader	1500
TRptGroup:cOldValue	1500
TRptGroup:cValue	1501

TRptGroup:Ieject	1501
TRptGroup:nClrText	1501
TRptGroup:nCounter	1501
TRptGroup:nFont	1502
TRptGroup:nFooterHeight	1502
TRptGroup:nHeaderHeight	1502
TRptGroup:nOrder	1502
TRptGroup:oReport	1502
TRptGroup:Events	1503
TRptGroup:OnPrintFooter	1503
TRptGroup:OnPrintHeader	1503
TRptGroup:OnPrintTotal	1503
FastReport for Xailer	1504
Sergey Spirin compatibility	1507
Register	1508
TFastReport	1508
TFastReport:Properties	1508
TFastReport:aDatasets	1509
TFastReport:cDllPath	1509
TFastReport:cFileName	1509
TFastReport:cFRLicense	1510
TFastReport:cLanguage	1510
TFastReport:cLastError	1510
TFastReport:cReslco	1511
TFastReport:cResName	1511
TFastReport:cTitle	1511
TFastReport:cXALicense	1511
TFastReport:IAbortOnErrors	1512
TFastReport:IDelFRVars	1512
TFastReport:ISynchroData	1512
TFastReport:nLastError	1512
TFastReport:oEngineOptions	1513
TFastReport:oPreviewOptions	1513
TFastReport:oPrintOptions	1513
TFastReport:oReportOptions	1513
TFastReport:Methods	1514
TFastReport:AddArray	1515
TFastReport:AddCategory	1515
TFastReport:AddDataset	1515
TFastReport:AddDbf	1516
TFastReport:AddFunction	1516
TFastReport:AddVariable	1517
TFastReport:Calc	1517
TFastReport:CategoriesList	1517
TFastReport:Clear	1518
TFastReport:ClearData	1518
TFastReport:ClearMasterDetail	1518
TFastReport:ClearVariables	1518
TFastReport:DelDataset	1519
TFastReport:DelError	1519
TFastReport>DeleteCategory	1519
TFastReport>DeleteVariable	1519
TFastReport:DesignReport	1520
TFastReport:DoExport	1520

TFastReport:FieldValue.....	1520
TFastReport:GetErrors	1521
TFastReport:GetProperty	1521
TFastReport:GetVariable.....	1522
TFastReport:LoadFPFile	1522
TFastReport:LoadFromFile.....	1522
TFastReport:LoadFromResource.....	1522
TFastReport:LoadFromString	1523
TFastReport:LoadLangFile.....	1523
TFastReport:LoadStyleSheetFromFile	1523
TFastReport:LoadStyleSheetFromString.....	1523
TFastReport:PrepareReport.....	1524
TFastReport:PreviewClose.....	1524
TFastReport:Print	1524
TFastReport:RestoreVariables	1524
TFastReport:SaveToFile	1524
TFastReport:SaveToFPFile.....	1525
TFastReport:SaveToString.....	1525
TFastReport:SaveVariables	1525
TFastReport:SetADODConnectStr.....	1525
TFastReport:SetFormatSettings	1526
TFastReport:SetIcon	1528
TFastReport:SetLicense.....	1528
TFastReport:SetProperty.....	1529
TFastReport:SetTitle	1529
TFastReport:SetVariable	1530
TFastReport:ShowPreparedReport.....	1530
TFastReport:ShowReport	1530
TFastReport:StopReport.....	1530
TFastReport:VariablesList.....	1531
TFastReport:Events.....	1531
TFastReport:OnAfterLoad.....	1532
TFastReport:OnAfterPrint.....	1532
TFastReport:OnAfterPrintReport.....	1532
TFastReport:OnBeforeCreate	1532
TFastReport:OnBeforeConnect.....	1532
TFastReport:OnBeforeLoad.....	1533
TFastReport:OnBeforePrint	1533
TFastReport:OnBeginDoc.....	1533
TFastReport:OnBtnGenPdf.....	1533
TFastReport:OnBtnPrint.....	1534
TFastReport:OnBtnSendMail.....	1534
TFastReport:OnClickObject	1534
TFastReport:OnDbClickObject.....	1534
TFastReport:OnDesignLoadReport.....	1535
TFastReport:OnDesignSaveReport.....	1535
TFastReport:OnEndDoc	1535
TFastReport:OnEndPreview	1535
TFastReport:OnGetValue.....	1535
TFastReport:OnMouseOverObject.....	1536
TFastReport:OnPageChanged.....	1536
TFastReport:OnPreview.....	1536
TFastReport:OnPrintPage	1536
TFastReport:OnPrintReport.....	1537

TFastReport:OnProgress	1537
TFastReport:OnProgressStart	1537
TFastReport:OnProgressStop	1538
TFrDataset	1538
TFrDataset:Properties	1538
TFrDataset:aRelationFields	1539
TFrDataset:cName	1539
TFrDataset:lLoadOnDemand	1539
TFrDataset:nLoaded	1540
TFrDataset:nMaxRecsOnDesign	1540
TFrDataset:oDsMaster	1540
TFrDataset:oReport	1540
TFrDataset:Methods	1541
TFrDataset:IsActive	1541
TFrDataset:IsLoaded	1541
TFrDataset:Refresh	1541
TFrDataset:SetMaster	1541
TFrDataset:Events	1542
TFrDataset:OnAfterLoad	1542
TFrDataset:OnClose	1542
TFrDataset:OnFirst	1542
TFrDataset:OnNext	1543
TFrDataset:OnOpen	1543
TFrDataset:OnPrior	1543
TFrXailerDataset	1543
TFrXailerDataset:Properties	1543
TFrXailerDataset:aFields	1544
TFrXailerDataset:oDataset	1544
TFrDbfDataset	1544
TFrDbfDataset:Properties	1544
TFrDbfDataset:aFields	1544
TFrArrayDataset	1545
TFrArrayDataset:Properties	1545
TFrArrayDataset:aData	1545
TFrArrayDataset:aFields	1545
TFrEngineOptions	1546
TFrEngineOptions:Properties	1546
TFrEngineOptions:cTempDir	1546
TFrEngineOptions:nConvertNulls	1547
TFrEngineOptions:nDoublePass	1547
TFrEngineOptions:nIgnoreDevByZero	1547
TFrEngineOptions:nPrintIfEmpty	1547
TFrEngineOptions:nSilentMode	1547
TFrEngineOptions:nUseFileCache	1548
TFrEngineOptions:nMaxMemSize	1548
TFrEngineOptions:nNewSilentMode	1548
TFrPreviewOptions	1549
TFrPreviewOptions:Properties	1549
TFrPreviewOptions:nAllowEdit	1549
TFrPreviewOptions:nDoubleBuffered	1549
TFrPreviewOptions:nMaximized	1550
TFrPreviewOptions:nModal	1550
TFrPreviewOptions:nOutlineExpand	1550
TFrPreviewOptions:nOutlineVisible	1550

TFrPreviewOptions:IPictureCacheInFile	1550
TFrPreviewOptions:IShowCaptions	1551
TFrPreviewOptions:nButtons	1551
TFrPreviewOptions:nOutlineWidth	1551
TFrPreviewOptions:nZoom	1552
TFrPreviewOptions:Methods	1552
TFrPreviewOptions:SetBounds	1552
TFrPrintOptions	1552
TFrPrintOptions:Properties	1553
TFrPrintOptions:cPageNumbers	1553
TFrPrintOptions:cPrinter	1553
TFrPrintOptions:ICollate	1553
TFrPrintOptions:IReverse	1554
TFrPrintOptions:IShowDialog	1554
TFrPrintOptions:nCopies	1554
TFrPrintOptions:nPrintPages	1554
TFrPrintOptions:Methods	1554
TFrPrintOptions:ClearOptions	1555
TFrReportOptions	1555
TFrReportOptions:Properties	1555
TFrReportOptions:aDescription	1555
TFrReportOptions:cAuthor	1556
TFrReportOptions:cInitString	1556
TFrReportOptions:cPassword	1556
TFrReportOptions:ICompresed	1556
TFrPreview	1557
TFrPreview:Properties	1557
TFrPreview:IBtnEdit	1558
TFrPreview:IBtnClose	1558
TFrPreview:IBtnEmail	1558
TFrPreview:IBtnExport	1558
TFrPreview:IBtnFind	1559
TFrPreview:IBtnNav	1559
TFrPreview:IBtnOpen	1559
TFrPreview:IBtnOutline	1559
TFrPreview:IBtnPage	1560
TFrPreview:IBtnPDF	1560
TFrPreview:IBtnPrint	1560
TFrPreview:IBtnSave	1560
TFrPreview:IBtnThumb	1560
TFrPreview:IBtnZoom	1561
TFrPreview:IStatusBar	1561
TFrPreview:IToolbar	1561
TFrPreview:nBorderStyle	1561
TFrPreview:nHeight	1562
TFrPreview:nPage	1562
TFrPreview:nPageCount	1562
TFrPreview:nWidth	1562
TFrPreview:nZoomMode	1562
TFrPreview:oReport	1563
TFrPreview:Methods	1563
TFrPreview:Close	1564
TFrPreview:Edit	1564
TFrPreview:Email	1564

TFrPreview:Export	1565
TFrPreview:ExportBIF	1565
TFrPreview:ExportBMP	1565
TFrPreview:ExportCSV	1566
TFrPreview:ExportEMF	1566
TFrPreview:ExportExcel	1566
TFrPreview:ExportGIF	1566
TFrPreview:ExportHTML	1566
TFrPreview:ExportJPEG	1567
TFrPreview:ExportMail	1567
TFrPreview:ExportODS	1567
TFrPreview:ExportODT	1567
TFrPreview:ExportPDF	1568
TFrPreview:ExportRTF	1568
TFrPreview:ExportTIFF	1568
TFrPreview:ExportTXT	1568
TFrPreview:Find.....	1568
TFrPreview:NavFirst.....	1569
TFrPreview:NavLast	1569
TFrPreview:NavNext.....	1569
TFrPreview:NavPrior	1569
TFrPreview:Open	1570
TFrPreview:Outline	1570
TFrPreview:Page.....	1570
TFrPreview:Print	1570
TFrPreview:Save	1570
TFrPreview:Thumb	1571
TFrPreview:ZoomMinus	1571
TFrPreview:ZoomPlus.....	1571
Sumatra PDF	1571
TSumatraViewer	1573
TSumatraViewer:Properties	1573
TSumatraViewer:cExeFile.....	1573
TSumatraViewer:cExePath	1573
TSumatraViewer:cFileName.....	1574
TSumatraViewer:cLastError	1574
TSumatraViewer:IHideToolBar.....	1574
TSumatraViewer:IRestrictMode.....	1574
TSumatraViewer:nInitialPage	1575
TSumatraViewer:nInitialView	1575
TSumatraViewer:nInitialZoom.....	1575
TSumatraViewer:Methods.....	1576
TSumatraViewer:FileOpen.....	1576
TSumatraPrinter	1576
TSumatraPrinter:Properties.....	1576
TSumatraPrinter:cBin	1577
TSumatraPrinter:cExeFile	1577
TSumatraPrinter:cExePath.....	1577
TSumatraPrinter:cPrinter.....	1577
TSumatraPrinter:cRange	1578
TSumatraPrinter:IExitWhenDone	1578
TSumatraPrinter:IShowDialog.....	1578
TSumatraPrinter:ISilent	1578
TSumatraPrinter:nColor.....	1579

TSumatraPrinter:nCopies	1579
TSumatraPrinter:nDuplex	1579
TSumatraPrinter:nPageSize	1580
TSumatraPrinter:nPageType	1580
TSumatraPrinter:nScale	1580
TSumatraPrinterr:Methods	1581
TSumatraPrinter:Print	1581
PDF Sign tool	1581
TPdfSignTool	1582
TPdfSign:Properties	1583
TPdfSignTool:cCertFile	1583
TPdfSignTool:cContact	1583
TPdfSignTool:cError	1584
TPdfSignTool:cInputFile	1584
TPdfSignTool:cLocation	1584
TPdfSignTool:cOutputFile	1584
TPdfSignTool:cPassword	1585
TPdfSignTool:cReason	1585
TPdfSignTool:cTimeStampAccount	1585
TPdfSignTool:cTimeStampPassword	1585
TPdfSignTool:cTimeStampURL	1585
TPdfSignTool:cToolPath	1586
TPdfSignTool:IMultiSign	1586
TPdfSignTool:IShowProcess	1586
TPdfSignTool:IShowResult	1586
TPdfSignTool:IShowSignature	1587
TPdfSignTool:nErrorCode	1587
TPdfSignTool:nSignColumns	1587
TPdfSignTool:nSignHeight	1587
TPdfSignTool:nSignHorzOffset	1587
TPdfSignTool:nSignHorzSep	1588
TPdfSignTool:nSignVertOffset	1588
TPdfSignTool:nSignVertSep	1588
TPdfSignTool:nSignWidth	1588
TPdfSignTool:Methods	1588
TPdfSignTool:Run	1589
10 Multithread programming	1589
Harbour Multitasking	1594
Futures or Promises	1597
TThread	1600
TThread:Properties	1601
TThread:nStatus	1601
TThread:Methods	1601
TThread:Lock	1601
TThread:Notify	1602
TThread:NotifyThread	1602
TThread:Pause	1602
TThread:Quit	1603
TThread:Resume	1603
TThread:Run	1603
TThread:Signal	1604
TThread:Stop	1604
TThread:Unlock	1604
TThread:WaitSignal	1604

TThread:Events	1605
TThread:OnEnd	1605
TThread:OnNotify	1605
TThread:OnNotifyThread	1605
TThread:OnPause	1606
TThread:OnResume	1606
TThread:OnStart	1606
11 TFuture	1606
TFuture:Properties	1607
TFuture:aThreadTasks	1607
TFuture:LastAwaitTask	1607
TFuture:nState	1607
TFuture:Methods	1608
TFuture:AddThreadTask	1608
TFuture:RunSynchroTask	1608
TFuture:CreateFrom	1609
TFuture:CurrentTask	1609
TFuture:Stop	1609
TFuture:TaskCount	1609
TFuture:Events	1610
TFuture:OnComplete	1610
TFuture:OnCompleteTask	1610
TFuture:OnError	1610
12 TFutureTask	1611
TFutureTask:Properties	1611
TFutureTask:nState	1611
TFutureTask:oError	1611
TFutureTask:oParent	1611
TFutureTask:ReturnValue	1612
13 Animations	1612
TAnimation	1614
TAnimation:Propiedades	1614
TAnimation:IAutoReverse	1614
TAnimation:IEnabled	1614
TAnimation:ILoop	1615
TAnimation:IReverse	1615
TAnimation:IStartFromCurrent	1615
TAnimation:nDelay	1615
TAnimation:nDuration	1616
TAnimation:nInterpolation	1616
TAnimation:nSynclnSec	1616
TAnimation:Methods	1616
TAnimation:Start	1617
TAnimation:Stop	1617
TAnimation:Events	1617
TAnimation:OnFinish	1617
TAnimation:OnProcess	1617
TAnimation:OnStart	1618
TAniNumProperty	1618
TAniNumProperty:cProperty	1618
TAniNumProperty:IForceRefresh	1618
TAniNumProperty:INcaRefresh	1619
TAniNumProperty:nStartValue	1619

TAniNumProperty:nStopValue	1619
TAniNumProperty:oControl	1619
TAniControlSize	1620
TAniControlSize:lForceRefresh	1620
TAniControlSize:lNcaRefresh.....	1620
TAniControlSize:aStartValues.....	1620
TAniControlSize:oControl.....	1621
14 TAniControlSize:aStopValues	1621
15 DLL creation	1621
16 Functions	1623
Arithmetical bit operation functions	1623
Shl.....	1624
Shr.....	1624
lAnd.....	1624
lOr	1625
lXOr	1625
nAnd.....	1625
nExclude.....	1626
nNot.....	1626
nOr	1626
nXor.....	1626
Clipboard functions	1627
Communications functions	1627
SendWMCopyData	1627
MySql_Ping.....	1628
Conversion functions	1628
AnsiToWide.....	1629
Atranspose.....	1629
DtoSql.....	1629
DTToSql	1630
Hex.....	1630
HexToDec.....	1630
HexToString	1630
RGBInverse	1631
RGBToHTML.....	1631
StringToHex	1631
StrMySql	1632
StrSql.....	1632
StrSqlLite	1632
WideToAnsi.....	1633
W2Hex	1633
XA_DateTimeZ2Local	1633
XA_UrlEncode	1633
XA_UrDecode.....	1634
XA_UrDecodeToAnsi.....	1634
Date and time functions	1634
Debug functions	1635
LogDebug.....	1635
LogDisplay	1635
LogFile.....	1636
OutDebug.....	1636
OutDebugDec.....	1636
OutDebugHex.....	1637

Dialog functions	1637
Calculator	1638
ChooseColorDlg	1639
ChooseFontDlg	1639
CreateFontDlg	1641
FileOpenDlg	1642
FileSaveDlg	1643
MsgAlert	1644
MsgCancelOK	1644
MsgDebug	1645
MsgInfo	1645
MsgNoYes	1646
MsgOkCancel	1646
MsgRetryCancel	1647
MsgStop	1647
MsgSysError	1648
MsgYesNo	1648
MsgYesNoCancel	1649
PageSetupDlg	1649
PickIconDlg	1650
PrintDlg	1651
RunFolder	1653
ShAboutDlg	1654
ShExitWindowsDlg	1655
ShFileProperty	1656
ShFindComputerDlg	1657
ShFindFilesDlg	1658
ShNetConnectionDlg	1659
ShRestartWindowsDlg	1659
DLLs and execution functions	1660
Execute	1661
GetFileVersionInfo	1661
IsOcxInstalled	1662
IsRunning	1662
KillProcess	1662
XA_GpfHandler	1663
XA_isGDIPPlus	1663
XA_ResourceSearchOrder	1663
EMF functions (Enhanced meta files)	1664
File functions	1664
GetTempFileName	1666
GetDrives	1667
FileDateTime	1667
FileDateTimeLocal	1667
SetFileDateTime	1668
Graphic functions	1668
AngleArc	1670
Arc	1670
ArcTo	1671
Chord	1671
ColorBlend	1672
Ellipse	1672
ExtFloodFill	1673
GetBmpFromColor	1673

HeightToSize.....	1673
Pie.....	1674
SizeToHeight.....	1674
XA_CaptureBitmap.....	1675
XA_DrawRoundedRect.....	1675
XA_FillRoundedRect.....	1675
XA_GetColorization.....	1676
XA_GradientFillH.....	1676
XA_GradientFillV.....	1676
XA_LoadImageAndResize.....	1677
XA_Lightness.....	1677
XA_Luminance.....	1678
XA_PaintCross.....	1678
XA_SetBmpDisableMode.....	1678
Keyboard functions	1679
XA_SetKey.....	1679
Menu functions	1679
Multi-language functions	1680
GetWindowsLanguage.....	1681
i18n.....	1682
i18n_Language.....	1682
LT.....	1683
LT2.....	1683
SetLanguage.....	1684
Multimedia functions	1684
ODBC functions	1684
Printing functions	1685
Resource functions	1685
String functions	1685
Ajoin.....	1686
Decrypt.....	1686
Encrypt.....	1686
IsAllDigits.....	1687
IsCharAlphaNumeric.....	1687
IsStringNumeric.....	1687
ReadLine.....	1688
ReplaceChars.....	1688
ToString.....	1689
XA_Aes256Decrypt.....	1689
XA_Aes256Encrypt.....	1689
XA_LineCount.....	1690
XA_RemoveTags.....	1690
XA_RTFToTxt.....	1690
String Lists functions	1690
XA_WLCreate.....	1691
XA_WLAdd.....	1692
XA_WLSearch.....	1692
XA_WLFree.....	1693
XA_SLCreate.....	1693
XA_SLSearch.....	1694
XA_SLFree.....	1694
System information functions	1694
EnumWindows.....	1695
GetComputerName.....	1695

GetEnvironmentVariable	1695
GetUserName	1696
GetVersionEx	1696
GetWinPlatform	1696
IsElevated	1696
IsUserAnAdmin	1697
SetEnvironmentVariable	1697
SHGetFolder	1697
XA_IsWin7	1699
XA_IsWin8	1699
XA_IsWin9x	1699
XA_IsWin10	1700
XA_IsWin11	1700
XA_IsWinNT	1700
XA_IsWow64	1700
XA_IsWinVista	1701
XA_IsVistaOrHigher	1701
XA_IsXPOrHigher	1701
XailerVersion	1701
TSystemVer class	1702
Useful functions	1703
XA_VarsEqual	1703
Timers & Threads functions	1703
Windows functions	1704
GetActiveForm	1705
ProcessMessages	1705
17 Commands	1706
18 More Help	1716
19 Appendix	1717
Colors	1717
Keyboard Shortcuts	1718
Mouse Shortcuts	1720
Special Characters	1720
Predefined resources	1720
 Index	 1723

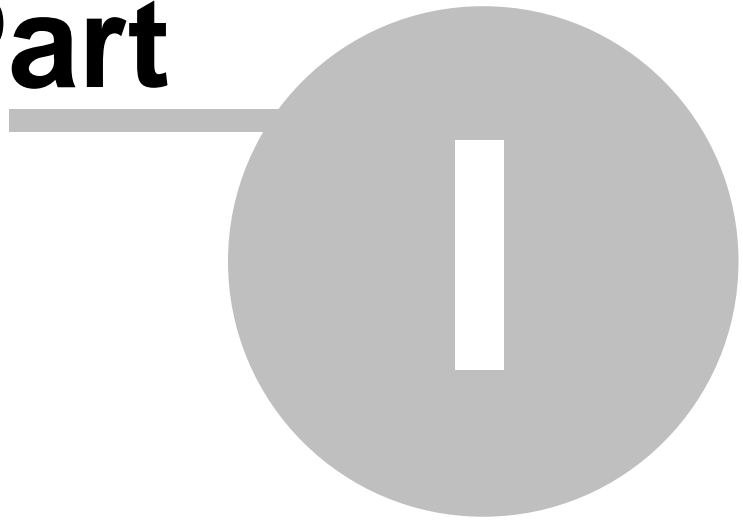
Foreword

This is just another title page
placed between table of contents
and topics

Top Level Intro

This page is printed before a new
top-level chapter starts

Part



1 Xailer Guide

Harbour docs on the web:

<https://harbour.github.io/doc/>

<https://harbour.wiki/>

http://www.marinass-gui.org/projects/harbour_manual/index.html

<https://vivaclipper.wordpress.com/>

<http://www.elektrosoft.it/tutorials/Harbour-Reference-Guide/harbour-reference-guide.htm>

<http://www.kresin.ru/en/index.html>

1.1 User license

XAILER DEMO VERSION NO-NONSENSE LICENSE STATEMENT

It is permitted the use of this software product ("Software"), including its documentation, with the following conditions:

This Software is owned by Xailer and is protected by copyright law and international copyright treaty. Xailer only gives you the right to use this software including its documentation for evaluation purposes and for a limited period. Therefore, is explicitly forbidden the use of this software to create your own applications, neither commercial, free or for your own use.

You are not entitled to transfer, rent, lease, lend, copy, modify, translate, sublicense, time-share or electronically transmit or receive the Software, media or documentation.

Xailer DEMO version does not offer any type of warranty due it is a free evaluation version.

XAILER PERSONAL, XAILER PROFESIONAL AND XAILER ENTERPRISE NO-NONSENSE LICENSE STATEMENT

It is permitted the use of this software product ("Software"), including its documentation, with the following conditions:

This Software is owned by Xailer and is protected by copyright law and international copyright treaty. Therefore, you must treat this Software like any other copyrighted material (e.g., a book), except that you may either make one copy of the Software solely for backup or archival purposes or transfer the Software to a single hard disk provided you keep the original solely for backup or archival purposes.

You may transfer the Software and documentation on a permanent basis provided you retain no copies and the recipient agrees to the terms of this license statement. Except as provided in this statement, you are not entitled to transfer, rent, lease, lend, copy, modify, translate, sublicense, time-share or electronically transmit or receive the Software, media or documentation.

If you have purchased an upgrade version of the Software, it constitutes a single product with the Xailer software that you upgraded. You may use or transfer the upgrade version of the Software only in accordance with this license statement.

Xailer grants to you as an individual, a personal, nonexclusive license to install and use the Software for the sole purposes of designing, developing, testing, and deploying application programs which you create. You may install a copy of the Software on a computer and freely move the Software

from one computer to another, provided that you are the only individual using the Software. If you are an entity, Xailer grants you the right to designate one individual within your organization ("Named User") to have the right to use the Software in the manner provided above.

Xailer may offer several upgrades of this actual software. Xailer has the right to demand an additional payment for them. Obviously you may decline those upgrades.

You accept the existence of a registry procedure individualized by machine to avoid software piracy. The impossibility to make this registry by technical causes will solely give you the right to receive the price of the product except the shipping freight, whenever the claim is made latest 15 days after the date of purchase.

ADDITIONAL LICENSE TERMS FOR XAILER PERSONAL

Xailer will offer free updates of this product during the first six months after the date of purchase. After that period you may be able to acquire a new upgrade extending in six months the period of gratuitous updates.

Xailer allows you to have installed and active the software in a maximum of two computers, and therefore only two of your computers will be able to accede to any update of the product. In order to be able to activate a third computer necessarily you will have to deactivate Xailer in one of the active computers and from then the version of Xailer of this computer will be unusable and without possibility of updates anymore, and you will not be able to activate the software in this computer until you perform a complete format on your local hard disk.

ADDITIONAL LICENSE TERMS FOR XAILER PROFESIONAL AND XAILER ENTERPRISE

Xailer will offer free updates of this product during the first twelve months as of the date of purchase. After that period you may be able to acquire a new upgrade extending in twelve months the period of gratuitous updates.

Xailer allows you to have installed and active the software in a maximum of three computers, and therefore only three of your computers will be able to accede to any update of the product. In order to be able to activate a fourth computer necessarily you will have to deactivate Xailer in one of the active computers and from then the version of Xailer of this computer it will be unusable and without possibility of updates anymore, and you will not be able to activate the software in this computer until you perform a complete format on your local hard disk.

ADDITIONAL LICENSE TERMS FOR ANY XAILER VERSION

You acknowledge that the Software in source code form remains a confidential trade secret of Xailer and therefore you agree not to modify the Software or attempt to decipher, decompile, disassemble or reverse engineer the Software, except to the extent applicable laws specifically prohibit such restriction.

Is specifically prohibited the use of any file including in Xailer for the creation of competitive product with Xailer, as well as its use with development tools different from Personal or Professional Xailer.

Xailer requires for its operation of tools from third party manufacturers or free distribution tools, not becoming responsible for the possible badly operation of the same ones.

GENERAL TERMS THAT APPLY TO COMPILED PROGRAMS AND REDISTRIBUTABLES (Only applicable to Personal, Professional and Enterprise version)

You may write and compile your own application programs using the Software, including any

libraries and source code included for such purpose with the Software. You may reproduce and distribute, in executable form only, programs which you create using the Software without additional license or fees, subject to all of the conditions in this statement.

Xailer products may include certain files ("Redistributables") intended for distribution by you to the users of programs you create. Redistributables include, for example, those files identified in the accompanying printed or on-line documentation as redistributable files, those files preselected for deployment by an install utility provided with the Software (if any), or those files pre-selected by a third party install utility which operates under control of an install script which Xailer has certified (if any) for use by licensed users of this Software for deploying applications. In any event, the Redistributables for the Software are only those files specifically designated as such by Xailer. From time to time, Xailer may designate other files as Redistributables. You should refer to the documentation, including any "readme" or "deploy" files included with the Software, for additional information.

Subject to all of the conditions in this statement, you may reproduce and distribute exact copies of the Redistributables, provided that such copies are made from the original copy of the Software or the copy transferred to the single hard disk. Copies of Redistributables may only be distributed with and for the sole purpose of executing application programs permitted under this statement that you have created using the Software. Under no circumstances, any copies of Redistributables may be distributed separately. Only you as the Named User have the right to use the Redistributables for developing programs created with the Software; accordingly, you may not share copies of the Redistributables with other co-developers. You may not reproduce or distribute any Xailer documentation without Xailer's permission.

The license granted in this statement for you to create your own compiled programs and distribute your programs and the Redistributables (if any) is subject to all of the following conditions: (i) all copies of the programs you create must bear a valid copyright notice, either your own or the Xailer copyright notice that appears on the Software; (ii) you may not remove or alter any Xailer copyright, trademark or other proprietary rights notice contained in any portion of Xailer libraries, source code, Redistributables or other files that bear such a notice; (iii) Xailer provides no warranty at all to any person, other than the Limited Warranty provided to the original purchaser of the Software, and you will remain solely responsible to anyone receiving your programs for support, service, upgrades, or technical or other assistance, and such recipients will have no right to contact Xailer for such services or assistance; (iv) you will indemnify and hold Xailer, its related companies and its suppliers harmless from and against any claims or liabilities arising out of the use, reproduction or distribution of your programs; (v) your programs must be written using a licensed, registered copy of the Software; (vi) your programs may not be merely a set or subset of any of the libraries, code, Redistributables or other files of the Software; and (vii) you may not use Xailer's or any of its suppliers' names, logos, or trademarks to market your programs, except to state that your program was written using the Software.

All Xailer libraries, source code, Redistributables and other files remain Xailer's exclusive property. Regardless of any modifications that you make, you may not distribute any files (particularly Xailer source code and other non-executable files) except those that Xailer has expressly designated as Redistributables. Nothing in this license statement permits you to derive the source code of files that Xailer has provided to you in executable form only, or to reproduce, modify, use, or distribute the source code of such files. You are not, of course, restricted from distributing source code that is entirely your own.

LIMITED WARRANTY

Except with respect to the Redistributables, which are provided "as is," without warranty of any kind, Xailer warrants that the Software, as updated and when properly used, will perform substantially in accordance with the accompanying documentation, and the Software media will be free from defects in materials and workmanship, for a period of ninety (90) days from the date of receipt.

Any implied warranties on the Software are limited to ninety (90) days. Some states/jurisdictions do not allow limitations on duration of an implied warranty, so the above limitation may not apply to you.

Xailer's entire liability and your exclusive remedy shall be, at Xailer's option, either (a) return of the price paid, or (b) repair or replacement of the Software that does not meet Xailer's Limited Warranty and which is returned to Xailer with a copy of your receipt. **DO NOT RETURN ANY PRODUCT UNTIL YOU HAVE CALLED THE XAILER CUSTOMER SERVICE DEPARTMENT AND OBTAINED A RETURN AUTHORIZATION NUMBER.** This Limited Warranty is void if failure of the Software has resulted from accident, abuse, or misapplication. Any replacement Software will be warranted for the remainder of the original warranty period or thirty (30) days, whichever is longer.

TO THE MAXIMUM EXTENT PERMITTED BY APPLICABLE LAW, XAILER DISCLAIMS ALL OTHER WARRANTIES, EITHER EXPRESS OR IMPLIED, INCLUDING, BUT NOT LIMITED TO, IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS FOR A PARTICULAR PURPOSE AND WARRANTIES OF NON-INFRINGEMENT, WITH REGARD TO THE SOFTWARE AND THE ACCOMPANYING DOCUMENTATION. THIS LIMITED WARRANTY GIVES YOU SPECIFIC LEGAL RIGHTS. YOU MAY HAVE OTHERS, WHICH VARY FROM STATE/JURISDICTION TO STATE/JURISDICTION.

TO THE MAXIMUM EXTENT PERMITTED BY APPLICABLE LAW, IN NO EVENT SHALL XAILER BE LIABLE FOR ANY DAMAGES WHATSOEVER (INCLUDING, WITHOUT LIMITATION, DAMAGES FOR LOSS OF BUSINESS PROFITS, BUSINESS INTERRUPTION, LOSS OF BUSINESS INFORMATION, OR ANY OTHER PECUNIARY LOSS) ARISING OUT OF THE USE OF OR INABILITY TO USE THIS XAILER PRODUCT, EVEN IF XAILER HAS BEEN ADVISED OF THE POSSIBILITY OF SUCH DAMAGES. BECAUSE SOME STATES/JURISDICTIONS DO NOT ALLOW THE EXCLUSION OR LIMITATION OF LIABILITY FOR CONSEQUENTIAL OR INCIDENTAL DAMAGES, THE ABOVE LIMITATION MAY NOT APPLY TO YOU.

HIGH RISK ACTIVITIES

The Software is not fault-tolerant and is not designed, manufactured or intended for use or resale as on-line control equipment in hazardous environments requiring fail-safe performance, such as in the operation of nuclear facilities, aircraft navigation or communication systems, air traffic control, direct life support machines, or weapons systems, in which the failure of the Software could lead directly to death, personal injury, or severe physical or environmental damage ("High Risk Activities"). Xailer specifically disclaims any express or implied warranty of fitness for High Risk Activities.

GENERAL PROVISIONS

This statement may only be modified in writing signed by you and an authorized officer of Xailer. If any provision of this statement is found void or unenforceable, the remainder will remain valid and enforceable according to its terms. If any remedy provided is determined to have failed for its essential purpose, all limitations of liability and exclusions of damages set forth in the Limited Warranty shall remain in effect.

This statement shall be construed, interpreted and governed by the laws of Spain. This statement gives you specific legal rights; you may have others which vary from state to state and from country to country. Xailer reserves all rights not specifically granted in this statement.

In case that the present contract was available in several languages, he will solely be valid the one in Spanish language, being the rest, mere translations.

1.2 Firsts Steps

1.2.1 Introduction

An IDE in programming is a very useful and productive tool that greatly eases the most mechanic and unpleasant jobs, giving us more time to dedicate it to our real goal as programmers: to focus in algorithms and routines needed by our programs.

Those programmers who have used an IDE before (i.e. Delphi) are aware to appreciate immediately the advantages of this way of programming, and will start using it without the need to learn something new.

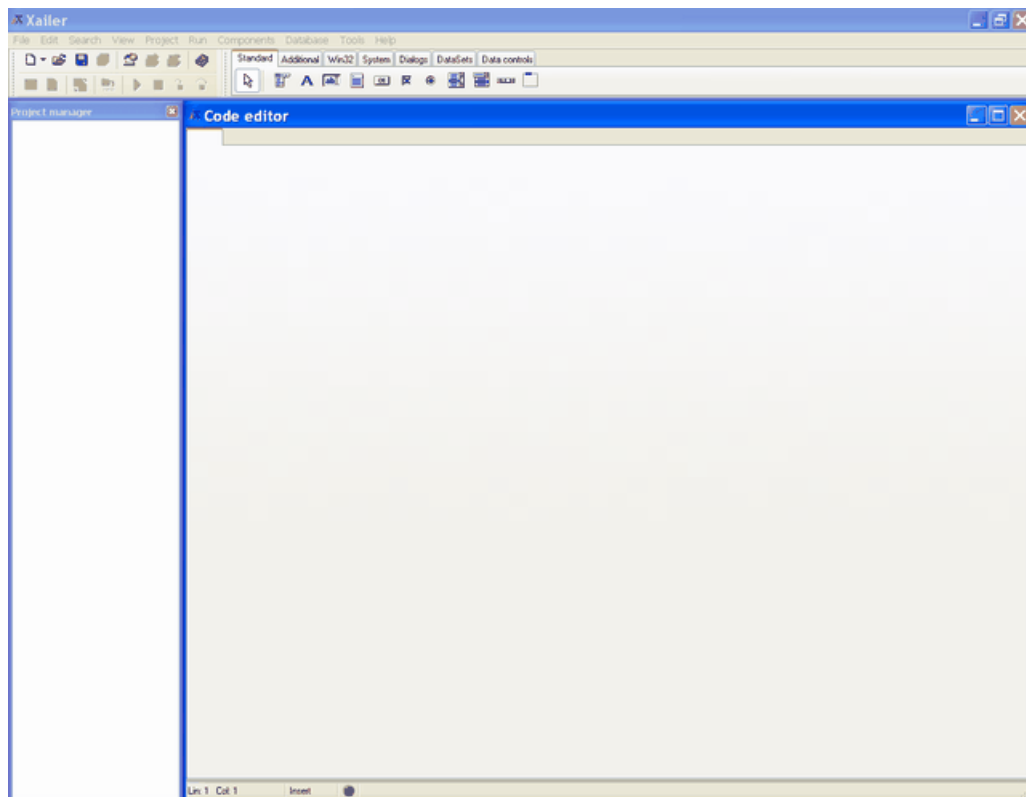
But those who haven't had contact with this kind of tools before will find themselves a little lost.

This document is dedicated to those programmers who don't know an IDE, and need that "little push" to start. We are pretty sure that once you feel comfortable with this way of programming all of us will win a lot in productivity and will speed our projects.

However, and IDE doesn't design the programs for us. No matter that it allows us to design in a visual way the windows of our application without writing any code for this, a full program is not only a set of windows more than less pretty, we have to write the necessary code to do what is needed to do. So, don't think that the IDE is an application generator, is a tool that eases our job, but doesn't do our work.

1.2.2 Running Xailer

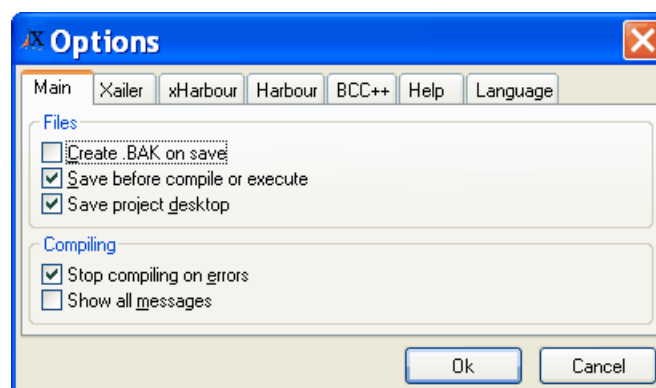
The first time you run the Xailer's IDE you will see this:



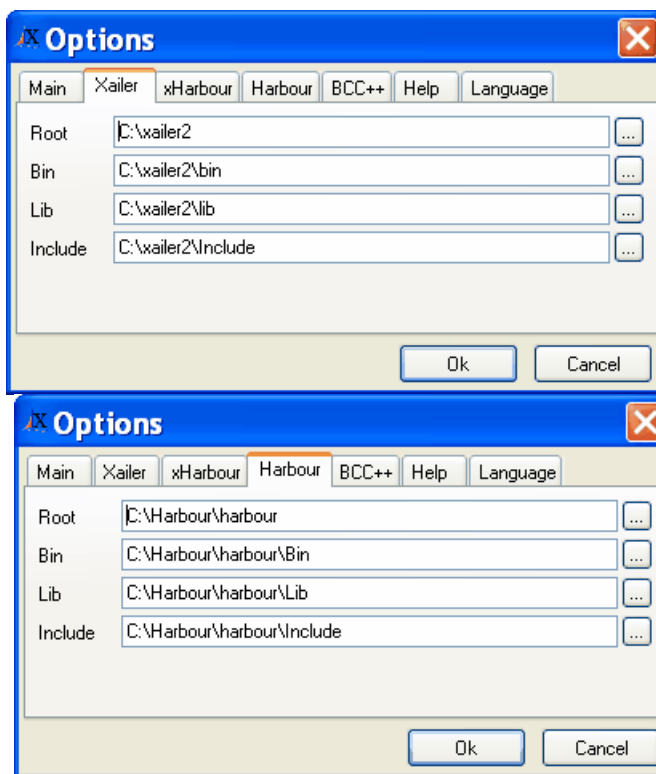
Please notice that Xailer doesn't try to create a new project nor open an existing one, that's why the Project Manager and the Source Code Editor are blank and some of the buttons in the toolbar are disabled.

1.2.3 Xailer Configuration

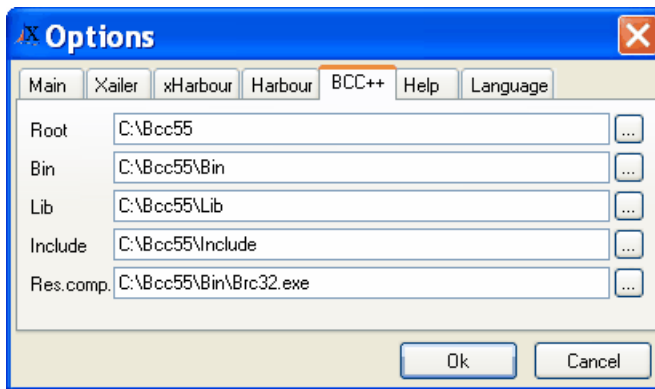
Before we start to work with Xailer, we have to configure it. For this task we'll use the Options... item in the Tools menu.



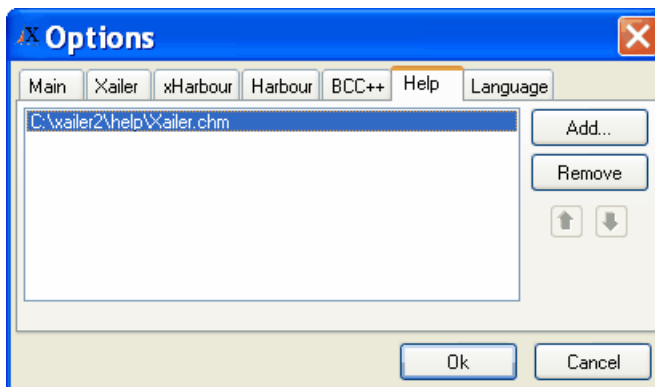
- Create .BAK on save: Every time a file is saved, a copy of the last file is done with the extension .bak
- Save before compile or execute: Before compiling or running the project, all the modified files are saved.
- Save project desktop: Along with the project, the state of the files actually opened in the source code editor is saved.
- Stop compiling on errors: When a compiling error is detected, the full process is stopped and no more modules are compiled.
- Show all messages: Xailer captures the output messages of the different compilers, and only shows the messages of interest, informative, warnings or errors. When this box is checked, all the compilers' output messages will be shown, not only the ones captured by Xailer.



In the tabs Xailer, xHarbour and Harbour we can configure the directories where these tools are installed.




In the tab BCC++ you can configure the directories where the Borland C++ 5.5 compiler is installed. You can specify also the resource compiler you want to use, by default it'll be the resource compiler included in the Borland C++ 5.5 (brc32.exe), however the use of the Microsoft resource compiler (rc.exe) is highly recommended, it's included in the MS Platform SDK and in other Microsoft tools, this is because it let you to use 32 bits icons and images with alpha mask (Windows XP compatibles).



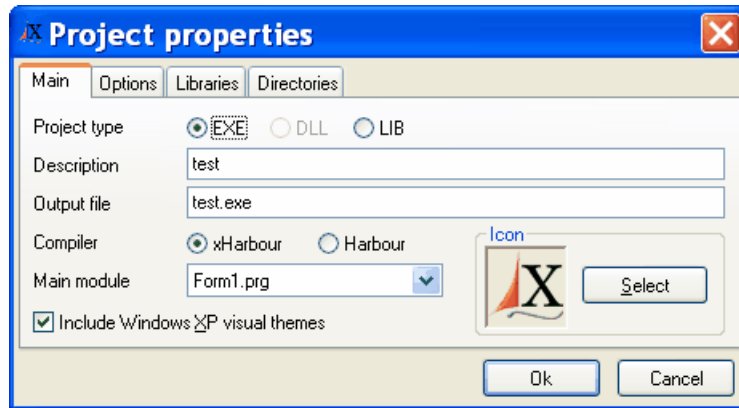
In the Help tab you can set up the help files you want to have at hand. Firstly we have to set the HTML files (.chm) because due to Xailer's way of work, it can search a topic or word inside them, and if it is not found, skip into the next topic. On the other hand, with traditional help files (.hlp) if a topic is not found, an index is shown, but Xailer cannot detect that the topic you were looking for was not found, and it cannot keep looking into the next topic.

1.2.4 Creating a new project

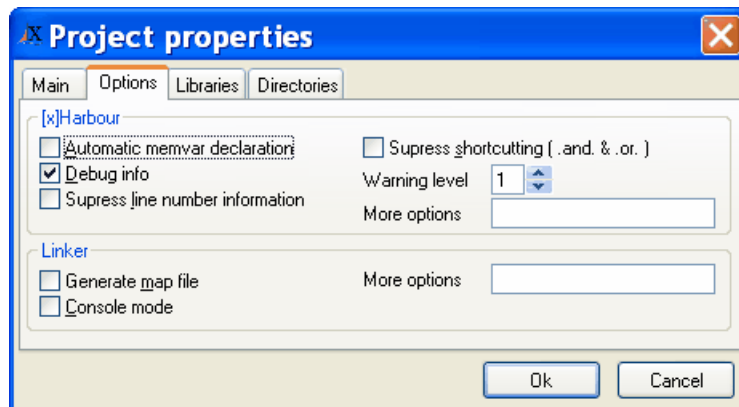
Once you had configured Xailer, you are ready to create a new project. Select NewProject... from the File menu or push the button . The Create a new project dialog will be shown, you have to give a name to the file that will host the project, with the .xproj extension.

Tip: It's recommended to create a new folder for every the project, to avoid mixing files of other projects. This new directory will be the project's root directory.

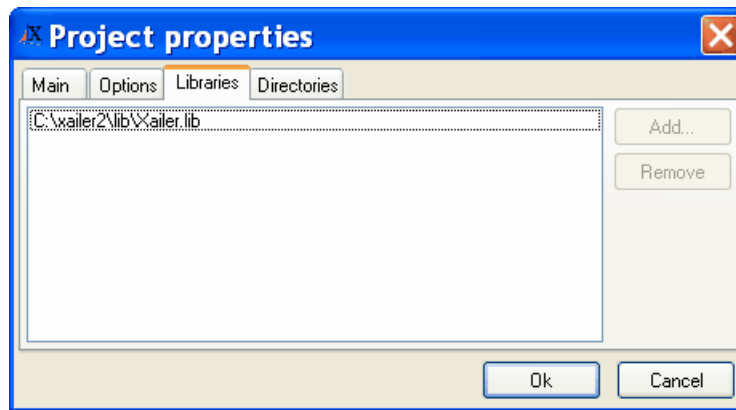
After giving a name to the file, Xailer will show a window with the project's options.



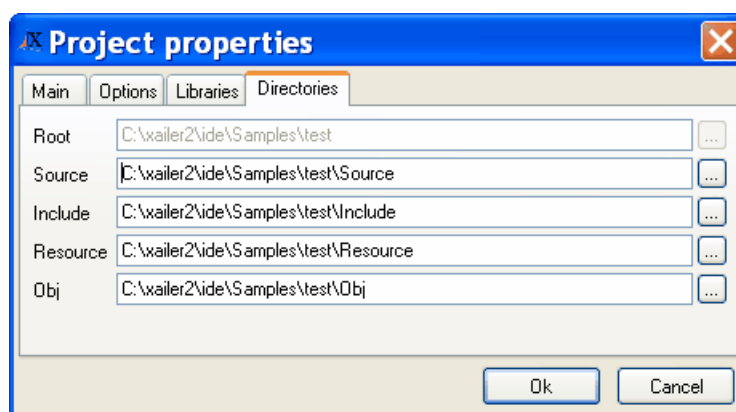
- Project type: EXE, DLL (not yet available) or LIB.
- Description: Brief project's description.
- Output file: Name of the executable or library output file.
- Compiler: xHarbour or Harbour. In spite of you can select Harbour, you cannot compile correctly with it.
- Main module: Here you can indicate the application's main module (once you have added more modules to the project). If it's a form, Xailer will write automatically the source code to create and show it, but if it's not a form, Xailer will look for the first function or procedure and it will generate the source code to call it.
- Include Windows XP visual themes: Indicates if we want to automatically link the manifest file for the application to show XP themes.



- [x]Harbour: These are the compiling options for [x]Harbour, they are equivalent to /a, /b, /l, /x y /w1 flags. In the More options text box you can indicate other compilation options.
- Linker: These are the linker options. The Console Mode option doesn't mean that you can generate an application in console mode, but indicates the inclusion of the -aa flag in linking time to verify non free memory blocks and other output messages of [x]Harbour.



In this tab you can include the third party libraries you may need, at this time this feature is not available. If you want to add a new library, you can add it directly in the project's files.



In this tab we can indicate the Xailer's project directories:

- Root: This is the project's root directory, the files that don't belong to other directories and the project file (.xpj) must exist in this directory. This is also the directory where the executable file will be created. It's disabled because it's here where the .xpj file resides.
- Source: It will contain the project's source code files (.prg, .xfm and .c).
- Include: It will contain the project's header files (.ch and .h).
- Resource: It will contain the project's resource files (.rc, .bmp, .ico, etc.).
- Obj: It will contain the object files generated by the compiling process (.obj and .res).

If these directories don't exist, Xailer will create them automatically.

1.2.5 The project manager


Once created the project, Xailer will create automatically a source code file with the same name of the project and .prg extension. This file will be the entry module of the application, and in spite of it's xbase code, is recommended not to open or modify it directly.


In this file you'll find the Main() procedure, which contains the calls to the main module of the

project and to the methods of the Application object that makes the application works.

From this point, you are now able to create forms and modules using the item New in the File menu or the buttons




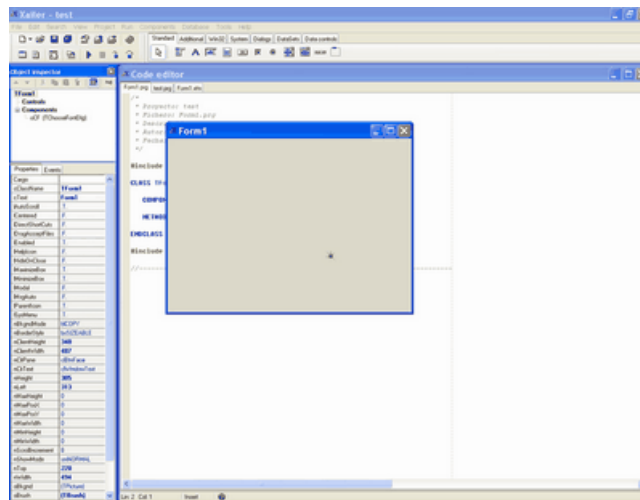
You can also add previous existing files to the project, using the option Add file to project... in the Project menu or using the button . You can add files .prg, .c, .ch, .h, .rc, .obj, .res and .lib. Any other kind of file added to the project will not be used by Xailer, but it will be included.

You can remove files from the project with the option Remove file from project... in the Project menu or using the button .

All the items in the project manager have a pop up menu (clicking the right button of the mouse) with the most common used options. You can also drag files from Windows explorer and drop them into the project manager window, and they'll be included in the project.

1.2.6 Creating a form

To create a new form, just click the button , or choose the option New, in the Form item into the File menu, you will see this:



As you can see, a form in design mode and the object inspector are shown, also some lines of code are now written in the source code editor.

Let's see in detail the written source code:

```
/*
 * Proyecto: Ejemplo
 * Fichero: Form1.prg
 * Descripción:
 * Autor:
 * Fecha: 08/04/2004
 */
```

```
#include "Xailer.ch"
```

```
CLASS TForm1 FROM TForm
```

```
    METHOD CreateForm()
```

```
ENDCLASS
```

```
#include "Form1.xfm"
```

```
//-----
```

Please note that apart from some comment lines and the inclusion of the Xailer's header file, you find a class declaration:

```
CLASS TForm1 FROM TForm
```

In Xailer, all the forms created with the IDE are classes inherited of TForm class. This could be something difficult and twisted to understand, but with the practice you will see that this way of work is appropriated and flexible. All the components and the event's source code of the form are members of the class, allowing you, among other things, to completely isolate a form from other forms inside the same application, within the same class or inside other class.

The first method you find in the class is CreateForm(), it will contain all the necessary source code to create the form, the controls and the components that this form has. This method is not written inside the same source code file, Xailer places it inside a file with the same name of the current module, but with extension .xfm.

The last line you will find is:

```
#include "Form1.xfm"
```

Which will include the CreateForm() method previously written in the .xfm file.


If you are curious, you can open the .xfm file and see what is inside it, you will see that is simply xbase source code used to set up the properties of the form and to create all and every one of the components. But, you shouldn't modify this source code, because Xailer will discard your changes every time you save the form in the IDE. This is what it contains now:

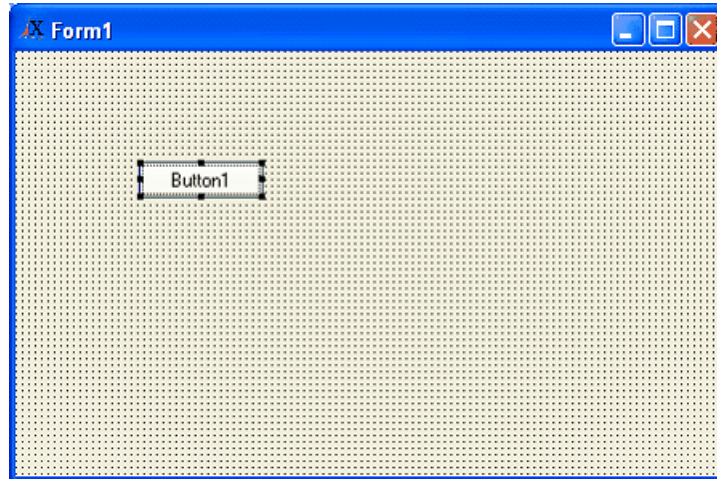
```
METHOD CreateForm() CLASS TForm1
```

```
    Super::CreateForm()  
    ::nLeft := 273  
    ::nTop := 126  
    ::nWidth := 450  
    ::nHeight := 319  
    ::cText := "Form1"  
    ::nClientWidth := 443  
    ::nClientHeight := 286  
    ::Create()
```

```
RETURN Self
```

1.2.7 The forms editor

Now you can place all the controls and components that you want inside this form, selecting them from the component palette and then clicking over the form in the place you want to put it. As a simple practice let's place a button, using the component  in the Standard palette:




Once placed the component in the form, you can move it using the mouse or the keyboard, pressing at the same time the Shift and the arrows keys, it will move pixel by pixel, by pressing the Control key, it will move from 20 to 20 pixels. You can also modify its size using the dots around the control when it is selected.


Right clicking with the mouse over the control will show a pop up menu with the most common used options.

You can also select several controls at the same time to move, copy, delete, align them, etc. Some operations require that all the selected controls be in the same level, this means, that they belong to the same parent, the form itself or any other container control.

To select several controls at the same time we can do it in several ways:

- Clicking over any point inside the form and then dragging the mouse over the controls we want to select.
- Clicking and at the same time pressing the Shift key over every one of the controls we want to select.
- Pressing the Control key at the same time you click over a container control, the container control will be selected along with all the controls contained inside it.

Whenever you are selecting controls to modify their properties or events, it's very easy that you move them without wanted to. To avoid this, you can "lock" the controls of the form, pressing the button  in the object inspector.

Also, if you are using Windows XP operating system, you may be interested on see how your form will look in older Windows versions, without themes, for this, you can press the button  in the object inspector.

1.2.8 The objects inspector

To modify the properties of a control or component, you'll use the objects inspector. This is divided in three areas: a button bar, a tree structure showing the components of the form and a folder with two tabs (Properties and Events)

The button bar let you to perform some operations, such modify the order of the form's controls and components, to use the clipboard, see how the form will look with and without XP themes (this feature is only available in Windows XP) and lock the controls inside the form for not to move or resize them.

The components tree let you to select any component of the form, including the same form, and has a pop up menu with the most common used options.

The properties and events folder, let you modify the components' properties and add the events you may need.

1.2.9 The properties

In the properties list you can see the design properties of the component. The properties that contain a value different from the default one will be shown in bold font. There are several kinds of properties, the most of them can be edited directly in the list, someothers will open a properties editor to modify them. Basically the cells in the list can be shown like these:


.

cText	Button1
-------	---------

 : normal properties, you can edit directly its value.


.

IDefault	F
----------	---

 : properties which their values depend of the values stored in a list. In this case, when pushing the button , a list of values will be shown. Double clicking over a value, will change it to the next value of the list, except for those with color properties, which will show the color selection dialog

.

nAnchors	akTopLeft
----------	-----------

 : properties with a more complex properties editor. Pressing the button  or double clicking over this value will make the appropriate editor appear.

In our example, we are going to modify the value of the control's cText property, simply you have to write the new value in adecuate cell.

cText	Aceptar
-------	---------

This will change the text of the button we have placed before in the form.

Note: Some properties of some components don't have developed their properties editor yet, so you won't be able to modify such properties. In this case, you can always change their values using the Form's OnInitialize event, thus it will be explained later in this file.

1.2.10 The events

A program is useless if it's not aware of the user's actions, then you have to write the code to be executed when this happens. In Xailer, this job is performed by the events.

An event is a kind of property, where you assign the code to be executed. This code can be assigned in two different ways:

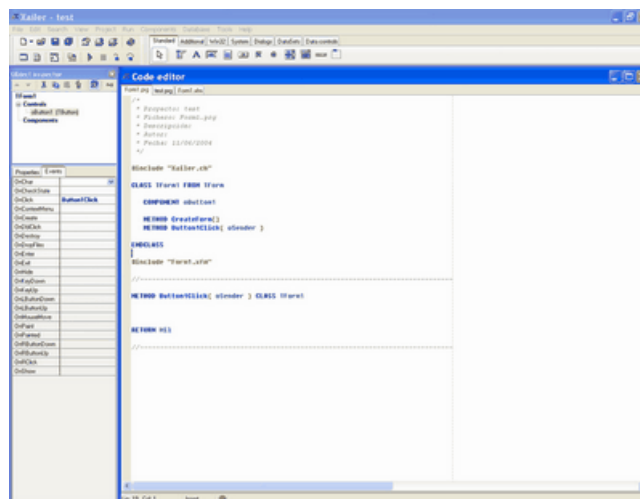
- A code block: Whenever a code block is assigned as an event, this is executed every time the event happens, and it receive all the necessary parameters.
- A method's name inside the form: In this case, the method is executed, receiving the necessary parameters.

In Xailer is always suggested to use the second way, because it runs faster, and at the same time, the source code is more readable, however, if the code to be executed is quite simple, it could be interesting to assing a code block instead.

The way you have to assign a code block is simple: write it in the adequate cell, in the tab Events in the objects inspector. The main disadvantage of this, is that you don't know in advance the parameters that the code block will receive, so you will have to remember them, or search for them in the help file (only if it's available).

To assign a form's method, just write down the name you want to give to the method in the cell and Xailer will declare it in the class definition and will create the module's body, with all its parameters. You can also make double click over the cell, and Xailer will create the method with a generic name.

For our sample, we are going to assign button's event OnClick, just double click on the event's cell. Xailer will create the method Button1Click(), and will show you the source code editor, with the cursor pointing to such method, now you are ready to write the code.



As you can see, Xailer has declared the oSender parameter. This parameter will always be the first parameter of any event, and it's the same object that called the method. In this sample, oSender is the button itself, who has called the event.

Let's write some code, a simple MsgInfo(), and then your code looks like:


```
METHOD Button1Click( oSender ) CLASS TForm1
```


```
    MsgInfo( "Click" )
```

```
RETURN Nil
```

You can assign the same form's method to different events of the same control, using the same parameters in the same order. You can also assign it to the same event or to different events of

other controls inside the same form. In these cases the `oSender` parameter let you know which control has triggered the event.

In this way, pressing the button  of one event, all the methods of the form with the same parameters for the event will be shown, and you can select the one you want, without writing it again.

So, our form sample is ready to save and compile. To save it, just click the button  or push Control + S. The first time you save the form, Xailer will ask you for the file name, and will change all the references to the file in the module and in the project manager. If this is the project's first module or form, it'll be established also as the main module in the project properties.

For our sample, we are going to save it as `Form1.prg`.

As stated before, there are actually some properties that cannot be changed from the objects inspector, so they have to be assigned from source code. The way to do it, is to create the form's event `OnInitialize`, and write the necessary code, this is due, because this event is triggered right after the form and all its components are created, but before of showing it.

For example, if we want to assign an image to a `TBtnBmp` control, the event would look like this:

```
METHOD FormInitialize(oSender) CLASS TForm1
```

```
    ::oBtnBmp1:oBitmaps := "Ok"
```

```
RETURN Nil
```

1.2.11 The source code editor

Until now, you have seen and used the source code editor, but it would be convenient to review a little bit over its main features.

This is a multidocument editor, with color syntax and the most of the features you are used to have in a source code editor. Actually all the editor's options are hardcoded inside Xailer's IDE, but in future versions you will be able to set up them at your own preferences.


The editor supports some other functions than the current ones, such autocomplete, loops and structures checking, etc. we'll be implementing them in the future.

The most common used key are (apart from the ones for navigation and selection):

Ctrl + A	Select all text
Ctrl + C	Copy
Ctrl + D	Duplicate current line
Ctrl + F	Search text
Ctrl + Shift + F	Search in all the project's files.
Ctrl + G	Go to line...
Ctrl + L	Erase current line
Ctrl + N	New module
Ctrl + O	Open file
Ctrl + R	Find and replace text

Ctrl + S	Save document
Ctrl + T	Switch current line with previous one.
Ctrl + V	Paste
Ctrl + X	Cut
Ctrl + Y	Re do
Ctrl + Z	Un do
Ctrl + +	Increase text size
Ctrl + -	Decrease text size
Ctrl + /	Back to normal text size
Ctrl + F1	Help with the current word
Ctrl + F2	Mark set
F2	Go to next mark
Shift + F2	Go to previous mark
Ctrl + Shift + F2	Delete all marks
F3	Find next
Tab	Indent selected block
Shift + Tab	Un indent selected block

1.2.12 Compiling and running the project

To compile the project is as easy as to press the button  or to press the Control + F9 keys. Xailer will perform all the necessary tasks to compile the .prg files with [x]Harbour, then the C files with BCC++ and finally the .rc file with the previously selected resource compiler, etc., It also calls the linker to generate the executable file. You don't have to worry about how to create a .mak file, Xailer includes its own compiling engine.




When compiling, Xailer will show an area with the compiler messages right down the source code editor window:

```

i Compiling: Form1.prg...
i Linking: test.exe...
0 Warnings, 0 Errors

```

Three kind of messages can be shown in this area:

- Informative: are shown with the image , and inform you the current compiling process.
- Warnings: are shown with the image , and inform you the warning messages during the compiling process.
- Errors: are shown with the image , and inform you the errors when compiling.

Some other messages would appear without any associated image, these messages are also informative.

If you double click over a warning or error line, Xailer will show the line with error or warning in the source code editor, if you double click over an informative line, the source code file will be open in the editor.


You can also compile pressing the F9 Key, and Xailer will compile and run the project if not compiling errors are found.

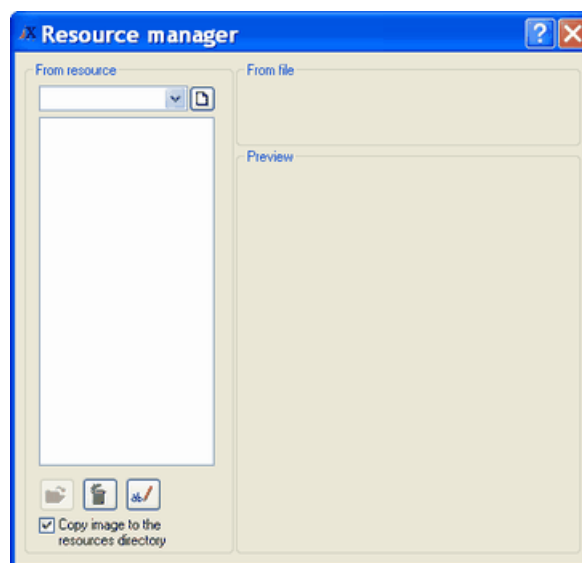
In all cases, Xailer will compile only those files modified after the last compilation process, however you can press **Alt + F9** to compile all files in the project no matter which files were modified or not.


On the other hand, if you press the **Shift** key at the same time you press the compiling button or the **F9**, **Control + F9** or **Alt + F9** keys, Xailer will show all the output messages of the different compilers, instead of filter them and show only the necessary ones, as it does normally.


1.2.13 The resource manager

Xailer can retrieve images from files, however it's always more practical to include all the necessary images in resources that later will be linked into the final executable file. For not to handle directly the resource files (.rc), Xailer includes its own Resource manager.

For example: let's place an image as background for the previously created form. For doing that, you have to modify the form's property `oBkgnd`, clicking the button  then the Resource manager will be shown:



As you can see, the first time it's empty. The first thing you have to do, is to create a new resource file, clicking on the button . The name given by Xailer is the same of the project's, but with `.rc` extension. If you need to create more than one resource file, do it the same way, but Xailer will not give automatically a default name.

Once created the resource file, the button  is enabled, use this button to add images to this file. The checkbox `Copy image to the resources directory` is used to indicate Xailer to make a copy of the image from one directory into the project's resource directory, so, all the images are stored in the same place. When you press the button, the Open Image dialog will appear.

After selecting the image, it will be shown in the Preview box. If the image is bigger than the box, you can make the dialog bigger.

Now, the other two buttons are enabled, the one to delete an image from the resource file, and the one used to change the resource name.

The group box `From file` allows you to add an image from file, instead of retrieve it from resources.

It also has a checkbox Copy image to the application directory to have all the images stored in the same place.

The None button lets you to delete an image from a previously assigned property.

Now press the Ok button, the image will appear as the form's background. Please notice that if the image is bigger than the form, it will not be fully shown, on the other hand, if it's smaller, it doesn't fill all the available space, if you want to make the image fit inside the form, change the property `nBkgndMode` by `blSTRETCH`.

1.2.14 A more complex sample

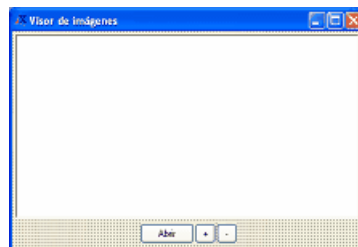
Now that you've seen the Xailer's IDE basics, there are still a lot of doubts and questions to ask for: how do I call another form ?, how do I assign properties in run-time ?, how can I use the non-visual components ?, etc.

Following, we will build a sample to answer you the most common questions about developing an application with Xailer. For this sample, let's build a simple image viewer.

First of all we are going to create a new project, as you have done before, let's name it `visor.xpj`. Next, let's create a new form, and change the following properties:

<code>cClassName</code>	<code>TVisor</code>
<code>cText</code>	Image viewer
<code>nShowMode</code>	<code>smMAXIMIZE</code>

Next we include a `TImage` control, and three `TButton` controls, as you can see here:



And finally, we add a `TFileOpenDlg` component (taken from the Dialogs palette). Please notice that this component is not visible inside the form. Other languages (such as Delphi), shown a button in the form to represent the non-visual components, but in Xailer we have decided not to do it, to keep the form's appearance in design mode as you will see it in run time. When you need to access the non-visual components properties, you can use the tree structure in the objects inspector.

Now is time to set up the properties for every object:

Objet	Property	Value
<code>oImage1 (TImage)</code>	<code>nAnchors</code>	<code>akALL</code>
<code>oButton1 (TButton)</code>	<code>cText</code>	Open
	<code>nAnchors</code>	<code>akBOTTOM</code>
<code>oButton2 (TButton)</code>	<code>cText</code>	+
	<code>nAnchors</code>	<code>akBOTTOM</code>

oButton3 (TButton)	cText	-
	nAnchors	akBOTTOM
oFileOpenDlg1 (TFileOpenDlg)	cFilter	Images files *.bmp;*.jpg;*.jpeg;*.gif All files *.*
	cTitle	Open image

And now, just the events are missing. Let's assign the `OnClick` event for every one of the buttons, our source code should look like this (remember: you have to make double click in the events of every one of the controls):

```
//-----
METHOD Button1Click( oSender ) CLASS TVisor

  IF ::oFileOpenDlg1:Run()
    ::oImage1:oPicture := ::oFileOpenDlg1:cFullFileName
  ENDIF

RETURN Nil

//-----

METHOD Button2Click( oSender ) CLASS TVisor

  ::oImage1:nZoom *= 2

RETURN Nil

//-----

METHOD Button3Click( oSender ) CLASS TVisor

  ::oImage1:nZoom /= 2

RETURN Nil

//-----
```

Well, now you can compile and test the project. You'll be surprised on all the things you have done with so little effort.

And now, let's complete the sample a little more to make possible to zoom the image pressing + and - keys, writing the form's `OnChar` event:

```
METHOD FormChar( oSender, nKey, nFlags ) CLASS TVisor

  IF nKey == Asc( "+" )
    ::Button2Click()
  ELSEIF nKey == Asc( "-" )
    ::Button3Click()
  ENDIF

RETURN Nil
```

We may also be interested on adding drag and drop support to our image viewer from the Windows Explorer, so, we have to set the `oImage1` control property `IDragAcceptFiles` to `.T.` and also we have

to write the OnDropFiles event:

```
METHOD Image1DropFiles( oSender, aFiles , aPoint ) CLASS TVisor
    ::oImage1:oPicture := aFiles[1]
RETURN Nil
```

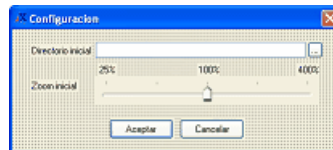
And as a last feature, we can include the ability to pass the name of the image file to be opened as a parameter to the program. For this, we are going to use the form's event OnInitialize:

```
METHOD FormInitialize( oSender ) CLASS TVisor
    IF !Empty( Application:aArguments )
        ::oImage1:oPicture := Application:aArguments[1]
    ENDIF
RETURN Nil
```

Well, until now we have written a very simple image viewer with some goodies, and we had a look over some Xailer 's IDE features that we may need in our own projects: to assign properties from the IDE and from source code, to write events, to use non-visual components, etc.

But we still have a couple of important issues left behind: to give a menu to the program, and to call other forms:

We are going to design a menu with a few options and a little configuration form. In first place, we have to create a new form, and include some controls, like this (along with a TBrowseForFolderDlg component):



With the following properties:

Object	Property	Value
Form	cClassName	TConfig
	cText	Configuration
	lCentered	.T.
	nBorderStyle	bsDIALOG
oLabel1 (TLabel)	cText	Root directory
oEdit1 (TEdit)	Value	(left blank)
oButton1 (TButton)	cText	...
oLabel2 (TLabel)	cText	Initial Zoom
oTrackBar1 (TTrackBar)	nMax	2
	nMin	-2
oLabel3 (TLabel)	cText	25%
oLabel4 (TLabel)	cText	100%
oLabel5 (TLabel)	cText	400%
oButton2 (TButton)	cText	Ok
	lDefault	.T.
oButton3 (TButton)	cText	Cancel
	lCancel	.T.

oBrowseForFolderDlg1 (TBrowseForFolderDlg)	cStatusText	Select the root directory
---	-------------	---------------------------

And now we write the events OnInitialize for the form and OnClick for oButton1 and oButton2:

```
//-----
METHOD FormInitialize( oSender ) CLASS TConfig

    WITH OBJECT TIni():New( ".\Visor.ini" )
        :oEdit1:Value := :GetEntry( "CONFIG", "Directorio", "" )
        :oTrackBar1:nValue := :GetEntry( "CONFIG", "Zoom", 0 )
    END

RETURN Nil

//-----
METHOD Button1Click( oSender ) CLASS TConfig

    IF ::oBrowseForFolderDlg1:Run()
        :oEdit1:Value := ::oBrowseForFolderDlg1:cRoot
    ENDIF

RETURN Nil

//-----
METHOD Button2Click( oSender ) CLASS TConfig

    WITH OBJECT TIni():New( ".\Visor.ini" )
        :SetEntry( "CONFIG", "Directory", ::oEdit1:Value )
        :SetEntry( "CONFIG", "Zoom", ::oTrackBar1:nValue )
    END
    :Close()

RETURN Nil

//-----
```

As you can see, we are using an .INI file to save the configuration, with only one section (CONFIG) and two entries (Directory and Zoom). This file is read in the form's OnInitialize event, and it's saved with the OnClick event of oButton2. In the OnClick event of oButton1, we simply make a call to the standard Window's dialog BrowseForFolder using the component oBrowseForFolder and then we copy the selected folder name into the oEdit1 control.

We are done by now with the configuration form. Now is time to complete the main form with a menu, that among other things, it's going to call to the configuration form. It's also necessary to retrieve the configuration at the beginning of the program.

Actually it's not possible to design a menu from within the IDE, so, to write it we are going to use the form's event OnInitialize to write it down. The syntax is quite similar to FW's one. We also take advantage of the same event to read the configuration, so the event would look like this:

```
METHOD FormInitialize( oSender ) CLASS TVisor

    LOCAL nZoom

    MENU ::oMenu
```

```
MENUITEM "&Archivo"
MENU
  MENUITEM "&Abrir"      ACTION ::Button1Click()
  MENUITEM "&Configurar" ACTION TConfig():New( Self ):ShowModal()
  SEPARATOR
  MENUITEM "&Salir"      ACTION ::Close()
ENDMENU
ENDMENU

WITH OBJECT TIni():New( ".\Visor.ini" )
  :oFileOpenDlg1:cInitialDir := :GetEntry( "CONFIG", "Directory", "" )
  nZoom := :GetEntry( "CONFIG", "Zoom", 0 )
END
DO CASE
  CASE nZoom == -2
    :oImage1:nZoom := 25
  CASE nZoom == -1
    :oImage1:nZoom := 50
  CASE nZoom == 1
    :oImage1:nZoom := 200
  CASE nZoom == 2
    :oImage1:nZoom := 400
ENDCASE

IF !Empty( Application:aArguments )
  :oImage1:oPicture := Application:aArguments[1]
ENDIF

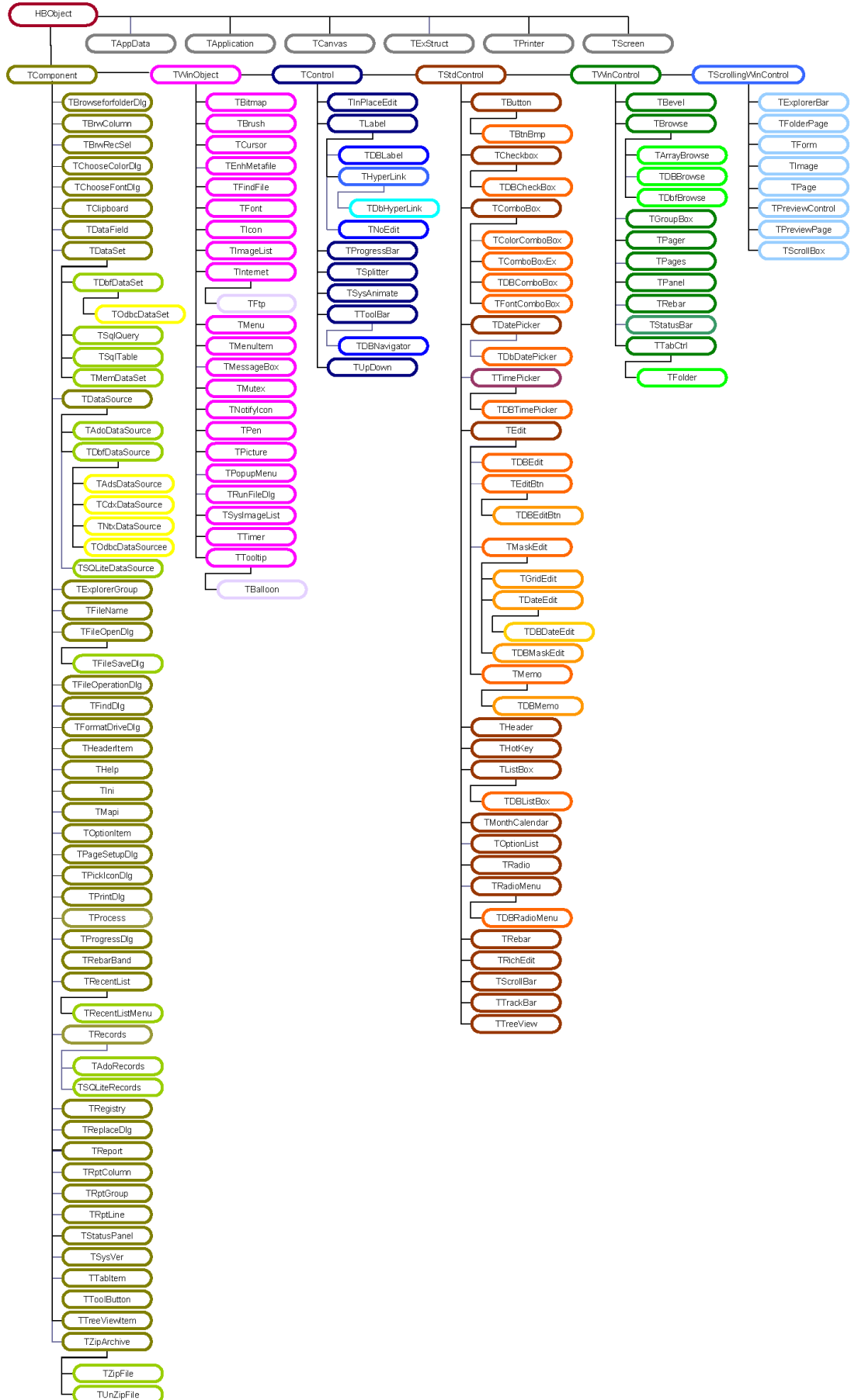
RETURN Nil
```

And that's all. Compile and test the program, you will notice that even this is a very simple program, it is fully operative and we have invested just a little effort to write it, the best of all is: we have done everything from the Xailer's IDE without using any external tool.

We hope that this guide had been useful for you to know and start working with Xailer. Of course this is not a reference manual, but it is a little document to introduce you in the use of this tool. Be aware that no matter the IDE is a very useful tool for programming, a program will do nothing unless you write the necessary code to make it work.

1.3 Classes Hierarchy

JERARQUIA PROVISIONAL DE CLASES GUI DE XAILER (Última actualización 2007.04.24)



1.3.1 Object Oriented Programming

Xailer provides an engine to develop systems, and all its functionality is based in Classes. Xailer takes full advantage of the [x]Harbour Object Oriented Programming capacities and in some aspects improve those capacities.

1.3.1.1 Previous knowledge

You need to have basic OPP knowledge to start working with Xailer. Otherwise the learning curve can be slow. If you don't have any previous experience with Object Oriented Programming, we recommend you to read *generic* OOP information (and not OOP information related to one specific programming language). Then, you can read the Clipper and/or [x]Harbour syntax to use classes to have an idea. You don't need to read in detail the Clipper/[x]Harbour classes because it might be that you will not use them.

1.3.1.2 The classes structure

Almost all the components and Xailer hierarchy classes are implemented using two main classes: the T???? class and the X???? class.

- The X???? class develops the component (it includes all the code).
- The T???? class is empty and it simply come from it X???? corresponding class.

Xailer manages the classes in this way to allow to the user to add or modify the properties or methods of any class, without the need to modify the original code source.

For example, let's suppose that we want to modify the TEdit control to have always a yellow background color. You need only to add in our program the following:

```
CLASS TEdit FROM XEdit
PUBLISHED:
    PROPERTY nClrPane INIT clYellow
ENDCLASS
```

And after that, you can use the TEdit class normally.

1.3.1.3 The properties

The properties are class members that contain any [x]Harbour value type, including other objects. They are declared with the reserved word PROPERTY.

The main difference with the normal DATA is that the properties support the READ and WRITE clauses, that will have the code that must be executed when the property is read or assigned. Normally this code will be only a call to the class' method (Set???? for WRITE clause and Get???? for READ). The WRITE clause will receive the value parameter that will have the value that is assigned to the property.

With the properties, always there will be an associated DATA, with the same property's name, but starting with F and a private scope. It is only accessible through the same class where the

property it is declared. This associated data is needed because if you try to assign the property's value from the Set??? method, it will be calling itself, recurrently, and the property's value will never be assigned. In this way, when it is needed to save the property's value from the Set???? method, it will be done assigning its F???? DATA associated.

Forexample:

```
PROPERTY cText INIT "" READ GetText WRITE SetText

METHOD GetText()
  IF ::Handle != 0
    ::FcText := GetWindowText( ::Handle )
  ENDIF
RETURN ::FcText

METHOD SetText( cText )
  ::FcText := cText
  IF ::Handle != 0
    SetWindowText( ::Handle, ::cText )
  ENDIF
RETURN ::FcText
```

In this case, the cText property and its private associated FcText DATA are created. When it tries to read the property, it calls to the GetText() method. When it tries to assign, it calls to the SetText() method. In both methods we can see that the value that we want to assign to the property is stored in ::FcText.

Besides the READ and WRITE clauses, the properties support the following clauses as well:

- **INIT:**
Allows to indicate the initial property's value, that is assigned automatically every time that the class object is instantiated.
- **AS ????:**
Allows to indicate the data type that it will contain the property. It does not mean that it might have another data type (it is not allowed by the language). It is used by the IDE only to know the property's value type.
- **VALUES:**
Indicates the possible property's list of values. This information is only used by the IDE.
- **EDITOR:**
Allows to indicate the properties editor to be used from the IDE. This information is only used by the IDE.

The READ and WRITE clauses, can be assigned in three different ways:

- **READ METHOD <cMethodName> and WRITE METHOD <cMethodName>.** Through this syntax, the method name is passed as a literal. It is not needed to indicate the receiving parameters. The pointing methods are should be developed on the same class. This assign form provides the fastest execution time.

```
PROPERTY cText READ METHOD GetText WRITE METHOD SetText
```

- **READ <cMethodName> and WRITE <cMethodName>.** It needs to be used when the pointing methods have been developed in a ancestor class. The speed offered by this method is good,

but lower than the first described form.

```
PROPERTY oFont READ GetFont WRITE SetFont
```

- The third form assigns directly the code to be executed on-line. This is the less efficient form in terms of speed.. However its use is very common due it avoids to create an additional method and besides that there is not any implication, like the first 2 forms (see note below).

```
PROPERTY cTag READ INLINE ::GetTag() WRITE INLINE ::FcTag :=
::SetTag( Value )
```

Important Note:

Take into consideration that if you use any of the 2 first assign forms, if you overload those methods in lower level classes, you should overload also the property. Otherwise, the lower level method will never be executed.

The properties can have the following scopes:

- PUBLISHED:
They are accessible from any part of the program and they appear in the IDE object inspector.
- PUBLIC:
They are accessible from any part of the program, but they will not appear in the IDE object inspector.
- PROTECTED:
They are accessible from inside the class and from any part that is derived from it, but they are not accessible from any other part of the program.
- PRIVATE:
They are only accessible from inside the class, and they are not accessible neither from other inherit classes or from any other part of the program.

And they can be READONLY. It means that they can be only assigned from inside the class, and out of the class they can be only read.

For the other side, the properties are "inherited". It means, any derived class, inherits all its properties. The PRIVATE properties will not be accessible. This inheritance includes all the properties characteristics and if they are declared in the 'daughter' class, only the declared characteristics will be overwritten, keeping the other characteristics intact. This include the property accessibility.

Forexample:

```
CLASS TFather
PUBLIC:
    PROPERTY First INIT One WRITE ::SetFirst( Value ) VALUES One, Two,
    Three
PRIVATE:
    PROPERTY Second INIT ""
ENDCLASS

CLASS TDaughter FROM TFather
PUBLIC:
    PROPERTY First INIT Two // It initialises with other value and
    keeps // the WRITE and VALUES clauses
```

```
PROPERTY Second WRITE ::SetSecond( Value ) // It changes the scope
to PUBLIC,                                     // Adds the WRITE
clause and                                     // Keeps the INIT
clause
ENDCLASS
```

In all the Xailer documentation you will see that we use the following symbology to define the properties and methods types. Take into consideration that the property section covers all the public or published data class, for both DATA and PROPERY types.

1.3.1.4 The components

The components are basically DATAS but they differ in some aspects:

- They do not admit any clause on its declaration
- You can only declare one component at a time
- They are displayed boldface on the debugger
- The IDE itself use them to distinguish from user DATAS.

They are declared with the reserved word **COMPONENT**.

1.3.1.5 The events

The events are an special class members. They are assignable like DATA and they are executable like the methods, but they can not be read.

They are declared with the reserved word **EVENT**.

They are used in the program to indicate a code object that need to be executed in an specific moment (normally a control status change or a user action). They are used from the classes to execute that code.

It is needed to indicate the event parameters, but they are used by the IDE to generate the corresponding method in the form.

The first event parameter is always **oSender**. This is the object that cause the event. In other words, the class object that has the event declared. This parameter is set by Xailer directly, even when the class that call the event does not pas the parameter (it does not really need to pass it).

An event can be assigned with a codeblock, a form's method name or a pointer to a method in the form.

When the codeblock is assigned, it can have the code we want, referencing to anything, in any scope visibility, but always will receive oSender as first parameter.

When a method name or a method pointer is assigned, it needs to belong to the class that is assigning the event. Normally is a form that comes from the TForm class.

If you assign a method name or a method pointer, it will execute faster that if you assign a codeblock. In any case, the third method (a pointer to the method) is a little bit complicated, and

we recommend that you don't use it.

The IDE allows to assign a codeblock if you write a string that begins with ({}). If you double click the event cell, or if you write an identifier, the IDE will create automatically a method in the form to write the event's code, and it assigns the name of this method to the event.

Forexample:

In the clause:

```
CLASS TButton FROM TStdControl
    ...
    EVENT OnClick( oSender )
    ...
ENDCLASS
```

```
METHOD Click() CLASS TButton
    :OnClick()
RETURN Nil
```

In the form:

```
CLASS TMyForm FROM TForm
    CONSTRUCTOR Create( oParent )
    METHOD Button1Click( oSender )
ENDCLASS

METHOD Create( oParent )
    ...
    // Method 1: codeblock
    :oButton1:OnClick := { | oSender | :Button1Click( oSender ) }
    // Method 2: Form's methods's name
    :oButton1:OnClick := "Button1Click"
    // Method 3: pointer to the form's method
    // You need to make it in this way to make it works in xHarbour
    :oButton1:_OnClick( @TMyForm_Button1Click )
RETURN Self

METHOD Button1Click( oSender )
    MsgInfo( "Click!" )
RETURN Nil
```

1.3.1.6 The forms

The forms are any kind of windows, including dialogs, managed by Xailer.

The IDE always create a form as a inherit **TForm** class, that is the base Xailer form class. It can be created using the DEFINE FORM command.

The dialogs are another kind of forms, only setting the property nBorderStyle = bsDIALOG. Xailer does not use the native windows dialogs, and the coordinate system is measured in pixels, same as the rest of the form types. In this way it does not need to support the double control behavior in windows and in dialogs, and all will behave in the same way.

The forms created by the IDE always have the `CreateForm()` method, that contains all the form controls, components, properties and methods. This method is developed in a file with the same name than the `.PRG` file but with the `.XFM` extension. You can open this text file and you will see the method's source code. Please don't modify this file, because it can produce some errors when the IDE tries to load it. Besides that, If the file is open from the IDE, in the moment that you save the file, all the changes you have made manually in the file will be disregarded, because the IDE generates the file based in the form designed. If you need to assign any control's property that the IDE does not allows to do it, it is possible to use the `OnInitialize` form event, that is executed just after all the components are created, but before to show the form.

In the form's class declaration, you will see all the form components and they are identified with the reserved `COMPONENT` word. This reserved word is `DATA` synonym, but `COMPONENT` is used to differentiate from other `DATA` in the form.

You will see in the class the methods that correspond to the events that have been assigned, due the IDE creates them in this form by default.

The form inherits from the `TForm` class, and we can overload all the methods we need, but we recommend not to do it, and use the events for this purpose, unless you know very well what you are doing.

1.3.1.7 Symbology

Properties:

■	Read Only property. It can not modified neither in design and run-time.
■	Property that can be assignable in both design and run-time.
■	Property that can be only assigned in design time o before to call its Create method. It is possible that the system allows to modify the property during run-time, however it will not provide the desired effect. You will need to destroy the object and create it again.
■	Property that can be only assigned during run-time.

Methods:

■	Class constructor method.
■	Standard method that can be used in any moment after the object has been instanced with its basic constructor (normally New).
■	Method that can only be executed in the moment that the object has been instanced and created with its Create method.

1.3.2 TComponent

Is the base class of all the Xailer class (the mother of all the classes)

Description:

The TComponent class provides to Xailer objects the following functionality:

- Property persistency
- Capacity to appear in the component palette and possibility to be manageable by the form designer and the object inspector.

TComponent can be used like base class to create new non visual components that need persistency or can be managed from the component palette.

To create non visible components in run-time, use the TControl class as base class. To create visual windows type components, use as base class TStdControl. If it needs to contain other controls, then use the TWinControl class.

Hierarchy Inherits from HObject ([x]Harbour basic class)
See also TControl, TStdControl, TWinControl
File name \source\Component.prg

1.3.2.1 TComponent.Properties

■ read Only ■ Assignable ■ Design assignable ■ Run-time assignable

Scope	Name	Type	Initial value
■	Cargo	Any	NIL
■	ICreated	Logic	.F.
■	oParent	Object	NIL

1.3.2.1.1 TComponent:Cargo

Class member for discrete use.

Scope:	Assignable
Type:	Any
Initial value:	NIL

Description:

Data container defined by the user. It can store any data type and is ignored by the object.

1.3.2.1.2 TComponent:ICreated

Indicates if the object has been create. In other words, if its Create() method has been called. Specifies the control name as is referenced in the source code.

Scope:	read Only
Type:	Logic
Initial value:	.F.

Description:

All the Xailer objects have a instance system in two phases. The first phase is responsible to create an object instance and initializes its properties and data values. The constructor method `New()` is used always in this phase. The second phase allows to execute a second create operation that might affect other control or object that depend from this, or even, affect the value initialization in a particular way in each object. The method `Create()` is used in this second phase. The `ICreated` property allows to know if the object has developed his complete instance. In other words, if the `Create()` method has been already executed.

1.3.2.1.3 TComponent:oParent

Proprietary object or present object's father.

Scope:	Assignable
Type:	Object
Initial value:	NIL

Description:

The `oParent` property references to the proprietary object that created the object. For example, the `oParent` data from a control, will be the form which it belongs.

1.3.2.2 TComponent.Methods

■ Constructor ■ Standard ■ Only after Create()

Typ	Name
■	Create
■	Destroy
■	End
■	EventAssigned
■	Free
■	New

1.3.2.2.1 TComponent:New

Default class constructor for all Xailer classes.

Type	Constructor
Parameters	[<oParent>]: Reference to an container object

Return value	Self Reference
---------------------	----------------

Description:

This method is the constructor normally used to instance any Xailer object.

1.3.2.2.2 TComponent:Create

Additional constructor for all the Xailer classes based in this component.

Type	Constructor
Parameters	[<oParent>]: Reference to the container object
Return value	Self Reference

Description:

This construct method is different than the `New()` constructor, because in the inherited classes the construct method is the responsible to instance the possible Windows component that manages the object. In other words, the `New()` constructor creates the object at Xailer's level, but is when the `Create()` method is executed when the windows object is really created.

As this method is a constructor as well, it is possible to instance directly an object, but it prevents the assignment to some properties that can only be assigned before to call the `Create()` method.

When this method has been executed, the `ICreated` takes a `.T.` value.

1.3.2.2.3 TComponent:Destroy

Destroys the windows component associated to the object, if any.

Type	Estándar
Parameters	None
Return value	NIL

Description:

This method destroys the Windows component associated to the Xailer's object, but it does not destroy the Xailer object. Destroy is opposite to the `TComponent:Create` method.

If you want to destroy the object completely, you should use the `TComponent:End` method.

1.3.2.2.4 TComponent:End

Destroys the object.

Type	Standard
Parameters	None
Return value	NIL

Description:

This method is the generic Xailer's object destroyer. It frees all the memory used by the object.

End() calls internally to the TComponent:Destroy method, just in case that a Windows component needs to be destroyed.

The compilers used by Xailer might not have the functionality of the destroyer methods, and is the **trash** who is in charge to release the object memory that are out of scope. Xailer includes this destroyer to manage memory releases that the trash could not manage and in the case that in the future a compiler might support this functionality.

1.3.2.2.5 TComponent:EventAssigned

Indicates if an event has been assigned.

Type	Standard
Parameters	Methodname
Return value	Logic

Description:

This method us used to check if an class event has been assigned.

1.3.2.2.6 TComponent:Free

This method is the responsible to release all the resources and the GDI objects that the object has assigned.

Type	Standard
Parameters	None
Return value	NIL

Description:

This method is generic Xailer resource releaser.

Free() is called internally during the destroy process of WMDestroy() controls.

You need to **overload** this method in its own controls to free all the GDI objects that you can create for your own use.

1.3.3 TWinObject

This is the base class for all the objects that share a handle to other internal system component.

Description:

The TWinObject class allows to the Xailer objects to share a reference number (handle) to the internal operating system components. This handle is used internally by descending classes to manage any Windows native component.

To create visible components during run-time, use the TControl as base class. To create visual Windows type controls, use TStdControl. If it contains other controls, then use the TWinControl as base class.

Hierarchy	Inherits from TComponent
See also	TControl, TStdControl, TWinControl
File name	\source\WinObject.prg

1.3.3.1 TWinObject.Properties

■ read Only ■ Assignable ■ Design Assignable ■ Run-time assignable

Scope	Name	Type	Initial value
■	Handle	Numeric	0

1.3.3.1.1 TWinObject.Handle

Provides access to the internal component from the operating system associated to this handle.

Scope	read Only
Type	Numeric
Initial value	0

Description:

Use the handle if you need to make a direct call to the Windows API functions to manage the operating system component associated to the object. Use the **Handle** property as a parameter when you make calls to the Windows API functions.

Try to use always an method or property from Xailer, if it exists, before to make a direct call to the Windows API functions.

1.3.4 TControl

This is the base class for all the Xailer's visual objects.

Description:

The TControl class allows to Xailer's objects the needed base to use any Windows visual component. This class is not used directly.

TControl can be used as the base class to create new components that can support other operating system visual components.

To create Window type visual components, use as base class the TStdControl and if besides this must contain other controls, use the TWinControl class.

Hierarchy Inherits from TWinObject
See also TStdControl, TWinControl
File name \source\Control.prg

1.3.4.1 TControl.Properties

■ read Only ■ Assignable ■ Design assignable ■ Run-time assignable

Scope	Name	Type	Initial value
■	HelpID	Numeric or Character	NIL
■	cBalloon	Character	""
■	cText	Character	""
■	cTooltip	Character	""
■	cTooltipTitle	Character	""
■	IAlignKeepSpace	Logic	.F.
■	IAlignShiftOnRotate	Logic	.F.
■	IEnabled	Logic	.T.
■	IFocused	Logic	.F.
■	IParentFont	Logic	.T.
■	IRedraw	Logic	.T.
■	IVisible	Logic	.T.
■	nAlign	Numeric	aNONE
■	nAlignMarginBottom	Numeric	0
■	nAlignMarginLeft	Numeric	0
■	nAlignMarginRight	Numeric	0
■	nAlignMarginTop	Numeric	0
■	nAlignWeight	Numeric	0
■	nAlignWeightSibling	Numeric	0
■	nAnchors	Numeric	akLEFTTOP
■	nBorderSides	Numeric	bsALL
■	nClientHeight	Numeric	CW_USEDEFAULT

■	nClientLeft	Numeric	CW_USEDEFAULT
■	nClientTop	Numeric	CW_USEDEFAULT
■	nClientWidth	Numeric	CW_USEDEFAULT
■	nClrPane	Numeric	cIBTNFACE
■	nClrText	Numeric	cWINDOWTEXT
■	nHeight	Numeric	CW_USEDEFAULT
■	nLeft	Numeric	CW_USEDEFAULT
■	nToolTipIcon	Numeric	tiNOICON
■	nTop	Numeric	CW_USEDEFAULT
■	nWidth	Numeric	CW_USEDEFAULT
■	oBrush	Object	NIL
■	oCursor	Object	NIL
■	oFont	Object	NIL
■	oForm	Object	NIL
■	oParent	Object	NIL

1.3.4.1.1 TControl:HelpID

External help identifier for every control in the form.

Scope:	Assignable
Type:	Numeric o Character
Initial value:	NIL

Description:

This property allows to identify a control with a unique help system ID that can be used by the application. See also THelp for more information.

1.3.4.1.2 TControl:cBalloon

Text to show in a balloon type windows when the user stops the mouse pointer over the control for certain period of time.

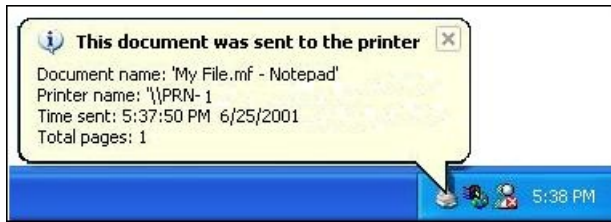
Scope:	Assignable
Type:	Character
Initial value:	""

Description:

The balloon windows type are standard operating system windows used to display context information.

See also the cTooltip, cTooltipTitle and nToolTipIcon properties

Image:



1.3.4.1.3 TControl:cText

Control's visible text.

Scope:	Assignable
Type:	Character
Initial value:	""

Description:

This property is used to indicate the text to be visible in the control and it is used for its descending classes to show the text in the screen. Depending on the control type, the text can be shown in different ways. For example, a TForm control type, descending from TControl, will show the value 'cText' in the upper window bar. On the other side, it is possible that some inherited controls from TControl, even when they have this property, it will not be used, like TUpDown.

1.3.4.1.4 TControl:cTooltip

Text to be shown in a rectangle window type with context information, when the user stops the mouse pointer on the control for a certain period of time.

Scope:	Assignable
Type:	Character
Initial value:	""

Description:

The Tooltip window type are standard operating system windows used to show context information.

See also the cBalloon, cTooltipTitle and nToolTipIcon properties.

Image:



1.3.4.1.5 TControl:cTooltipTitle

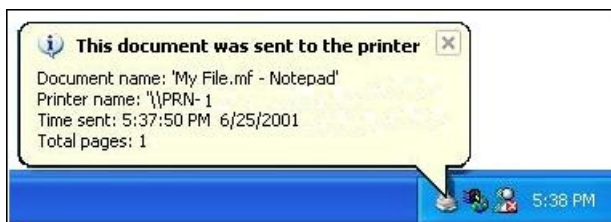
Indicates the possible title for the Tooltip or Balloon windows type.

Scope:	Assignable
Type:	Character
Initial value:	""

Description:

The Tooltip or Balloon Windows are standard operating system windows that are used to show context information. Those windows will show by default only a text (cBallon or cTooltip), but it is possible that they might have a title assigning this property.

See also the cBallon, cTooltip and nToolTiplcon properties

Image:

1.3.4.1.6 TControl:IAlignKeepSpace

If true, even if the control is not visible, its space will be maintained for alignment purposes on its container.

Scope:	Assignable
Type:	Logic
Initial value:	.F.

See also: nAlign, nAlignWeight, IAlignShiftOnRotate

1.3.4.1.7 TControl:IAlignShiftOnRotate

If true, when the screen orientation changes, its alignment also changes this way:

- aLEFT => aTOP
- aTOP => aLEFT
- aRIGHT => aBOTTOM
- aBOTTOM => aRIGHT
- aNONE and aCLIENT are not affected

Initially alignment is for landscape screens, where width is greater than height.

Scope:	Assignable
Type:	Logic
Initial value:	.F.

See also: `nAlign`, `nAlignWeight`, `lAlignKeepSpace`

1.3.4.1.8 TControl:Enabled

Property that indicates if the control is enabled.

Scope:	Assignable
Type:	Logic
Initial value:	.T.

Description:

This property allows to enable or disable the control state. When the control is disabled it is not possible to select it and even when it is visible in some cases it is possible that its visual presentation changes.

1.3.4.1.9 TControl:Focused

This property indicates if the control currently has the focus.

Scope:	readOnly
Type:	Logic
Initial value:	.F.

Description:

This is a read Only property that allows to know if the control has the focus, it means, if it is the current selected control and it received, for example, all the pressed keys.

1.3.4.1.10 TControl:ParentFont

Indicates if the control uses its `oParent` object container font.

Scope:	Design assignable
Type:	Logic
Initial value:	.T.

Description:

Every control can have its own font to be used to visualize any text that it might have. However, by default is used the font used by its oParent container object, if exists, or if this property is set to .F..

This technique of using the same font from its container object allows to reduce the use of GDI font type in an application.

Take into consideration that if you modifies the control font and its IParentFont property is set to .T., you are modifying not only the font for this control but its container object font and all the control that will use it.

1.3.4.1.11 TControl:IRedraw

Property that indicates if the control must be repainted when it receives a repaint notification.

Scope:	Assignable
Type:	Logic
Initial value:	.T.

Description:

This property allows to cancel temporarily the control's repaint operation, that can be very useful when there are multiple modification in the same that might affect its visual aspect, for example, when we add multiple columns in an Browse type object.

1.3.4.1.12 TControl:IVisible

Indicates if the control is visible or not.

Scope:	Assignable
Type:	Logic
Initial value:	.T.

Description:

This property allows to modify the control status to visible or invisible.

1.3.4.1.13 TControl:nAlign

Control alignment in its oParent object container.

Scope:	Assignable
Type:	Numeric
Initial value:	aINONE
Possible	aINONE, aLEFT, aTOP, aRIGHT, aBOTTOM,

values: aICLIENT

Description:

This property allows to the controls to adjust its dimensions and position to its oParent object container. The alignment can be:

- **None:** Default value
- **Left:** The control is aligned to the left of its oParent control and it takes the height of its client container.
- **Upper:** The control is aligned to the top of its oParent control and it takes the width of its client container.
- **Right:** The control is aligned to the right of its oParent control and it takes the height of its client container.
- **Bottom:** The control is aligned to the button of its oParent control and it takes the width of its client container.
- **Client:** The control is aligned the client area of its oParent control adjusts itself its size.

1.3.4.1.14 TControl:nAlignMarginBottom

Control bottom margin when is aligned with the property nAlign.

Scope:	Assignable
Type:	Numeric
Initial value:	0

See also: nAlignMarginLeft, nAlignMarginRight, nAlignMarginTop

1.3.4.1.15 TControl:nAlignMarginLeft

Control left margin when is aligned with the property nAlign.

Scope:	Assignable
Type:	Numeric
Initial value:	0

See also: nAlignMarginBottom, nAlignMarginRight, nAlignMarginTop

1.3.4.1.16 TControl:nAlignMarginRight

Control right margin when is aligned with the property nAlign.

Scope:	Assignable
Type:	Numeric

Initial value: 0

See also: `nAlignMarginLeft`, `nAlignMarginBottom`, `nAlignMarginTop`

1.3.4.1.17 TControl:nAlignMarginTop

Control top margin when is aligned with the property `nAlign`.

Scope: Assignable

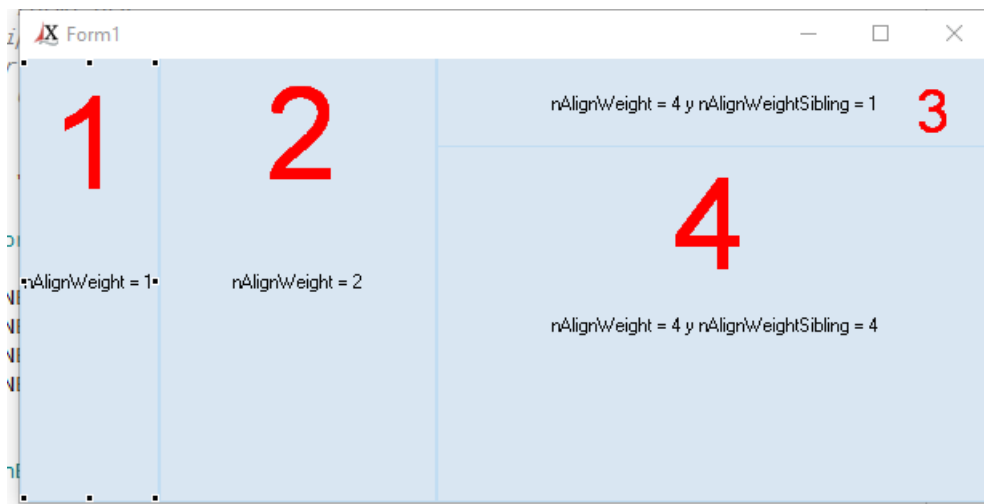
Type: Numeric

Initial value: 0

See also: `nAlignMarginLeft`, `nAlignMarginRight`, `nAlignMarginBottom`

1.3.4.1.18 TControl:nAlignWeight

It sets the 'relative weight' of the control over other controls that have the same alignment as it and in is opposite dimension (and that are created correlatively), in its container object.



When the **nAlignWeight** property is set, its non-automatic size property loses utility and is also calculated based on the weight indicated with this property. An example is the most appropriate to explain it:

In the image above you can see four panels aligned to the left. When the form has only one left panel, the height of the form is calculated automatically and therefore its **nHeight** property is as if it were read only, because any value you want to set it is useless. However its **nWidth** property works correctly and we can set the width of the panel to whatever we want. However, it is a fixed size and does not adapt to any changes in its container form.

As soon as you set the **nAlignWeight** property to any value other than zero, the **nWidth** property will be useless as it will be calculated automatically. In the image you can see the first three

panels with values on this property of 1, 2 and 4, which means That the third panel will occupy four times the first and twice what the second. That is, the indicated weights are distributed among all the panels with the same alignment. And in all of them their **nWidth** property happens to be useless.

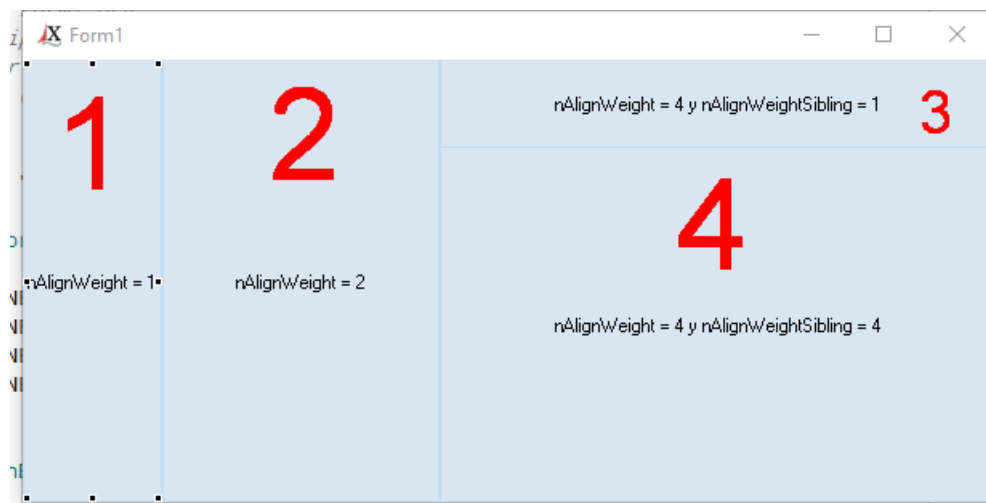
Note on the system of weights: The system of weights allows to establish by percentages the relative weight of each panel in the dimension that we want to adjust. In the example, for **nAlignWeight**, the weights are 1, 2, and 4. Therefore the panels are to be distributed based on a maximum weight of 7 which is the sum of the weights of the three panels. Therefore 7 would be 100%, 1 would be 14.28%, 2 would be 28.57% and 4 would be 57.14%.

Scope:	Assignable
Type:	Numeric
Initial value:	0

See also: **nAlignWeightSibling**, **nAlign**, **IAlignKeepSpace**, **IAlignShiftOnRotate**

1.3.4.1.19 TControl:nAlignWeightSibling

It establishes the 'relative weight' of the control over other controls that have the **same** alignment as it (and that are created correlatively), in its container object in the same dimension that marks its alignment.



When the **nAlignWeightSibling** property is set, its size property related with its alignment is also calculated based on the weight indicated with this property.

Note on the system of weights: The system of weights allows to establish by percentages the relative weight of each panel in the dimension that we want to adjust. In the example, for **nAlignWeightSibling**, the weights are 1 and 4. Therefore the panels are to be distributed based on a maximum weight of 5 which is the sum of the weights of the two panels. Therefore 5 would be 100%, 1 would be 20.00% and 4 would be 80.00%.

Consult **nAlignWeight** property for further information.

Scope:	Assignable
Type:	Numeric
Initial value:	0

See also: `nAlignWeight`, `nAlign`, `lAlignKeepSpace`, `lAlignShiftOnRotate`

1.3.4.1.20 TControl:nAnchors

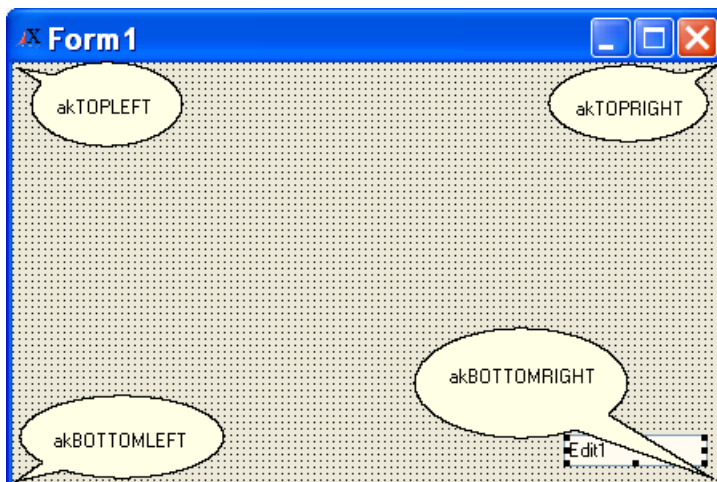
Anchors for the control in its oParent container object.

Scope:	Assignable
Type:	Numeric
Initial value:	<code>akLEFTTOP</code>

Description:

This property allows to the control to use different values as visualization coordinates, different to the classic row and column. Normally every control is anchored in an specific row and column, showing the control from left to right respecting to the indicated column, and from top to down respecting to the row indicated.

The `nAnchors` property allows to anchor the control to other different coordinates and instead to use the upper-left corner as a reference point, it is possible to use, for example the lower-right corner and it will allow that the control is moved in the same relative position when we modify the coordinates of its form.



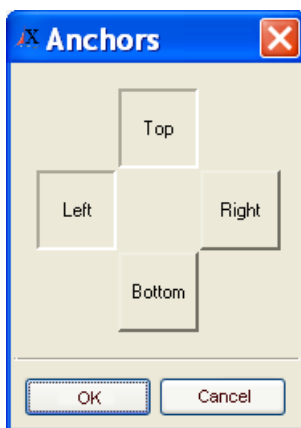
And besides that, it is possible to establish more than one anchor point and the control can change its size when its container object changes.

The possible values for this property are:

<code>akNONE</code>	0
<code>akLEFT</code>	1
<code>akTOP</code>	2

akRIGHT	4
akBOTTOM	8
akLEFTTOP	3
akLEFTRIGHT	5
akLEFTBOTTOM	9
akTOPLEFT	3
akTOPRIGHT	6
akTOPBOTTOM	10
akRIGHTLEFT	5
akRIGHTTOP	6
akRIGHTBOTTOM	12
akBOTTOMLEFT	9
akBOTTOMTOP	10
akBOTTOMRIGHT	12
akLEFTTOPRIGHT	7
akBOTTOMTOPRIGHT	14
akBOTTOMTOPLEFT	11
akBOTTOMRIGHTLEFT	13
akALL	15

However you might not need to indicate this properties directly, due the IDE provides an specific property editor for this property and it is very intuitive and easy to use. You only need to indicate the extremes that you need to use as anchors



1.3.4.1.21 TControl:nBorderSides

Border to paint when **nBorderStyle** is different from bvNONE.

Scope:	Assignable
Type:	Numeric
Initial value:	bsALL
Possible values:	bsNONE, bsLEFT, bsTOP, bsLEFTTOP, bsRIGHT, bsLEFTRIGHT, bsTOPRIGHT, bsLEFTTOPRIGHT, bsBOTTOM, bsLEFTBOTTOM, bsTOPBOTTOM, bsLEFTTOPBOTTOM, bsRIGHTBOTTOM, bsLEFTRIGHTBOTTOM, bsTOPRIGHTBOTTOM, bsALL

This property, as well the **nBorderStyle** property is only visible on some descendant classes, like: TLabel or TBevel.

1.3.4.1.22 TControl:nClientHeight

Control's client zone height. The client zone corresponds to its interior.

Scope:	Assignable
Type:	Numeric
Initial value:	CW_USEDEFAULT

See also nClientTop, nClientWidth, nClientLeft

1.3.4.1.23 TControl:nClientLeft

Y coordinate from the control's client zone.

Scope:	Assignable
Type:	Numeric
Initial value:	CW_USEDEFAULT

See also nClientHeight, nClientTop, nClientWidth

1.3.4.1.24 TControl:nClientTop

X coordinate from the control's client zone.

Scope:	Assignable
Type:	Numeric
Initial value:	CW_USEDEFAULT

See also nClientHeight, nClientLeft, nClientWidth

1.3.4.1.25 TControl:nClientWidth

Control's client zone width.

Scope:	Assignable
Type:	Numeric
Initial value:	CW_USEDEFAULT

See also `nClientHeight`, `nClientTop`, `nClientLeft`

1.3.4.1.26 TControl:nClrPane

Control background color.

Scope:	Assignable
Type:	Numeric
Initial value:	<code>clBTNFACE</code>

Consult the appendix for the list of available colors

1.3.4.1.27 TControl:nClrText

Control text color.

Scope:	Assignable
Type:	Numeric
Initial value:	<code>clWINDOWTEXT</code>

Consult the appendix for the list of available colors

1.3.4.1.28 TControl:nHeight

Indicates the control height.

Scope:	Assignable
Type:	Numeric
Initial value:	<code>CW_USEDEFAULT</code>

1.3.4.1.29 TControl:nLeft

Control Y coordinate.

Scope:	Assignable
Type:	Numeric
Initial value:	<code>CW_USEDEFAULT</code>

Description:

This property indicates the control visualization row.

See also `nTop`, `nWidth`, `nHeight`

1.3.4.1.30 TControl:nTooltipIcon

Icon to be used for the Tooltip and Balloon Pop Up windows.

Scope:	Assignable
Type:	Numeric
Initial value:	tiNOICON
Possible values:	tiNOICON, tiINFO, tiWARNING, tiERROR

Description:

The Tooltip or Balloon Windows are standard operating system windows that are used to show context information. This icon will be shown when there is a windows title defined with the 'cTooltipTitle' property.

See also the properties cTooltip, cTooltipTitle y cBalloon.

1.3.4.1.31 TControl:nTop

Control X coordinate.

Scope:	Assignable
Type:	Numeric
Initial value:	CW_USEDEFAULT

Description:

This property indicates the control visualization column.

See also nLeft, nWidth, nHeight

1.3.4.1.32 TControl:nWidth

Indicates the control width.

Scope:	Assignable
Type:	Numeric
Initial value:	CW_USEDEFAULT

1.3.4.1.33 TControl:oBrush

TBrush to be used to paint the control's background.

Scope:	Assignable
Type:	Object
Initial value:	NIL

Description:

This property allows to establish a 'Brush' object to paint the control background.

1.3.4.1.34 TControl:oCursor

TCursor to be used in the control.

Scope:	Assignable
Type:	Object
Initial value:	NIL

Description:

This property allows to change the cursor that will be used in the control. This cursor will be visible when the mouse pointer is located on the control.

1.3.4.1.35 TControl:oFont

TFont to be used for the text.

Scope:	Assignable
Type:	Object
Initial value:	NIL

Description:

This property allows to change the font to be used in the control when it is painted. By default, every control always will use its container object font if the IParentFont property is set to .T..

See also the TFont class for more information.

1.3.4.1.36 TControl:oForm

TForm control's container object.

Scope:	read Only
---------------	-----------

Type:	Object
Initial value:	NIL

Description:

This property returns the TForm control's container object. The oForm object does not need necessarily to be the same that the oParent object. The oParent object points to the container object, but this can be any control type descending from TWinControl, like a TFolder. However the oFont object always points to the form that has the control.

1.3.4.1.37 TControl:oParent

oParent control's container object.

Scope:	read Only
Type:	Object
Initial value:	NIL

Description:

This property returns the control's container object. The oParent object does not need necessarily to be the same that the oForm object. The oForm object points to the form that contains the object. The oParent object points to the container object, but this can be any control type descending from TWinControl, like a TFolder.

1.3.4.2 TControl.Methods

■ Constructor ■ Standard ■ Only after Create()

Type	Name
■	BeginPaint
■	Create
■	Destroy
■	Disable
■	Enable
■	EndPaint
■	GetSiblingControl
■	GoNextControl
■	GoPrevControl
■	Hide
■	HideBalloon
■	HideTooltip
■	New
■	PostMsg
■	Redraw
■	Refresh
■	SendMsg

❑	SetBounds
❑	SetFocus
❑	Show
❑	ShowBalloon
❑	ShowPopupMenu
❑	ShowTooltip
❑	Update

1.3.4.2.1 TControl:BeginPaint

Method that marks the beginning of a painting process on the control. It is used exclusively from the the WMPaint() method on the self made controls.

Type	Standard
Parameters	<hDC>: Handle to device context
Return value	<hDC>

See also EndPaint method.

Sample:

```

METHOD WMPaint( hDC ) CLASS TMyControl

    LOCAL aRect := GetClientRect( ::Handle )
    LOCAL oCanvas

    hDC := ::BeginPaint( hDC )

    WITH OBJECT oCanvas := TCanvas()
        :hDC := hDC
        :FillRect( aRect, ::nClrPane )
    END

    RETURN ::EndPaint()

```

1.3.4.2.2 TControl:Create

Object constructor and instanter of the possible Windows component associated to the control.

Type	Constructor
Parameters	[<oParent>]: Reference to a container object
Return value	Self reference (Self)

Description:

This constructor method is different to the `New` method because instantiates the associated Windows component.

You can use this method instead of use `New` before, however, this will not allow you to modify some control properties that can only be modified before to instantiate the Windows associated component.

1.3.4.2.3 TControl:Destroy

Destroys the associated object Windows component, if any.

Type	Standard
Parameters	None
Return value	NIL

Description:

This method destroys the associated Windows component to the Xailer object, but it does not destroy the Xailer object. Destroy is inverse to the `Create` method.

If you want to destroy completely the object, you need to use the `End` method.

1.3.4.2.4 TControl:Disable

Disables the control and does not allows its selection with the mouse and most probably changes its aspect to show that condition.

Type	Only after <code>Create()</code>
Parameters	None
Return value	.F.

Description:

This method is called internally for the `IEnabled` property and always is better to use the property directly, so you will barely use this method.

See also the `Enable` method.

1.3.4.2.5 TControl:Enable

Enables the control and allows to select it with the mouse, receives the focus and process events.

Type	Only after Create()
Parameters	None
Return value	.T.

Description:

This method is called internally by the IEnabled property and always is better to use the property directly, then you will barely use this method.

See also the Disable method.

1.3.4.2.6 TControl:EndPoint

Method that marks the end of a painting process on the control. It is used exclusively from the the WMPaint() method on the self made controls.

Type	Standard
Parameters	None
Return value	None

See also BeginPaint method.

Sample:

```
METHOD WMPaint( hDC ) CLASS TMyControl

    LOCAL aRect := GetClientRect( ::Handle )
    LOCAL oCanvas

    hDC := ::BeginPaint( hDC )

    WITH OBJECT oCanvas := TCanvas()
        :hDC := hDC
        :FillRect( aRect, ::nClrPane )
    END

RETURN ::EndPaint()
```

1.3.4.2.7 TControl:GetSiblingControl

Returns the sibling control from itself based on its ordinal creation position.

Type	Only after Create()
Parameters	<nPos> Sibling position. Negative values means before itself, positive values means after itself.
Return value	<oControl> Sibling control or NIL

1.3.4.2.8 TControl:GoNextControl

Selects / changes the focus to the next control in its container form.

Type	Only after Create()
Parameters	None
Return value	NIL

Description:

This method allows to select the next form control with its ITabStop property set to .T..

See also the GoPreviousControl method.

1.3.4.2.9 TControl:GoPrevControl

Selects / changes the focus to the previous control in its container form.

Type	Only after Create()
Parameters	None
Return value	NIL

Description:

This method allows to select the previous form control with its ITabStop property set to .T..

See also the GoNextControl method.

1.3.4.2.10 TControl:Hide

Hides the control.

Type	Only after Create()
Parameters	None
Return value	NIL

Description:

This method is called internally for the IVisible property and always is better to use directly the property, then you barely will use this method directly.

See also the Show method.

1.3.4.2.11 TControl:HideBalloon

Hides the TBalloon control showed by the method ShowBalloon.

Type	Only after Create()
Parameters	None
Return value	NIL

1.3.4.2.12 TControl:HideTooltip

Hides the TTooltip control showed by the method ShowTooltip.

Type	Only after Create()
Parameters	None
Return value	NIL

1.3.4.2.13 TControl:New

Object constructor.

Type	Constructor
Parameters	[<oParent>]: Reference to an container object
Return value	Self reference (Self)

Description:

This constructor method is different to the `Create` constructor because it does not instantiate the associated Windows component and then is possible to modify any control property before to instantiate the Windows component.

1.3.4.2.14 TControl:PostMsg

Puts in the associated application message queue a message and returns immediately without waiting for the system to process the message.

Type	Only after <code>Create()</code>
Parameters	<nMsg> Message identifier <nWParam> First parameter of additional specific message information <nLParam> Second parameter of additional specific message information
Return value	nValue Any value different to zero means success

Description:

This method is exclusively to advanced users and needs a deep Windows API knowledge. You will barely use it but it is used for other Xailer controls in a very specific situations.

This method basically calls to the **PostMessage()** Windows API function. Check it to get more information.

See also the `SendMsg` method.

1.3.4.2.15 TControl:Redraw

Forces immediately a repaint operation in the control's visible area.

Type	Only after <code>Create()</code>
Parameters	None
Return value	ISuccess

Description:

This method allows to force a immediate control repaint calling the **RedrawWindow** Windows API function. Even when the control has not changed or it's current visible area is not invalidated, this method forces a repaint operation. You will barely use this method and always is better to allow to Windows to repaint the control through the `Refresh()` method.

See also Refresh and Update methods.

1.3.4.2.16 TControl:Refresh

Invalidates all the control's visible area to be repainted when the rest of messages have been processed.

Type	Only after Create()
Parameters	None
Return value	ISuccess

Description:

This method allows to invalidate the control painting area calling to the **InvalidateRect** Windows API function. The control will be repainted after other possible messages have been processed. In other words, the painting operation will be the same message to be processed. This is the deal method to use for a normal control repaint.

See also Redraw and Update methods.

1.3.4.2.17 TControl:SendMessage

Puts in the associated application message queue a message and does not return until it is processed.

Type	Only after Create()
Parameters	<nMsg> Message identifier <nWParam> First parameter of additional specific message information <nLParam> Second parameter of additional specific message information
Return value	nValue Any value different to zero means success

Description:

This method is exclusively to advanced users and needs a deep Windows API knowledge. You will barely use it but it is used for other Xailer controls in a very specific situations.

This method basically calls to the **SendMessage()** Windows API function. Check it to get more information.

See also the PostMsg method.

1.3.4.2.18 TControl:SetBounds

Establishes the control coordinates.

Type	Only after Create()
Parameters	<nLeft> Column <nTop> Row <nWidth> Width <nHeight> Height <IRedraw> If .T., it forces a repaint operation
Return value	NIL

Description:

This method allows to change the control coordinates. It is better to use this method instead to change the nLeft, nTop, nWidth, and nHeight properties because the operation is faster, and when the control has defined any alignment or anchor.

1.3.4.2.19 TControl:SetFocus

Establishes the focus on the control. Is possible that the focus can not be changed due the existence of a trapped OnExit event on the control who has focus that rejects it. On that case you may force the change sending a parameter .T. to the method.

Type	Only after Create()
Parameters	[<IForce>] If true it will force the change. By default false
Return value	<IFocused> Returns true if the focus is received

1.3.4.2.20 TControl:Show

Shows the control.

Type	Only after Create()
Parameters	None
Return value	NIL

Description:

This method is called internally for the `IVisible` property and it is recommend that you use the property directly, because you will use barely this method.

See also the `Hide` method.

1.3.4.2.21 TControl:ShowBalloon

Method to show a `TBalloon` on the control. To hide the Balloon you should use the method `HideBalloon`.

Type	Only after <code>Create()</code>
Parameters	<p>[<nPosX>] Mouse X coordinate. By default coordinate 0 of the control</p> <p>[<nPosY>] Mouse Y coordinate. By default coordinate 0 of the control</p> <p><cText> Text to show. Corresponds with property <code>cTooltip</code>.</p> <p>[<cTitle>] Title. Corresponds with property <code>cTooltipTitle</code>.</p> <p>[<nIcon>] Icon to use. Corresponds with property <code>nTooltipIcon</code>.</p>
Return value	<oBalloon> A reference to a <code>TBalloon</code> object

1.3.4.2.22 TControl:ShowPopupMenu

Method to show a on context menu in the control.

Type	Only after <code>Create()</code>
Parameters	<p><oPopup>: Object menu.</p> <p><nPosX>: Mouse pointer X coordinate</p> <p><nPosY> Mouse pointer Y coordinate</p> <p>[<ICanceled>] Parameter passed by reference that becomes <code>.T.</code> when the user cancels the menu with out making any selection</p>
Return value	Logic.

Description:

This method is used to show a Pop Up menu and normally is used together with the

OnContextMenu event.

The returned value is used to now if the Pop Up menu is still active, but it does not mean that we can call the ShowPopupMenu with any parameter.

1.3.4.2.23 TControl:ShowTooltip

Method to show a TTooltip on the control. To hide the Balloon you should use the method HideTooltip.

Type	Only after Create()
Parameters	<p>[<nPosX>] Mouse X coordinate. By default coordinate 0 of the control</p> <p>[<nPosY>] Mouse Y coordinate. By default coordinate 0 of the control</p> <p><cText> Text to show. Corresponds with property cTooltip.</p> <p>[<cTitle>] Title. Corresponds with property cTooltipTitle.</p> <p>[<nIcon>] Icon to use. Corresponds with property nTooltipIcon.</p>
Return value	<oTooltip> A reference to a TTooltip object

1.3.4.2.24 TControl:Update

Forces an immediate update in the visible control's area.

Type	Only after Create()
Parameters	None
Return value	ISuccess

Description:

This method allows to force an instant control paint operation calling to the **UpdateWindow()** Windows API function. Take into consideration that is needed to invalidate the area before with the Refresh method.

See also the Refresh and Redraw methods

1.3.4.3 TControl.Events

Name
OnBeforeCreate
OnBeginDrag
OnClick
OnContextMenu
OnCreate
OnDestroy
OnDestroyed
OnDispTooltip
OnDragOver
OnEndDrag
OnHide
OnLButtonDown
OnLButtonUp
OnMouseMove
OnPaint
OnPainted
OnRButtonDown
OnRButtonUp
OnShow

1.3.4.3.1 TControl:OnBeforeCreate

Event that is produced just before the native API control creation.

Parameters	<oSender>:
:	Object that triggers the event
Return value:	NIL

Description:

This event is produced just when the native API control is going to be created. Is the perfect event to change any control style, that if already created will be impossible to do it, like the alignment on a TLabel control for example.

1.3.4.3.2 TControl:OnBeginDrag

Drag & Drop request operation.

Parameters	<oSender>:
:	Object that triggers the event
	<nPosX>:
	X coordinate (client relative)

	<nPosY>: Y coordinate (client relative)
Return value:	<IOk>: If returns true it means that the operation is permitted and the mouse cursor changes

This event is produced when the user pushes the mouse left button over a control and tries to drag it. If you want to change the default cursors used for Drag & Drop operations you should modify the properties `oCursorDropYes` and `oCursorDropNo` from the public object `Screen`.

See also `OnEndDrag` and `OnDragOver`.

1.3.4.3.3 TControl:OnClick

Event produced when the mouse left button is pushed and also released over the control.

Parameters	<oSender>: Object that triggers the event
:	<nFlags>: Virtual key identifier status. Any combination of the following values: MK_CONTROLCTRL key pushed MK_LBUTTON Mouse left button pushed MK_MBUTTON Mouse center button pushed MK_RBUTTON Mouse right button pushed MK_SHIFT Key CTRL pushed Use <code>lAnd()</code> function to retrieve it
	<nXPos>: X coordinate from the mouse pointer
	<nYPos>: Y coordinate from the mouse pointer.
Return value:	NIL

Some Windows API controls provide this event itself. On the rest of cases Xailer generates the event automatically.

1.3.4.3.4 TControl:OnContextMenu

Request a context menu.

Parameters	<oSender>: Object that triggers the event
:	<nPosX>: Mouse pointer X coordinate
	<nPosY>: Mouse pointer Y coordinate
Return	NIL

value:

Description:

This event is produced when the user right-clicks the mouse over the control requesting the on context menu. Logically this event is used to show the Pop up menu and the coordinates reported are to situate the menu just in that position.

See also ShowPopupMenu.

1.3.4.3.5 TControl:OnCreate

Event that is produced in the control creation.

Parameters	<oSender>:
:	Object that triggers the event
Return value:	NIL

Description:

This event is produced when the control has been created and after creating any native Windows component associated to the control. This event is a good candidate to feed the object with the information needed when the Windows control already exists.

1.3.4.3.6 TControl:OnDestroy

Event that is produced just before the control is destroyed.

Parameters	<oSender>:
:	Object that triggers the event
Return value:	NIL

Description:

This event is produced when the control is going to be destroyed. This method is a good candidate to save some information that is not available when the control is fully destroyed.

1.3.4.3.7 TControl:OnDestroyed

Event that is produced when the control is fully destroyed.

Parameters	<oSender>:
:	Object that triggers the event
Return	NIL

value:

Description:

This event is produced when the control is destroyed and in the resource release process that is done in the 'Free' method. This method is a good candidate to destroy any additional resource that the control will not release automatically.

1.3.4.3.8 TControl:OnDispTooltip

Event that is fired when a Tooltip is going to be shown. This event is only triggered if at least the property cTooltip or cBalloon is already assigned.

Parameters	<oSender>: Object that triggers the event <@cText>: Tooltip text. Passed by reference <@cTitle>: Tooltip title. Passed by reference <@nIcon>: Tooltip icon. Passed by reference <nItem>: If the object that fires the event has 'Items', like TStatusBar for example, it indicates the element number on the items array.
Return value:	NIL

1.3.4.3.9 TControl:OnDragOver

Event produced when the mouse cursor is moved over the control when there is a Drag & Drop operation in process.

Parameters	<oSender>: Object that triggers the event <oFrom>: Control that began the Drag & Drop operation <nPosX>: X coordinate (client relative) <nPosY>: Y coordinate (client relative)
Return value:	<IOk>: If it returns true means that the Drop operation can be done and the mouse cursor changes with the style indicated by oCursorDropYes

If you want to change the default cursors used for Drag & Drop operations you should modify the properties oCursorDropYes and oCursorDropNo from the public object Screen

See also OnBeginDrag y OnEndDrag.

1.3.4.3.10 TControl:OnEndDrag

Event produced when the mouse cursor is released over the control when there is a Drag & Drop operation in process.

Parameters	<oSender>: Object that triggers the event
:	<oFrom>: Control that began the Drag & Drop operation
	<nPosX>: X coordinate (client relative)
	<nPosY>: Y coordinate (client relative)
Return value:	NIL

If you want to change the default cursors used for Drag & Drop operations you should modify the properties oCursorDropYes and oCursorDropNo from the public object Screen

See also OnBeginDrag y OnDragOver.

1.3.4.3.11 TControl:OnHide

Event that is produced when the control is hidden.

Parameters	<oSender>: Object that triggers the event
Return value:	NIL

Description:

This event is produced when the user assigns the IVisible property to .F., or when there is a call to the Hide method.

1.3.4.3.12 TControl:OnLButtonDown

Event that is produced when the user clicks the mouse left button.

Parameters	<oSender>: Object that triggers the event
:	<nPosX>:

	Mouse pointer X coordinate <nPosX> Mouse pointer Y coordinate
Return value:	NIL

Description:

The event is produced after the user click the mouse left button and received the control that triggers the event and the mouse pointer coordinates.

1.3.4.3.13 TControl:OnLButtonUp

Event that is produced when the user releases the mouse left button.

Parameters	<oSender>: Object that triggers the event
:	<nPosX>: Mouse pointer X coordinate
	<nPosY>: Mouse pointer Y coordinate
Return value:	NIL

Description:

The event is produced when the user releases the mouse left button and besides receives the control that triggers the event and the mouse pointer coordinates.

1.3.4.3.14 TControl:OnMouseMove

Event that is produced when the user moves the mouse pointer over the control.

Parameters	<oSender>: Object that triggers the event
:	<nPosX>: Mouse pointer X coordinate
	<nPosY>: Mouse pointer Y coordinate
Return value:	NIL

Description:

The event is produced every time that the mouse is moved over the control, however, this is an event that can be produced several times, then it is important that the code assigned to this events is executed quickly.

1.3.4.3.15 TControl:OnPaint

Event that is produced when the control will be painting.

Parameters	<oSender> : Object that triggers the event
:	<hDC> : Device handle
	<@cPaintStruct> : PaintStruct structure from the Windows API needed for the painting process
Return value:	NIL

Description:

This event allows to change the control's paint behaviour. It is possible to add more details to the control or change any of its properties every time that the control is painted.

To use this event you need to have enough Windows API knowledge, because in most of the cases you will need to use the painting primitives that are provided by Windows API.

1.3.4.3.16 TControl:OnPainted

Event that is produced after the control is painted.

Parameters	<oSender> : Object that triggers the event
:	<hDC> : Device handle
	<@cPaintStruct> : PaintStruct structure from the Windows API needed for the painting process
Return value:	NIL

Description:

This event allows to change the control's paint behaviour. It is possible to add more details to the control or change any of its properties every time that the control is painted.

To use this event you need to have enough Windows API knowledge, because in most of the cases you will need to use the painting primitives that are provided by Windows API.

1.3.4.3.17 TControl:OnRButtonDown

Event that is produced when the user clicks the mouse right button

Parameters	<oSender> : Object that triggers the event
:	<nPosX> : Mouse pointer X coordinate
	<nPosY> : Mouse pointer X coordinate
Return value:	NIL

Description:

The event is produced when the user clicks the mouse right button and receives the control that triggers the event and the mouse pointer coordinates

1.3.4.3.18 TControl:OnRButtonUp

Event that is produced when the user releases the mouse right button.

Parameters	<oSender> : Object that triggers the event
:	<nPosX> : Mouse pointer X coordinate
	<nPosY> : Mouse pointer X coordinate
Return value:	NIL

Description:

The event is produced when the user releases the mouse right button and receives the control that triggers the event and the mouse pointer coordinates.

1.3.4.3.19 TControl:OnShow

Event that is produced when the control is shown.

Parameters	<oSender> : Object that triggers the event
Return value:	NIL

Description:

This event is triggered every time that the control is shown after it has been created by a call to the Show method.

1.3.5 TStdControl

This is the base class of all the Xailer visual objects.

Description:

The TStdControl provides to the Xailer object the capacity to show the visual controls with the difference with their ascending classes that they can receive the focus (ITabStop property) and the possibility of assigning a border (IBorder property).

To create components that are always visible during runtime, use the TControl class as base class. To create Windows type visual controls that will contain other controls, use the TWinControl class.

Hierarchy	Inherits from TControl
See also	TControl, TWinControl
File name	\source\StdControl.prg

1.3.5.1 TStdControl:Properties

■ read Only ■ Assignable ■ Design assignable ■ Run-time assignable

Scope	Name	Type	Initial value
■	aDropFiles	Array	{}
■	cMessage	Character	""
■	IBorder	Logic	.F.
■	IDragAcceptFiles	Logic	.F.
■	ITabStop	Logic	.T.

1.3.5.1.1 TStdControl:aDropFiles

Array with all the files dropped in the control with the mouse through the Windows explorer.

Scope:	read Only
Type:	Array
Initial value:	{}

1.3.5.1.2 TStdControl:cMessage

Message that appears in the status bar (TStatusBar) when the control receives the focus.

Scope:	Assignable
---------------	------------

Type:	Character
Initial value:	""

1.3.5.1.3 TStdControl:IBorder

Indicates if the control has a border or not.

Scope	Design assignable
Type	Logic
Initial value	.F.

This property is not enabled in some controls that descend from TStdControl due they use the **nBorderStyle** property that offers better possibilities.

1.3.5.1.4 TStdControl:IDragAcceptFiles

Activates the Drag&Drop support in the control.

Scope	Design assignable
Type	Logic
Initial value	.F.

1.3.5.1.5 TStdControl:ITabStop

Indicates if the control receives the focus when the user presses the TAB key.

Scope	Design assignable
Type	Logic
Initial value	.T.

1.3.5.2 TStdControl:Methods

■ Constructor ■ Standard ■ Only after Create()

Typ	Name
■	RequestState

1.3.5.2.1 TStdControl:RequestState

Forces to evaluate the OnCheckState event in the control and all its son controls that process it.

Type	Only after Create()
Parameters	None
Return value	NIL

The OnCheckState event is produced every time that the control focus is changed normally pressing the TAB key or selecting other control with the mouse pointer.

1.3.5.3 TStdControl:Events

Name
OnChar
OnCheckState
OnEnter
OnExit
OnDropFiles
OnKeyDown
OnKeyUp

1.3.5.3.1 TStdControl:OnChar

Event that is produced when the users presses a character type key.

Parameters	<oSender>: Reference to the object that triggers the event <nKey>: Char type key pressed <nFlags>: Additional information
Return value:	<NIL>: Follows the default Windows processing <0>: Stops the key processing. The effect is like there is no key pressed.

The Windows API differentiates when a key is pressed and released (with the OnKeyUp and OnKeyDown events) and when a character key is pressed. Every key when is pressed generates the events OnKeyUp and OnKeyDown, and they receive as parameters the virtual identifier for the pressed key. Later, if the key pressed is a character key, it will receive the event OnChar.

We recommend to check the [WM_CHAR](#) message from the Windows API documentation, to get more information about the parameters received.

1.3.5.3.2 TStdControl:OnCheckState

Event that is produced every time that there is a focus change in the form. The event is received by all controls descending from TStdControl. If you want, the event can return a logic value, and in that case, if it returns `.F.` the control is not enable and if it returns `.T.` the control is enabled.

Parameters	<oSender> : Reference to the object that triggers the event
Return value:	<NIL> : No additional action <.T.> : The control is enabled <.F.> : The control is disabled

This event allows to simulate the original WHEN clause from the [x]Harbour GET command used in console mode.

If you want to force to execute this event, for example, the first time that a form is started, you can use the `RequesState` method from the own `TForm` object.

1.3.5.3.3 TStdControl:OnEnter

Event that is produced every time that the control receives the focus.

Parameters	<oSender> : Reference to the object that triggers the event <oPrevCtl> : Object that loose the focus
Return value:	<NIL> : It is not processed

1.3.5.3.4 TStdControl:OnExit

Event that is produced every time that the control loose the focus.

Parameters	<oSender> : Reference to the object that triggers the event <oNextCtl> : Object that will receive the focus
Return value:	<NIL> : No action <.F.> : The focus loose is canceled in the control and it

will not receive **oNextCtl**

1.3.5.3.5 TStdControl:OnDropFiles

Event that is produced when the users Drops one or more files to the control through the Windows explorer.

Parameters	<oSender>: Reference to the object that triggers the event
:	<aFiles>: Array with the filename dropped
	<aPoint> Array with the mouse pointer coordinates in {x, y} format, when the files were dropped
Return value:	NIL

1.3.5.3.6 TStdControl:OnKeyDown

Event that is produced when the user presses a key in the keyboard..

Parameters	<oSender>: Reference to the object that triggers the event
:	<nKey>: Virtual code for the pressed key
	<nFlags>: Additional information
Return value:	<NIL>: Follows the default Windows processing
	<0>: Stops the key processing. The effect is like there is not a key pressed

The Windows API differentiates when a key is pressed and released (with the OnKeyUp and OnKeyDown events) and when a character key is pressed. Every key when is pressed generates the events OnKeyUp and OnKeyDown, and they receive as parameters the virtual identifier for the pressed key. Later, if the key pressed is a character key, it will receive the event OnChar.

We recommend to check the [WM_KEYDOWN](#) message from the Windows API documentation, to get more information about the parameters received.

1.3.5.3.7 TStdControl:OnKeyUp

Event that is produced when the user releases a key in the keyboard.

Parameters	<oSender>: Reference to the object that triggers the event <nKey>: Virtual code for the pressed key <nFlags>: Additional information
Return value:	<NIL>: Follows the default Windows processing <0>: Stops the key processing. The effect is like there is not a key pressed

The Windows API differentiates when a key is pressed and released (with the OnKeyUp y OnKeyDown events) and when a character key is pressed. Every key when is pressed generates the events OnKeyUp and OnKeyDown, and they receive as parameters the virtual identifier for the pressed key. Later, if the key pressed is a character key, it will receive the event OnChar.

We recommend to check the [WM_KEYUP](#) message from the Windows API documentation, to get more information about the parameters received.

1.3.6 TWinControl

This is the base of all the Xailer's visual objects that might contain other visual objects as well. It has also support to have an image in the background.

Description:

The TWinControl class is the most specialized Xailer class to manage visual components (except TScrollingWinControl) due they can contain other visual components as well.

TWinControl can be used as class base to create new components that at the same time are supported by other visual components from the operating system and can contain more visual components.

To create components that are visible during run-time use the TControl class as base class. To create Windows type visual components use TStdControl as base class and if this needs to contain other controls, use the TWinControl class.

Hierarchy	Inherits from TStdControl
See also	TControl, TStdControl, TScrollingWinControl
File name	\source\WinControl.prg

1.3.6.1 TWinControl:Properties

■ read Only ■ Assignable ■ Design assignable ■ Run-time assignable

Scope	Name	Type	Initial value
■	aControls	Array	{}
■	nClrPaneEnd	Numeric	clBtnFace
■	nGradient	Numeric	grNONE
■	nBkGndMargin X	Numeric	0
■	nBkGndMargin Y	Numeric	0
■	nBkGndMode	Numeric	blCOPY
■	oActiveControl	Object	Nil
■	oBkGnd	Object	Nil
■	oMenu	Object	Nil

1.3.6.1.1 TWinControl:aControls

Array will all the dependent visual controls.

Scope:	read Only
Type:	Array
Initial value:	{}

1.3.6.1.2 TWinControl:nClrPaneEnd

Control background end color when used in conjunction with the nGradient property.

Scope:	Assignable
Type:	Numeric
Initial value:	clBtnFace

Consult the appendix for the list of available colors

1.3.6.1.3 TWinControl:nGradient

Gradient type to used in conjunction with the nClrPaneEnd property.

Scope:	Assignable
Type:	Numeric
Initial value:	grNONE
Possible	grNONE, grHORIZONTAL, grVERTICAL

values:

Important note: This property is incompatible with the oBkGnd property. When a background image is set this property is useless.

Consult the appendix for the list of available colors

1.3.6.1.4 TWinControl:nBkGndMarginX

Inner margin on X axis to be used when displaying the oBkGnd property.

Scope	Assignable
Type	Numeric
Initial value	0

1.3.6.1.5 TWinControl:nBkGndMarginY

Inner margin on Y axis to be used when displaying the oBkGnd property.

Scope	Assignable
Type	Numeric
Initial value	0

1.3.6.1.6 TWinControl.nBkGndMode

Indicates the way that the background image will be shown through the oBkGnd property.

Scope	Assignable
Type	Numeric
Initial value	bICOPY
Possible values	bICOPY, bITOPLEFT, bITOPRIGHT, bIBOTTOMLEFT, bIBOTTOMRIGHT, bICENTER, bIFIT, bIFITSMOOTH, bITILED, bISTRETCH, bISTRETCHSMOOTH, bIFILL, bIFILLSMOOTH

1.3.6.1.7 TWinControl:oActiveControl

Indicates the current active control. The control must be one of the controls from the aControls array.

Scope	Assignable
--------------	------------

Type	Object
Initial value	Nil

1.3.6.1.8 TWinControl:oBkGnd

TPicture object with the image to show as the control's background. It is used together with the nBkGndMode property.

Scope	Assignable
Type	Object
Initial value	Nil

1.3.6.1.9 TWinControl:oMenu

TMenu object to include an option menu in the control.

Scope	Assignable
Type	Object
Initial value	Nil

1.3.6.2 TWinControl:Methods

■ Constructor ■ Standard ■ Only after Create()

Type	Name
■	GoFirstControl
■	GoNextControl
■	GoPrevControl
■	InsertControl
■	Redraw
■	RemoveControl
■	RequestState

1.3.6.2.1 TWinContol:GoFirstControl

Establishes as active control the first control. It corresponds to the aControls array's first element.

Type	Only after Create()
Parameters	None
Return value	<oCtl> Reference to the active control or NIL if has not

been possible to activate any control

1.3.6.2.2 TWinControl:GoNextControl

Establishes as active control, the next control respecting to the current one.

Type	Only after Create()
Parameters	None
Return value	<hNext> Handle for the new active control or zero if has not been possible to change the active control

1.3.6.2.3 TWinControl:GoPrevControl

Establishes as active control, the previous control respecting to the current one.

Type	Only after Create()
Parameters	None
Return value	<hNext> Handle for the new active control or zero if has not been possible to change the active control

1.3.6.2.4 TWinControl:InsertControl

Adds a new control to the aControls list of controls.

Type	Standard
Parameters	<oCtl>: New control to include
Return value	NIL

1.3.6.2.5 TWinControl:Redraw

Forces to redraw the control. The main difference with the Refresh method is that Redraw does not return until the control is completely painted.

Type	Standard
Parameters	None
Return	NIL

value	
--------------	--

1.3.6.2.6 TWinControl:RemoveControl

Removes a control from the aControls list of controls.

Type	Only after Create()
Parameters	<oCtl>: Control to remove
Return value	NIL

1.3.6.2.7 TWinControl:RequestState

Asks to every control from the aControls array about its state evaluating the OnCheckState event of every single one.

The OnCheckState method is used to enable or disable a control during run-time. This event is triggered every time that the user changes the active control. If you want to force to trigger this event manually, you can call it directly.

Type	Only after Create()
Parameters	NIL
Return value	NIL

1.3.7 TScrollingWinControl

This is the base of all the Xailer visual objects that might contain other visual objects and its dimension can be bigger than the window where it is shown, showing in that case the classic scroll bars to be able to see the full control.

Hierarchy	Inherits from TWinControl
See also	TWinControl, TStdControl, TControl
File name	\source\ScrlWinControl.prg

1.3.7.1 TScrollingWinControl:Properties

■ read Only ■ Assignable ■ Design assignable ■ Run-time assignable

Scope	Name	Type	Initial value
■	IAutoScroll	Logic	.T.
■	IHideScrollBars	Logic	.F.

■	nScrollIncrement	Numeric	8
■	nScrollShowDelay	Numeric	1000
■	nClientLeft	Numeric	0
■	nClientTop	Numeric	0
■	nVirtualHeight	Numeric	0
■	nVirtualWidth	Numeric	0

1.3.7.1.1 TScrollingWinControl:IAutoScroll

If it is .T. the scroll bars will appear automatically when are needed.

Scope:	Assignable
Type:	Logic
Initial value:	.T.

1.3.7.1.2 TScrollingWinControl:IHideScrollBars

If true will hide the scrollbars and only show them when the mouse gets close to the areas where it should be.

Scope:	Assignable
Type:	Logic
Initial value:	.T.

1.3.7.1.3 TScrollingWinControl:nScrollIncrement

Vertical / horizontal control scroll increment specified in pixels.

Scope:	Assignable
Type:	Numeric
Initial value:	8

1.3.7.1.4 TScrollingWinControl:nScrollShowDelay

Delay in milliseconds to show the scrollbars when the property IHideScrollBars is set to .T.

Scope:	Assignable
Type:	Numeric
Initial value:	8

1.3.7.1.5 TScrollingWinControl:nClientLeft

Indicates the control's left virtual corner based in the visible corner. Its value is always zero or negative.

Scope:	Assignable
Type:	Numeric
Initial value:	0

1.3.7.1.6 TScrollingWinControl:nClientTop

Indicates the control's top virtual corner based in the visible corner. Its value is always zero or negative.

Scope:	Assignable
Type:	Numeric
Initial value:	0

1.3.7.1.7 TScrollingWinControl:nVirtualHeight

Indicates the usable control height. It is the same as the virtual control size.

Scope:	Assignable
Type:	Numeric
Initial value:	0

1.3.7.1.8 TScrollingWinControl:nVirtualWidth

Indicates the usable control width. It is the same as the virtual control size.

Scope:	Assignable
Type:	Numeric
Initial value:	0

1.3.7.2 TScrollingControl:Events

Name
OnScrollDown
OnScrollLeft
OnScrollPageDown
OnScrollPageLeft
OnScrollPageRight
OnScrollPageUp
OnScrollRight
OnScrollUp
OnThumbPosH
OnThumbPosV
OnThumbTrackH
OnThumbTrackV

1.3.7.2.1 TScrollingWinControl:OnScrollDown

Event that is produced when the indicator is moved down.

Parameters	<oSender> : Reference to the object that triggers the event
:	<nPos> : New indicator position
Return value:	<nPos> : If returns a value different than NIL, this must be the new indicator position. When the value is NIL the indicator increments automatically for the value indicated in the nScrollIncrement property.

1.3.7.2.2 TScrollingWinControl:OnScrollLeft

Event that is produced when the indicator is moved to the left.

Parameters	<oSender> : Reference to the object that triggers the event
:	<nPos> : New indicator position
Return value:	<nPos> : If returns a value different than NIL, this must be the new indicator position. When the value is NIL the indicator decrements automatically for the value indicated in the nScrollIncrement property.

1.3.7.2.3 TScrollingWinControl:OnScrollPageDown

Event that is produced when the indicator is moved one page down.

Parameters	<oSender>: Reference to the object that triggers the event
:	<nPos>: New indicator position
Return value:	<nPos>: If returns a value different than NIL, this must be the new indicator position. When the value is NIL the indicator increments automatically one page

1.3.7.2.4 TScrollingWinControl:OnScrollPageLeft

Event that is produced when the indicator is moved one page to the left.

Parameters	<oSender>: Reference to the object that triggers the event
:	<nPos>: New indicator position
Return value:	<nPos>: If returns a value different than NIL, this must be the new indicator position. When the value is NIL the indicator decrements automatically one page

1.3.7.2.5 TScrollingWinControl:OnScrollPageRight

Event that is produced when the indicator is moved one page to the right.

Parameters	<oSender>: Reference to the object that triggers the event
:	<nPos>: New indicator position
Return value:	<nPos>: If returns a value different than NIL, this must be the new indicator position. When the value is NIL the indicator increments automatically one page

1.3.7.2.6 TScrollingWinControl:OnScrollPageUp

Event that is produced when the indicator is moved one page up.

Parameters	<oSender>: Reference to the object that triggers the event
:	<nPos>: New indicator position
Return value:	<nPos>: If returns a value different than NIL, this must be the new indicator position. When the value is NIL the indicator decrements automatically one page

1.3.7.2.7 TScrollingWinControl:OnScrollRight

Event that is produced when the indicator is moved to the right.

Parameters	<oSender>: Reference to the object that triggers the event
:	<nPos>: New indicator position
Return value:	<nPos>: If returns a value different than NIL, this must be the new indicator position. When the value is NIL the indicator increments automatically for the value indicated in the nScrollIncrement property.

1.3.7.2.8 TScrollingWinControl:OnScrollUp

Event that is produced when the indicator is moved up.

Parameters	<oSender>: Reference to the object that triggers the event
:	<nPos>: New indicator position
Return value:	<nPos>: If returns a value different than NIL, this must be the new indicator position. When the value is NIL the indicator decrements automatically for the value indicated in the nScrollIncrement property.

1.3.7.2.9 TScrollingWinControl:OnThumbPosH

Event that is activated when the user stops to drag the horizontal indicator.

Parameters	<oSender>: Reference to the object that triggers the event
:	<nPos>: New indicator position
Return value:	<nPos>: Scroll value to be applied. If it is NIL it assigns the default value

1.3.7.2.10 TScrollingWinControl:OnThumbPosV

Event that is activated when the user stops to drag the vertical indicator.

Parameters	<oSender>: Reference to the object that triggers the event
:	<nPos>: New indicator position
Return value:	<nPos>: Scroll value to be applied. If it is NIL it assigns the default value

1.3.7.2.11 TScrollingWinControl:OnThumbTrackH

Event that is activated when the horizontal indicator is dragged.

Parameters	<oSender>: Reference to the object that triggers the event
:	<nPos>: New indicator position
Return value:	<nPos>: Scroll value to be applied. If it is NIL it assigns the default value

1.3.7.2.12 TScrollingWinControl:OnThumbTrackV

Event that is activated when the vertical indicator is dragged.

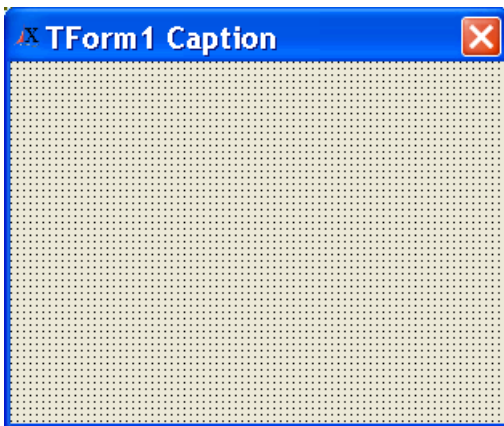
Parameters	<oSender>: Reference to the object that triggers the event
:	

	<nPos> : New indicator position
Return value:	<nPos> : Scroll value to be applied. If it is NIL it assigns the default value

1.3.8 TForm

This class represents a form. A form is a window used in windows. It can be a main application window a normal window, a dialog window or a tools window.

In Xailer don't exist the dialogs. They are forms with an special style (nBordeStyle = bsDIALOG) and its coordinate system is measured in pixels like the other windows styles.



Hierarchy Inherits from TScrollingWinControl
File name \source\Form.prg

1.3.8.1 TForm:Properties

■ read Only ■ Assignable ■ Design assignable ■ Run-time assignable

Scope	Name	Type	Initial value
■	aComponents	Array	{}
■	IActive	Logic	.F.
■	IAppWindow	Logic	.F.
■	ICentered	Logic	.F.
■	ICloseBox	Logic	.T.
■	IDirectShortCuts	Logic	.F.
■	IHelpIcon	Logic	.F.
■	IHideOnClose	Logic	.F.
■	IMaximizeBox	Logic	.T.

■	IMinimizeBox	Logic	.T.
■	IModal	Logic	.F.
■	IMsgAuto	Logic	.F.
■	IParentIcon	Logic	.T.
■	ISysMenu	Logic	.T.
■	nBorderStyle	Numeric	bsSIZEABLE
■	nFormType	Numeric	ftNORMAL
■	nMaxHeight	Numeric	0
■	nMaxPosX	Numeric	0
■	nMaxPosY	Numeric	0
■	nMaxWidth	Numeric	0
■	nMinHeight	Numeric	0
■	nMinWidth	Numeric	0
■	nModalResult	Numeric	0
■	nShowMode	Numeric	smNORMAL
■	oActiveControl	Object	NIL
■	oCancelButton	Object	NIL
■	oDefaultButton	Object	NIL
■	olcon	Object	NIL
■	oMDIMenu	Object	NIL
■	oMDIClient	Object	NIL
■	oMsgBar	Object	NIL
■	oPopup	Object	NIL
■	oPrevWnd	Object	NIL
■	XHtmlHelp	Character	""

1.3.8.1.1 TForm:aComponents

Array with all the components defined in the form. The components are all the non visual control that can be included in a form and they are in the IDE under the System or DataSet tabs.

Scope	Design assignable
Type	Array
Initial value	{}

1.3.8.1.2 TForm:IActive

If it is .T. indicate that the form is active.

Scope	read Only
Type	Logic
Initial value	.F.

1.3.8.1.3 TForm:IAppWindow

If it is .T. indicate that the form is like a main application form. You will be able to see the the window in the system task bar. In order to work correctly its oParent property should be Application.

Scope	Design assignable
Type	Logic
Initial value	.F.

1.3.8.1.4 TForm:ICentered

If it is .T. the form will be shown centered on the desktop.

Scope	Assignable
Type	Logic
Initial value	.F.

1.3.8.1.5 TForm:ICloseBox

If it is .F. the form close button will be disabled.

Scope	Design assignable
Type	Logic
Initial value	.T.

1.3.8.1.6 TForm:IDirectShortCuts

If it is .T. the keyboard shortcuts of all the controls included in the form will be activated without the need to press the ALT key. In other words, they will activate directly when the user presses the shortcut. To establish a control keyboard shortcut it is only needed to include a '&' in the control text, or in the text of the previous control according to its creation order.

Scope	Assignable
Type	Logic
Initial value	.F.

1.3.8.1.7 TForm:IHelpIcon

If it is `.T.` the form will show the help icon in its bar title. The property `nBorderStyle` should be set to `bsDIALOG`.

Scope	Design assignable
Type	Logic
Initial value	<code>.F.</code>



1.3.8.1.8 TForm:IHideOnClose

If it is `.T.`, the form will hide instead to be destroyed when it is close by the user, or call its `Close` method. Xailer destroys the forms by default when are closed, but you can avoid it with this property.

This property is very useful for several reasons:

- Allows to keep th complete form status to be used later
- Allows to get information from the form controls that is not possible to obtain when the form has been destroyed, like for example, all the form linked `DataControls`.

It is important that you don't forget to destroy the form when you don't needed anymore, with its `End` destructor method.

Scope	Design assignable
Type	Logic
Initial value	<code>.F.</code>

1.3.8.1.9 TForm:IMaximizeBox

If it is `.T.` the form will show the maximize icon in its bar title.

Scope	Design assignable
Type	Logic
Initial value	<code>.T.</code>

This property can be used only when the `nBorderStyle` form property is `bsSIZEABLE`.

1.3.8.1.10 TForm:IMinimizeBox

If it is .T. the form will show the minimize icon in its bar title.

Scope	Design assignable
Type	Logic
Initial value	.T.

This property can be used only when the nBorderStyle form property is bsSIZEABLE.

1.3.8.1.11 TForm:IModal

If it is .T. the form will be shown in modal mode when it is activated. It means that will not be able to change to other application form until you close this form. You can also use the ShowModal method instead the Show method to get the same result.

Scope	Design assignable
Type	Logic
Initial value	.F.

1.3.8.1.12 TForm:IMsgAuto

If it is .T. and the form has a TStatusBar, it will show in the TStatusBar default panel the control cMessage property every time that it has the focus.

Scope	Assignable
Type	Logic
Initial value	.F.

1.3.8.1.13 TForm:IParentIcon

If it is .T. the form will use the same icon as its parent objet. In the case that the parent object is the first application, it will use the icon from the Application class.

Scope	Design assignable
Type	Logic
Initial value	.T.

1.3.8.1.14 TForm:ISysMenu

If it is .T. the form will show a system menu when the user right-clicks the mouse pointer in its title bar.

Scope	Design assignable
Type	Logic
Initial value	.T.

1.3.8.1.15 TForm:nBorderStyle

Indicates the form border type. Besides the visual look change that is produced when is chosen different border types, it affect the form behavior as well.

Scope	Design assignable
Type	Numeric
Initial value	bsSIZEABLE
Possible values	<p>bsNONE: No border, no title bar, it can not be moved or resized and only can be close by code</p> <p>bsSINGLE: Single border, it can not be resized</p> <p>bsSIZEABLE: Normal border, it can be moved, resized and it shows maximize and minimize icons in the title bar</p> <p>bsDIALOG: Identical visual style as bsSINGLE but it behaves always as a modal form, and it is possible to exit pressing the ESC key. This emulates the classic Windows API dialogs</p> <p>bsTOOLWINDOW: Identical visual style as bsSINGLE but the title bar height is lower</p> <p>bsSIZETOOLWIN: Identical visual style as bsTOOLWINDOW but allows to resize the form</p> <p>bsSPLASH: Splash border, with no title bar. It can not be moved or resized and it can be closes only by code. This window type usually is used in the classic "About..." windows</p>

1.3.8.1.16 TForm:nFormType

Indicates the form type.

Scope	Design assignable
Type	Numeric
Initial value	ftNORMAL
Possible values	ftNORMAL: Normal style ftMDIFRAME: MDI (Multiple document interface) container form that might contain more from on it ftMDICHILD: MDI (Multiple document interface) child form to be used with other form with ftMDIFRAME style

1.3.8.1.17 TForm:nMaxHeight

Indicates the maximum form height. A zero value indicates that there is no limitation.

Scope	Assignable
Type	Numeric
Initial value	0

1.3.8.1.18 TForm:nMaxPosX

Forces the maximum X coordinate form position when is maximized. A zero value indicates that there is no limitation.

Scope	Assignable
Type	Numeric
Initial value	0

1.3.8.1.19 TForm:nMaxPosY

Forces the maximum X coordinate form position when is maximized. A zero value indicates that there is no limitation.

Scope	Assignable
Type	Numeric
Initial value	0

1.3.8.1.20 TForm:nMaxWidth

Indicates the maximum form width. A zero value indicates that there is no limitation.

Scope	Assignable
Type	Numeric
Initial value	0

1.3.8.1.21 TForm:nMinHeight

Indicates the minimum form height. A zero value indicates that there is no limitation.

Scope	Assignable
Type	Numeric
Initial value	0

1.3.8.1.22 TForm:nMinWidth

Indicates the minimum form width. A zero value indicates that there is no limitation.

Scope	Assignable
Type	Numeric
Initial value	0

1.3.8.1.23 TForm:nModalResult

Property that indicates how the form has been closed after it was shown in modal way.

This property is a numeric value that allows to know the way that the form has been closed. It is used together with the nModalResult property in the TButton and TBtnBmp buttons.

Scope	Assignable
Type	Numeric
Initial value	0

1.3.8.1.24 TForm:nShowMode

Property that indicates the way that the form will be show. This property only is usable if the nBorderStyle property is set to bsSIZEABLE.

Scope	Design assignable
Type	Numeric
Initial value	smNORMAL
Possible values	smNORMAL: Normal form smMAXIMIZE: Maximized form ftMINIMIZE: Minimized form

1.3.8.1.25 TForm:oActiveControl

Indicates and establishes the active control. The indicated control is an existing control in the form.

Scope	Assignable
Type	Object
Initial value	NIL

1.3.8.1.26 TForm:oCancelButton

Property that references to a button that might be inside the form with its ICancel property set to .T..

Scope	read Only
Type	Object
Initial value	NIL

1.3.8.1.27 TForm:oDefaultButton

Property that references to a button that might be inside the form with is IDefault property set to .T..

Scope	read Only
Type	Object
Initial value	NIL

1.3.8.1.28 TForm:olcon

Specifies the icon to shown in the form title bar and when this is minimized, if it is possible.

Scope:	Design assignable
Type:	Object
Initial value:	Standard Xailer TIcon

1.3.8.1.29 TForm:oMDIMenu

TMenu object that receives the list of all the MDIChild existing windows. This property only makes sense when it is used together with forms with the nFormType property set to ftMDIFRAME.

Scope:	Assignable
Type:	Object
Initial value:	NIL

1.3.8.1.30 TForm:oMDIClient

Hidden control inherited from TWinControl that is responsible to manage all the existing MDIChild forms. This property only makes sense when it is used together with the forms with the nFormType property set to ftMDIFRAME.

Scope:	read Only
Type:	Object
Initial value:	NIL

This control type is automatically created by Xailer when the form is create with the ftMDIFRAME style and its use by the user is very limited. However, it offers several properties and methods to manage the different existing forms inside the MDIFrame form like:

Properties: (read Only)

aForms	Array with all the MDIChild forms
oActiveForm	Current active MDIChild form

Methods:

Arrangelcons	Organizes the icons for the MDIChild forms tat are iconized
Cascade	Organizes the MDIChild form in cascade mode
CloseAll	Closes all the MDIChild forms
MaximizeAll	Maximizes all the MDIChild forms

MinimizeAll	Minimizes all the MDIChild forms
Next	Activates the next MDIChild form
Previous	Activates the previous MDIChild form
TileHorizontal	Organizes the forms in horizontally mode
TileVertical	Organizes the form in vertically mode

1.3.8.1.31 TForm:oMsgBar

Property that references to a TStatusBar that might be inside the form.

Scope	readOnly
Type	Object
Initial value	NIL

1.3.8.1.32 TForm:oPopup

Property that references to a Pop-up TMenu that might be inside the form.

Scope	readOnly
Type	Object
Initial value	NIL

1.3.8.1.33 TForm:oPrevWindow

Property that references to a form that might be active before to show it.

Scope	readOnly
Type	Object
Initial value	NIL

1.3.8.1.34 TForm:XHtmlHelp

Indicator for advanced Html help file. Is better to click on three dots button so the help manager can do all the work for you. For further information consult this link:

Blog article

By default, the HTML help uses the TWebBrowser control to show its content. If you want to use TWebView (only available in Xailer Enterprise) you must include the following code on any of the applications PRG's:

```

CLASS TXHtmlHelpView FROM XXHtmlHelpView
  CLASSDATA lWebView INIT .T.
END CLASS

```

Only available in Xailer Professional and Xailer Enterprise.

Scope	Assignable
Type	Character
Initial value	""

1.3.8.2 TForm:Methods

Constructor
 Standard
 Only after Create()()

Typ e	Name
<input type="checkbox"/>	Close
<input type="checkbox"/>	Maximize
<input type="checkbox"/>	Minimize
<input type="checkbox"/>	Restore
<input type="checkbox"/>	RestoreState
<input type="checkbox"/>	SaveState
<input type="checkbox"/>	SetFocus
<input type="checkbox"/>	Show
<input type="checkbox"/>	ShowModal

1.3.8.2.1 TForm:Close

Closes the form. If the property `lHideOnClose` is true the form is hided, otherwise is completely destroyed.

If you need to consult any form control property after the form is closed, is preferable that you use the property `lHideOnClose` to true to avoid that certain information of the control stops from being accesible, in case contrary the form controls will be also destroyed. When you use this property is important that you destroy the form calling its **End** method after you got all the information you need.

Type	Only after Create()
Parameters	<IDontAsk> : It true the event <code>OnClose</code> will not be triggered closing directly the form
Return value	<ISuccess> True if success

1.3.8.2.2 TForm:Maximize

Shows the form in maximized mode. It can be used only when its `nBorderStyle` is `bsSIZEABLE`.

Type	Only after Create()
Parameters	None
Return value	NIL

1.3.8.2.3 TForm:Minimize

Shows the form in minimized mode. It can be used only when its `nBorderStyle` is `bsSIZEABLE`.

Type	Only after Create()
Parameters	None
Return value	NIL

1.3.8.2.4 TForm:Restore

Restores the form state to normal. It makes sense to use only when the form is minimized. It can be used only when its `nBorderStyle` is `bsSIZEABLE`.

Type	Only after Create()
Parameters	None
Return value	NIL

1.3.8.2.5 TForm:RestoreState

Restores the full form state saved with the `SaveState` method.

Type	Only after Create()
Parameters	<code><cState></code> Form state returned by the <code>SaveState</code> method <code><ISetPositionOnly></code> If true only the position and dimensions of the form is restored, not its state: Hidden, normal, minimized or maximized
Return value	NIL

1.3.8.2.6 TForm:SaveState

Returns in a string the current form status: coordinates, visibility and visualization state (normal, maximized or minimized). It is used together with the RestoreState method.

Type	Only after Create()
Parameters	None
Return value	<cState>

1.3.8.2.7 TForm:SetFocus

Activates the form establishing the focus on it.

Type	Only after Create()
Parameters	None
Return value	NIL

1.3.8.2.8 TForm:Show

Shows the form.

Type	Only after Create()
Parameters	[<nShowMode>]: Show mode. Correspond with the nShowMode property [<ICenter>] If true the form will be centered on the desktop. By default its property ICentered
Return value	NIL

1.3.8.2.9 TForm:ShowModal

Shows the form on modal style. This means no other application form can receive focus until this form is close and the application flow is stopped until the form is closed. For this reason the return value is important since it reports how the form has been closed.

Type	Only after Create()
Parameters	[<nShowMode>]: Show mode. Correspond with the nShowMode property

	[<ICenter>] If true the form will be centered on the desktop. By default its property ICentered
Return value	<nModalResult> How the form is closed. See the property nModalResult for further information

1.3.8.3 TForm:Events

Name
OnActivate
OnChangeFocus
OnClose
OnCopyData
OnCreate
OnDeactivate
OnHelp
OnHelpClick
OnInitialize
OnMenuSelect
OnMove
OnSize

1.3.8.3.1 TForm:OnActivate

Event that is produced every time that the form is activated.

Parameters	<oSender>: Self reference
:	<oForm>: Previous form that gives the current status to the form or NIL
Return value:	NIL

1.3.8.3.2 TForm:OnChangeFocus

Event that is produced very time that the focus changes among the form controls.

Parameters	<oSender>: Self reference
:	<oFrom>: Origin control
	<oTo>:

	Destination control
Return value:	NIL

1.3.8.3.3 TForm:OnClose

Event that is produced when the form is closed.

Parameters	<oSender>: Self reference
:	<IClose>: Variable passed by reference that allows to abort the process if it has a .F. value
Return value:	<IClose>: If returns .F, the close process is aborted

1.3.8.3.4 TForm:OnCopyData

Event that is produced when a WM_COPYDATA message is received.

Parameters	<oSender>: Self reference
:	<cString>: Text message
	<nMsgId>: Message identifier (optional)
Return value:	<IResult>: If returns .T, the message is processed
See also	Function SendWMCopyData

1.3.8.3.5 TForm:OnCreate

Event that is created when the form is created. Take into consideration that in that moment the control that will contain the form have not been created.

Parameters	<oSender>: Self reference
:	
Return value:	NIL

1.3.8.3.6 TForm:OnDeactivate

Event that is produced every time that the form is deactivated.

Parameters	<oSender> : Self reference
:	<oForm> : Current active form or NIL
Return value:	NIL

1.3.8.3.7 TForm:OnHelp

Event that is fired every time a help request is made.

Parameters	<oSender> : A reference to the object that fired the request
:	<HelpId> : HelpId property value of the control or form that request the help
	<nPosX> : Mouse X coordinate when the request proceed from a mouse click
	<nPosY> : Mouse Y coordinate when the request proceed from a mouse click
Return value:	NIL

The help system works as follows:

- The control that receives the help request checks that its property HelpId has a value. If Nil, then the request will be sent to its parent. If not Nil, then calls the help system of its container form. If on the form OnHelp event is assigned, then is triggered. If is not assigned then the TApplication help systems is called.
- If the TApplication OnHelp event is assigned, then is triggered. If is not assigned and its property oHelp has a valid THelp object, then it shows the help from the topic HelpId of the control or form that request the help.
- The **oSender** parameter of TForm or TApplication **OnHelp** event is the control that request the help whose HelpId had a value.

1.3.8.3.8 TForm:OnHelpClick

Event that is triggered every time the help button of the form title bar is pressed. To enable the help button the property IHelpIcon must be set to true.

Parameters	<oSender> : A reference to the form
Return value:	<NIL> : To maintain the default behavior which consists that the mouse cursor turns into an arrow plus the question mark, expecting a mouse click on any of the form controls to perform a help request. <0> : To break the default behaviour.

Consult the OnHelp event for further information.

1.3.8.3.9 TForm:OnInitialize

Event that is produced when the form is initialized after all its controls have been created.

Parameters	<oSender> : Self reference
Return value:	NIL

1.3.8.3.10 TForm:OnMenuSelect

Event that is produced when the user navigates along the different form menu options.

Parameters	<oSender> : Self reference <nFlags> : Additional information about the menu status provided by the Windows API. It is the same as HIWORD(wParam) from the WM_MENUSELECT message <hMenu> : Internal API handle to the menu element <cMessage> : Text that shows the menu element or NIL <oMenuItem> : Active menu item
Return value:	NIL

1.3.8.3.11 TForm:OnMove

Event that is produced when the form changes its position.

Parameters	<oSender> : Self reference <nLeft> : New X coordinate <nTop> : New Y coordinate
Return value:	NIL

1.3.8.3.12 TForm:OnSize

Event that is produced when the form changes its dimensions.

Parameters	<oSender> : Self reference <nSizeType> : Operation type. It can be: SIZE_MAXHIDE: (4) Other windows has been maximized SIZE_MAXIMIZED: (2) The form has been maximized SIZE_MAXSHOW: (3) Other windos has been maximized SIZE_MINIMIZED: (1) The form has been minimized SIZE_RESTORED: (0) The form has been restored <nClientWidth> : New client area width <nClientHeight> : New client area height
Return value:	NIL

1.4 Public Objects

1.4.1 TApplication

Unique and automatic instance class for the application control, form management and control navigation. It provides important information about the environment for the running application.

■ read Only ■ Assignable ■ Design assignable ■ Run-time assignable

Scope	Name	Type	Initial value
■	aArguments	Array	{}
■	aForms	Array	{}
■	cComments	Character	""
■	cCompany	Character	""
■	cCopyright	Character	""
■	cDescription	Character	""
■	cDirectory	Character	""
■	cFilename	Character	""
■	cTitle	Character	""
■	cTradeMarks	Character	""
■	cVersion	Character	""
■	lAutoScale	Logic	.F.
■	lBusy	Logic	.F.
■	lDpiAware	Logic	.F.
■	lSingleInstance	Logic	.F.
■	lUseReturn	Logic	.F.
■	nBuild	Numeric	0
■	nScale	Numeric	100
■	nRelease	Numeric	0
■	nVersion	Numeric	0
■	nWinPlatform	Numeric	0
■	nWinVer	Numeric	0
■	oActiveControl	Object	NIL
■	oActiveForm	Object	NIL
■	oFont	Object	TFont (Standard)
■	oHelp	Object	NIL
■	olcon	Object	TIcon (Xailer)
■	oMainForm	Object	TForm

■ Constructor ■ Standard

Type	Name
■	CurrentDir
■	CurrentDrive
■	Initialize
■	DecimalSeparator
■	MessageBox
■	Run

- SetOverlayIcon
- SetProgress
- Terminate

Nombre

OnHelp

OnQueryEndSession

Description:

This class is created automatically from an INIT PROCEDURE and it is always available through the public variable "Application". It is not needed to instantiate it from the application.

If the application has several INIT PROCEDURES, avoid to use any Xailer function or class from there. And if for any reason, it is absolutely needed, then you must call to TApplication():Initialize() at the beginning of that INIT PROCEDURE.

Hierarchy Inherits from nobody
File Name \source\Application.prg

1.4.1.1 TApplication.Properties

■ read Only ■ Assignable ■ Design assignable ■ Run-time assignable

Scope	Name	Type	Initial value
■	aArguments	Array	{}
■	aForms	Array	{}
■	cComments	Character	""
■	cCompany	Character	""
■	cCopyright	Character	""
■	cDescription	Character	""
■	cDirectory	Character	""
■	cFilename	Character	""
■	cTitle	Character	""
■	cTradeMarks	Character	""
■	cVersion	Character	""
■	lAutoScale	Logic	.F.
■	lBusy	Logic	.F.
■	lDpiAware	Logic	.F.
■	lSingleInstance	Logic	.F.
■	lUseReturn	Logic	.F.
■	nBuild	Numeric	0
■	nScale	Numeric	100
■	nRelease	Numeric	0
■	nVersion	Numeric	0
■	nWinPlatform	Numeric	0
■	nWinVer	Numeric	0
■	oActiveControl	Object	NIL

■	oActiveForm	Object	NIL
■	oFont	Object	TFont (Standard)
■	oHelp	Object	NIL
■	oIcon	Object	TIcon (Xailer)
■	oMainForm	Object	TForm

1.4.1.1.1 TApplication:aArguments

Array with applications arguments passed to the application through the command line or a windows shortcut. To separate arguments, use the space.

Scope:	readOnly
Type:	Array
Initial value:	{}

Example:

MiXailer.Exe one two three

The aArguments Array will have the values : { "one", "two", "three" }

1.4.1.1.2 TApplication:aForms

Array with all the forms created by the application.

Scope:	readOnly
Type:	Array
Initial value:	{}

Description:

All the forms created by the application are stored in the aForms array. Regardless they are hidden or visible the created forms will become part of this array. When a form is destroyed using its method End(), then it is deleted from the aForms array.

1.4.1.1.3 TApplication:cComments

Version information: Comments. This property can be set on the IDE through the option 'Project properties-Version'.

Scope:	readOnly
Type:	Character
Initial value:	""

1.4.1.1.4 TApplication:cCompany

Version information: Company. This property can be set on the IDE through the option 'Project properties-Version'.

Scope:	read Only
Type:	Character
Initial value:	""

1.4.1.1.5 TApplication:cCopyright

Version information: Copyright. This property can be set on the IDE through the option 'Project properties-Version'.

Scope:	read Only
Type:	Character
Initial value:	""

1.4.1.1.6 TApplication:cDescription

Version information: Description. This property can be set on the IDE through the option 'Project properties-Version'.

Scope:	read Only
Type:	Character
Initial value:	""

1.4.1.1.7 TApplication:cDirectory

Application's startup directory

Scope:	read Only
Type:	Character
Initial value:	""

1.4.1.1.8 TApplication:cFileName

Executable file name (including path).

Scope:	read Only
Type:	Character
Initial value:	""

1.4.1.1.9 TApplication:cTitle

Indicates the title or application's name.

Scope:	Design assignable
Type:	Character
Initial value:	""

1.4.1.1.10 TApplication:cTradeMarks

Version information: Trademarks. This property can be set on the IDE through the option 'Project properties-Version'.

Scope:	read Only
Type:	Character
Initial value:	""

1.4.1.1.11 TApplication:cVersion

Version information: Version as character. This property can be set on the IDE through the option 'Project properties-Version'.

Scope:	read Only
Type:	Character
Initial value:	""

1.4.1.1.12 TApplication:!AutoScale

This property, if true, autoscales the forms and its controls when using large fonts in Windows, or any font size different than 96 d.p.i. See also nScale.

Scope:	Asignable
Type:	Logic
Initial value:	.F.

1.4.1.1.13 TApplication:IBusy

If .T., the cursor changes to the hourglass icon for all the applications.

Scope:	Asignable
Type:	Logic
Initial value:	.F.

1.4.1.1.14 TApplication:IDpiAware

Makes the program recognize the true resolution of the monitor.

When windows is running on a high-resolution monitor and scaling other than 100% is set, programs receive "adjusted" resolution feedback, not the actual resolution. In the same way, when a program sets certain coordinates in a window, windows readjusts those coordinates according to the value of the scale that it has established. For example, if windows is set to 150% scaling, when a program sets the dimensions of a window to 400x400 pixels, windows resets it to an actual size of 600x600 pixels, including all controls it contains.

This means that we do not have to worry about the scale that Windows has configured, but it has the drawback that the texts and images look blurred. When we set Application:IDpiAware to .T., our program will work with real coordinates, with no adjustments by windows. The images and texts will look perfectly sharp and in their original size, even if this means that they will appear smaller than expected. In these cases, we can use the Application:nScale property to adjust the desired scale without losing the sharpness of images and texts.

NOTE: Application:IDpiAware := .T. it can only be assigned once, and it must be done before displaying a form or any other visual element. Any attempt to change Application:IDpiAware to .F. it produces no effect.

Scope:	Asignable
Type:	Logic
Initial value:	.F.

1.4.1.1.15 TApplication:ISingleInstance

If it is .T., only one instance of the program will be allowed to run in the same computer. In the case that the user tries to open a second instance of the same program, the focus will move to the first instance running. The input of this value is made through the project properties menu, in the 'Main' tab, checking the "Allow multiple application's instances".

Scope:	Design assignable
Type:	Logic
Initial value:	.F.

Important note: In order to work correctly, the property cTitle must be given a value since its used for the mutex creation.

1.4.1.1.16 TApplication:lUseReturn

If it is .T., the **Return** key can be used to move through the different controls or fields. This is not the standard windows way, due the **Tab** key is used to do that. However, several Xbase users would appreciate to work using this kind of navigation system.

Take into consideration that there are controls that process the **Return** key by themselves. For example, if you use the Return key in a '**Memo**' control type, the **Return** key is processed by the control. In this case the focus will not be changed to the next available control.

Scope:	Design assignable
Type:	Logic
Initial value:	.F.

1.4.1.1.17 TApplication:nBuild

Version information: Build number. This property can be set on the IDE through the option 'Project properties-Version'.

Scope:	read Only
Type:	Numeric
Initial value:	0

1.4.1.1.18 TApplication:nRelease

Version information: Release number. This property can be set on the IDE through the option 'Project properties-Version'.

Scope:	read Only
Type:	Numeric
Initial value:	0

1.4.1.1.19 TApplication:nScale

This property sets the scale factor to use as a percentage for forms and controls. A factor of 100 corresponds with its original value. You can indicate values less or greater than 100, to increase or decrease the forms size respectively. See also IAutoScale.

When setting IAutoScale to true, the value of this property is changed automatically to adapt the forms to the new font sizes.

If after changing IAutoScale to true, you change the property **nScale**, this new value will be used, but the fonts will be scaled taking care of the font size change in Windows itself. For example, if you are using large fonts (120dpi, or 125% greater), when changing IAutoScale to .T. then **nScale** value will change to 125. If after that you change the value of **nScale** to 100, then the font size will be reduced a 80%, so you will get the same look as you were using normal fonts.

Scope:	Design assignable
Type:	Numeric
Initial value:	100

1.4.1.1.20 TApplication:nVersion

Version information: Version number. This property can be set on the IDE through the option 'Project properties-Version'.

Scope:	read Only
Type:	Numeric
Initial value:	0

1.4.1.1.21 TApplication:nWinPlatform

Shows the Windows platform used by the application.

Scope:	read Only
Type:	Numeric
Initial value:	0

1.4.1.1.22 TApplication:nWinVer

Shows the Windows version.

Scope:	read Only
Type:	Numeric
Initial value:	0

1.4.1.1.23 TApplication.oActiveControl

Return the current active control.

Scope:	read Only
Type:	Object
Initial value:	NIL

1.4.1.1.24 TApplication.oActiveForm

Return the current active form.

Scope:	read Only
Type:	Object
Initial value:	NIL

1.4.1.1.25 TApplication.oFont

Specify the current font used for the application.

Scope:	Design assignable
Type:	Object
Initial value:	Standard Xailer font (TFont)

1.4.1.1.26 TApplication.oHelp

Indicated the THelp object to use for the help system.

Scope:	Assignable
Type:	Objeto
Initial value:	NIL

Consult the OnHelp event for further information.

1.4.1.1.27 TApplication.olcon

Specifies the icon to be shown.

Scope:	Design assignable
Type:	Object
Initial value:	Standard Xailer icon (TIcon)

1.4.1.1.28 TApplication.oMainForm

Main applications form.

Scope:	Design assignable
Type:	Object
Initial value:	NIL

1.4.1.2 TApplication.Methods

■ Constructor ■ Standard

Type	Name
■	CurrentDir
■	CurrentDrive
■	Initialize
■	DecimalSeparator
■	MessageBox
■	Run
■	SetOverlayIcon
■	SetProgress
■	Terminate

1.4.1.2.1 TApplication.CurrentDir

Return the current work directory.

Type	Standard
Parameters	None
Valor	Character
Retorno	

1.4.1.2.2 TApplication:CurrentDrive

Returns the current work drive.

Type	Standard
Parameters	None
Return value	Character

1.4.1.2.3 TApplication:Initialize

Constructor de la clase. Es llamado automáticamente por un procedimiento al principio de la aplicación creando una instancia de la misma que puede ser accedida a través de la variable pública 'Application'..

Class constructor. It is automatically called at the beginning of the applications creating an instance that can be access trough the public variable ***Application***.

Type	Constructor
Parameters	None
Return value	Self reference

1.4.1.2.4 TApplication:DecimalSeparator

Returns the decimal separator string used by the system

Type	Standard
Parameters	None
Return value	Character

1.4.1.2.5 TApplication:Run

Starts the applications showing the main application's form.

Type	Standard
Parameters	None
Return value	NIL

1.4.1.2.6 TApplication:SetOverlayIcon

Sets an overlay icon to be shown on the main application icon on the task bar.

Type	Standard
Parameters	<ulcon> : Icon object, resource name or identifier <cDescription> : Additional description for the overlay icon
Return value	<ISuccess> True if success

Important note: Only available on Windows 7 and above.

1.4.1.2.7 TApplication:SetProgress

Shows a progress bar on the application icon under the task bar

Type	Standard
Parameters	<nStatus> : Progress bar status: psNOPROGRESS, psNORMAL, psPAUSED, psERROR, psINDETERMINATE <nValue> : Current progress bar value <nTotal> : Total progress bar value
Return value	<ISuccess> True if success

Important note: Only available on Windows 7 and above.

1.4.1.2.8 TApplication:MessageBox

Shows the classic Windows message screen through the Windows API MessageBox() function. Review the MessageBox() function in the Windows API to get the complete relation of flags available.

Type	Standard
Parameters	<cName> : Text to be shown [<cTitle>]: Window title, Default value: the applications title (cTitle) [<nFlags>]: Dialog flags according to the MessageBox() API Windows function. Default value:

	MB_ICONINFORMATION. To pass more than one flag, you need to use the nOrg() function.
Return value	Return a MessageBox() API Windows value

Dialog buttons flags:

Flag	Significado
MB_ABORTRETRYIGNORE	Includes 3 buttons: Abort, Retry and Ignore.
MB_OK	Includes only one button: OK.
MB_OKCANCEL	Includes 2 buttons: OK and Cancel.
MB_RETRYCANCEL	Includes 2 buttons: Retry and Cancel.
MB_YESNO	Includes 2 buttons: Yes and No.
MB_YESNOCANCEL	Includes 3 buttons: Yes, No and Cancel.

Dialog icons flags:

Flag	Meaning
MB_ICONEXCLAMATION, MB_ICONWARNING	Includes the <i>exclamation</i> icon.
MB_ICONINFORMATION, MB_ICONASTERISK	Includes the <i>information</i> icon.
MB_ICONQUESTION	Includes the <i>question mark</i> icon.
MB_ICONSTOP, B_ICONERROR, B_ICONHAND	Includes a <i>stop</i> icon.

For more information, check the Windows API function [MessageBox\(\)](#)

1.4.1.2.9 TApplication:Terminate

Ends the application.

Type	Standard
Parameters	None
Return value	NIL

1.4.1.3 TApplication:Events

Nombre
OnHelp
OnQueryEndSession

1.4.1.3.1 TApplication:OnHelp

Event that is fired every time a help request is made.

Parameters:	<oSender>: A reference to the object that fired the request <HelpId>: HelpId property value of the control or form that request the help <nPosX> Mouse X coordinate when the request proceed from a mouse click <nPosY> Mouse Y coordinate when the request proceed from a mouse click
Return value:	NIL

The help system works as follows:

- The control that receives the help request checks that its property HelpId has a value. If Nil, then the request will be sent to its parent. If not Nil, then calls the help system of its container form. If on the form OnHelp event is assigned, then is triggered. If is not assigned then the **TApplication** help systems is called.
- If the **TApplication** OnHelp event is assigned, then is triggered. If is not assigned and its property oHelp has a valid THelp object, then it shows the help from the topic HelpId of the control or form that request the help.
- The **oSender** parameter of TForm or **TApplication** OnHelp event is the control that request the help whose HelpId had a value.

1.4.1.3.2 TApplication:OnQueryEndSession

Event that is fired every time a session logoff is done.

Parameters:	<oSender>: A reference to the object that fired the request <nSource>: Reservaed <nLogOff>: One or more of this options. Use nOr() ENDSESSION_CLOSEAPP (0x00000001) ENDSESSION_CRITICAL (0x40000000) ENDSESSION_LOGOFF (0x80000000)
	For further information follow this link
Return value:	A return value of zero tries to interrupt the session logoff

1.4.2 TAppData

This class contains all the application properties and public data.

Description:

The TApplication class creates a TAppData instance when the application initiates and makes it accessible through the public variable APPDATA. TAppData can be instantiated only once in the application. The use of his class is mainly like data container and offers a method to add new properties in run-time to the class.

Hierarchy Inherits from nobody
File Name \source\AppData.prg

1.4.2.1 TAppData.Methods

■ Constructor ■ Standard ■ Only after Create()

Typ	Name
■	AddData
■	Destroy
■	Initialize

1.4.2.1.1 TAppData:AddData

Creates a new DATA with the name <cName> in the TAppData class.

Type	Standard
Parameters	<cName> Property name [<xValorInicial>] Initial value for the property. (Optional)
Return value	NIL

Description:

This method allows to create new DATA in the TAppData class to be used later in any part of the program. The public variable AppData allows to have a reference to the TAppData created by the application. It is better to use this method to create data containers instead of use multiple public variables, because it does not force its declaration in all the modules where it will be used. besides that, the OnExit event allows to make operations over the data containers at the end of the application.

1.4.2.1.2 TAppData:Destroy

Generic class destroyer.

Type	Standard
Parameters	None
Return value	NIL

Description:

Class destroyer method. When the application ends, it is called automatically through the public object AppData.

1.4.2.1.3 TAppData:Initialize

Class constructor. It does not have any parameter.

Type	Constructor
Parameters	None
Return value	Self reference

Description:

This constructor is automatically called for the TApplication object when the application starts. Its reference is stored in the public variable AppData and can be accessible in any part of the application.

1.4.2.2 TAppDatEvents

Name	OnClose
-------------	---------

1.4.2.2.1 TAppData:OnClose

Event that is produced when the application is going to end.

Parameters	<oSender>: Object that triggers the event.
Return value:	NIL

Description:

This event allows to make any operation when the application is going to end. Its main objective is to save in external files any existing data that can be in different DATA created in the TAppData

class.

1.4.3 TPrinter

This class manages the different printers installed in the system.

Description:

Xailer creates automatically the public Printer variable, that contains an object from this class, with the default printer. It is possible to create more instances of this class if is needed, or modify the current active printer.

Hierarchy Inherits from nobody

File Name \source\Printer.prg

1.4.3.1 TPrinter:Properties

■ read Only ■ Assignable ■ Design assignable ■ Run-time assignable

Scope	Name	Type	Initial value
■	aPrinterName	Array	{}
■	cJobTitle	Character	""
■	cPort	Character	""
■	cPrinterName	Character	""
■	hDC	Numeric	0
■	IAborted	Logic	.F.
■	ICollate	Logic	.F.
■	IPreview	Logic	.F.
■	IPreviewModal	Logic	.T.
■	IPrintWithNext	Logic	.F.
■	nColor	Numeric	0
■	nCopies	Numeric	0
■	nDefaultBin	Numeric	0
■	nDuplex	Numeric	0
■	nOrientation	Numeric	0
■	nPageNumber	Numeric	0
■	nPaperLength	Numeric	0
■	nPaperSizeType	Numeric	0
■	nPaperWidth	Numeric	0
■	nPDFPrinter	Numeric	0
■	nPreviewShowMode	Numeric	smNORMAL
■	nPrinterIndex	Numeric	0
■	nPrintQuality	Numeric	0
■	nStatus	Numeric	0
■	oCanvas	Object	NIL
■	oExportInfo	Object	NIL

■	oPageSetupDlg	Object	NIL
■	oPreviewDC	Object	NIL
■	oPrnDlg	Object	NIL
■	oWndOwner	Object	NIL

1.4.3.1.1 TPrinter:aPrinterNames

Lists the printers installed in the system.

Scope:	readOnly
Type:	Array
Initial value:	{}

1.4.3.1.2 TPrinter:cJobTitle

Printer job title. This text will be shown in the printer queue manager and in the preview window's title.

Scope:	Assignable
Type:	Character
Initial value:	""

1.4.3.1.3 TPrinter:cPort

Printer port for the selected printer.

Scope:	readOnly
Type:	Character
Initial value:	""

1.4.3.1.4 TPrinter:cPrinterName

Printer's name for the selected printer.

Scope:	readOnly
Type:	Character
Initial value:	""

1.4.3.1.5 TPrinter:hDC

Handle device control needed to print through Windows. Its initial value is 0, and its value can be changed through the StartDoc method.

Scope:	read Only
Type:	Numeric
Initial value:	0

1.4.3.1.6 TPrinter:IAborted

.T, if the print process has been aborted using the AbortDoc() method.

Scope:	read Only
Type:	Logic
Initial value:	.F.

1.4.3.1.7 TPrinter:ICollate

.T. if multiple copies of printer output are collated (if the active printer supports this feature).

Scope:	Assignable
Type:	Logic
Initial value:	.F.

1.4.3.1.8 TPrinter:IPreview

.T. if a preview window will be shown.

Scope:	Assignable
Type:	Logic
Initial value:	.F.

1.4.3.1.9 TPrinter:IPreviewModal

.T. if the preview window will be shown in modal mode. See also the IPreview property.

Scope:	Assignable
Type:	Logic

Initial value: .T.

1.4.3.1.10 TPrinter:IPrintWithNext

If .T. if current document will not be show when preview is calles. Its content will be accumulated with next document.

Scope:	Assignable
Type:	Logic
Initial value:	.F.

1.4.3.1.11 TPrinter:nColor

Colors supported by the active printer.

Scope:	Assignable
Type:	Numeric
Initial value:	0
Possible values:	DMCOLOR_MONOCHROME, DMCOLOR_COLOR

1.4.3.1.12 TPrinter:nCopies

Indicates the number of copies to print.

Scope:	Assignable
Type:	Numeric
Initial value:	0

1.4.3.1.13 TPrinter:nDefaultBin

Indicates the default printer tray.

Scope:	Assignable
Type:	Numeric
Initial value:	0
Possible values:	DMBIN_FIRST 1
	DMBIN_UPPER 1
	DMBIN_ONLYONE 1

DMBIN_LOWER	2	
DMBIN_MIDDLE	3	
DMBIN_MANUAL	4	
DMBIN_ENVELOPE	5	
DMBIN_ENVMANUAL	6	
DMBIN_AUTO	7	
DMBIN_TRACTOR	8	
DMBIN_SMALLFMT	9	
DMBIN_LARGEFORMT	10	
DMBIN_LARGECAPACITY		11
DMBIN_CASSETTE	14	
DMBIN_FORMSOURCE	15	

Every printer has its own sets of paper bins that may not be the same of the above list. To consult the paper bins available on any printer you may use the methods `GetBinSourceNames` and `GetBinSourceTypes`. The first method gives you a list with all the paper bins names, and the second method reports the numerical identifier of every bin. That number is the one you should use on this property.

1.4.3.1.14 TPrinter:nDuplex

Duplex printer system for the active printer.

Scope:	Assignable
Type:	Numeric
Initial value:	0
Possible values:	DMDUP_SIMPLEX, DMDUP_VERTICAL, DMDUP_HORIZONTAL

1.4.3.1.15 TPrinter:nOrientation

Indicates the page orientation.

Scope:	Assignable
Type:	Numeric
Initial value:	0
Possible values:	DMORIENT_PORTRAIT, DMORIENT_LANDSCAPE

1.4.3.1.16 TPrinter:nPageNumber

Page number. This property is initialized on every printing process.

Scope:	Assignable
---------------	------------

Type:	Numeric
Initial value:	0

1.4.3.1.17 TPrinter:nPaperLength

Paper length indicated in millimeters x 10. When this information is specified, it overwrites the nPaperSizeType property and the paper length is specified manually (if and only if it is supported by the printer).

Scope:	Assignable
Type:	Numeric
Initial value:	0

1.4.3.1.18 TPrinter:nPaperSizeType

Indicates the paper size.

Scope:	Assignable		
Type:	Numeric		
Initial value:	0		
Possible values:	DMPAPER_FIRST	DMPAPER_LETTER	
	DMPAPER_LETTER	1 /* Letter 8 1/2 x 11 in	*/
	DMPAPER_LETTERSMALL	2 /* Letter Small 8 1/2 x 11 in	*/
	DMPAPER_TABLOID	3 /* Tabloid 11 x 17 in	*/
	DMPAPER_LEDGER	4 /* Ledger 17 x 11 in	*/
	DMPAPER_LEGAL	5 /* Legal 8 1/2 x 14 in	*/
	DMPAPER_STATEMENT	6 /* Statement 5 1/2 x 8 1/2 in	*/
	DMPAPER_EXECUTIVE	7 /* Executive 7 1/4 x 10 1/2 in	*/
	DMPAPER_A3	8 /* A3 297 x 420 mm	*/
	DMPAPER_A4	9 /* A4 210 x 297 mm	*/
	DMPAPER_A4SMALL	10 /* A4 Small 210 x 297 mm	*/
	DMPAPER_A5	11 /* A5 148 x 210 mm	*/
	DMPAPER_B4	12 /* B4 (JIS) 250 x 354	*/
	DMPAPER_B5	13 /* B5 (JIS) 182 x 257 mm	*/
	DMPAPER_FOLIO	14 /* Folio 8 1/2 x 13 in	*/
	DMPAPER_QUARTO	15 /* Quarto 215 x 275 mm	*/
	DMPAPER_10X14	16 /* 10x14 in	*/
	DMPAPER_11X17	17 /* 11x17 in	*/
	DMPAPER_NOTE	18 /* Note 8 1/2 x 11 in	*/
	DMPAPER_ENV_9	19 /* Envelope #9 3 7/8 x 8 7/8	*/
	DMPAPER_ENV_10	20 /* Envelope #10 4 1/8 x 9 1/2	*/
	DMPAPER_ENV_11	21 /* Envelope #11 4 1/2 x 10 3/8	*/
	DMPAPER_ENV_12	22 /* Envelope #12 4 1/2 x 11	*/
	DMPAPER_ENV_14	23 /* Envelope #14 5 x 11 1/2	*/
	DMPAPER_CSHEET	24 /* C size sheet	*/
	DMPAPER_DSHEET	25 /* D size sheet	*/
	DMPAPER_ESHEET	26 /* E size sheet	*/
	DMPAPER_ENV_DL	27 /* Envelope DL 110 x 220mm	*/
	DMPAPER_ENV_C5	28 /* Envelope C5 162 x 229 mm	*/
	DMPAPER_ENV_C3	29 /* Envelope C3 324 x 458 mm	*/
	DMPAPER_ENV_C4	30 /* Envelope C4 229 x 324 mm	*/
	DMPAPER_ENV_C6	31 /* Envelope C6 114 x 162 mm	*/

DMPAPER_ENV_C65	32	/* Envelope C65 114 x 229 mm	*/
DMPAPER_ENV_B4	33	/* Envelope B4 250 x 353 mm	*/
DMPAPER_ENV_B5	34	/* Envelope B5 176 x 250 mm	*/
DMPAPER_ENV_B6	35	/* Envelope B6 176 x 125 mm	*/
DMPAPER_ENV_ITALY	36	/* Envelope 110 x 230 mm	*/
DMPAPER_ENV_MONARCH	37	/* Envelope Monarch 3.875 x 7.5 in	*/
DMPAPER_ENV_PERSONAL	38	/* 6 3/4 Envelope 3 5/8 x 6 1/2 in	*/
DMPAPER_FANFOLD_US	39	/* US Std Fanfold 14 7/8 x 11 in	*/
DMPAPER_FANFOLD_STD_GERMAN	40	/* German Std Fanfold 8 1/2 x 12 in	*/
DMPAPER_FANFOLD_LGL_GERMAN	41	/* German Legal Fanfold 8 1/2 x 13 in	*/
DMPAPER_ISO_B4	42	/* B4 (ISO) 250 x 353 mm	*/
DMPAPER_JAPANESE_POSTCARD	43	/* Japanese Postcard 100 x 148 mm	*/
DMPAPER_9X11	44	/* 9 x 11 in	*/
DMPAPER_10X11	45	/* 10 x 11 in	*/
DMPAPER_15X11	46	/* 15 x 11 in	*/
DMPAPER_ENV_INVITE	47	/* Envelope Invite 220 x 220 mm	*/
DMPAPER_RESERVED_48	48	/* RESERVED--DO NOT USE	*/
DMPAPER_RESERVED_49	49	/* RESERVED--DO NOT USE	*/
DMPAPER_LETTER_EXTRA	50	/* Letter Extra 9 \275 x 12 in	*/
DMPAPER_LEGAL_EXTRA	51	/* Legal Extra 9 \275 x 15 in	*/
DMPAPER_TABLOID_EXTRA	52	/* Tabloid Extra 11.69 x 18 in	*/
DMPAPER_A4_EXTRA	53	/* A4 Extra 9.27 x 12.69 in	*/
DMPAPER_LETTER_TRANSVERSE	54	/* Letter Transverse 8 \275 x 11 in	*/
DMPAPER_A4_TRANSVERSE	55	/* A4 Transverse 210 x 297 mm	*/
DMPAPER_LETTER_EXTRA_TRANSVERSE	56	/* Letter Extra Transverse 9 \275 x 12 in	*/
DMPAPER_A_PLUS	57	/* SuperA/SuperA/A4 227 x 356 mm	*/
DMPAPER_B_PLUS	58	/* SuperB/SuperB/A3 305 x 487 mm	*/
DMPAPER_LETTER_PLUS	59	/* Letter Plus 8.5 x 12.69 in	*/
DMPAPER_A4_PLUS	60	/* A4 Plus 210 x 330 mm	*/
DMPAPER_A5_TRANSVERSE	61	/* A5 Transverse 148 x 210 mm	*/
DMPAPER_B5_TRANSVERSE	62	/* B5 (JIS) Transverse 182 x 257 mm	*/
DMPAPER_A3_EXTRA	63	/* A3 Extra 322 x 445 mm	*/
DMPAPER_A5_EXTRA	64	/* A5 Extra 174 x 235 mm	*/
DMPAPER_B5_EXTRA	65	/* B5 (ISO) Extra 201 x 276 mm	*/
DMPAPER_A2	66	/* A2 420 x 594 mm	*/
DMPAPER_A3_TRANSVERSE	67	/* A3 Transverse 297 x 420 mm	*/
DMPAPER_A3_EXTRA_TRANSVERSE	68	/* A3 Extra Transverse 322 x 445 mm	*/
DMPAPER_DBL_JAPANESE_POSTCARD	69	/* Japanese Double Postcard 200 x 148 mm	*/
DMPAPER_A6	70	/* A6 105 x 148 mm	*/
DMPAPER_JENV_KAKU2	71	/* Japanese Envelope Kaku #2	*/
DMPAPER_JENV_KAKU3	72	/* Japanese Envelope Kaku #3	*/
DMPAPER_JENV_CHOU3	73	/* Japanese Envelope Chou #3	*/
DMPAPER_JENV_CHOU4	74	/* Japanese Envelope Chou #4	*/
DMPAPER_LETTER_ROTATED	75	/* Letter Rotated 11 x 8 1/2 11 in	*/
DMPAPER_A3_ROTATED	76	/* A3 Rotated 420 x 297 mm	*/
DMPAPER_A4_ROTATED	77	/* A4 Rotated 297 x 210 mm	*/
DMPAPER_A5_ROTATED	78	/* A5 Rotated 210 x 148 mm	*/
DMPAPER_B4_JIS_ROTATED	79	/* B4 (JIS) Rotated 364 x 257 mm	*/
DMPAPER_B5_JIS_ROTATED	80	/* B5 (JIS) Rotated 257 x 182 mm	*/
DMPAPER_JAPANESE_POSTCARD_ROTATED	81	/* Japanese Postcard Rotated 148 x 100 mm	*/
DMPAPER_DBL_JAPANESE_POSTCARD_ROTATED	82	/* Double Japanese Postcard Rotated 148 x 200 mm	*/
DMPAPER_A6_ROTATED	83	/* A6 Rotated 148 x 105 mm	*/
DMPAPER_JENV_KAKU2_ROTATED	84	/* Japanese Envelope Kaku #2 Rotated	*/
DMPAPER_JENV_KAKU3_ROTATED	85	/* Japanese Envelope Kaku #3 Rotated	*/
DMPAPER_JENV_CHOU3_ROTATED	86	/* Japanese Envelope Chou #3 Rotated	*/
DMPAPER_JENV_CHOU4_ROTATED	87	/* Japanese Envelope Chou #4 Rotated	*/
DMPAPER_B6_JIS	88	/* B6 (JIS) 128 x 182 mm	*/
DMPAPER_B6_JIS_ROTATED	89	/* B6 (JIS) Rotated 182 x 128 mm	*/
DMPAPER_12X11	90	/* 12 x 11 in	*/
DMPAPER_JENV_YOU4	91	/* Japanese Envelope You #4	*/
DMPAPER_JENV_YOU4_ROTATED	92	/* Japanese Envelope You #4 Rotated	*/
DMPAPER_P16K	93	/* PRC 16K 146 x 215 mm	*/
DMPAPER_P32K	94	/* PRC 32K 97 x 151 mm	*/

DMPAPER_P32KBIG	95	/* PRC 32K(Big) 97 x 151 mm	*/
DMPAPER_PENV_1	96	/* PRC Envelope #1 102 x 165 mm	*/
DMPAPER_PENV_2	97	/* PRC Envelope #2 102 x 176 mm	*/
DMPAPER_PENV_3	98	/* PRC Envelope #3 125 x 176 mm	*/
DMPAPER_PENV_4	99	/* PRC Envelope #4 110 x 208 mm	*/
DMPAPER_PENV_5	100	/* PRC Envelope #5 110 x 220 mm	*/
DMPAPER_PENV_6	101	/* PRC Envelope #6 120 x 230 mm	*/
DMPAPER_PENV_7	102	/* PRC Envelope #7 160 x 230 mm	*/
DMPAPER_PENV_8	103	/* PRC Envelope #8 120 x 309 mm	*/
DMPAPER_PENV_9	104	/* PRC Envelope #9 229 x 324 mm	*/
DMPAPER_PENV_10	105	/* PRC Envelope #10 324 x 458 mm	*/
DMPAPER_P16K_ROTATED	106	/* PRC 16K Rotated	*/
DMPAPER_P32K_ROTATED	107	/* PRC 32K Rotated	*/
DMPAPER_P32KBIG_ROTATED	108	/* PRC 32K(Big) Rotated	*/
DMPAPER_PENV_1_ROTATED	109	/* PRC Envelope #1 Rotated 165 x 102 mm	*/
DMPAPER_PENV_2_ROTATED	110	/* PRC Envelope #2 Rotated 176 x 102 mm	*/
DMPAPER_PENV_3_ROTATED	111	/* PRC Envelope #3 Rotated 176 x 125 mm	*/
DMPAPER_PENV_4_ROTATED	112	/* PRC Envelope #4 Rotated 208 x 110 mm	*/
DMPAPER_PENV_5_ROTATED	113	/* PRC Envelope #5 Rotated 220 x 110 mm	*/
DMPAPER_PENV_6_ROTATED	114	/* PRC Envelope #6 Rotated 230 x 120 mm	*/
DMPAPER_PENV_7_ROTATED	115	/* PRC Envelope #7 Rotated 230 x 160 mm	*/
DMPAPER_PENV_8_ROTATED	116	/* PRC Envelope #8 Rotated 309 x 120 mm	*/
DMPAPER_PENV_9_ROTATED	117	/* PRC Envelope #9 Rotated 324 x 229 mm	*/
DMPAPER_PENV_10_ROTATED	118	/* PRC Envelope #10 Rotated 458 x 324 mm	*/
DMPAPER_USER	256		

1.4.3.1.19 TPrinter:nPaperWidth

Indicates paper width specified in millimeters x 10. When this information is specified, it overwrites the nPaperSizeType property and the paper width is specified manually (if and only if it is supported by the printer).

Scope:	Assignable
Type:	Numeric
Initial value:	0

1.4.3.1.20 TPrinter:nPDFPrinter

Printer ordinal value on the aPrinterNames array that represents a PDF printer. If assigned, from the printing preview window and on its tool bar a new button will appear to directly print with the PDF printer.

When the application starts, the Printer public object checks if there is any printer which has the word 'PDF' on its name, In case there is, this property is automatically initialized with its ordinal position on the aPrinterNames array.

Scope:	Assignable
Type:	Numeric
Initial value:	0

1.4.3.1.21 TPrinter:nPrinterIndex

Number of active printer, in the range specified by the aPrinterNames printers.

Scope:	Assignable
Type:	Numeric
Initial value:	0

1.4.3.1.22 TPrinter:nPrintQuality

Indicates the print quality.

Scope:	Assignable
Type:	Numeric
Initial value:	0
Possible values:	DMRES_DRAFT, DMRES_LOW, DMRES_MEDIUM, DMRES_HIGH

1.4.3.1.23 TPrinter:nPreviewShowMode

Indicates how the preview window will be shown.

Scope:	Assignable
Type:	Numeric
Initial value:	smNORMAL
Possible values:	smNORMAL, smMAXIMIZE, smMINIMIZE

1.4.3.1.24 TPrinter:nStatus

Indicates printer state. This property normally is only accessible in local printers. For network printers, you must use the ADSI API through the [IADsPrintQueueOperations::Get_Status\(\)](#) COM component.

Scope:	read Only
Type:	Numeric
Initial value:	0
Possible values:	Any combination from the following values: PRINTER_STATUS_PAUSED PRINTER_STATUS_ERROR PRINTER_STATUS_PENDING_DELETION

```

PRINTER_STATUS_PAPER_JAM
PRINTER_STATUS_PAPER_OUT
PRINTER_STATUS_MANUAL_FEED
PRINTER_STATUS_PAPER_PROBLEM
PRINTER_STATUS_OFFLINE
PRINTER_STATUS_IO_ACTIVE
PRINTER_STATUS_BUSY
PRINTER_STATUS_PRINTING
PRINTER_STATUS_OUTPUT_BIN_FULL
PRINTER_STATUS_NOT_AVAILABLE
PRINTER_STATUS_WAITING
PRINTER_STATUS_PROCESSING
PRINTER_STATUS_INITIALIZING
PRINTER_STATUS_WARMING_UP
PRINTER_STATUS_TONER_LOW
PRINTER_STATUS_NO_TONER
PRINTER_STATUS_PAGE_PUNT
PRINTER_STATUS_USER_INTERVENTION
PRINTER_STATUS_OUT_OF_MEMORY
PRINTER_STATUS_DOOR_OPEN
PRINTER_STATUS_SERVER_UNKNOWN
PRINTER_STATUS_POWER_SAVE

```

Note: You should use the `lAnd()` function to check the possible printer states.

```
lPaperOut := lAnd( Printer:nStatus, PRINTER_STATUS_PAPER_OUT )
```

1.4.3.1.25 TPrinter:oCanvas

TCanvas object used to print. This object is created automatically then it calls the `StartDoc()` method.

Scope:	Assignable
Type:	Logic
Initial value:	NIL

1.4.3.1.26 TPrinter:oExportInfo

TPrnExportInfo object that holds all the available export options from the print preview window. This object is common to all TPrinter instances and its created automatically when the `Create` method is calle.

Scope:	Assignable
Type:	Object
Initial value:	NIL

TPrnExportInfo members:

Name:	clmgPDFLicense
Description :	Sets the Image2PDF DLL license code. If not set the library will work on Demo mode showing a 'Evaluation mode' notice on every generated page. Visit Utility warrior for further information.
Type:	Character
Initial value:	""
Name:	clmgPDFPassword
Description :	Sets a password to open the PDF file.
Type:	Character
Initial value:	""
Name:	lImgPDFButton
Description :	If true the PDF export button will be shown on the Print preview window. If the file Image2PDF.dll is not found the button will be shown disabled.
Type:	Logical
Initial value:	.F.

Name:	lImgPDFCanCopy
Description :	If true the PDF can be copied.
Type:	Logical
Initial value:	.T.
Name:	lImgPDFCanEdit
Description :	If true the PDF can be edited.
Type:	Logical
Initial value:	.F.
Name:	lImgPDFCanPrint
Description :	If true the PDF can be printed.
Type:	Logical
Initial value:	.T.

Name:	lMailButton
Description :	If true the send Email button will be shown on the Print preview window. If the file Image2PDF.dll is not found the button will be shown disabled.
Type:	Logical
Initial value:	.F.

Sample:

```
WITH OBJECT Printer:oExportInfo
  :cImgPDFLicense := "... "
  :lImgPDFButton := .T.
END WITH
```

1.4.3.1.27 TPrinter:oPageSetupDlg

Reference to the TPageSetupDlg object type when the PageSetup() method is executed when the user specifies the page setup.

Scope:	Assignable
Type:	Logic
Initial value:	NIL

1.4.3.1.28 TPrinter:oPreviewDC

Reference to the internal TPreviewDC object type that manages the print preview documents. This object will be created automatically when it calls the StartDoc() method and the IPreview is set to .T.

Scope:	Assignable
Type:	Logic
Initial value:	NIL

1.4.3.1.29 TPrinter:oPrnDlg

Reference to the TPrintDlg object type when the Setup() is executed when the users sets the printer specifications.

Scope:	Assignable
Type:	Logic
Initial value:	NIL

1.4.3.1.30 TPrinter:oWndOwner

Reference to the proprietary Tform object type used by the document to print. It is assigned in the StartDoc(). method.

Scope:	Assignable
Type:	Logic
Initial value:	NIL

1.4.3.2 TPrinter:Methods

■ Constructor ■ Standard

Type	Name
■	AbortDoc
■	Attributes
■	CanPrintToPdf
■	Comment
■	CountBinSource
■	CountPapers
■	Create
■	Destroy
■	DocumentSetup
■	DriverName
■	EndDoc
■	EndPage
■	GetBinSourceNames
■	GetBinSourceTypes
■	GetPaperNames
■	GetPaperSizes
■	GetPaperTypes
■	GetPrinterDefault
■	GetPrinterFonts
■	GetPrinterNames
■	GetResolutions
■	Initialize
■	IsDuplex
■	IsPrinting
■	Location
■	PageSetup
■	PaperPhysicalSize
■	PaperRes
■	PaperSize
■	PhysicalOffset
■	Port
■	Preview
■	PreviewControl
■	PrintProcessor
■	Resolution
■	Run
■	SepFile
■	ServerName
■	Setup
■	ShareName
■	StartDoc
■	StartDocPdf
■	StartPage

■ WriteData

1.4.3.2.1 TPrinter:AbortDoc

Aborts the current printer process. The IAborted is set to .T.

Type	Standard
Parameters	None
Return value	Logic .T. if the operation is successful

1.4.3.2.2 TPrinter:Attributes

Attributes of the active printer.

Type	Standard
Parameters	None
Return value	String attributes

1.4.3.2.3 TPrinter:CanPrintToPdf

Returns .T. if there is any PDF printer that can send to file directly any job.

Type	Standard
Parámetros	None
Return value	Logic

1.4.3.2.4 TPrinter:Comment

Comments of the active printer.

Type	Standard
Parameters	None
Return value	String comment

1.4.3.2.5 TPrinter:CountBinSources

Total number of trays supported by the active printer.

Type	Standard
Parameters	None
Return value	Tray number

1.4.3.2.6 TPrinter:CountPapers

Number of paper types supported by the active printer.

Type	Standard
Parameters	None
Return value	Number of paper types.

1.4.3.2.7 TPrinter:Create

Class constructor.

Type	Constructor
Parámetros	None
Return value	Self Reference

1.4.3.2.8 TPrinter:Destroy

Class destroyer. It ends the current document, if it exists.

Type	Standard
Parámetros	None
Return value	NIL

1.4.3.2.9 TPrinter:DocumentSetup

Executes the standard document configuration standard dialog.

Type	Standard
Parámetros	None
Return value	NIL

1.4.3.2.10 TPrinter:DriverName

Drives name for the active printer.

Type	Standard
Parámetros	None
Return value	Character

1.4.3.2.11 TPrinter:EndDoc

Finalizes to print the active document.

Type	Standard
Parámetros	None
Return value	NIL

1.4.3.2.12 TPrinter:EndPage

Indicates the end of page from the active document print operation.

Type	Standard
Parámetros	None
Return value	True if the operation is successful

1.4.3.2.13 TPrinter:GetBinSourceNames

Returns an array with all the available tray names in the active printer.

Type	Standard
Parámetros	None
Return value	Array

Consult also the property nDefaultBin.

1.4.3.2.14 TPrinter:GetBinSourceTypes

Returns an array all the available tray identifiers in the active printer.

Type	Standard
Parámetros	None
Return value	Array

Consult also the property nDefaultBin.

1.4.3.2.15 TPrinter:GetPaperNames

Returns an array with all the available paper names in the active printer.

Type	Standard
Parámetros	None
Return value	Array

1.4.3.2.16 TPrinter:GetPaperSizes

Returns an array with all the available paper size in the active printer with the { width, height } format specified in millimeters x 10 (portrait orientation).

Type	Standard
Parameters	None
Return value	Array

1.4.3.2.17 TPrinter:GetPaperTypes

Returns an array with all the available paper types identifiers in the active printer.

Type	Standard
Parámetros	None
Return value	Array

1.4.3.2.18 TPrinter:GetPrinterDefault

Default printer name. Equivalent to cPrinterName.

Type	Standard
Parámetros	None
Return value	Character

1.4.3.2.19 TPrinter:GetPrinterFonts

Returns an array with all the available font names in the active printer. It is normal that the printer does not have any specific font.

Type	Standard
Parameters	None
Return value	Array

1.4.3.2.20 TPrinter:GetPrinterNames

Returns an array with all the printer names. It is equivalent to aPrinterNames.

Type	Standard
Parámetros	None
Return value	Array

1.4.3.2.21 TPrinter:GetResolutions

Returns an array with all available resolutions in the active printer with the { width, height } format specified in points per inch .

Type	Standard
Parámetros	None
Return value	Array

1.4.3.2.22 TPrinter:IsDuplex

Returns .T. if the printer supports the full duplex printer mode.

Type	Standard
Parámetros	None
Return value	Logic

1.4.3.2.23 TPrinter:IsPrinting

Returns .T. if there is a printing process running.

Type	Standard
Parámetros	None
Return value	Logic

1.4.3.2.24 TPrinter:Location

Location of the active printer.

Type	Standard
Parameters	None
Return value	String location

1.4.3.2.25 TPrinter:PageSetup

Executes the standard page configuration API dialog.

Type	Standard
Parámetros	<oForm>: Proprietary dialog form
Return value	NIL

1.4.3.2.26 TPrinter:PaperPhysicalSize

Returns an array { width, height } with the physical paper dimensions specified in millimeters x 10.

Type	Standard
Parameters	None
Return value	Array

1.4.3.2.27 TPrinter:PaperRes

Returns an array { width, height } with the available paper resolutions indicated in pixels.

Type	Standard
Parámetros	None
Return value	Array

1.4.3.2.28 TPrinter:PaperSize

Returns an array { width, height } with the available paper dimensions indicated in millimeters x 10.

Type	Standard
Parameters	None
Return value	Array

1.4.3.2.29 TPrinter:PhysicalOffset

Returns an array { nXOffset, nYOffset } with the printer offsets indicated in millimeters x 10.

Type	Standard
Parameters	None
Return value	Array

1.4.3.2.30 TPrinter:Port

Returns the printer name used by the active printer.

Type	Standard
Parámetros	None
Return value	Character

1.4.3.2.31 TPrinter:Preview

Previews a document in a window.

Type	Standard
Parámetros	<p>[<nShowMode>] Display mode. By default the property nShowMode <IModal>: If the document preview will be done over a modal window. Default value: .T. [<oParent>]: Window owner of the dialog. By default, application main window [<bOnClose>]: Codeblock to execute when the preview form is closed</p>
Return value	NIL

1.4.3.2.32 TPrinter:PreviewControl

Returns a TPreviewControl object to be used to show on screen any printing made on the **TPrinter** object.

Type	Standard
-------------	----------

Parameters	[<oParent>]: Window owner of the dialog. By default, application main window
Return value	TPreviewControl object

For comfort the value returned by this method is not a reference to the form itself, but directly to the TPreviewControl object which inside the form.

This method can be used instead of the Preview method to change any of its properties before the form is shown.

Sample:

```
with object Printer:PreviewControl()
  :nZoomIndex := 8
  :oParent:Show()
end with
```

1.4.3.2.33 TPrinter:PrintProcessor

Print processor of the active printer.

Type	Standard
Parameters	None
Return value	String description

1.4.3.2.34 TPrinter:Resolution

Returns an array { nLogPixelX, nLogPixelY } with the active printer resolution specified in pixels per inch.

Type	Standard
Parámetros	None
Return value	Array

1.4.3.2.35 TPrinter:Run

Executes the standard TPrintDlg API dialog to start the printout.

Type	Standard
Parámetros	None

Return value	<IOk> : If true, the user did exit from the dialog using the 'OK' button. On that case a new printing process is started in the same way you do with the StartDoc method but using the configuration done by the user through the dialog.
---------------------	---

1.4.3.2.36 TPrinter:SepFile

Separator page file name of the active printer.

Type	Standard
Parameters	None
Return value	string file name

1.4.3.2.37 TPrinter:ServerName

Returns the printer server name.

Type	Standard
Parámetros	None
Return value	Character

1.4.3.2.38 TPrinter:Setup

Executes the standard API printer configuration dialog (TPrintDlg)

Type	Standard
Parámetros	<oForm> : Proprietary form dialog.
Return value	<IResult> True if success

1.4.3.2.39 TPrinter:ShareName

Returns the share printer name.

Type	Standard
-------------	----------

Parámetros	None
Return value	Character

1.4.3.2.40 TPrinter:StartDoc

Starts to print a new document in the active printer.

Type	Standard
Parámetros	[< cJobTitle >]: Print job name. Default value: the 'cJobTitle' property from the class [< oWndOwner >]: Proprietary working window. Default value: active window
Return value	.T. if the operation is successful

1.4.3.2.41 TPrinter:StartDocPdf

Starts to print a new document directly to file with a available PDF printer.

Type	Standard
Parámetros	< cOutput >: File name [< cJobTitle >]: Print job name. Default value: the 'cJobTitle' property from the class [< oWndOwner >]: Proprietary working window. Default value: active window
Return value	.T. if the operation is successful

1.4.3.2.42 TPrinter:StartPage

Starts to print a new page from the active document.

Type	Standard
Parámetros	None
Return value	.T. if the operation is successful

1.4.3.2.43 TPrinter:WriteData

Prints directly in the printer the data buffer cData without the printer manager interference. This method executes by itself all the operations to print (creates a new document, writes the cData and closes the document).

Type	Standard
Parámetros	<cData>: Binary data to print [<cJobTitle>]: Printer job name. Default value: 'cJobTitle' from the class.
Return value	.T. if the operation is successful

1.4.3.3 TPrinter:Events

Name	
	OnPreview

1.4.3.3.1 TPrinter:OnPreview

Evento fired when the preview windows is going to be shown.

Parametros	<oSender> Object that fires the event
:	<nShowMode> View mode. See nShowMode property for more information
	<IModal> True if the preview was requested modal when the Preview method was called
Valor Retorno:	If FALSE is returned the default preview window will not be shown giving the opportunity to shown your own preview form.

1.4.4 TScreen

The TScreen class offer information about the display where the application is executed.

Description:

The TScreen class provides display information like dimensions, resolution, etc.

Xailers instances automatically a TScreen object class when it starts the application, and can be

access through the public variable SCREEN (global).

Hierarchy Inherits from nobody
See also TPrinter, TApplication
File Name \source\Screen.prg

1.4.4.1 TScreen.Properties

■ read Only ■ Assignable ■ Design assignable ■ Run-time assignable

Scope	Name	Type	Initial value
■	aMonitors	Array	{}
■	Handle	Numeric	0
■	hDC	Numeric	0
■	ITouch	Logic	.F.
■	nBitsPerPixel	Numeric	0
■	nCaptionHeight	Numeric	0
■	nClientHeight	Numeric	0
■	nClientLeft	Numeric	0
■	nClientTop	Numeric	0
■	nClientWidth	Numeric	0
■	nColorPlanes	Numeric	0
■	nHeight	Numeric	0
■	nMonitors	Numeric	0
■	nPixelsPerInch	Numeric	0
■	nWidth	Numeric	0
■	oCursorArrow	Objeto	TCursor
■	oCursorDropNo	Objeto	TCursor
■	oCursorDropYes	Objeto	TCursor

1.4.4.1.1 TScreen:aMonitors

Array of **TDisplayMonitor** objects with the information of all available monitors.

Scope	read Only
Type	Array
Initial value	{}

The **TDisplayMonitor** class only contains read only properties of the monitor:

Handle	Handle
cName	Name
IPrimary	True if is the primary monitor
nLeft	Monitor left coordinate. Always 0 when is primary
nTop	Monitor top coordinate. Always 0 when is primary
nWidth	Monitor width in pixels

nHeight	Monitor height in pixels
nClientLeft	Monitor client area left coordinate
nClientTop	Monitor client area top coordinate
nClientWidth	Monitor client area width in pixels
nClientHeight	Monitor client area height in pixels

1.4.4.1.2 TScreen:Handle

Windows desktop screen handle. It will return a numeric value.

Scope	read Only
Type	Numeric
Initial value	0

1.4.4.1.3 TScreen:hDC

Windows desktop device context handle. It will return a numeric value. This handle is created and deleted with the methods CreateDC and DeleteDC.

Scope	read Only
Type	Numeric
Initial value	0

1.4.4.1.4 TScreen:ITouch

If true the device has a touch screen.

Scope	read Only
Type	Logic
Initial value	.F.

1.4.4.1.5 TScreen:nBitsPerPixel

Indicates the number of bits per pixel in the screen.

Scope	read Only
Type	Numeric

Initial value	0
----------------------	---

1.4.4.1.6 TScreen:nCaptionHeight

Indicates the windows title's height

Scope	read Only
Type	Numeric
Initial value	0

1.4.4.1.7 TScreen:nClientHeight

Indicates the client screen height.

Scope	read Only
Type	Numeric
Initial value	0

1.4.4.1.8 TScreen:nClientLeft

Indicates the left coordinate for the client zone screen .

Scope	read Only
Type	Numeric
Initial value	0

1.4.4.1.9 TScreen:nClientTop

Indicates the top coordinate for the client zone screen .

Scope	read Only
Type	Numeric
Initial value	0

1.4.4.1.10 TScreen:nClientWidth

Indicates the client zone screen width.

Scope	read Only
Type	Numeric
Initial value	0

1.4.4.1.11 TScreen:nColorPlanes

Indicates the number of color planes.

Scope	read Only
Type	Numeric
Initial value	0

1.4.4.1.12 TScreen:nHeight

Indicates the screen height.

Scope	read Only
Type	Numeric
Initial value	0

1.4.4.1.13 TScreen:nMonitors

Indicates the number of monitors availables.

Scope	read Only
Type	Numeric
Initial value	0

1.4.4.1.14 TScreen:nPixelsPerInch

Indicates the number of pixels per inch.

Scope	read Only
Type	Numeric
Initial value	0

1.4.4.1.15 TScreen:nWidth

Indicates the screen width.

Scope	read Only
Type	Numeric
Initial value	0

1.4.4.1.16 TScreen:oCursorArrow

TCursor object with the classic arrow cursor (IDC_ARROW).

Scope	read Only
Type	Object
Initial value	TCursor():New(IDC_ARROW)

1.4.4.1.17 TScreen:oCursorDropNo

TCursor object with the classic prohibited cursor (IDC_NO) to be used on Drag & Drop operations on controls that do not admit Drop.

Scope	Assignable
Tipo	Object
Initial value	TCursor():New(IDC_NO)

Note: The assignment of a new cursor does not destroy the current one. If you are not going to restore the old cursor you should destroy it before assigning the new value.

1.4.4.1.18 TScreen:oCursorDropYes

TCursor object used for Drag & Drop operations between control that do admit Dropping. The cursor is not a system cursor but a predefined Xailer cursor: 'XA_CUR_DRAG'.

Scope	Assignable
Tipo	Object
Initial value	TCursor():New('XA_CUR_DRAG')

Note: The assignment of a new cursor does not destroy the current one. If you are not going to

restore the old cursor you should destroy it before assigning the new value.

1.4.4.2 TScreen.Methods

■ Constructor ■ Standard

Type	Name
■	CreateDC
■	DeleteDC
■	Inititalize
■	PaperPhysicalSize
■	PaperRes
■	PaperSize
■	Resolution

1.4.4.2.1 TScreen:CreateDC

Creates a device context handle. Its value is saved on the hDC property.

Type	Standard
Parameters	None
Return value	NIL

1.4.4.2.2 TScreen>DeleteDC

Deletes a device context handle created with CreateDC.

Type	Standard
Parameters	None
Return value	NIL

1.4.4.2.3 TScreen:Initialize

Class constructor

Type	Constructor
Parameters	None
Return value	Self reference

1.4.4.2.4 TScreen:PaperPhysicalSize

Returns an array (width, height) with the physical screen dimensions specified in millimeters x 10

Type	Standard
Parameters	None
Return value	Array

1.4.4.2.5 TScreen:PaperRes

Returns an array (width, height) indicating the screen resolution in pixels.

Type	Standard
Parameters	None
Return value	Array

1.4.4.2.6 TScreen:Resolution

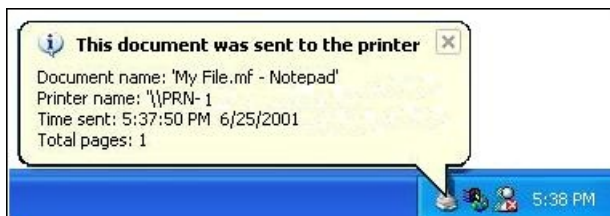
Returns an array { nLogX, nLogY } indicating the screen resolution in pixels per inch.

Type	Standard
Parameters	None
Return value	Array

1.5 Windows Objects

1.5.1 TBalloon

This class represents a balloon Windows Tooltip .



A balloon is a pop-up window with text that normally is shown by the controls with the cBalloon

assigned when the mouse pointer is stopped over the control for a certain period of time.

This class allows to manage the standard Windows API Tooltips and it may be that the user does not need to access to it.

Hierarchy Inherits from TToolTip
File Name \source\Tooltip.prg

1.5.1.1 TBalloon:Methods

■ Constructor ■ Standard

Type	Name
■	AddTool
■	SetTool

1.5.1.1.1 TBalloon:AddTool

Assigns a Tooltip to the specified control.

Type	Standard
Parameters	<oControl> Control where the Tooltip will be assigned. <aZone> Coordinate array that indicate the position where the different Tooltips for the same control will be shown. <nId> Tooltip identifier
Return value	NIL

1.5.1.1.2 TBalloon:SetToolInfo

Modifies a control's Tooltip.

Type	Standard
Parameters	<oControl> Control where the Tooltip is assigned to
Return value	NIL

1.5.2 TBitmap

This class represents a Windows bitmap object.

Description:

The TBitmap class allows to access native Windows bitmap image objects.

Hierarchy Inherits from TWinObject
File Name \source\Bitmap.prg

1.5.2.1 TBitmap:Properties

■ read Only
 ■ Assignable
 ■ Design assignable
 ■ Run-time assignable

Scope	Name	Type	Initial value
■	cName	Character	""
■	nHeight	Numeric	0
■	nWidth	Numeric	""

1.5.2.1.1 TBitmap:cName

Indicates the bitmap name. it can be a filename, a bitmap from a resource file or a standard system bitmap.

Scope	Assignable
Type	Character
Initial value	""

1.5.2.1.2 TBitmap:nHeight

Indicates the bitmap height.

Scope	read Only
Type	Numeric
Initial value	0

1.5.2.1.3 TBitmap:nWidth

Indicates the bitmap width.

Scope	read Only
Type	Numeric

Initial value 0

1.5.2.2 TBitmap:Methods

■ Constructor ■ Standard ■ Only after Create()

Typ Name	e
■ CreateFromHandle	
■ CreateNew	
■ Destroy	
■ LoadFromFile	
■ LoadFromResource	
■ New Create Load	
■ Paint	
■ Save	

1.5.2.2.1 TBitmap:CreateFromHandle

Creates a bitmap object from other bitmap handle.

Type	Constructor
Parameters	<nHandle> Bitmap handle to copy
Return value	<Self> Self reference

An additional bitmap is not really created. If the bitmap is destroyed it will destroy the original object where it was copied from.

1.5.2.2.2 TBitmap:CreateNew

Creates an empty bitmap.

Type	Constructor
Parameters	<nWidth> bitmap width <nHeight> bitmap height <nColorBits> Bits per pixel: <ul style="list-style-type: none"> • 1: Monochrome • 4: 16 colours • 8: 256 colours • 24: 16 millions of colours

	<ul style="list-style-type: none"> • 32: savem as 24 plus Alfa mask
Return value	<Self> Self reference

Sample:

```

WITH OBJECT oBitmap := TBitmap():CreateNew( 2200, 1500, 1 )
  WITH OBJECT oCanvas := TCanvas():Create()
    :hDC := CreateCompatibleDC()
    SelectObject( :hDC, oBitmap:Handle )
    :FillRect( { 0, 0, 2200, 1500 }, clWhite )
    :TextOut( 100, 100, "Hello!!!",, clBlack )
    DeleteDC( :hDC )
  END
:Save( ::oFileSaveDlg1:cFileName )
:Destroy()
END

```

1.5.2.2.3 TBitmap:Destroy

Releases the resources used by the bitmap object.

Type	Only after Create()
Parameters	None
Return value	NIL

An additional bitmap is not really created. If the bitmap is destroyed it will destroy the original object where it was copied from.

1.5.2.2.4 TBitmap:LoadFromFile

Loads the specified bitmap from a file.

Type	Constructor
Parameters	<cName> Filename that contains the bitmap.
Return value	<Self> Self reference

1.5.2.2.5 TBitmap:LoadFromResource

Loads the specified bitmap from a resource file.

Type	Constructor
Parameters	<cName> Resource name where the bitmap is loaded from
Return value	<Self> Self reference

1.5.2.2.6 TBitmap:New

Loads the specified bitmap in memory.

Type	Constructor
Parameters	<cName> Bitmap name. It can be a filename, a bitmap from a resource file or an standard system bitmap.
Return value	<Self> Self reference

The **New** and **Create** constructors are equivalent for this class.

1.5.2.2.7 TBitmap:Paint

Show the bitmap in the device indicated by **hDC** in the specified coordinates.

Type	Only after Create()()
Parameters	<hDC> Device <nLeft> Top column <nTop> Top row <nRight> bottom column <nBottom> Bottom row
Return value	NIL

1.5.2.2.8 TBitmap:Save

Saves the bitmap to file.

Type	Only after Create()()
Parameters	<cFileName> Bitmap filename
Return value	<ISuccess> True if success

1.5.3 TBrush

This class allows to manage the Windows API **"Brush"** object type. The Brush objects allow to establish the background painting for all kind of controls. It can be from a simple paint operation using any color to the use of bitmaps in mosaic mode to fill the background.

To specify the control's background color, it is not needed to instantiate a TBrush object, due we only need to indicate the color in the nClrPane property. The Brush object allows to establish besides the color, other style properties and even bitmaps that can be used o create more sophisticated backgrounds.

Hierarchy	Inherits from TWinObject
File name	\source\Brush.prg

1.5.3.1 TBrush:Properties

■ read Only ■ Assignable ■ Design assignable ■ Run-time assignable

Scope	Name	Type	Initial value
■	hBitmap	Numeric	0
■	nColor	Numeric	0
■	nStyle	Numeric	0

1.5.3.1.1 TBrush:hBitmap

Bitmap handle used like a brush pattern.

Scope:	Assignable
Type:	Numeric
Initial value:	0

Specify the handle to an existing bitmap. To get this handle you can use the TBitmap class. The indicated image will be shown in the background in mosaic mode.

1.5.3.1.2 TBrush:nColor

Solid color used by the brush to paint.

Scope:	Assignable
Type:	Numeric
Initial value:	0

Consult the appendix for the list of available colors

1.5.3.1.3 TBrush:nStyle

Specifies the brush style.

Scope:	Assignable
Type:	Numeric
Initial value:	0

Description:

Specify the brush plot:

Value	Description
HS_BDIAGONAL	Diagonal plot (Top, from left to right)
HS_CROSS	Cross plot vertical-horizontal
HS_DIAGCROSS	Diagonal 45 degrees Plot
HS_FDIAGONAL	Diagonal plot (Down, from left to right)
HS_HORIZONTAL	Horizontal plot
HS_VERTICAL	Vertical plot

1.5.3.2 TBrush:Methods

■ Constructor ■ Standard ■ Only after Create()

Typ Name	e
■ Clone	
■ Create	
■ Destroy	

1.5.3.2.1 TBrush:Clone

Creates a new Brush object based in the current one.

Type	Standard
Parameters	None
Return	oBrush

value	
-------	--

1.5.3.2.2 TBrush:Create

Creates a brush based in the parameters received.

Type	Constructor
Parameters	[<nStyle> Brush style. Optional <nColor> Solid color used by the brush to paint. Optional. <Handle> Bitmap handle used like brush pattern. Optional.
Return value	Self

1.5.3.2.3 TBrush:Destroy

Destroys the object and release system resources.

Type	Standard
Parameters	None
Return value	NIL

1.5.4 TCanvas

This class allows to draw/paint on the specified device (screen or printer) using pens, brushes, fonts, lines, colors, rectangles and images.

Description:

The object **oDevice** specifies the device where the different coordinates will be calculated. Normally is a TPrinter or TPreviewDC object.

The **hDC** property is aligned with the device that will be used and it must be different that zero. Normally this **hDC** matches with the device's **hDC**, but not always: one exception is in a Preview Page action, because in this case the device used to make the calculations is the printer, but the real printout operation is done in other device (display). Every time that the Printer object instantiate a TCanvas object, it assigns automatically the **oDevice** object and **hDC** property.

Before to establish any paint operation, it is needed to establish the coordinate system to be used, with the **nMapMode** property.

This class is instantiated automatically for the TPrinter class during the printing process, calling the StartPage() method, and it is accessible through the oCanvas object.

Hierarchy Inherits from nobody
File Name \source\Canvas.prg

1.5.4.1 TCanvas:Properties

■ read Only ■ Assignable ■ Design assignable ■ Run-time assignable

Scope	Name	Type	Initial Value
■	aPenPos	Array	{ 0, 0 }
■	hDC	Numeric	0
■	lTransparent	Logic	.T.
■	nClrPane	Numeric	0
■	nMapMode	Numeric	mmPIXELS
■	nTextAlignment	Numeric	taLEFT
■	nTextVAlignme nt	Numeric	vaCENTER
■	oDevice	Object	NIL
■	oFont	Object	NIL
■	oPen	Object	NIL

1.5.4.1.1 TCanvas:aPenPos

Two dimension numeric array that indicates the current printer coordinate (x, y). The measure unit is defined by the nMapMode property value.

Scope:	Assignable
Type:	Array
Initial value:	{ 0, 0 }

1.5.4.1.2 TCanvas:hDC

Handle identifier for a Windows' API. When the Printer object instantiates an TCanvas object, it assign this property automatically.

Scope:	Assignable
Type:	Numeric
Initial value:	0

Description:

The hDC is related to the device to be used to print. Normally this device will be the same to be used to get the coordinate calculations and consequently has the same effect to use oDevice:hDC or TCanvas:hDC. However, there are some cases that this is not true, for example, in the Print Preview process, due that in this case the device used to make the calculations is the printer, but

the real printout operation is done in other device.

1.5.4.1.3 TCanvas:ITransparent

Affects mainly to the text painting operations. If its value is .T., the following painting operation will be done using a transparent background color.

Scope:	Assignable
Type:	Logic
Initial value:	.T.

1.5.4.1.4 TCanvas:nClrPane

Background color to be used for the next painting operations.

Scope:	Assignable
Type:	Numeric
Initial value:	NIL

If the value is NIL, it will assign the white color for the painting operation.

Consult the appendix for the list of available colors

1.5.4.1.5 TCanvas:nMapMode

Measure unit to be used for the next painting operations.

Scope:	Assignable
Type:	Numeric
Initial value:	mmPIXELS
Possible values:	mmPIXELS, mmHIMETRICS, mmHIENGLISH, mmSIMULCHAR

Description:

This property allows to establish the coordinate system to be used in the next painting operations. The possible values are:

- mmPIXELS: Pixels
- mmHIMETRICS: Tenth of millimeters
- mmHIENGLISH: Inches x 10
- mmSIMULCHAR: DOS print out simulation. One unit in the X coordinate matches with the "B" character width from the current selected font. One unit in the Y coordinate corresponds to the height from the current selected font, including the upper and lower margin spaces.

1.5.4.1.6 TCanvas:nTextAlignment

Defines the way that the text will be aligned in the next painting operations.

Scope:	Assignable
Type:	Numeric
Initial value:	taLEFT
Possible values:	taLEFT, taRIGHT, taCENTER

The possible values are:

- taLEFT: Text aligned to the left
- taRIGHT: Text aligned to the right
- taCENTER: Text centered

1.5.4.1.7 TCanvas:nTextVAlignment

Defines the vertical text alignment for the next painting operations.

Scope:	Assignable
Type:	Numeric
Initial value:	vaCENTER
Possible values:	vaTOP, vaCENTER, vaBOTTOM

The possible values are:

- vaTOP: Text aligned to the Top
- vaCENTER: Text aligned to the Center
- vaBOTTOM: Text aligned to the Bottom

1.5.4.1.8 TCanvas:oDevice

Object to be used to calculate the coordinates in a print out operation. Normally is an object from the TPrinter or TPreviewDC classes.

Scope:	Assignable
Type:	Numeric
Initial value:	0

Description:

The **oDevice** object specifies the device to be used to calculate the coordinates in a printout operation. Normally this device will be the same to be used to get a hDC and therefore it will have the same effect to use oDevice:hDC or TCanvas:hDC. However, there are cases when this is different, for example, in a Print Preview action, due that the Printer device is used to make all the calculations but the printout is done in other device (display).

1.5.4.1.9 TCanvas:oFont

Font object to be used in the next text painting operations.

Scope:	Assignable
Type:	Object
Initial value:	NIL

Description:

The Font object used must not be referenced to any device, because it will be done automatically. Any existing assigned object with this property will be destroyed when it is reassigned. When the Tcanvas object is destroyed, the Font object is destroyed as well. See also the SelectFont() method.

1.5.4.1.10 TCanvas:oPen

Pen object to be used in the next painting operations

Scope:	Assignable
Type:	Object
Initial value:	NIL

1.5.4.2 TCanvas:Methods

■ Constructor ■ Standard

Type	Name
■	AngleArc
■	Arc
■	ArcTo
■	Chord
■	Create
■	DrawPicture
■	DrawRichEdit
■	Ellipse

■	ExtFloodFill
■	FillRect
■	LineTo
■	MoveTo
■	New
■	NewFromDC
■	Pie
■	PixelsToMapMode
■	PixelsToTwips
■	PointToPixels
■	Polygon
■	Rectangle
■	RectToPixels
■	RoundRect
■	SelectFont
■	SelectPen
■	TextExtent
■	TextHeight
■	TextLines
■	TextOut
■	TextRect
■	TextWidth

1.5.4.2.1 TCanvas:AngleArc

Draws a line segment and an arc.

Type	Standard
Parameters	<p><nX>: Circle center X coordinate with measure units specified by 'nMapMode'</p> <p><nY>: Circle center Y coordinate with measure units specified by 'nMapMode'</p> <p><nRadius>: Circle radius</p> <p><nStartAngle>: Start angle in grades realtive to X axe</p> <p><nSweepAngle>: End angle in grades realtive to X axe</p>
Return value	<p>ISuccess .T. if the operation is successful</p>

1.5.4.2.2 TCanvas:Arc

Draws an elliptical arc.

Type	Standard
Parameters	<aRect> : Rectangle (Left, Top, Right, Bottom) coordinates. The measure units are specified by 'nMapMode' <nXStart> : Radius initial X point that defines the arc <nYStart> : Radius initial Y point that defines the arc <nXEnd> : Radius final X point that defines the arc <nYEnd> : Radius final Y point that defines the arc
Return value	ISuccess .T. if the operation is successful

1.5.4.2.3 TCanvas:ArcTo

Draws an elliptical arc and updates Canvas position.

Type	Standard
Parameters	<aRect> : Rectangle (Left, Top, Right, Bottom) coordinates. The measure units are specified by 'nMapMode' <nXStart> : Radius initial X point that defines the arc <nYStart> : Radius initial Y point that defines the arc <nXEnd> : Radius final X point that defines the arc <nYEnd> : Radius final Y point that defines the arc
Return value	ISuccess .T. if the operation is successful

1.5.4.2.4 TCanvas:Chord

Draws a chord.

Type	Standard
Parameters	<aRect> :

	Rectangle (Left, Top, Right, Bottom) coordinates. The measure units are specified by 'nMapMode' <nXRadial1> : Initial X point that defines the beginning of the area <nYRadial1> : Initial Y point that defines the beginning of the area <nXRadial2> : Final X point that defines the beginning of the area <nYRadial2> : Final Y point that defines the beginning of the area [<oBrush>] : TBrush object to be used
Return value	ISuccess .T. if the operation is successful

1.5.4.2.5 TCanvas:Create

Generic class' constructor. Create and New are equivalent for this class.

Type	Constructor
Parameters	<oDevice>
Return value	Self reference

1.5.4.2.6 TCanvas:DrawPicture

This method paints a graphic object in the coordinates and dimensions specified in aRect.

Type	Standard
Parameters	<aRect> : Rectangle (Left, Top, Right, Bottom) coordinates. The measure units are specified by 'nMapMode' <oGraphic> : Graphic object to print. It can be a TImage or TEnhMetafile object.
Return value	ISuccess .T. if the operation is successful

1.5.4.2.7 TCanvas:DrawRichEdit

Paints a TRichEdit control in the area specified.

Type	Standard
Parameters	<aRect> : Rectangle (Left, Top, Right, Bottom) coordinates. The measure units are specified by 'nMapMode' <oRichEdit> : TRichEdit object to print [<nFrom>] : First initial Richedit position to print. By default 0.
Return value	<nTo> Really printed text. Returns -1 when the complete text has been printed.

1.5.4.2.8 TCanvas:Ellipse

This method paints an ellipse according to the coordinates aRect with the object oBrush that normally will be a color, but it can be other graphic element. If the parameter oBrush is not passed, then the brush used will be the color indicated in the nClrPane property.

Type	Standard
Parameters	<aRect> : Rectangle (Left, Top, Right, Bottom) coordinates. The measure units are specified by 'nMapMode' [<oBrush nColor >] : Brush object or color to be used to paint.
Return value	ISuccess .T. if the operation is successful

Consult the appendix for the list of available colors

1.5.4.2.9 TCanvas:ExtFloodFill

Fills an area with a specific color

Type	Standard
Parameters	<nX> : Start X coordinate with measure units specified by 'nMapMode' <nY> : Start Y coordinate with measure units specified by 'nMapMode' <nColor>

	Color to be used <nMode> 0: The fill area is bounded by the color specified by the nColor parameter. 1: The fill area is defined by the color that is specified by nColor. Filling continues outward in all directions as long as the color is encountered. This style is useful for filling areas with multicolored boundaries.
Return value	ISuccess .T. if the operation is successful

1.5.4.2.10 TCanvas:FillRect

This method paints an area according to the coordinates aRect with the interior filled based in the object oBrush that normally will be a color, but it can be other graphic element. If the parameter oBrush is not passed, then the brush used will be the color indicated in the nClrPane property.

Type	Standard
Parameters	<aRect> : Rectangle (Left, Top, Right, Bottom) coordinates. The measure units are specified by 'nMapMode' [<oBrush nColor >] : Brush object or color to be used to paint.
Return value	ISuccess .T. if the operation is successful

Consult the appendix for the list of available colors

1.5.4.2.11 TCanvas:LineTo

Paints a line from the current print coordinates to { nX, nY } using the Pen specified in the 'oPen' property

Type	Standard
Parameters	<nX> : X coordinate with measure units specified by 'nMapMode' <nY> : Y coordinate with measure units specified by 'nMapMode'
Return value	ISuccess .T. if the operation is successful

1.5.4.2.12 TCanvas:MoveTo

Moves the pointer from the current print coordinate to { nX, nY }.

Type	Standard
Parameters	<nX>: X coordinate with measure units specified by 'nMapMode' <nY>: Y coordinate with measure units specified by 'nMapMode'
Return value	ISuccess .T. if the operation is successful

1.5.4.2.13 TCanvas:NewFromDC

Class constructor providing a device context.

Type	Constructor
Parameters	<hDC> Device context handle
Return value	Self reference

1.5.4.2.14 TCanvas:Pie

Draws a pie.

Type	Standard
Parameters	<aRect>: Rectangle (Left, Top, Right, Bottom) coordinates. The measure units are specified by 'nMapMode' <nXRadial1>: Initial X point that defines the beginning of the area <nYRadial1>: Initial Y point that defines the beginning of the area <nXRadial2>: Final X point that defines the beginning of the area <nYRadial2>: Final Y point that defines the beginning of the area [<oBrush>]: TBrush object to be used
Return value	ISuccess .T. if the operation is successful

1.5.4.2.15 TCanvas:PixelsToMapMode

Transforms the nX and nY pixel coordinates in current Map mode established by the 'nMapMode' property. If the parameters are passed by reference they are also updated.

Type	Standard
Parameters	<@nX> : X coordinate in pixels <@nY> : Y coordinate in pixels
Return value	<aRect> Array with coordinates in current Map mode (nWidth, nHeight)

1.5.4.2.16 TCanvas:PixelToTwips

Transforms the Array coordinates from pixels to twips. If the aRectOrg parameter is passed by reference, the array values will be updated.

Type	Standard
Parameters	<aRectOrg> : Array coordinates in Pixels (nWidth, nLeft, nRight, nBottom)
Return value	<aRectDes> Array coordinates in Twips (nWidth, nLeft, nRight, nBottom)

1.5.4.2.17 TCanvas:PointToPixels

Transforms the nX and nY coordinates in pixels. If the nX and nY parameters are passed by reference, they will be updated as well.

Type	Standard
Parameters	<nX> : X coordinate with measure units specified by 'nMapMode' <nY> : Y coordinate with measure units specified by 'nMapMode'
Return value	<aRect> Array with coordinates specified in Pixels (nWidth, nLeft, nRight, nBottom)

1.5.4.2.18 TCanvas:Polygon

Draws a polygon.

Type	Standard
Parameters	<aRect>: Array of points (x,y). The measure units are specified by 'nMapMode' [<oBrush nColor>]: TBrush object or color RGB to be used. By default nClrPanecolor or black if not defined.
Return value	ISuccess .T. if the operation is successful

1.5.4.2.19 TCanvas:Rectangle

This method paints a rectangle based in the current 'Pen' object selected, according to the aRect coordinates and filled based in the oBrush object that normally will be a color, but it can be other graphic element. If the oBrush parameter is not specified, the brush color will be specified by the 'nClrPane' property.

Type	Standard
Parameters	<aRect>: Array with coordinates specified by 'nMapMode' (Left, Top, Right, Bottom) [<oBrush>]: oBrush object to be used to paint
Return value	<ISuccess> .T. if the operation is successful

1.5.4.2.20 TCanvas:RectToPixels

Transforms the Array coordinates to pixels. If the aRectOrg parameter is passed by reference, the Array values will be updated as well.

Type	Standard
Parameters	<aRectOrg>: Array with coordinates specified by 'nMapMode' (Left, Top, Right, Bottom)
Return value	<aRectDes>: Array in pixels (nWidth, nLeft, nRight, nBottom)

1.5.4.2.21 TCanvas:RoundRect

This method paints a rectangle with rounded borders based in the current 'Pen' object selected and according to the aRect coordinates specified.

Type	Standard
Parameters	<aRect> : Array with coordinates units specified by 'nMapMode' (Left, Top, Right, Bottom) <nEllipseWidth> : Ellipse width to be used to paint the borders <nEllipseHeight> : Ellipse height to be used to paint the borders
Return value	<ISuccess> .T. if the operation is successful

1.5.4.2.22 TCanvas:SelectFont

Establishes the font to be used in the next pain operations.

Type	Standard
Parameters	<oFont> : New font object
Return value	NIL

Description:

Opposite to the oFont property assignment, the oFont object will not be destroyed if it is assigned using this method. It is not created a cloned TFont object for the oDevice canvas object neither. It is the programmer responsibility to create a oFont object according to the device where it will be used. The oFont object is destroyed when the TCanvas object is destroyed.

Example:

```

WITH OBJECT oFont := TFont():New()
  :Name := "Arial"
  :oDevice := Printer
  :Create()
END WITH

WITH OBJECT oDevice:oCanvas
  :SelectFont(oFont)
  .....
END WITH

```

1.5.4.2.23 TCanvas:SelectPen

Establishes the Pen to be used in the next paint operations.

Type	Standard
Parameters	<oPen> : New pen object
Return value	NIL

Description:

Opposite to the oPen property assignment, the oPen object will not be destroyed if it is assigned using this method. It is not created a cloned TPen object for the oDevice canvas object neither. It is the programmer responsibility to create a oPen object according to the device where it will be used. The oPen object is destroyed when the TCanvas object is destroyed.

Example:

```
WITH OBJECT oPen := TPen():New()
  :nSize := 10
  :oDevice := Printer
  :Create()
END WITH

WITH OBJECT oDevice:oCanvas
  :SelectPen(oPen)
  .....
END WITH
```

1.5.4.2.24 TCanvas:TextExtent

Returns in a two dimensional array with the 'cText' text width and height based in the current font selected, specified in pixels.

Type	Standard
Parameters	<cText> : Text to measure
Return value	<{ nWidth, nHeight }> Two dimension Array with the text width and height specified in pixels

1.5.4.2.25 TCanvas:TextHeight

Returns the 'cText' text height based in the current font selected.

Type	Standard
-------------	-----------------

Parameters	<cText> : Text to measure
Return value	<nHeight> Height specified in pixels

See also: TCanvas:TextExtent, TCanvas:TextWidth

1.5.4.2.26 TCanvas:TextLines

Returns the number of lines of text based in the current font selected.

Type	Standard
Parameters	None
Return value	<nLines> Lines of text

See also: TCanvas:TextExtent, TCanvas:TextWidth, TCanvas:TextHeight

1.5.4.2.27 TCanvas:TextOut

Prints the 'cText' text with the color 'nColor' in the coordinates specified.

Type	Standard
Parameters	<nX> : X coordinate with measure units specified by 'nMapMode' <nY> : Y coordinate with measure units specified by 'nMapMode' <cText> : Text to print [<nWidth>] : Text width with measure units specified by 'nMapMode'. Only needed when alignment diferent than taLEFT [<nColor>] : Text color. By default nClrPane
Return value	<ISuccess> .T. if the operation is successful

Description:

The background color used will be specified by the 'nClrPane' property if and only if the 'ITransparent' property is not set to .T.. The text alignment will be specified by the 'nTextAlignment' property. The nX coordinate will be corrected by the current alignment selected and the nWidth value. In this way, the nX coordinate will always be specified based in the initial text position.

Consult the appendix for the list of available colors

1.5.4.2.28 TCanvas:TextRect

This method prints a rectangle based in the 'aRect' coordinates with the 'cText' text inside using the 'nColor' color.

Type	Standard
Parameters	<aRect> : Array with coordinates units specified by 'nMapMode' (Left, Top, Right, Bottom) <cText> : Text to print <nColor> : Text color [<IWordBreak>] : Breaks words. Lines are automatically broken between words if a word would extend past the edge of the rectangle. By default .T. [<nFlag>] : Additional flag to pass to the DrawText() API function
Return value	<nHeight> Height used by the text

Description:

The background color to be used will be specified by the 'nClrPane' property. The rectangle coordinates will specify the exact zone to print. The method will make the calculations to paint the text inside the rectangle based in the upper left rectangle coordinate and depending to the normal and vertical text alignment

Consult the appendix for the list of available colors

1.5.4.2.29 TCanvas:TextWidth

Returns the 'cText' text width in pixels based in the current font selected.

Type	Standard
Parameters	<cText> : Text to measure
Return value	<nWidth> Text width in pixels

See also: TCanvas:TextExtent, TCanvas:TextHeight

1.5.5 TClipboard

Class to manage the Windows clipboard.

Description:

The TClipboard class allows full access to the Windows clipboard.

Hierarchy Inherits from TComponent
File Name \source\Clipboard.prg

1.5.5.1 TClipboardMethods

■ Constructor ■ Standard ■ Only after Create()

Type	Name
■	Close
■	CountFormats
■	Empty
■	EnumFormats
■	EnumFormatsName
■	GetData
■	GetDropFiles
■	GetFormatName
■	GetOwner
■	GetText
■	GetWindow
■	HasData
■	IsEmpty
■	IsPrivateFormat
■	IsSystemFormat
■	Open
■	RegisterFormat
■	SetData
■	SetDropFiles
■	SetText

1.5.5.1.1 TClipboard:Close

Closes the Windows clipboard.

Type	Standard
Parameters	None
Return value	<ISuccess> .T. if the operation is successful

1.5.5.1.2 TClipboard:CountFormats

Number of data formats managed by the Windows clipboard.

Type	Standard
Parameters	None
Return value	<nFormats>

1.5.5.1.3 TClipboard:Empty

Clears out the Windows clipboard.

Type	Standard
Parameters	None
Return value	NIL

1.5.5.1.4 TClipboard:EnumFormats

Returns an array with a list of format identifiers supported by the Windows clipboard, or {} in case of error.

Type	Standard
Parameters	None
Return value	<aLista>

Predefined clipboard types:

CF_TEXT	1
CF_BITMAP	2
CF_METAFILEPICT	3
CF_SYLK	4
CF_DIF	5
CF_TIFF	6
CF_OEMTEXT	7
CF_DIB	8
CF_PALETTE	9
CF_PENDATA	10
CF_RIFF	11
CF_WAVE	12
CF_UNICODETEXT	13

CF_ENHMETAFILE	14
CF_HDROP	15
CF_LOCALE	16
CF_DIBV5	17
CF_MAX	18
CF_OWNERDISPLAY	128
CF_DSPTEXT	129
CF_DSPBITMAP	130
CF_DSPMETAFILEPICT	131
CF_DSPENHMETAFILE	142

1.5.5.1.5 TClipboard:EnumFormatNames

Returns an array with a list of format names supported by the Windows clipboard, or {} in case of error.

Type	Standard
Parameters	None
Return value	<aLista>

1.5.5.1.6 TClipboard:GetData

Returns the information contained in the Windows clipboard in the format specified by nFormatID.

Type	Standard
Parameters	<nFormatID>: Data format. The possible values for this variable corresponds with the CF_ system constants.
Return value	<uData>: Information recovered from the Windows clipboard.

Predefined clipboard types:

CF_TEXT	1
CF_BITMAP	2
CF_METAFILEPICT	3
CF_SYLK	4
CF_DIF	5
CF_TIFF	6
CF_OEMTEXT	7
CF_DIB	8
CF_PALETTE	9
CF_PENDATA	10
CF_RIFF	11
CF_WAVE	12
CF_UNICODETEXT	13

CF_ENHMETAFILE	14
CF_HDROP	15
CF_LOCALE	16
CF_DIBV5	17
CF_MAX	18
CF_OWNERDISPLAY	128
CF_DSPTEXT	129
CF_DSPBITMAP	130
CF_DSPMETAFILEPICT	131
CF_DSPENHMETAFILE	142

1.5.5.1.7 TClipboard:GetDropFiles

Returns the file list contained in the Windows clipboard after a Drag & Drop operation among applications, or {} if there is not any file.

Type	Standard
Parameters	None
Return value	<aLista>

1.5.5.1.8 TClipboard:GetFormatName

Returns the name associated to an format identifier.

Type	Standard
Parameters	<nFormatID>: Data format. The possible values for this variable corresponds with the CF_ system constants.
Return value	<cName>: Format name

Predefined clipboard types:

CF_TEXT	1
CF_BITMAP	2
CF_METAFILEPICT	3
CF_SYLK	4
CF_DIF	5
CF_TIFF	6
CF_OEMTEXT	7
CF_DIB	8
CF_PALETTE	9
CF_PENDATA	10
CF_RIFF	11
CF_WAVE	12
CF_UNICODETEXT	13
CF_ENHMETAFILE	14

CF_HDROP	15
CF_LOCALE	16
CF_DIBV5	17
CF_MAX	18
CF_OWNERDISPLAY	128
CF_DSPTEXT	129
CF_DSPBITMAP	130
CF_DSPMETAFILEPICT	131
CF_DSPENHMETAFILE	142

1.5.5.1.9 TClipboard:GetOwner

Returns the form manager that is taking action over the Windows clipboard.

Type	Standard
Parameters	None
Return value	<Handle>: Form handle

1.5.5.1.10 TClipboard:GetText

Returns the Windows clipboard content in text format.

Type	Standard
Parameters	None
Return value	<cText>: Clipboard text.

1.5.5.1.11 TClipboard:GetWindow

Returns the form manager that has the Windows clipboard open.

Type	Standard
Parameters	None
Return value	<Handle>: Form handle

1.5.5.1.12 TClipboard:HasData

Returns if there are data in the Windows clipboard in the specified nFormatID format.

Type	Standard
Parameters	<nFormatID> : Data format. The possible values for this variable corresponds with the CF_ system constants..
Return value	<ISuccess> : .T. if the Windows clipboard is not empty

1.5.5.1.13 TClipboard:IsEmpty

Returns the Windows clipboard status.

Type	Standard
Parameters	None
Return value	.T. if the Windows clipboard is empty

1.5.5.1.14 TClipboard:IsPrivateFormat

Returns .T. if the specified format belongs to the applications that manages the Windows clipboard.

Type	Standard
Parameters	<nFormatID> : Format identifier returned by RegisterFormat
Return value	<ITrue> : .T. or .F.

1.5.5.1.15 TClipboard:IsSystemFormat

Returns .T. if the specified format belongs to the operating system.

Type	Standard
Parameters	<nFormatID> : Data format. The possible values for this variable corresponds with the CF_ system constants.
Return value	<ITrue> : .T. or .F.

Predefined clipboard types:

CF_TEXT	1
CF_BITMAP	2
CF_METAFILEPICT	3
CF_SYLK	4
CF_DIF	5
CF_TIFF	6
CF_OEMTEXT	7
CF_DIB	8
CF_PALETTE	9
CF_PENDATA	10
CF_RIFF	11
CF_WAVE	12
CF_UNICODETEXT	13
CF_ENHMETAFILE	14
CF_HDROP	15
CF_LOCALE	16
CF_DIBV5	17
CF_MAX	18
CF_OWNERDISPLAY	128
CF_DSPTEXT	129
CF_DSPBITMAP	130
CF_DSPMETAFILEPICT	131
CF_DSPENHMETAFILE	142

1.5.5.1.16 TClipboard:Open

Opens the Windows clipboard.

Type	Standard
Parameters	None
Return value	<ISucces>: .T. if the operation is successful

1.5.5.1.17 TClipboard:RegisterFormat

Registers a proprietary format that can be managed by the Windows clipboard.

Type	Standard
Parameters	<cFormat>: Format name
Return value	<nFormatID>: New format identifier

1.5.5.1.18 TClipboard:SetData

Assigns the information to the Windows clipboard in the specified format.

Type	Standard
Parameters	<nFormatID>: Data format. The possible values for this variable corresponds with the CF_ system constants. <uData>: Data to insert into the clipboard with the format specified with the nFormatID parameter.
Return value	<ISuccess>: .T. if the operation is successful

Predefined clipboard types:

CF_TEXT	1
CF_BITMAP	2
CF_METAFILEPICT	3
CF_SYLK	4
CF_DIF	5
CF_TIFF	6
CF_OEMTEXT	7
CF_DIB	8
CF_PALETTE	9
CF_PENDATA	10
CF_RIFF	11
CF_WAVE	12
CF_UNICODETEXT	13
CF_ENHMETAFILE	14
CF_HDROP	15
CF_LOCALE	16
CF_DIBV5	17
CF_MAX	18
CF_OWNERDISPLAY	128
CF_DSPTEXT	129
CF_DSPBITMAP	130
CF_DSPMETAFILEPICT	131
CF_DSPENHMETAFILE	142

1.5.5.1.19 TClipboard:SetDropFiles

Assigns a file name list to the Windows clipboard to be used in a Drag & Drop operation among applications.

Type	Standard
Parameters	<aFiles>: Array with file names. This variable can specify a file name or a string text with the format "File +

	(chr(0)) + File + ...
Return value	<ISuccess>: .T. if the operation is successful

1.5.5.1.20 TClipboard:SetText

Assigns the text to the Windows clipboard.

Type	Standard
Parameters	<cText>: Text to be assigned
Return value	<ISuccess>: .T. if the operation is successful

Note: Is necessary to call the Empty() method before using this method.

1.5.6 TCursor

This class allows to manage '**Cursor**' objects type from the Windows API. The cursor type objects are used to indicate the image that will be used by the mouse pointer in an specific situation, for example, when the cursor is inside a window.

There are a lot of predefined cursors in the Windows API. Besides that you can create your own cursors and include them in your application like a resource or external file (.ICO).

Hierarchy	Inherits from TWinObject
File name	\source\Cursor.prg

1.5.6.1 TCursor:Properties

■ read Only ■ Assignable ■ Design assignable ■ Run-time assignable

Scope	Name	Type	Initial value
■	cName	Character	""

1.5.6.1.1 TCursor:cName

Indicates the cursor name.

Scope:	Design assignable
Type:	Character
Initial value	""

Cursor name. It can be a file name, a cursor from a resource file or a standard system cursor. The possible value from the system cursors are:

IDC_APPSTARTIN G	Normal arrow + Hourglass
IDC_ARROW	Normal arrow
IDC_CROSS	Cross
IDC_HAND	Hand
IDC_HELP	Arrow and question mark
IDC_IBEAM	Edition cursor
IDC_NO	Circle crossed off
IDC_SIZEALL	North, south, east and west arrows
IDC_SIZENESW	Northeast and southwest arrows
IDC_SIZENS	North and south arrows
IDC_SIZENWSE	Northwest and southeast arrows
IDC_SIZEWE	West and east arrows
IDC_UPARROW	Up arrow
IDC_WAIT	Hourglass

The resource editor included in Xailer simplifies a lot the cursor selection.

1.5.6.2 TCursor:Methods

■ Constructor ■ Standard ■ Only after Create()

Typ	Name
■	Clone
■	Create
■	Destroy
■	Load
■	LoadFromFile
■	LoadFromResource
■	New

1.5.6.2.1 TCursor:Clone

Creates a new cursor object from the current one.

Type	Standard
Parameters	None
Return value	oBrush

1.5.6.2.2 TCursor:Create

Creates a cursor with the parameter specified. For more information, see the cName property.

Type	Constructor
Parameters	<cName> Cursor name
Return value	Self

1.5.6.2.3 TCursor:Destroy

Destroys the object and releases system resources.

Type	Standard
Parameters	None
Return value	NIL

1.5.6.2.4 TCursor:LoadFromFile

Creates a cursor with the <cFileName> file. For more information, see also the cName property.

Type	Constructor
Parameters	<cFileName> Filename that contains the cursor
Return value	Self

1.5.6.2.5 TCursor:LoadFromResource

Creates a cursor with the <cResourceName> resource. For more information, see also the cName property.

Type	Constructor
Parameters	<cResourceName> Resource name that contains the cursor
Return value	Self

1.5.7 TEnhMetafile

This class allows to manage the '**Enhanced Metafile**' object types from the Windows API.

The Metafile object type are used normally to show vectorial graphics that can even include any text or bitmap image. The metafile file is basically a set of operations to be done on the Windows API. However, the Metafile files does not save inside any information about external resources that are used to paint, like fonts or images. To reproduce a metafile object is mandatory that all the resources that were used to create it, will be available when they are reproduced.

Be sure to differentiate between the file that stores the metafile and the metafile itself. A metafile is a set of actions to be done over GDI object that finally can be saved in disk. if you want, but it is not needed. To create a metafile does not imply the creation of any file on disk. In the same way, if you destroy a metafile it does not imply that you need to delete a file from disk.

Advanced level:

Besides to reproduce any metafile saved in an external file (normally with the WFM or EMF extension), Xailer allows to create your own metafiles. To do that, you need to create an empty metafile with the Create method and later you should make all the painting operations at lower level over the new device using its hNewDC property. This technique is used for the report preview included in Xailer.

Hierarchy Inherits from TWinObject
File Name \source\EnhMetafile.prg

1.5.7.1 TEnhMetafile:Properties

■ read Only ■ Assignable ■ Design assignable ■ Run-time assignable

Scope	Name	Type	Initial value
■	cDescription	Character	""
■	cName	Character	""
■	hNewDC	Numeric	0
■	nHeight	Numeric	0
■	nWidth	Numeric	0

1.5.7.1.1 TEnhMetafile:cDescription

Metafile description.

Scope	read Only
Type	Character
Initial value	""

1.5.7.1.2 TEnhMetafile:cName

Metafile name to load.

Scope:	assignable
Type:	Character
Initial value:	""

If the TEnhMetafile object had previously another Metafile it will be destroyed.

1.5.7.1.3 TEnhMetafile:hNewDC

Device handle created by the Metafile.

Scope	read Only
Type	Numeric
Initial value	0

1.5.7.1.4 TEnhMetafile:nHeight

Indicates the metafile height.

Scope	read Only
Type	Numeric
Initial value	0

1.5.7.1.5 TEnhMetafile:nWidth

Indicates the metafile width.

Scope	read Only
Type	Numeric
Initial value	0

1.5.7.2 TEnhMetafile:Methods

■ Constructor ■ Standard ■ Only after Create()

Typ	Name
■	Close

■	Copy
■	Create
■	Destroy
■	Load
■	Paint
■	Play

1.5.7.2.1 TEnhMetafile:Close

Closes the current metafile. When the metafile is closed, the metafile handle will be destroyed and all its related information can be saved ether in memory or in an external file.

Type	Standard
Parameters	None
Return value	<ISuccess> .T. if the operation is successful

1.5.7.2.2 TEnhMetafile:Copy

Copies the current metafile to a disk file.

Type	Standard
Parameters	<cFilename> File name where the metafile will be copied to.
Return value	<ISuccess> .T. if the operation is successful

1.5.7.2.3 TEnhMetafile:Create

Creates a new metafile.

Type	Standard
Parameters	<hDC> Device handle where the metafile will work. [<cName>] File name where the metafile will be saved. If this parameter is not specified, the metafile will be created in memory only [<nWidth>] Metafile width [<nHeight>] Metafile height [<cDescription>] Metafile description

Return value	<Self> Self reference
---------------------	---------------------------------------

1.5.7.2.4 TEnhMetafile:Destroy

Destroys the current metafile.

Type	Standard
Parameters	None
Return value	NIL

1.5.7.2.5 TEnhMetafile:Load

Loads a Metafile from disk.

Type	Standard
Parameters	<cFilename> Metafile file name
Return value	<Self> Self reference

1.5.7.2.6 TEnhMetafile:Paint

Paints or reproduces the Metafile in the specified device.

Type	Standard
Parameters	<hDC> Device where the metafile will be painted <nLeft> Left coordinate <nTop> Top coordinate <nRight> Right coordinate <nBottom> Bottom coordinate
Return value	<ISuccess> .T. if the operation is successful

The Play method makes the same operation, but the coordinates are passed in an array to the method.

1.5.7.2.7 TEnhMetafile:Play

Paints or reproduces the metafile in the specified device.

Type	Standard
Parameters	<hDC> Device where the metafile will be painted {<nLeft>, <nTop>, <nRight>, <nBottom>} Rectangle paint area
Return value	<ISuccess> .T. if the operation is successful

The Paint methods makes the same operation, but the coordinates are passed like numeric values to the method.

1.5.8 TFont

Class to manipulate Windows FONT Objects.

Description:

The TFont class represents a Windows FONT object.

Hierarchy	Inherits from TWinObject
File name	\source\Font.prg

1.5.8.1 TFont:Properties

■ read Only ■ Assignable ■ Design assignable ■ Run-time assignable

Scope	Name	Type	Initial value
■	cName	Character	"Ms Sans Serif"
■	IBold	Numeric	.F.
■	lItalic	Logic	.F.
■	lStrikeOut	Logic	.F.
■	lUnderline	Logic	.F.
■	nCharSet	Numeric	csDEFAULT
■	nHeight	Numeric	0
■	nOrientation	Numeric	0
■	nSize	Numeric	8
■	nStyle	Numeric	0
■	nWeight	Numeric	400
■	nWidth	Numeric	0
■	oDevice	Object	NIL

1.5.8.1.1 TFont:cName

Indicates the Font name.

Scope	Assignable
Type	Character
Initial value	"Ms Sans Serif"

1.5.8.1.2 TFont:lBold

If it is .T., the text will be in bold.

Scope	Assignable
Type	Logic
Initial value	.F.

1.5.8.1.3 TFont:lItalic

If it is .T., the text will be in italic.

Scope	Assignable
Type	Logic
Initial value	.F.

1.5.8.1.4 TFont:lStrikeOut

If it is .T., the text will be in strike out.

Scope	Assignable
Type	Logic
Initial value	.F.

1.5.8.1.5 TFont:lUnderline

If it is .T., the text will be underlined.

Scope	Assignable
Type	Logic
Initial value	.F.

1.5.8.1.6 TFont:nCharSet

Preselects the font character set.

Scope	Assignable
Type	Numeric
Initial value	csDEFAULT

The possible values, depending of the operating system, are:

- csANSI
- csDEFAULT
- csSYMBOL
- csSHIFTJIS
- csGB2312
- csHANGEUL
- csCHINESEBIG5
- csOEM

1.5.8.1.7 TFont:nHeight

Height font specified in logic units.

Scope	Assignable
Type	Numeric
Initial value	0

The positive values indicate the font size including the upper and lower blank lines that all the fonts have. The negative values indicate the font size from the first to the last printing position.

Internally, the TFont class converts the value specified in nSize to its equivalent **nHeight** (negative) and nWidth =0. This allows to create standard fonts very easy.

1.5.8.1.8 TFont:nOrientation

Degree of inclination between the horizontal base and the X axis in the device, specified in tenth of degrees.

Scope	Assignable
Type	Numeric
Initial value	0

1.5.8.1.9 TFont:nSize

Logic font size.

Scope	Assignable
Type	Numeric
Initial value	8

The logic size coincides with the standard ChooseFont dialog from the Windows API. It is possible to indicate the font size using the nHeight and nWidth as well.

1.5.8.1.10 TFont:nStyle

Mask with the font styles.

Scope	Assignable
Type	Numeric
Initial value	0

Possible values:

- 1 = Italic
- 2 = Underline
- 4 = Strikeout

1.5.8.1.11 TFont:nWeight

Mask with the font weight.

Scope	Assignable
Type	Numeric
Initial value	400

Possible values:

- 100 = Thin
- 300 = Light
- 400 = Normal
- 700 = Bold
- 900 = Heavy

1.5.8.1.12 TFont:nWidth

Font width specified in logic units.

Scope	Assignable
Type	Numeric
Initial value	0

1.5.8.1.13 TFont:oDevice

Device where the font is created.

Scope	Assignable
Type	Object
Initial value	NIL

This data is normally passed as parameter in the Create constructor, but it is only needed when the font will be used in other device than the display (Screen).

You can use the Clone method to duplicate a font in any device.

1.5.8.2 TFont:Methods

■ Constructor ■ Standard

Type	Name
■	Clone
■	Compare
■	Create
■	Destroy
■	GetTextHeight
■	GetTextSize
■	GetTextWidth
■	RestoreFromText
■	SaveToText

1.5.8.2.1 TFont:Clone

Returns a duplicate from the TFont object.

Type	Standard
Parameters	[<oDevice>] Device where the new font will be create. Default

	value: the display (Screen)
Return value	<oFont> TFont object

1.5.8.2.2 TFont:Compare

Compares the current Font with the last Font passed by reference, returning .T. if both are exactly the same.

Type	Standard
Parameters	<oFont> Object to compare
Return value	<IOK> .T. if both are the same

1.5.8.2.3 TFont:Create

Creates a new TFont object.

Type	Constructor
Parameters	[<cName>] Font name: Default value: 'Ms Sans Serif' [<nSize>] Font size. Default value: the nSize property value [<nSyle>] Font style. Default value: nStyle property value [<nWeight>] Mask with the font weight. Default value: nWeight property value [<nOrientation>] Angle between the horizontal font base and the X axis. Default value: nOrientation property value [<oDevice>] Device where the font belongs to [<nCharSet>] Char set font. Default value: nCharSet property value
Return value	<Self> Self reference (Self)

The **nSize** parameter might be an array with the width and height values that correspond with the nWidth and nHeight properties. In that case, the nSize object property will take the logic size calculated after the width and height values are applied.

1.5.8.2.4 TFont:Destroy

Destroys the font and releases the memory used.

Type	Standard
Parameters	None
Return value	NIL

1.5.8.2.5 TFont:GetTextHeight

Calculates the text height on the device.

Type	Standard
Parameters	<cText> Text string <oCtl> Device where the text will be written
Return value	<nHeight> Text height on the device specified in pixels

1.5.8.2.6 TFont:GetTextSize

Calculates the text dimensions on the device.

Type	Standard
Parameters	<cText> Text string <oCtl> Device where the text will be written
Return value	<{nWidth, nHeight}> Text width and height on the device specified in pixels

1.5.8.2.7 TFont:GetTextWidth

Calculates the text width on the device.

Type	Standard
Parameters	<cText> Text string

	<oCtl> Device where the text will be written
Return value	<nWidth> Text width specified in pixels on the device

1.5.8.2.8 TFont:RestoreFromText

Restores the object configuration from a string variable.

Type	Standard
Parameters	<cText> Text information
Return value	<ISuccess> True if success
See also	SaveToText

1.5.8.2.9 TFont:SaveToText

Return a string with the current object configuration.

Type	Standard
Parameters	None
Return value	<cText> Text information
See also	RestoreFromText

1.5.9 THelp

This component allows to manage the help files, in .HLP, .CHM and new Xailer 8.1 .XHLP formats.

Only one instruction is needed to make it work in your application:

```
Application.oHelp := THelp().New( Filename )
```

Description:

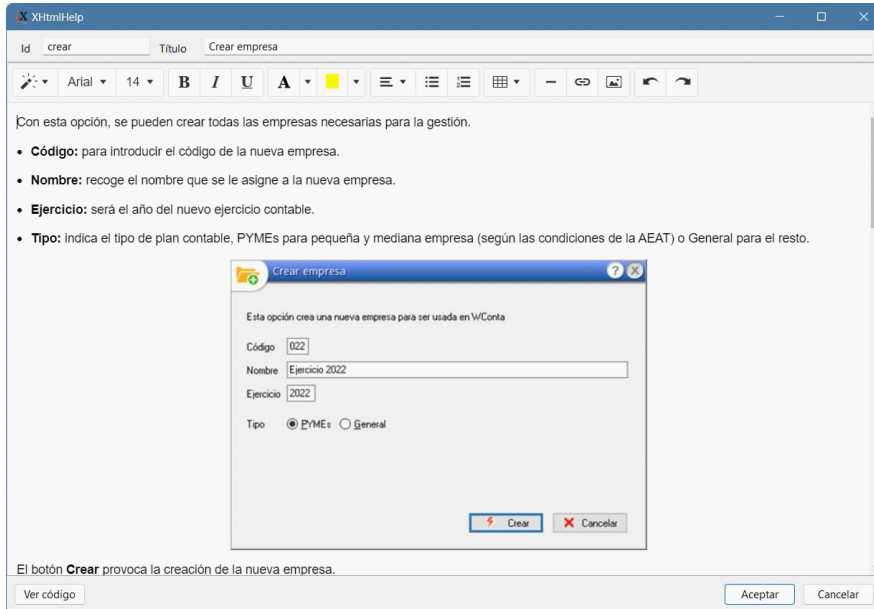
This component allows to manage help files, in .HLP and .CHM formats.

Hierarchy	Inherits from TComponent
File name	\source\Help.prg

1.5.9.1 XHTML Helpeditor

Xailer 8.1 incorporates a complete help generator in HTML format that allows creating the help of any management application in a very simple and powerful way.

As of version 8.1, the TForm control (form) includes a new property named XHtmlHelp. When you click on the edit button in the property inspector you will be shown this HTML editing screen:

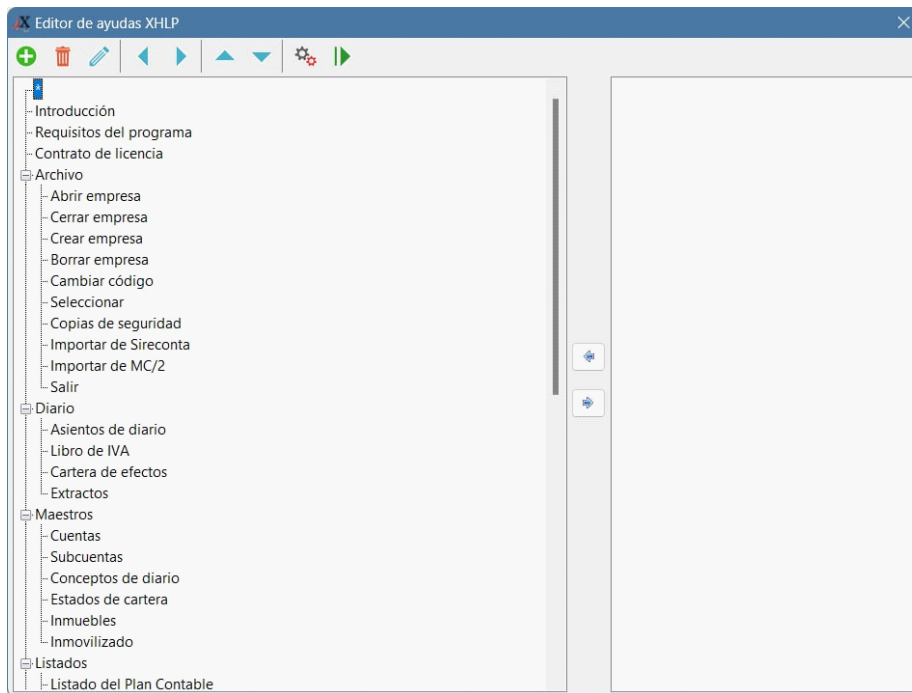


The ID corresponds to the identifier of the help page for that form and the title is the text that will be displayed at the top of the page. It is mandatory to enter some ID, however, the title is not mandatory. If you leave it blank, that page will simply not have a title and the area that the title occupies will disappear.

For each help you create, Xailer will create a file with the same name as the source, but with an .XHH extension in the 'source' directory with the HTML content of the newly created document, including all the images you have embedded.

Our XVC version control system automatically manages the existence of these files and includes them in the repository, without you having to do anything else. If you're using SVN, you'll need to add the files manually via Tortoise SVN or via the command line.

Clicking on the menu option Project->Project Helps will show us the help manager, which looks like the following:



On the right side, all the help assigned to each of the forms are initially shown and it is your responsibility to move them to the left side to build the table of contents. Once moved to the table of contents (left side), you can use the arrow buttons to move up, down or change its nesting in a simple way. It is not necessary that all help pages that you create appear in the table of contents.

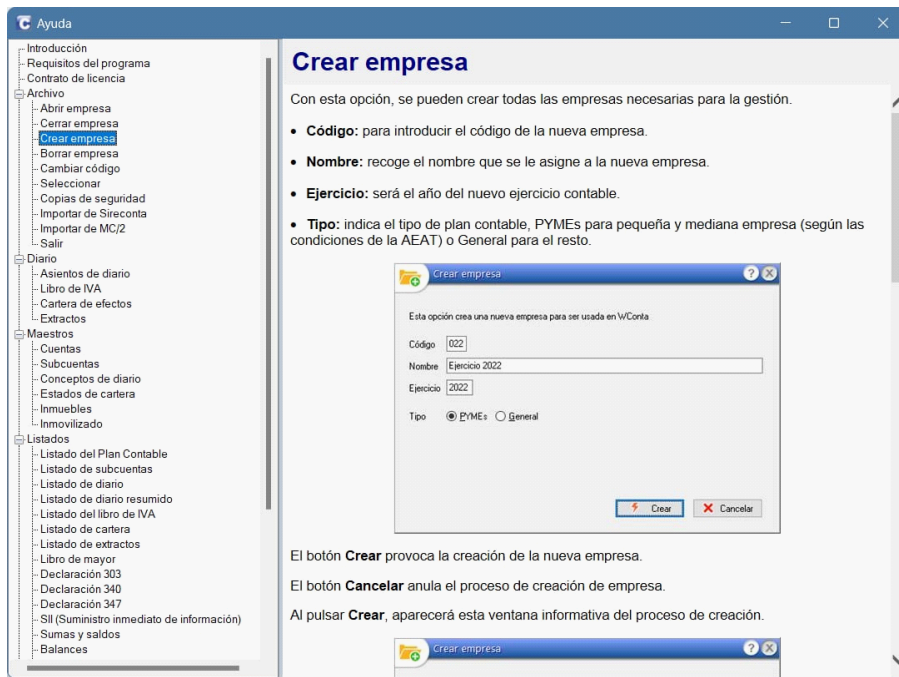
Note that there is a button with some gears that allows you to compile the help in a single file with the extension .XHLP. Logically, said file should be incorporated as another file of the application that you want to distribute or deliver to your client. Xailer creates with all the aids a single compressed file, with a proprietary format, only accessible through our aid manager.

In the editor you can include all the help text. It is possible to incorporate images, tables, text styles and links to web pages or any other help page. For the latter, you must use the word 'help://' as a protocol. For example, if you wanted to put a link to the page 'NewCustomer', you would write the link as: 'help://NewCustomer'. That is all.

For our entire help system to work, you only have to add a line of code at the beginning of our application, which is:

```
oApplication:oHelp := THelp():New( "MiFichero.XHLP" )
```

When the user presses the F1 key on any form, the help for that particular form will be displayed. For example:



1.5.9.2 THelp:Properties

■ read Only ■ Assignable ■ Design assignable ■ Run-time assignable

Scope	Name	Type	Initial value
■	cFileName	Character	""
■	IModal	Logical	.F.
■	IShowTopics	Logical	.T.
■	nType	Numeric	htHLP

1.5.9.2.1 THelp:cFilename

Indicates help filename and path.

Scope	Assignable
Type	Character
Initial value	""

1.5.9.2.2 THelp:IModal

If true the help window will be modal. Only for format XHTML.

Scope	Assignable
Type	Logical
Initial value	.F.

1.5.9.2.3 THelp:IShowTopics

If false the TOC area will be hidden. Only for format XHTML.

Scope	Assignable
Type	Logical
Initial value	.T.

1.5.9.2.4 THelp:nType

Indicates the help file type managed by the object: HLP, CHM or XHTML

Scope	Assignable
Type	Numeric
Initial value	htHLP
Possible values	htHLP, htHTML, htXHTML

1.5.9.3 THelp:Methods

■ Constructor ■ Standard

Typ	Name
■	Close
■	Create
■	New
■	Popup
■	SetFile
■	ShowContext
■	ShowFinder
■	ShowIndex
■	ShowKeyword
■	ShowSearch
■	ShowTOC
■	ShowTopic

1.5.9.3.1 THelp:Close

Closes the help window.

Type	Standard
Parameters	None
Return value	NIL

1.5.9.3.2 THelp:Create

Class constructor.

Type	Constructor
Parameters	<cFile> Help filename
Return value	Self reference (Self)

1.5.9.3.3 THelp:New

Class constructor.

Type	Constructor
Parameters	<cFile> Help filename
Return value	Self reference (Self)

1.5.9.3.4 THelp:PopUp

Shows the help text associated to a control in a pop-up window.

Type	Standard
Parameters	<oSender> Control that the help will refer to.
Return value	NIL

1.5.9.3.5 THelp:SetFile

Establishes the file that manages the object. It has the same effect to assign the cFileName property.

Type	Standard
Parameters	<cFileName> Filename and path for the help file
Return value	Self reference (Self)

1.5.9.3.6 THelp:ShowContext

Shows the help text associated to the identifier.

Type	Standard
Parameters	<nHelpID> Help identifier
Return value	NIL

1.5.9.3.7 THelp:ShowFinder

Shows the dialog "Help topics".

Type	Standard
Parameters	None
Return value	NIL

1.5.9.3.8 THelp:ShowIndex

Shows the help file index.

Type	Standard
Parameters	None
Return value	NIL

1.5.9.3.9 THelp:ShowKeyword

Shows the specified help page or the index if the page does not exist.

Type	Standard
Parameters	<cKeyword> Help page <IIndexOnFail> Shows the index if the page does not exist.
Return value	NIL

1.5.9.3.10 THelp:ShowSearch

Shows the search page.

Type	Standard
Parameters	None
Return value	NIL

1.5.9.3.11 THelp:ShowTOC

Shows the table of contents from the help file.

Type	Standard
Parameters	None
Return value	NIL

1.5.9.3.12 THelp:ShowTopic

Shows the specified topic from the help file.

Type	Standard
Parameters	<cURL> Specific topic from the help file
Return value	NIL

1.5.10 TIcon

Class to manipulate ICON Windows objects.

Description:

The TIcon class represents an ICON Windows object.

Hierarchy Inherits from TWinObject
File name \source\Icon.prg

1.5.10.1 TIcon:Properties

■ read Only ■ Assignable ■ Design assignable ■ Run-time assignable

Scope	Name	Type	Initial value
■	cName	Character	" "
■	nHeight	Numeric	32
■	nWidth	Numeric	32

1.5.10.1.1 TIcon:cName

Icon's name. it can be a filename, an icon embedded in a resource file or an standard system icon.

Scope	Assignable
Type	Character
Initial value	" "

1.5.10.1.2 TIcon:nHeight

Icon's height, specified in pixels.

Scope	Design assignable
Type	Numeric
Initial value	32

1.5.10.1.3 TIcon:nWidth

Icon's width, specified in pixels.

Scope	Design assignable
--------------	-------------------

Type	Numeric
Initial value	32

1.5.10.2 TIcon:Methods

■ Constructor ■ Standard

Type	Name
■	Clone
■	Create
■	Destroy
■	Load
■	LoadFromFile
■	LoadFromResource
■	Paint

1.5.10.2.1 TIcon:Clone

Returns an icon object duplicated.

Type	Standard
Parameters	None
Return value	<olcon> TIcon object

1.5.10.2.2 TIcon:Create

Class constructor.

Type	Constructor
Parameters	<cName> Resource file name. It can be a icon embedded in a resource file or an standard system icon [<nWidth> Permits to select the correct image indicanting its exact width [<nHeight> Permits to select the correct image indicanting its exact height
Return value	Self reference (self)

1.5.10.2.3 TIcon:Destroy

Releases the resources used by the object.

Type	Standard
Parameters	None
Return value	NIL

1.5.10.2.4 TIcon:Load

Loads the specified icon in memory.

Type	Standard
Parameters	<cName> Resource file name. It can be a icon embedded in a resource file or an standard system icon.
Return value	Self reference (self)

1.5.10.2.5 TIcon:LoadFromFile

Loads the specified icon from a file. Accepts the following file extensions: ICO, DLL, EXE.

Type	Standard
Parameters	<cName> Filename that contains the icon [<nIcon>]: Icon number. By default first icon in file
Return value	Self reference (self)

1.5.10.2.6 TIcon:LoadFromResource

Loads the specified icon from a resources file.

Type	Standard
Parameters	<cName> Resource file name. It can be a icon embedded in a resource file or an standard system icon.
Return value	Self reference (self)

1.5.10.2.7 TIcon:Paint

Shows the icon in the device indicated by hDC in the specified coordinates.

Type	Standard
Parameters	<hDC> Device handle <nLeft> Upper column <nTop> Lower row <nRight> Lower column <nBottom> Lower row
Return value	NIL

1.5.11 TIni

Class to manipulate Windows INI files.

Hierarchy	Inherits from TComponent
File name	\source\Ini.prg

1.5.11.1 TIni:Properties

read Only
 Assignable
 Design assignable
 Run-time assignable

Scope	Name	Type	Initial value
<input type="checkbox"/>	cFileName	Character	""
<input type="checkbox"/>	nBuffer	Numeric	32768

1.5.11.1.1 TIni:cFileName

Indicates the configuration file name.

Importante note: If no path is given the windows directory is assumed.

Scope	Standard
Type	Character
Initial value	""

1.5.11.1.2 TIni:nBuffer

Denotes the buffer size in bytes. Default value: 8 Kb.

Scope	Standard
Type	Numeric
Initial value	32768

1.5.11.2 TIni:Methods

■ Constructor ■ Standard ■ Only after Create()

Typ	Name
■	AddSection
■	Commit
■	Create
■	DeleteEntry
■	DeleteSection
■	DeleteSections
■	EnumSectionEntries
■	EnumSectionValues
■	GetEntry
■	GetArray
■	GetDate
■	GetLogical
■	GetNumeric
■	GetSection
■	GetSectionEx
■	GetSections
■	GetSectionsEx
■	New
■	RenameSection
■	SetEntry
■	SetSection
■	SortSection
■	SystemChanged

1.5.11.2.1 TIni:AddSection

Creates a new section in the INI file.

Type	Standard
Parameters	<cSection>

	Section name
Return value	NIL

1.5.11.2.2 TIni:Commit

Forces to write the file on disk. This operation depends of the operating system and version then it does not guarantee that the info will be written immediately.

Type	Standard
Parameters	None
Return value	NIL

1.5.11.2.3 TIni:Create

Class constructor.

Type	Standard
Parameters	<cFilename> INI File
Return value	Self Reference

1.5.11.2.4 TIni>DeleteEntry

Deletes an entry from the specified section.

Type	Standard
Parameters	<cSection> Section name <cEntry> Entry name
Return value	NIL

1.5.11.2.5 TIni>DeleteSection

Deletes the specified section.

Type	Standard
Parameters	<cSection> Section name
Return value	NIL

1.5.11.2.6 TIni>DeleteSections

Eliminates all the entries and sections from the INI file.

Type	Standard
Parameters	None
Return value	NIL

1.5.11.2.7 TIni:EnumSectionEntries

Returns a list with all the entry names that belong to an specific section.

Type	Standard
Parameters	<cSection> Section name
Return value	<aEntries> Array with all the entry names

1.5.11.2.8 TIni:EnumSectionValues

Returns a list with all the entry values that belong to an specific section.

Type	Standard
Parameters	<cSection> Section name
Return value	<aEntries> Array with all the entry values

1.5.11.2.9 TIni:GetEntry

Returns the value assigned to an specific entry in an specific section.

Type	Standard
Parameters	<cSection> Section name <cEntry> Entry name [<uDefault>] Default value in case that the entry section does not exist.
Return value	<uValue> Value taken from the file or ' uDefault '

1.5.11.2.10 TIni:GetArray

Returns an array with the value assigned to an specific entry in an specific section.

Type	Standard
Parameters	<cSection> Section name <cEntry> Entry name [<cSeparator>] Character use as input separator. Default (;)
Return value	<aValue> Entry value or {} in case that the value does not exist.

1.5.11.2.11 TIni:GetDate

Returns the value assigned to an specific entry in an specific section.

Type	Standard
Parameters	<cSection> Section name <cEntry> Entry name
Return value	<dValue> Entry value or current date if the value does not exist.

1.5.11.2.12 TIni:GetLogical

Returns the value assigned to an specific entry in an specific section.

Type	Standard
Parameters	<cSection> Section name <cEntry> Entry name
Return value	<IValue> Entry value or .F. it the value does not exist.

1.5.11.2.13 TIni:GetNumeric

Returns the value assigned to an specific entry in an specific section.

Type	Standard
Parameters	<cSection> Section name <cEntry> Entry name
Return value	<nValue> Entry value or 0 it the value does not exist.

1.5.11.2.14 TIni:GetSection

Returns an array with all the elements from an specific section.

Type	Standard
Parameters	<cSection> Section name
Return value	<aSection> List with the section elements

1.5.11.2.15 TIni:GetSectionEx

Returns a string with the content of an specific section.

Type	Standard
Parameters	<cSection>

	Section name
Return value	<cString> String with the section content

1.5.11.2.16 TIni:GetSections

Returns an array with all the section names.

Type	Standard
Parameters	None
Return value	<aSections> Section name array

1.5.11.2.17 TIni:GetSectionsEx

Returns a string with the section names.

Type	Standard
Parameters	None
Return value	<cString> String with section names

1.5.11.2.18 TIni:RenameSection

Renames a section.

Type	Standard
Parameters	<cSection> Section name <cNewSection> New Section name
Return value	NIL

1.5.11.2.19 Tlni:SetEntry

Saves the value in the **cEntry** entry that belongs to the **cSection** section.

Type	Standard
Parameters	<cSection> Section name <cEntry> Entry name <uValue> Value that will be saved in the file
Return value	NIL

1.5.11.2.20 Tlni:SetSection

Establishes the section content from a two dimension array that contains all the section elements.

Type	Standard
Parameters	<cSection> Section name <aValues> Two dimension array with the section elements
Return value	NIL

1.5.11.2.21 Tlni:SortSection

Sorts the section entries.

Type	Standard
Parameters	<cSection> Section name
Return value	NIL

1.5.11.2.22 Tlni:SystemChanged

Informs to the system and the applications that a section in a system file has been changed.

Type	Standard
Parameters	<cSection>

	Section name
Return value	NIL

1.5.12 TMutex

Class to encapsulates a Mutex semaphore.

Hierarchy	Inherits from TWinObject
File name	\source\Mutex.prg

1.5.12.1 TMutex:Properties

read Only
 Assignable
 Design assignable
 Run-time assignable

Scope	Name	Type	Initial value
<input type="checkbox"/>	cName	Character	" "
<input type="checkbox"/>	nTimeOut	Numeric	0

1.5.12.1.1 TMutex:cName

Semaphore unique name. The default value is Application:cTitle.

Scope	Assignable
Type	Character
Initial value	" "

1.5.12.1.2 TMutex:nTimeOut

Waiting time in milliseconds.

Scope	Assignable
Type	Numeric
Initial value	0

1.5.12.2 TMutex:Methods

Constructor
 Standard

Typ e	Name
-------	------

- Create
- Destroy

1.5.12.2.1 TMutex:Create

Creates a semaphore with the specified name and waits, if needed, the needed time to get a valid semaphore. Depending of the operation result, the OnCreate or OnFail event is triggered.

Type	Constructor
Parameters	<cName> Semaphore unique name <nTimeOut> Time in milliseconds
Return value	Self reference (Self)

1.5.12.2.2 TMutex:Destroy

Destroys the semaphore and releases the associated resources.

Type	Standard
Parameters	None
Return value	NIL

1.5.12.2.3 TMutex:Events

Name	OnCreate
	OnFail
	There are not inherited events in TWinObject
	There are not inherited events in TComponent

1.5.12.2.4 TMutex:OnCreate

Event that is triggered if the semaphore was created successfully.

Parameters	<oSender> Reference to the object that triggers the event
Return value	NIL

1.5.12.2.5 TMutex:OnFail

Event that is triggered if the semaphore was not created correctly.

Parameters	<oSender> Reference to the object that triggers the event
Return value	NIL

1.5.13 TPen

Class to manipulate Windows PEN objects..

Description:

The TPen Class represents a PEN Windows object. These type of objects are used to draw lines, rectangles or curves.

Hierarchy	Inherits from TWinObject
File name	\source\Pen.prg

1.5.13.1 TPen:Properties

■ read Only ■ Assignable ■ Design assignable ■ Run-time assignable

Scope	Name	Type	Initial value
■	nColor	Numeric	CLR_BLACK
■	nStyle	Numeric	PS_SOLID
■	nWidth	Numeric	1
■	oDevice	Object	NIL

1.5.13.1.1 TPen:nStyle

Pen's style.

Scope	Assignable
Type	Numeric
Initial value	PS_SOLID
Possible values	PS_SOLID: solid PS_DASH: Lines PS_DOT: dots PS_DASHDOT: line - dot PS_DASHDOTDOT: line - dot - dot

The possible values definitions are included in the file **WinGdi.api**.

Note: The styles different than PS_SOLID only accept widths of 1 or less. For greater widths it will be shown as a PS_SOLID style.

1.5.13.1.2 TPen:nColor

Pen's color.

Scope:	Assignable
Type:	Numeric
Initial value:	CLR_BLACK

Consult the appendix for the list of available colors

1.5.13.1.3 TPen:nWidth

Pen's width specified in pixels.

Scope	Assignable
Type	Numeric
Initial value	1

1.5.13.1.4 TPen:oDevice

Device object used to create the pen.

Scope	Read only
Type	Object
Initial value	NIL

1.5.13.2 TPen:Methods

■ Constructor ■ Standard

Typ	Name
■	Clone
■	Create New
■	Destroy

1.5.13.2.1 TPen:Clone

Clones the actual Pen to a new Pen object referred to a specific Device.

Type	Standard
Parameters	[<oDevice>] Device object to use the pen. By default the :: oDevice
Return value	<oPen> New Pen object

1.5.13.2.2 TPen:Create

Class constructor.

Type	Constructor
Parameters	[<nStyle>] Pen style. Default value: the nStyle property value. [<nWidth>] Pen width. Default value: the nWidth property value [<nColor>] Pen color. Default value: the nColor property value [<oDevice>] Device object to use the pen. By default the Screen
Return value	Self reference (Self)

1.5.13.2.3 TPen:Destroy

Releases the resources used by the object.

Type	Standard
Parameters	None
Return value	NIL

1.5.14 TPicture

Class to manipulate images. Supports the following formats:

- BMP
- ICO
- CUR

- WMF
- EMF
- JPG
- GIF
- PNG
- TIF

Description:

The TPicture class allows to manage several image formats.

Hierarchy Inherits from TWinObject
File name \source\Picture.prg

1.5.14.1 TPicture:Properties

■ read Only ■ Assignable ■ Design assignable ■ Run-time assignable

Scope	Name	Type	Initial value
■	aTypes	Array	{ ... }
■	cName	Character	""
■	hPalette	Numeric	0
■	IScalable	Logic	.F.
■	ITransparent	Logic	.F.
■	nHeight	Numeric	0
■	nImageCount	Numeric	0
■	nImageIndex	Numeric	0
■	nImageType	Numeric	-1
■	nLogicalHeight	Numeric	0
■	nLogicalWidth	Numeric	0
■	nTimeFrame	Numeric	0
■	nType	Numeric	0
■	nWidth	Numeric	0
■	nZoomH	Numeric	1
■	nZoomV	Numeric	1

1.5.14.1.1 TPicture:aTypes

Allows to establish the image types to be searched when is loaded a image from a resource file.

Scope	Assignable
Type	Array
Initial value	{"JPG", "JPEG", "GIF", "PNG", "TIF", "TIFF"}

1.5.14.1.2 TPicture:cName

File name or resource name that contains an image.

Scope	Assignable
Type	Character
Initial value	""

1.5.14.1.3 TPicture:hPalette

Image palette colors.

Scope	read Only
Type	Numeric
Initial value	0

1.5.14.1.4 TPicture:IScalable

The image can be shown with a different size than the original.

Scope	read Only
Type	Logic
Initial value	.F.

1.5.14.1.5 TPicture:ITransparent

The image contains transparent areas. The icons and metafiles are image types that support this attribute.

Scope	read Only
Type	Logic
Initial value	.F.

1.5.14.1.6 TPicture:nHeight

Image height specified in pixels.

Scope	read Only
Type	Numeric

Initial value 0

1.5.14.1.7 TPicture:nImageCount

Number of images on the control. Only available in TIFF and GIF formats.

Scope	read Only
Type	Numeric
Initial value	0

1.5.14.1.8 TPicture:nImageIndex

Active image on the control. Only available in TIFF and GIF formats.

Scope	Assignable
Type	Numeric
Initial value	0

1.5.14.1.9 TPicture:nImageType

Image type.

Scope	read Only
Type	Numeric
Initial value	0
Possible values	0 BMP 1 JPEG 2 GIF 3 PNG 4 TIFF

1.5.14.1.10 TPicture:nLogicalHeight

Image logical height

Scope	read Only
Type	Numeric
Initial value	0

1.5.14.1.11 TPicture:nLogicalWidth

Image logical width.

Scope	read Only
Type	Numeric
Initial value	0

1.5.14.1.12 TPicture:nTimeFrame

Frames between images on GIF animated images.

Scope	read Only
Type	Numeric
Initial value	0

1.5.14.1.13 TPicture:nType

If the image is standard, it contains the file type: IMG_BITMAP, IMG_ICON, IMG_METAFILE and IMG_ENHMETAFIELD.

Scope	read Only
Type	Numeric
Initial value	0

1.5.14.1.14 TPicture:nWidth

Indicates the image width specified in pixels.

Scope	read Only
Type	Numeric
Initial value	0

1.5.14.1.15 TPicture:nZoomH

Horizontal zoom.

Scope	Assignable
Type	Numeric

Initial value	1
----------------------	---

1.5.14.1.16 TPicture:nZoomV

Vertical zoom.

Scope	Assignable
Type	Numeric
Initial value	1

1.5.14.2 TPicture:Methods

■ Constructor ■ Standard

Type	Nombre
■	Clone
■	Create
■	CreateIndirect
■	Destroy
■	GetCaptureDateTime
■	GetPictureHandle
■	GetThumbnailSize
■	Load
■	LoadFromFile
■	LoadFromResource
■	LoadFromStream
■	IsBitmap
■	IsEnhMetafile
■	IsIcon
■	IsMetafile
■	IsPicture
■	MakeThumbNail
■	Paint
■	Save
■	SaveToStream
■	SetTypes

1.5.14.2.1 TPicture:Clone

Returns a duplicate image object.

Type	Standard
-------------	----------

Parameters	None
Return value	<oPicture> TPicture object

1.5.14.2.2 TPicture:Create

Loads the indicated image and builds a TPicture object.

Type	Constructor
Parameters	<cName> File name or resource that contains the image
Return value	Self reference (Self)

1.5.14.2.3 TPicture:CreateIndirect

Creates an Image from a pre-existing Handle.

Type	Constructor
Parameters	<nImageType> Image type. One of the following: <ol style="list-style-type: none"> 1. Bitmap 2. Metafile 3. Icon 4. Enhanced metafile <nHandle> Handle
Return value	Self reference (Self)

1.5.14.2.4 TPicture:Destroy

Destroys the resources managed by the image.

Type	Standard
Parameters	None
Return value	NIL

1.5.14.2.5 TPicture:GetCaptureDateTime

Gets the image capture date time if available. Requires the use of GDI+.

Type	Standard
Parameters	None
Return value	<cDateTime>

1.5.14.2.6 TPicture:GetPictureHandle

Gets the image handle assigned by the operating system

Type	Standard
Parameters	None
Return value	<Handle> Numeric value

1.5.14.2.7 TPicture:GetThumbnailSize

Calculates the ideal image dimensions to show it like a thumbnail with the Paint method.

Type	Standard
Parameters	<nWidth> Thumbnail width <nHeight> Thumbnail height
Return value	<aSize> Ideal image width and height

1.5.14.2.8 TPicture:Load

Loads the indicated image.

Type	Standard
Parameters	<cName> File name or resource that contains the image [<luseGdiPlus>] If true forces the use of GDI+
Return value	Self reference (Self)

This method tries to load the image with the following order:

1. Using the native API
2. Using Freeimage.dll if present
3. Using GDI+

1.5.14.2.9 TPicture:LoadFromFile

Loads the specified image from a file.

Type	Standard
Parameters	<cName> File name or resource that contains the image [<nImage> Imager number to load. Only available for TIFF and GIF images. By default 1 [<luseGdiPlus> If true forces the use of GDI+
Return value	Self reference (Self)

This method tries to load the image with the following order:

1. Using the native API
2. Using Freeimage.dll if present
3. Using GDI+

1.5.14.2.10 TPicture:LoadFromResource

Load the specified image from a resource file.

Type	Standard
Parameters	<cName> Resource name that contains the image <cType> Image type description saved in the resource file [<luseGdiPlus> If true forces the use of GDI+
Return value	Self reference (Self)

This method tries to load the image with the following order:

1. Using the native API
2. Using Freeimage.dll if present
3. Using GDI+

1.5.14.2.11 TPicture:LoadFromStream

Load the specified image from a data stream.

Type	Standard
Parameters	<cStream> Data stream [<luseGdiPlus>] If true forces the use of GDI+
Return value	Self reference (Self)

This method tries to load the image with the following order:

1. Using the native API
2. Using Freeimage.dll if present
3. Using GDI+

1.5.14.2.12 TPicture:IsBitmap

Indicates if the image contains a bitmap.

Type	Standard
Parameters	None
Return value	<IBitmap> .T. if the object has a bitmap

1.5.14.2.13 TPicture:IsEnhMetafile

Indicates if the object contains an enhanced metafile.

Type	Standard
Parameters	None
Return value	<IEnhMetafile> .T. if the object contains an enhanced metafile

1.5.14.2.14 TPicture:IsIcon

Indicates if the object contains an icon.

Type	Standard
Parameters	None
Return	<IIcon>

value	.T. if the object contains an icon
--------------	------------------------------------

1.5.14.2.15 TPicture:IsMetafile

Indicates if the object contains a metafile.

Type	Standard
Parameters	None
Return value	<IMetafile> .T. if the object has a metafile

1.5.14.2.16 TPicture:IsPicture

Indicates if the object contains a valid image.

Type	Standard
Parameters	None
Return value	<IPicture> .T. if the object has a valid image

1.5.14.2.17 TPicture:MakeThumbNail

Creates a thumbnail of the image.

Type	Standard
Parameters	[<nWidth>] Image width. By default 128 [<nHeight>] Image Height. By default 128 [<nImgFormat>] Image format. By default FIF_JPEG. Possible values: FIF_BMP, FIF_ICO, FIF_JPEG, FIF_JNG, FIF_KOALA, FIF_LBM, FIF_IFF, FIF_LBM, FIF_MNG, FIF_PBM, FIF_PBMRAW, FIF_PCD, FIF_PCX, FIF_PGM, FIF_PGMRAW, FIF_PNG, FIF_PPM, FIF_PPMRAW, FIF_RAS, FIF_TARGA, FIF_TIFF, FIF_WBMP, FIF_PSD, FIF_CUT, FIF_XBM, FIF_XPM, FIF_DDS, FIF_GIF, FIF_HDR, FIF_FAXG3, FIF_SGI, FIF_EXR, FIF_J2K, FIF_JP2 [<nFlags>] Additional creation parameters. Is an OR value of the possible values included at the end of this page. By default

JPEG_QUALITYNORMAL.

[<nBgColor>]

If used, then the result image will have the required dimensions. If the original image fits inside, it will be shown centered and the surrounding space will be filled with this color. It can be used also to reduce the size of the image. If this parameter is not passed then only the reduce feature is available

<nBgColor> must be different than -1, the parameter <ITransparent> should be false and the image should be loaded with GDI+. FreeImage.dll is not supported

Return value <cStream>
Stream of data

Important note:

Is necessary to use the free library **FreeImage.DLL** in order to use this method. You should also include the header file **freeimage.h** in all the modules that use this method.

For further information consult the **FreeImage** documentation.

```
// FreeImage Load / Save flag constants -----
#define BMP_DEFAULT      0
#define BMP_SAVE_RLE    1
#define CUT_DEFAULT      0
#define DDS_DEFAULT     0
#define EXR_DEFAULT     0          // save data as half with piz-based wavelet
compression
#define EXR_FLOAT        0x0001   // save data as float instead of as half (not
recommended)
#define EXR_NONE         0x0002   // save with no compression
#define EXR_ZIP          0x0004   // save with zlib compression, in blocks of 16 scan
lines
#define EXR_PIZ          0x0008   // save with piz-based wavelet compression
#define EXR_PXR24        0x0010   // save with lossy 24-bit float compression
#define EXR_B44          0x0020   // save with lossy 44% float compression - goes to 22%
when combined with EXR_LC
#define EXR_LC           0x0040   // save images with one luminance and two chroma
channels,
#define FAXG3_DEFAULT   0
#define GIF_DEFAULT     0
#define GIF_LOAD256    1          // Load the image as a 256 color image with unused palette
entries
#define GIF_PLAYBACK   2          // 'Play' the GIF to generate each frame (as 32bpp)
#define HDR_DEFAULT    0
#define ICO_DEFAULT     0
#define ICO_MAKEALPHA  1          // convert to 32bpp and create an alpha channel from the
AND-mask when loading
#define IFF_DEFAULT    0
#define J2K_DEFAULT    0          // save with a 16:1 rate
#define JP2_DEFAULT    0          // save with a 16:1 rate
#define JPEG_DEFAULT   0          // loading (see JPEG_FAST); saving (see
JPEG_QUALITYGOOD)
#define JPEG_FAST      0x0001   // load the file as fast as possible, sacrificing some
quality
#define JPEG_ACCURATE  0x0002   // load the file with the best quality, sacrificing
some speed
#define JPEG_CMYK      0x0004   // load separated CMYK "as is" (use | to combine
with other load flags)
#define JPEG_QUALITYSUPERB 0x80 // save with superb quality (100:1)
#define JPEG_QUALITYGOOD 0x0100 // save with good quality (75:1)
```

```

#define JPEG_QUALITYNORMAL      0x0200    // save with normal quality (50:1)
#define JPEG_QUALITYAVERAGE    0x0400    // save with average quality (25:1)
#define JPEG_QUALITYBAD        0x0800    // save with bad quality (10:1)
#define JPEG_PROGRESSIVE        0x2000    // save as a progressive-JPEG (use | to
combine with other save flags)
#define KOALA_DEFAULT           0
#define LBM_DEFAULT             0
#define MNG_DEFAULT             0
#define PCD_DEFAULT             0
#define PCD_BASE                1          // load the bitmap sized 768 x 512
#define PCD_BASEDIV4            2          // load the bitmap sized 384 x 256
#define PCD_BASEDIV16           3          // load the bitmap sized 192 x 128
#define PCX_DEFAULT             0
#define PNG_DEFAULT             0
#define PNG_IGNOREGAMMA         1          // avoid gamma correction
#define PNM_DEFAULT             0
#define PNM_SAVE_RAW            0          // If set the writer saves in RAW format (i.e. P4, P5
or P6)
#define PNM_SAVE_ASCII          1          // If set the writer saves in ASCII format (i.e. P1, P2
or P3)
#define PSD_DEFAULT            0
#define RAS_DEFAULT            0
#define SGI_DEFAULT            0
#define TARGA_DEFAULT          0
#define TARGA_LOAD_RGB888      1          // If set the loader converts RGB555 and ARGB8888 ->
RGB888.
#define TIFF_DEFAULT           0
#define TIFF_CMYK               0x0001    // reads/stores tags for separated CMYK (use |
to combine with compression flags)
#define TIFF_PACKBITS          0x0100    // save using PACKBITS compression
#define TIFF_DEFLATE           0x0200    // save using DEFLATE compression (a.k.a. ZLIB
compression)
#define TIFF_ADOBE_DEFLATE      0x0400    // save using ADOBE DEFLATE compression
#define TIFF_NONE               0x0800    // save without any compression
#define TIFF_CCITTFAX3         0x1000    // save using CCITT Group 3 fax encoding
#define TIFF_CCITTFAX4         0x2000    // save using CCITT Group 4 fax encoding
#define TIFF_LZW                0x4000    // save using LZW compression
#define TIFF_JPEG               0x8000    // save using JPEG compression
#define WBMP_DEFAULT           0
#define XBM_DEFAULT            0
#define XPM_DEFAULT            0

```

1.5.14.2.18 TPicture:Paint

Sows the image in the device handle indicated by hDC in the specified coordinates.

Type	Standard
Parameters	<hDC> Device handle <nLeft> Upper column <nTop> Upper row <nRight> Lower column <nBottom> Lower row
Return value	NIL

1.5.14.2.19 TPicture:Save

Saves the image in disk. The conversion among different image types is not supported. The image will keep the same format as read.

Type	Standard
Parameters	<cName> Image name [<nType>] Image type. By default the type set by the property nImageType [<nQuality>] Image quality. Possible values from 0 to 100. By default 50. Only applicable to JPEG styles
Return value	<lSuccess> .T. if the file is saved successfully

This method tries to save the image with the following order:

1. Using the native API if it was used to load it and its saved using the same type
2. Using GDI+

1.5.14.2.20 TPicture:SaveToStream

Saves the image to a data stream. The conversion among different image types is not supported.

Type	Standard
Parameters	[<nType>] Image type. By default the type set by the property nImageType [<nQuality>] Image quality. Possible values from 0 to 100. By default 50. Only applicable to JPEG styles
Return value	<cStream> Data stream

This method tries to save the image with the following order:

1. Using the native API if it was used to load it and its saved using the same type
2. Using GDI+

1.5.14.2.21 TPicture:SetTypes

Establish the image sort order in the resource file.

Type	Standard
Parameters	<aTypes> Array with the image types
Return value	<aOldTypes> Array with the previous image types

1.5.15 TSysVer

This class allows to get the operating system's version.

Description:

This class allows to get the operating system's version.

Hierarchy Inherits from TComponent
File name \source\SysVer.prg

1.5.15.1 TSysVer:Properties

■ read Only ■ Assignable ■ Design assignable ■ Run-time assignable

Scope	Name	Type	Initial value
■	cCSDVersion	Character	""
■	cSysName	Character	"Microsoft Windows"
■	cSysVer	Character	""
■	IIsWin10	Logic	.F.
■	IIsWin7	Logic	.F.
■	IIsWin98	Logic	.F.
■	IIsWinMe	Logic	.F.
■	IIsWin2000	Logic	.F.
■	IIsWinXP	Logic	.F.
■	IIsWinXP64	Logic	.F.
■	IIsWinServer2003	Logic	.F.
■	IIsWinServer2016	Logic	.F.
■	IIsWinServer2019	Logic	.F.
■	IIsWinVista	Logic	.F.
■	IIsWinHomeServer	Logic	.F.
■	IIsWow64	Logic	.F.
■	nBuildNumber	Numeric	0
■	nMajorVersion	Numeric	0

■	nMemoryRAM	Numeric	0
■	nMinorVersion	Numeric	0
■	nPlatformID	Numeric	0
■	nProcessorArch	Numeric	0
■	nProductType	Numeric	0
■	nReleaseID	Numeric	0
■	nServicePackMajor	Numeric	0
■	nServicePackMinor	Numeric	0
■	nSuiteMask	Numeric	0

1.5.15.1.1 TSysVer:cSDVersion

Text string that indicates the last "Service Pack" installed in the system, for example "Service Pack 1". In the case that no service pack is installed, the string will be empty.

Scope	read Only
Type	Character
Initial value	""

Windows 95/98/Me: String that provides additional version information. For example, "C" indicates Windows 95 OSR2 and "A" indicates Windows 98 Second Edition.

1.5.15.1.2 TSysVer:cSysName

String that identifies the operating system name. For example "Microsoft Windows XP Professional".

Scope	read Only
Type	Character
Initial value	"Microsoft Windows"

1.5.15.1.3 TSysVer:cSysVer

String that identifies the operating system's version. For example: "5.1.2600 Service Pack 1 Compilation 2600".

Scope	read Only
Type	Character
Initial value	""

1.5.15.1.4 TSysVer:cWinNTProduct

String that specifies the Windows NT product type.

Scope	read Only
Type	Character
Initial value	""

The types could be:

WINNT	Workstation
LANMANT	Server
SERVERNT	Advanced Server.

1.5.15.1.5 TSysVer:llsWin10

.T. if the operating system is Windows 10.

Scope	read Only
Type	Logic
Initial value	.F.

1.5.15.1.6 TSysVer:llsWin7

.T. if the operating system is Windows 7.

Scope	read Only
Type	Logic
Initial value	.F.

1.5.15.1.7 TSysVer:llsWin98

.T. if the operating system is Windows 98.

Scope	read Only
Type	Logic
Initial value	.F.

1.5.15.1.8 TSysVer:lisWinMe

.T. if the operating system is Windows Me.

Scope	read Only
Type	Logic
Initial value	.F.

1.5.15.1.9 TSysVer:lisWin2000

.T. if the operating system is Windows 2000.

Scope	read Only
Type	Logic
Initial value	.F.

1.5.15.1.10 TSysVer:lisWinXP

.T. if the operating system is Windows XP.

Scope	read Only
Type	Logic
Initial value	.F.

1.5.15.1.11 TSysVer:lisWinXP64

.T. if the operating system is Windows XP 64.

Scope	read Only
Type	Logic
Initial value	.F.

1.5.15.1.12 TSysVer:lisWinServer2003

.T. if the operating system is Windows Server 2003 family.

Scope	read Only
Type	Logic
Initial value	.F.

1.5.15.1.13 TSysVer:IsWinServer2016

.T. if the operating system is Windows Server 2016 family.

Scope	read Only
Type	Logic
Initial value	.F.

1.5.15.1.14 TSysVer:IsWinServer2019

.T. if the operating system is Windows Server 2019 family.

Scope	read Only
Type	Logic
Initial value	.F.

1.5.15.1.15 TSysVer:IsWinVista

.T. if the operating system is Windows Vista.

Scope	read Only
Type	Logic
Initial value	.F.

1.5.15.1.16 TSysVer:IsWinHomeServer

.T. if the operating system is Windows Home server.

Scope	read Only
Type	Logic
Initial value	.F.

1.5.15.1.17 TSysVer:lsWow64

Returns .T. if the system used is a 64 bits OS and the application is running on a 32 bits subsystem..

Scope	read Only
Type	Logic
Initial value	.F.

1.5.15.1.18 TSysVer:nBuildNumber

Indicates the operating system's "build" number.

Scope	read Only
Type	Numeric
Initial value	0

1.5.15.1.19 TSysVer:nMajorVersion

Indicates the "mayor" version number from the operating system.

Scope	read Only
Type	Numeric
Initial value	0

And the values can be:

3	Windows NT 3.51
4	Windows 95
4	Windows 98
4	Windows Me
4	Windows NT 4.0
5	Windows 2000
5	Windows XP
5	Windows Server 2003 family.

1.5.15.1.20 TSysVer:nMemoryRAM

RAM in kilobytes of the PC.

Scope	read Only
Type	Numeric
Initial value	0

1.5.15.1.21 TSysVer:nMinorVersion

Indicates the "minor" version number from the operating system.

Scope	read Only
Type	Numeric
Initial value	0

And the values can be:

10	Windows NT 3.1
50	Windows NT 3.5
51	Windows NT 3.51
0	Windows 95
10	Windows 98
90	Windows Me
0	Windows NT 4.0
0	Windows 2000
1	Windows XP
2	Windows Server 2003 family

1.5.15.1.22 TSysVer:nPlatformID

Indicates the operating system's platform.

Scope	read Only
Type	Numeric
Initial value	0

The values can be:

VER_PLATFORM_WIN32s	Win32s en Windows 3.1
VER_PLATFORM_WIN32_WINDOWS	Windows 95, 98 o Me

VER_PLATFORM_WIN32_NT	Windows NT, 2000, XP o Server 2003
-----------------------	------------------------------------

1.5.15.1.23 TSysVer:nProcessorArch

CPU architecture

Scope	read Only
Type	Numeric
Initial value	0

The values can be:

PROCESSOR_ARCHITECTURE_I NTEL	x86
PROCESSOR_ARCHITECTURE_I A64	Intel Itanium Processor Family (IPF)
PROCESSOR_ARCHITECTURE_A AMD64	x64 (AMD o Intel)
PROCESSOR_ARCHITECTURE_I A32_ON_WIN64	WOW64

1.5.15.1.24 TSysVer:nProductType

Additional system information.

NOTE: Only available in Windows NT 4.0 SP6/2000/XP/2003 or beyond.

Scope	read Only
Type	Numeric
Initial value	0

The values can be:

VER_NT_WORKSTATION	The system is running in Windows NT 4.0 Workstation, Windows 2000 Professional, Windows XP Home Edition, or Windows XP Professional.
VER_NT_SERVER	El system is a server

1.5.15.1.25 TSysVer:nReleaseID

Indicates the operating system's release. Only available on Windows 10.

Scope	read Only
--------------	-----------

Type	Numeric
Initial value	0

1.5.15.1.26 TSysVer:nServicePackMajor

'Major' version number from the last "Service Pack" installed in the system.

For example, for the "Service Pack 1", the "major" version number is 1. In the case that there is not "Service Pack" installed, the value is zero.

NOTE: Only available in Windows NT 4.0 SP6/2000/XP/2003 or beyond.

Scope	read Only
Type	Numeric
Initial value	0

1.5.15.1.27 TSysVer:nServicePackMinor

'Minor' version number from the last "Service Pack" installed in the system.

For example, for the "Service Pack 1", the "minor" version number is 0.

NOTE: Only available in Windows NT 4.0 SP6/2000/XP/2003 or beyond.

Scope	read Only
Type	Numeric
Initial value	0

1.5.15.1.28 TSysVer:nSuiteMask

Mask that identifies the product "suites" available in the system.

NOTE: Only available in Windows NT 4.0 SP6/2000/XP/2003 or beyond.

Scope	read Only
Type	Numeric
Initial value	0

It can be a combination of the following values:

VER_SUITE_BLADE	Windows Server 2003 Web edition
VER_SUITE_DATACENT	Windows 2000 or Windows Server 2003 "Datacenter"

ER	edition.
VER_SUITE_ENTERPRISE	Windows NT 4.0, Windows 2000 Advanced Server or Windows Server 2003 "Enterprise" edition.
VER_SUITE_PERSONAL	Windows XP "Home" edition

1.5.15.2 TSysVer:Methods

■ Constructor ■ Standard

Type	Name
■	Create

1.5.15.2.1 TSysVer:Create

Class constructor.

Type	Constructor
Parameters	None
Return value	Self Reference (Self)

1.5.16 TToolTip

This class represents a Windows Tooltip object.



A Tooltip is a small pop-up windows with text in those controls with the cTooltip assigned when the mouse pointer is stopped over them for a certain period of time.

This class allows to manage the standard Windows API Tooltips and it may be that the user does not need to access directly to it.

Hierarchy Inherits from **TWinObject**
File Name \source\Tooltip.prg

1.5.16.1 TToolTip:Methods

■ Constructor ■ Standard

Type	Name
■	AddTool
■	Create

- DelTool
- SetToolInfo

1.5.16.1.1 TToolTip:AddTool

Assign a Tooltip to the specified control.

Type	Standard
Parameters	<oControl> Control where the tooltip is assigned <aZone> Coordinate array that indicate the position where the Tooltips will be shown in the same control. <nId> Tooltip Id.
Return value	NIL

1.5.16.1.2 TToolTip:Create

Class constructor.

Type	Constructor
Parameters	None
Return value	<Self> Self Reference

1.5.16.1.3 TToolTip:DelTool

Deletes a Tooltip from a control.

Type	Standard
Parameters	<oControl> Tooltip control <nId> Tooltip identifier
Return value	NIL

1.5.16.1.4 TToolTip:SetToolTipInfo

Modifies a control's Tooltip.

Type	Standard
Parameters	<oControl> Control where the Tooltip belongs to
Return value	NIL

1.6 Controls

1.6.1 Standard

1.6.1.1 TButton

This class represents a Windows button control (Pushbutton or DefPushButton type).



Hierarchy	Inherits from TStdControl
See also	TBtnBmp
File name	\source\Button.prg

1.6.1.1.1 TButton:Properties

■ read Only ■ Assignable ■ Design assignable ■ Run-time assignable

Scope	Name	Type	Initial value
■	ICancel	Logic	.F.
■	IDefault	Logic	.F.
■	IMultiLine	Logic	.F.
■	IPushed	Logic	.F.
■	nAlignment	Numeric	taCENTER
■	nHeight	Numeric	25
■	nModalResult	Numeric	mrNONE
■	nVAlignment	Numeric	vaCENTER
■	nWidth	Numeric	80

1.6.1.1.1.1 TButton:ICancel

Indicates to trigger the OnClick event when the **ESC** key is pushed in its container form.

Scope: Design assignable

Type:	Logic
Initial value:	.F.

1.6.1.1.1.2 TButton:IDefault

Indicates to trigger the OnClick event when the **ENTER** key is pushed in its container form.

Scope:	Design assignable
Type:	Logic
Initial value:	.F.

1.6.1.1.1.3 TButton:IMultiLine

If it is .T., the button text can use several lines.

Scope:	Design assignable
Type:	Logic
Initial value:	.F.

1.6.1.1.1.4 TButton:IPushed

If it is .T. marks the button as pushed.

Scope:	Assignable
Type:	Logic
Initial value:	.F.

1.6.1.1.1.5 TButton:nAlignment

Indicates the button's horizontal text alignment.

Scope:	Design assignable
Type:	Numeric
Initial value:	taCENTER
Possible values:	taLEFT, taCENTER, taRIGHT

1.6.1.1.1.6 TButton:nHeight

Indicates the button height.

Scope:	Assignable
Type:	Numeric
Initial value:	25

1.6.1.1.1.7 TButton:nModalResult

Value that indicates the close mode from the modal form where the button belongs.

Scope:	Assignable
Type:	Numeric
Initial value:	mrNONE
Possible values:	mrNONE, mrOK, mrCANCEL, mrABORT, mrRETRY, mrIGNORE, mrYES, mrNO, mrCLOSE, mrHELP, mrTRYAGAIN, mrCONTINUE, mrALL, mrNOTOALL, mrYESTOALL

This property is very useful when it is used together with modal forms. When its value is different to mrNONE, the simple push button operation implicates to close the form and the value of this property is assigned to the nModalResult property of the form where the button belongs, and this value is returned as returned value in the Showmodal() method calls.

The possible values are basically the description of the classic operations:

Example:

```
With Object TMyForm():New()  
  If :ShowModal == mrOK  
    . . . .  
  Endif  
End With
```

1.6.1.1.1.8 TButton:nVAlignment

Vertical alignment for the button text.

Scope:	Design assignable
Type:	Numeric
Initial value:	vaCENTER
Possible values:	vaTOP, vaCENTER, vaBOTTOM

1.6.1.1.1.9 TButton:nWidth

Indicates the button width.

Scope:	Assignable
Type:	Numeric
Initial value:	80

1.6.1.1.2 TButton:Methods

■ Constructor ■ Standard ■ Only after Create()

Type	Name
■	Click

1.6.1.1.2.1 TButton:Click

Executes the action associated to the button.

Type	Only after Create()
Parameters	None
Return value	The OnClick returned value

1.6.1.1.3 TButton:Events

Name
OnClick

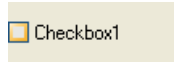
1.6.1.1.3.1 TButton:OnClick

Event that is produced when the button is pushed.

Parameters	<oSender>:
:	Reference to the object that triggers the event
Return value:	<IResult>:
	If nModalResult is different than mrNONE it can abort the close form process returning a .F. value in this event

1.6.1.2 TCheckBox

This class represents a Windows CheckBox control type.



Hierarchy Inherits from TStdControl
File name \source\CheckBox.prg

1.6.1.2.1 TCheckBox:Properties

■ read-Only ■ Assignable ■ Design assignable ■ Run-time assignable

Scope	Name	Type	Initial value
■	IChecked	Logic	.F.
■	IMultiLine	Logic	.F.
■	IPushLike	Logic	.F.
■	ITransparent	Logic	.T.
■	nAlignment	Numeric	taLEFT
■	nHeight	Numeric	18
■	nVAlignment	Numeric	vaCENTER
■	nWidth	Numeric	90

1.6.1.2.1.1 TCheckBox:IChecked

If it is .T., the control is checked or marked.

Scope:	Assignable
Type:	Logic
Initial value:	.F.

1.6.1.2.1.2 TCheckBox:IMultiLine

If it is .T. the control text can use several lines.

Scope:	Design assignable
Type:	Logic
Initial value:	.F.

1.6.1.2.1.3 TCheckBox:IPushLike

If it is .T., the control will look like a button.

Scope:	Design assignable
Type:	Logic
Initial value:	.F.

1.6.1.2.1.4 TCheckBox:ITransparent

If it is .T., the control will have transparent color.

Scope:	Design assignable
Type:	Logic
Initial value:	.T.

1.6.1.2.1.5 TCheckBox:nAlignment

Indicates the control horizontal alignment.

Scope:	Design assignable
Type:	Numeric
Initial value:	taLEFT
Possible values:	taLEFT, taRIGHT

1.6.1.2.1.6 TCheckBox:nHeight

Indicates the control height.

Scope:	Assignable
Type:	Numeric
Initial value:	18

1.6.1.2.1.7 TCheckBox:nVAlignment

Vertical alignment for the text control.

Scope:	Design assignable
---------------	-------------------

Type:	Numeric
Initial value:	vaCENTER
Possible values:	vaTOP, vaCENTER, vaBOTTOM

1.6.1.2.1.8 TCheckBox:nWidth

Indicates the button width.

Scope:	Assignable
Type:	Numeric
Initial value:	90

1.6.1.2.2 TCheckBox:Methods

Constructor Standard Only after Create()

Typ e	Name
<input type="checkbox"/>	Click
<input type="checkbox"/>	Toggle

1.6.1.2.2.1 TCheckBox:Click

Changes the control state, from marked to unmarked and vice versa and triggers the OnClick event.

Type	Only after Create()
Parameters	None
Return value	The returned OnClick value

1.6.1.2.2.2 TCheckBox:Toggle

Changes the control state, from marked to unmarked and vice versa.

Type	Only after Create()
Parameters	None
Return value	NIL

1.6.1.2.3 TCheckBox:Events

Name

OnChange

OnClick

1.6.1.2.3.1 TCheckBox:OnChange

Event that is produced when the control changes its state.

Parameters	<oSender>:
:	Reference to the object that triggers the event
Return value:	NIL

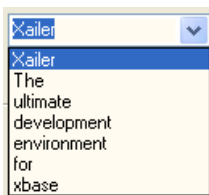
1.6.1.2.3.2 TCheckBox:OnClick

Event that is triggered when the user clicks the control

Parameters	<oSender>:
:	Reference to the object that triggers the event
Return value:	NIL

1.6.1.3 TComboBox

This class represents a ComboBox Windows control type.



Hierarchy	Inherits from TStdControl
File name	\source\ComboBox.prg

1.6.1.3.1 TComboBox:Properties

■ read Only ■ Assignable ■ Design assignable ■ Run-time assignable

Scope	Name	Type	Initial value
e			

■	altems	Array	{}
■	IAutoHScroll	Logic	.F.
■	IDropped	Logic	.F.
■	IExtendedUI	Logic	.F.
■	ISort	Logic	.F.
■	nClrPane	Numeric	clWindow
■	nDroppedHeight	Numeric	300
■	nDroppedWidth	Numeric	0
■	nEditSelEnd	Numeric	-1
■	nEditSelStart	Numeric	-1
■	nHeight	Numeric	100
■	nHorzExtent	Numeric	0
■	nIndex	Numeric	1
■	nMaxLength	Numeric	0
■	nStyle	Numeric	csDROPDOWN
■	nTopIndex	Numeric	1
■	nWidth	Numeric	100

1.6.1.3.1.1 TComboBox:altems

List of elements displayed by the control.

Scope:	Assignable
Type:	Array
Initial value	{}

1.6.1.3.1.2 TComboBox:IAutoHScroll

If this property is enabled, when the user type text in the combo field, it will automatically move the text beyond the combo boundaries. If this property is disabled, only will be possible to type text until the combo field width.

Scope:	Design assignable
Type:	Logic
Initial value	.F.

1.6.1.3.1.3 TComboBox:IDropped

This property gets or sets the control drop state.

Scope:	Run-time assignable
Type:	Logic
Initial value	.F.

1.6.1.3.1.4 TComboBox:ExtendedUI

If this property is enabled, activates the extended interface of the control. By default, pushing F4 opens and closes the list and the down key changes the item selected. With the extended interface the F4 key is useless and the 'Down' key opens the list.

Scope:	Assignable
Type:	Logic
Initial value	.F.

1.6.1.3.1.5 TComboBox:ISort

Sorts automatically the control's list of values.

Scope:	Design assignable
Type:	Logic
Initial value	.F.

1.6.1.3.1.6 TComboBox:nClrPane

Indicates the control background color.

Scope:	Assignable
Type:	Numeric
Initial value	clWindow

(See also the appendix to check the possible colors available)

1.6.1.3.1.7 TComboBox:nDroppedHeight

Indicates the maximum height used by the list when it is displayed, specified in pixels.

Scope:	Assignable
---------------	------------

Type:	Numeric
Initial value	300

1.6.1.3.1.8 TComboBox:nDroppedWidth

Indicates the maximum width used by the list when it is displayed, specified in pixels.

Scope:	Assignable
Type:	Numeric
Initial value	0

1.6.1.3.1.9 TComboBox:nEditSelEnd

Indicates the position from the last character selected in the edit control, if any. Otherwise, its value is -1.

Scope:	Assignable
Type:	Numeric
Initial value	-1

1.6.1.3.1.10 TComboBox:nEditSelStart

Indicates the position from the first character selected in the edit control, if any. Otherwise, its value is -1.

Scope:	Assignable
Type:	Numeric
Initial value	-1

1.6.1.3.1.11 TComboBox:nHeight

Indicates the control height.

Scope:	Assignable
Type:	Numeric
Initial value	100

1.6.1.3.1.12 TComboBox:nHorzExtent

Establishes the control width in pixels where the control can scroll horizontally. If the control width is lower than this value, it will show a scroll bar.

Scope:	Assignable
Type:	Numeric
Initial value	0

1.6.1.3.1.13 TComboBox:nIndex

Indicates the index of the current selected element.

Scope:	Assignable
Type:	Numeric
Initial value	1

1.6.1.3.1.14 TComboBox:nMaxLength

Indicates the maximum number of character that can be introduced in the edit control.

Scope:	Assignable
Type:	Numeric
Initial value	0

1.6.1.3.1.15 TComboBox:nStyle

Indicates the control style.

Scope:	Design assignable
Type:	Numeric
Initial value	csDROPDOWN
Possible values:	csDROPDOWN, csDROPDOWNLIST

1.6.1.3.1.16 TComboBox:nTopIndex

Forces to show the indicated element.

Scope:	Assignable
Type:	Numeric
Initial value	1

1.6.1.3.1.17 TComboBox:nWidth

Indicates the control width.

Scope:	Assignable
Type:	Numeric
Initial value	100

1.6.1.3.2 TComboBox:Methods

■ Constructor ■ Standard ■ Only after Create()

Typ Name	e
■ AddItem	
■ DeleteItem	
■ InsertItem	
■ ModifyItem	
■ SelectString	

1.6.1.3.2.1 TComboBox:AddItem

Adds a new element to the list of values to be displayed by the control.

Type	Standard
Parameters	<cItem>: Description for the element to be added
Return value	<ISuccess>: .T. if the operation is successful

1.6.1.3.2.2 TComboBox:DeleteItem

Deletes an element from the list of values to be displayed.

Type	Standard
Parameters	<nPos> : Element's position to be deleted
Return value	<ISuccess> : .T. if the operation is successful

1.6.1.3.2.3 TComboBox:InsertItem

Insert an element in the list of values to be displayed.

Type	Standard
Parameters	<nPos> : Position where the new element will be inserted <cItem> : Indicates the description for the element to be inserted
Return value	<ISuccess> : .T. if the operation is successful

1.6.1.3.2.4 TComboBox:ModifyItem

Modify an element from the list of values displayed.

Type	Standard
Parameters	<nPos> : Element's position number to be modified <cItem> : New element description
Return value	<ISuccess> : .T. if the operation is successful

1.6.1.3.2.5 TComboBox:SelectString

Searches in the list of values an element that starts with the characters indicated in the string and updates the control's edit control.

Type	Standard
Parameters	<cString> : String to search

	<nFrom>: Element position where the search operation will start from
Return value	<nPos>: Position found, or 0 if there is an error

1.6.1.3.3 TComboBox:Events

Name	
	OnChange
	OnCloseUp
	OnDbClick
	OnDropDown
	OnEditChange
	OnEditUpdate

1.6.1.3.3.1 TComboBox:OnChange

Event that is produced when the control changes its state.

Parameters	<oSender>: Reference to the object that triggers the event
:	<nItem>: New item selected
	<nOldItem>: Old item selected
Return value:	<IOk>: If .F. the change is aborted

1.6.1.3.3.2 TComboBox:OnCloseUp

Event that is triggered when the list of elements is going to be closed.

Parameters	<oSender>: Reference to the object that triggers the event
:	
Return value:	NIL

1.6.1.3.3.3 TComboBox:OnDbClick

Event that is produced when the user double-clicks the mouse over the control.

Parameters	<oSender> :
:	Reference to the object that triggers the event
Return value:	NIL

1.6.1.3.3.4 TComboBox:OnDropDown

Event that is triggered when the list of elements is going to be displayed.

Parameters	<oSender> :
:	Reference to the object that triggers the event
Return value:	NIL

1.6.1.3.3.5 TComboBox:OnEditChange

Event that is triggered when the user types a character in the control's edit control. It is triggered after it shows the text.

Parameters	<oSender> :
:	Reference to the object that triggers the event
Return value:	NIL

1.6.1.3.3.6 TComboBox:OnEditUpdate

Event that is triggered every time that the user types a character in the control's edit control. It is triggered before to show the text.

Parameters	<oSender> :
:	Reference to the object that triggers the event
Return value:	NIL

1.6.1.4 TEdit

This class represents a Windows Edit control used to edit simple text.



For multi line edition you should use the TMemo control. For edition using a template (the Clipper's picture clause), it is better to use the TMaskEdit control.

Hierarchy Inherits from TStdControl
See also TMemo, TMaskEdit
File name \source\Edit.prg

1.6.1.4.1 TEdit:Properties

■ read Only ■ Assignable ■ Design assignable ■ Run-time assignable

Scope	Name	Type	Initial value
■	cPasswordChar	Character	*
■	IAutoSelect	Logic or NIL	asDONTCARE
■	IBorder	Logic	.T.
■	IHScroll	Logic	.T.
■	IModify	Logic	.F.
■	INumeric	Logic	.F.
■	IPassword	Logic	.F.
■	IReadOnly	Logic	.F.
■	IShowSystemBalloon	Logic	.F.
■	nAlignment	Numeric	taLEFT
■	nCharCase	Numeric	ecNONE
■	nClrPaneDisabled	Numeric	cWindow
■	nClrPaneFocus	Numeric	cWindow
■	nClrTextDisabled	Numeric	cBtnFace
■	nClrTextFocus	Numeric	cWindowText
■	nHeight	Numeric	20
■	nMaxLength	Numeric	0
■	nWidth	Numeric	90
■	Value	Any	""

1.6.1.4.1.1 TEdit:cPasswordChar

Unique character to use as template/picture when it is established that the control is a password type control through the IPassword property.

Scope:	Design assignable
Type:	Character
Initial value:	*

1.6.1.4.1.2 TEdit:IAutoSelect

Establishes the auto text selection when the control receives the focus.

Scope:	Design assignable
Type:	Logic or NIL
Initial value:	asDONTCARE
Possible values:	asDONTCARE, asSELECTALL, asSELECTNONE

1.6.1.4.1.3 TEdit:IBorder

If it is .T., it will show the control's border.

Scope:	Design assignable
Type:	Logic
Initial value:	.T.

1.6.1.4.1.4 TEdit:IHScroll

If it is .T. the control will show an horizontal scroll bar.

Scope:	Design assignable
Type:	Logic
Initial value:	.F.

1.6.1.4.1.5 TEdit:IModify

Indicates if the control has been modified. It is possible to assign any logical value to this property.

Scope:	Run-time assignable
Type:	Logic
Initial value:	.F.

1.6.1.4.1.6 TEdit:INumeric

If it is .T., the control will accept only numeric characters.

Scope:	Design assignable
Type:	Logic
Initial value:	.F.

1.6.1.4.1.7 TEdit:IPassword

If it is .T. the control will show all the characters with a unique character specified in the cPasswordChar property. It is used to receive password type values when it is needed to hide the text that is being typed.

Scope:	Design assignable
Type:	Logic
Initial value:	.F.

1.6.1.4.1.8 TEdit:IReadOnly

If it is .T. the control will be read-only.

Scope:	Assignable
Type:	Logic
Initial value:	.F.

1.6.1.4.1.9 TEdit:IShowSystemBalloon

If it is .T. the control will not show the balloon tips when non valid characters are introduced. For example, letters on numeric controls or when 'Caps Lock' is activated when introducing passwords.

Scope:	Assignable
Type:	Logic
Initial value:	.F.

1.6.1.4.1.10 TEdit:nAlignment

Indicates the horizontal text alignment.

Scope:	Design assignable
Type:	Numeric
Initial value:	taLEFT
Possible values:	taLEFT, taCENTER, taRIGHT

1.6.1.4.1.11 TEdit:nCharCase

Indicates the character type set that will be shown by the control. They can be: any, only uppercase or only lowercase. In the case of uppercase or lowercase, the control will make automatically all the needed adjustments.

Scope:	Design assignable
Type:	Numeric
Initial value:	ecNONE
Possible values:	ecNONE, ecUPPERCASE, ecLOWERCASE

1.6.1.4.1.12 TEdit:nClrPaneDisabled

Background control color when it is disabled.

Scope:	Assignable
Type:	Numeric
Initial value:	clWindow

(See also the appendix to see possible colors)

1.6.1.4.1.13 TEdit:nClrPaneFocus

Background control color when it received the focus. Normally this color coincides with the normal background color(nClrPane), but it can be changed to show to the user what the current control active is..

Scope:	Assignable
Type:	Numeric
Initial value:	clWindow

(See also the appendix to see possible colors)

1.6.1.4.1.14 TEdit:nClrTextDisabled

Control text color, when it is disabled.

Scope:	Assignable
Type:	Numeric
Initial value:	clBtnFace

(See also the appendix to see possible colors)

1.6.1.4.1.15 TEdit:nClrTextFocus

Control text color, when it received the focus. Normally this color coincides with the normal text color (nClrText), but it is possible to change it to show to the user what the current control active is.

Scope:	Assignable
Type:	Numeric
Initial value:	clWindowText

(See also the appendix to see possible colors)

1.6.1.4.1.16 TEdit:nHeight

Indicates the control height specified in pixels.

Scope:	Assignable
Type:	Numeric
Initial value:	20

1.6.1.4.1.17 TEdit:nMaxLength

Indicates the maximum number of characters that the user will be allowed to input on the control. A zero value indicated unlimited length.

This property will initialize its value with the length of the first passed string with the property Value, but only if its current value is zero.

Scope:	Design assignable
Type:	Numeric
Initial value:	0

Note: This size limitation only affects the user input. By code you can assign any value of any length.

1.6.1.4.1.18 TEdit:nWidth

Indicates the control width in pixels.

Scope:	Assignable
Type:	Numeric
Initial value:	90

1.6.1.4.1.19 TEdit:Value

Indicates the control's text or value. This property allows any type value, but it is converted to a string character type. Is you need to keep the type value, it is better to use the TMaskEdit control.

Scope:	Assignable
Type:	Any
Initial value:	""

1.6.1.4.2 TEdit:Methods

■ Constructor ■ Standard ■ Only after Create()

Typ	Name
■	CanCopy
■	CanCut
■	CanPaste
■	CanUndo
■	EmptyUndoBuffer
■	GetSel
■	GetSelText
■	nPos
■	SelectAll
■	SelectNone
■	SetSel

1.6.1.4.2.1 TEdit:CanCopy

Returns .T. if is it possible to copy the text selected in the control to the clipboard.

Type	Only after Create()
Parameters	None
Return value	<ISuccess> Returns .T. if is it possible to copy the text selected in the control to the clipboard

1.6.1.4.2.2 TEdit:CanCut

Returns .T. if it is possible to cut the text selected in the control to the clipboard.

Type	Only after Create()
Parameters	None
Return value	<ISuccess>.T. if it is possible to cut the text selected in the control to the clipboard

1.6.1.4.2.3 TEdit:CanPaste

Returns .T. if it is possible to make a paste operation from the clipboard to the control.

Type	Only after Create()
Parameters	None
Return value	<ISuccess> .T. is if is possible to make paste operation from the clipboard

1.6.1.4.2.4 TEdit:CanUndo

Returns .T. if it is possible to make a undo operation in the control.

Type	Only after Create()
Parameters	None
Return value	<ISuccess> .T. is if is possible to make a undo operation

1.6.1.4.2.5 TEdit:EmptyUndoBuffer

Empty the control's buffer content used in undo operations.

Type	Only after Create()
Parameters	None
Return value	NIL

1.6.1.4.2.6 TEdit:GetSel

Returns the first and last positions from the selected text from the control in a two dimensional array. The first element receives the initial position and the second element the receives the final position.

The indicated values start from zero. It means that the first position starts with zero and not with one, as usual.

Type	Only after Create()
Parameters	None
Return value	<aPos>

1.6.1.4.2.7 TEdit:GetSelText

Returns the current selected text. If no text is selected returns an empty string.

Type	Only after Create()
Parameters	None
Return value	<cText>

1.6.1.4.2.8 TEdit:nPos

Indicates the current cursor position in the control.

Type	Only after Create()
Parameters	None
Return value	<nPos>

1.6.1.4.2.9 TEdit:SelectAll

Selects all the text in the control.

Type	Only after Create()
Parameters	None
Return value	NIL

1.6.1.4.2.10 TEdit:SelectNone

Cancels any existing selection in the control.

Type	Only after Create()
Parameters	None
Return value	NIL

1.6.1.4.2.11 TEdit:SetSel

Selects in the control a text range.

The indicated values start from zero. It means that the first position starts with zero and not with one, as usual.

Type	Only after Create()
Parameters	<nStart> : First character to be selected. A -1 value removes any existing selection <nEnd> : Last character to be selected. A -1 value when nStart value is 0 selects all the text. A -1 value when nStart is -1 moves the cursor at the end of the text
Return value	NIL

1.6.1.4.3 TEdit:Events

Name
OnChange

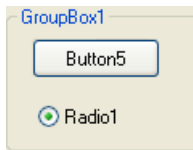
1.6.1.4.3.1 TEdit:OnChange

Event that is produced when the control changes its value.

Parameters	<oSender> : Reference to the object that triggers the event
Return value	NIL

1.6.1.5 TGroupBox

This class represents a GroupBox control. in other words, a frame with title that may contain other controls.



Hierarchy Inherits from TWinControl
See also TPanel
File name \source\GroupBox.prg

1.6.1.5.1 TGroupBox:Properties

read Only
 Assignable
 Design assignable
 Run-time assignable

Scope	Name	Type	Initial value
<input type="checkbox"/>	nAlignment	Numeric	taLEFT
<input type="checkbox"/>	nHeigth	Numeric	90
<input type="checkbox"/>	nWidth	Numeric	120

1.6.1.5.1.1 TGroupBox:nAlignment

Alignment for the text that is shows as the GroupBox title.

Scope	Assignable
Type	Numeric
Initial value	taLEFT
Possible values	taLEFT, taRIGHT, taCENTER

1.6.1.5.1.2 TGroupBox:nHeight

Indicates the control height.

Scope	Assignable
Type	Numeric
Initial value	90

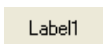
1.6.1.5.1.3 TGroupBox:nWidth

Indicates the control width.

Scope	Assignable
Type	Numeric
Initial value	120

1.6.1.6 TLabel

This class represents a text label control.



Hierarchy	Inherits from TControl
File name	\source\Label.prg

1.6.1.6.1 TLabel:Properties

read Only
 Assignable
 Design assignable
 Run-time assignable

Scope	Name	Type	Initial value
<input type="checkbox"/>	IAutosize	Logic	.T.
<input type="checkbox"/>	IMultiLine	Logic	.F.
<input type="checkbox"/>	IPathStyle	Logic	.F.
<input type="checkbox"/>	ITransparent	Logic	.T.
<input type="checkbox"/>	nAlignment	Numeric	taLEFT
<input type="checkbox"/>	nBorderStyle	Numeric	bvNONE
<input type="checkbox"/>	nVAlignment	Numeric	vaTOP

1.6.1.6.1.1 TLabel:IAutosize

Automatically adjust the area where the control is painted.

Scope	Assignable
Type	Logic
Initial value	.T.

1.6.1.6.1.2 TLabel:IMultiLine

Allows to the control to sow several lines of text.

Scope	Assignable
Type	Logic
Initial value	.F.

1.6.1.6.1.3 TLabel:IPathStyle

If the text can not be completely shown, a '...' will be displayed at the end of the string.

Scope	Assignable
Type	Logic
Initial value	.F.

1.6.1.6.1.4 TLabel:ITransparent

The control is painted in transparent mode over the parent form.

Scope	Assignable
Type	Logic
Initial value	.T.

1.6.1.6.1.5 TLabel:nAlignment

Horizontal text alignment. The possible values are taRIGHT and taLEFT.

Scope	Assignable
Type	Numeric
Initial value	taLEFT
Possible values	taLEFT, taRIGHT, taCENTER

1.6.1.6.1.6 TLabel:nBorderStyle

Indicates the control border style.

Scope	Assignable
--------------	------------

Type	Numeric
Initial value	bvNONE
Possible values	bvNONE, bvRAISED, bvSUNKEN, bvBUMP, bvETCHED, bvFLAT

1.6.1.6.1.7 TLabel:nVAlignment

Vertical text alignment. Shows the text box aligned to the top, centered or aligned to the button in reference to the text.

Scope	Assignable
Type	Numeric
Initial value	vaTOP
Possible values	vaTOP, vaBOTTOM, vaCENTER

1.6.1.6.2 TLabel:Methods

■ Constructor ■ Standard

Type	Name
■	Adjust

1.6.1.6.2.1 TLabel:Adjust

Automatically adjusts the area where the control is painted.

Type	Standard
Parameters	None
Return value	NIL

1.6.1.7 TLine

This class paints a vertical or horizontal line..

Hierarchy	Inherits from TControl
File name	\source\Line.prg

1.6.1.7.1 TLine:Properties

■ read Only
 ■ Assignable
 ■ Design assignable
 ■ Run-time assignable

Scope	Name	Type	Initial value
■	IVertical	Logic	.F.
■	nPenColor	Numeric	clLightGray
■	nPenSize	Numeric	1
■	nPenStyle	Numeric	psSolid

1.6.1.7.1.1 TLine:IVertical

If true the line is painted vertically.

Scope	Assignable
Type	Logic
Initial value	.T.

1.6.1.7.1.2 TLine:nPenColor

The line color.

Scope	Assignable
Type	Numeric
Initial value	clLightGray

Consult the appendix for the list of available colors

1.6.1.7.1.3 TLine:nPenSize

The pen size in pixels.

Scope	Assignable
Type	Numeric
Initial value	1

1.6.1.7.1.4 TLine:nPenStyle

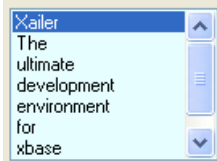
The pen style.

Scope	Assignable
Type	Numeric

Initial value	psSolid
Possible values	psSolid, psDash, psDot, psDashDot, psDashDotDot, psNull, psInsideFrame

1.6.1.8 TListBox

This class represents a ListBox Windows control type. This class shows a list of options and the user may select any of them. It is possible also to select more than one element, setting the IMultipleSel to .T..



Hierarchy	Inherits from TStdControl
File name	\source\ListBox.prg

1.6.1.8.1 TListBox:Properties

read Only
 Assignable
 Design assignable
 Run-time assignable

Scope	Name	Type	Initial value
<input type="checkbox"/>	altems	Array	{}
<input type="checkbox"/>	cText	Character	""
<input type="checkbox"/>	IAutoSort	Logic	.F.
<input type="checkbox"/>	IBorder	Logic	.F.
<input type="checkbox"/>	IHScroll	Logic	.F.
<input type="checkbox"/>	IIntegralHeight	Logic	.T.
<input type="checkbox"/>	IMultiColumn	Logic	.F.
<input type="checkbox"/>	IMultipleSel	Logic	.F.
<input type="checkbox"/>	IScrollBarAlways	Logic	.F.
<input type="checkbox"/>	IUseTabStops	Logic	.F.
<input type="checkbox"/>	nClrPane	Numeric	clWindow
<input type="checkbox"/>	nColsWidth	Numeric	100
<input type="checkbox"/>	nHeight	Numeric	100
<input type="checkbox"/>	nIndex	Numeric	1
<input type="checkbox"/>	nTopIndex	Numeric	1
<input type="checkbox"/>	nWidth	Numeric	75

1.6.1.8.1.1 TListBox:alItems

List of items to be displayed by the control.

Scope:	Assignable
Type:	Array
Initial value:	{}

1.6.1.8.1.2 TListBox:cText

Indicates the selected item. If this property has a value assigned the control will try to select the element from the list with that value. If the control didn't find any element, the control will keep its old value.

Scope:	Assignable
Type:	Character
Initial value:	""

1.6.1.8.1.3 TListBox:lAutoSort

Sorts automatically the control list of values.

Scope:	Design assignable
Type:	Logic
Initial value:	.F.

1.6.1.8.1.4 TListBox:lBorder

If it is .T., the control will show a border.

Scope:	Design assignable
Type:	Logic
Initial value:	.F.

1.6.1.8.1.5 TListBox:lHScroll

If it is .T., the control will show an horizontal scroll bar.

Scope:	Design assignable
---------------	-------------------

Type:	Logic
Initial value:	.F.

1.6.1.8.1.6 TListBox:IntegralHeight

Indicates if the list height must be resize to show all the elements correctly.

Scope:	Design assignable
Type:	Logic
Initial value:	.T.

1.6.1.8.1.7 TListBox:IMultiColumn

Indicates if the list presents the element in columns.

Scope:	Design assignable
Type:	Logic
Initial value:	.F.

1.6.1.8.1.8 TListBox:IMultipleSel

Allows to select more than one element in the control.

Scope:	Design assignable
Type:	Logic
Initial value:	.F.

1.6.1.8.1.9 TListBox:IScrollBarAlways

Shows the vertical scroll bar even when the number of elements in bigger than the list height.

Scope:	Design assignable
Type:	Logic
Initial value:	.F.

1.6.1.8.1.10 TListBox:UseTabStops

Expands the tabs to format the elements of the list.

Scope:	Design assignable
Type:	Logic
Initial value:	.F.

1.6.1.8.1.11 TListBox:nClrPane

Indicates the background color.

Scope:	Assignable
Type:	Numeric
Initial value:	clWindow

(Check the appendix to see the colors available)

1.6.1.8.1.12 TListBox:nColsWidth

Indicates the column width in pixels when the property IMultiColumn is set to true.

Scope:	Assignable
Type:	Numeric
Initial value:	100

1.6.1.8.1.13 TListBox:nHeight

Indicates the control height.

Scope:	Assignable
Type:	Numeric
Initial value:	100

1.6.1.8.1.14 TListBox:nHorzExtent

Establishes horizontal scroll width in pixels. If the control width is lower than this value, the control will show a scroll bar.

Scope:	Assignable
---------------	------------

Type:	Numeric
Initial value:	0

1.6.1.8.1.15 TListBox:nIndex

Indicates the number of the current selected element.

Scope:	Assignable
Type:	Numeric
Initial value:	1

1.6.1.8.1.16 TListBox:nTopIndex

Force to show the indicated element, if is not visible.

Scope:	Assignable
Type:	Numeric
Initial value:	1

1.6.1.8.1.17 TListBox:nWidth

Indicates the control width.

Scope:	Assignable
Type:	Numeric
Initial value:	75

1.6.1.8.2 TListBox:Methods

■ Constructor ■ Standard ■ Only after Create()

Typ	Name
e	

■	AddItem
---	---------

■	Deleteltem
---	------------

■	Deleteltems
---	-------------

■	GetCurSel
---	-----------

■	GetHorzExtent
---	---------------

■	GetSelltems
---	-------------

■	InsertItem
---	------------

■	ModifyItem
---	------------

- SelectString
- SetCurSel
- SetHorzExtent
- SetSel
- SetSelItems

1.6.1.8.2.1 TListBox:AddItem

Adds a new element to the control's list of values.

Type	Standard
Parameters	<cltem>: Description for the element to be added
Return value	<ISuccess>: .T. if the operation is successful

1.6.1.8.2.2 TListBox:DeleteItem

Deletes an element from the control's list of values.

Type	Standard
Parameters	<nPos>: Position for the element to be deleted
Return value	<ISuccess>: .T. if the operation is successful

1.6.1.8.2.3 TListBox:DeleteItems

Deletes all the elements from the control list.

Type	Standard
Parameters	None
Return value	<ISuccess>: .T. if the operation is successful

1.6.1.8.2.4 TListBox:GetCurSel

Recovers the number of the element selected in the list of values.

Type	Only after Create()
Parameters	None

Return value	<nItem> : Number in the list for the selected element
---------------------	---

1.6.1.8.2.5 TListBox:GetHorzExtent

Returns the horizontal scroll from the list when the element's width is bigger than the list width.

Type	Only after Create()
Parameters	None
Return value	<nSize> : Horizontal scroll specified in pixels

1.6.1.8.2.6 TListBox:GetSelItems

Returns the selected list of elements. If must be used with IMultipleSel set to .T..

Type	Only after Create()
Parameters	None
Return value	<altems> : Array with the selected elements

1.6.1.8.2.7 TListBox:InsertItem

Inserts an element in the control's list of values.

Type	Standard
Parameters	<nPos> : Position for the element to be inserted <cItem> : New element description
Return value	<ISuccess> : .T. if the operation is successful

1.6.1.8.2.8 TListBox:ModifyItem

Modifies an element in the control's list of values.

Type	Standard
Parameters	<nPos> : Position for the element to be modified

	<cItem> : New element description
Return value	<ISuccess> : .T. if the operation is successful

1.6.1.8.2.9 TListBox:SelectString

Searches in the list of values an element that starts with the characters indicated in the search operation and updates the edited control content.

Type	Only after Create()
Parameters	<cString> : Search expression <nFrom> : Initial position where the Search operation will start
Return value	<ISuccess> : .T. if the operation is successful

1.6.1.8.2.10 TListBox:SetCurSel

Selects and shows an element from the list of values.

Type	Only after Create()
Parameters	<nPos> : Element to select
Return value	<nLastPos> : Previous selected element

1.6.1.8.2.11 TListBox:SetHorzExtent

Establishes the horizontal scroll of the list when the element width is longer than the list width.

Type	Only after Create()
Parameters	<nSize> : Horizontal scroll size
Return value	NIL

1.6.1.8.2.12 TListBox:SetSel

Establishes the selection status of an element. It must be used with IMultipleSel.

Type	Only after Create()
Parameters	<nItem> : Element to be selected <IMode> : If .T. the element is selected, if .F., the element is de-selected
Return value	<ISuccess> : .T. if the operation is successful

1.6.1.8.2.13 TListBox:SetSelItems

Establishes the list of selected items. It must be used with IMultipleSel.

Type	Only after Create()
Parameters	<alItems> : Array with the elements to be selected
Return value	<ISuccess> : .T. if the operation is successful

1.6.1.8.3 TListBox:Events

Name	
OnChange	
OnDbClick	

1.6.1.8.3.1 TListoBox:OnChange

Event that is produced when the control changes its state.

Parameters	<oSender> : Reference to the object that triggers the event <nIndex> : Selected element
Return value:	NIL

1.6.1.8.3.2 TListBox:OnDbClick

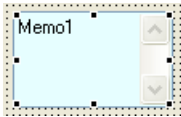
Event that is produced when the user double clicks the mouse pointer on the control.

Parameters	<oSender>:
:	Reference to the object that triggers the event
Return value:	NIL

1.6.1.9 TMemo

This class represents a TEdit control, specialized in the multi line text edition (memo).

The functionality of this control is limited, due only allows to establish a very basic format and for all the text. If you need more functionality, you can use the TRichEdit control.



Hierarchy	Inherits from TEdit
File name	\source\Memo.prg

1.6.1.9.1 TMemo:Properties

■ read Only ■ Assignable ■ Design assignable ■ Run-time assignable

Scope	Name	Type	Initial value
■	IAutoHScroll	Logic	.F.
■	IAutoVScroll	Logic	.T.
■	IHScroll	Logic	.F.
■	IVScroll	Logic	.T.
■	IWantReturn	Logic	.T.
■	IWantTab	Logic	.F.
■	nColumn	Numeric	0
■	nHeight	Numeric	60
■	nLine	Numeric	0
■	nLineCount	Numeric	0
■	nWidth	Numeric	100

1.6.1.9.1.1 TMem:IAutoHScroll

If it is .T., the horizontal scroll bar will be shown only when is needed.

Scope:	Design assignable
Type:	Logic
Initial value:	.F.

1.6.1.9.1.2 TMem:IAutoVScroll

If it is .T., the vertical scroll bar will be shown only when is needed.

Scope:	Design assignable
Type:	Logic
Initial value:	.T.

1.6.1.9.1.3 TMem:IHScroll

If it is .T. the horizontal scroll bar will be shown always.

Scope:	Design assignable
Type:	Logic
Initial value:	.F.

1.6.1.9.1.4 TMem:IVScroll

If it is .T. the vertical scroll bar will be shown always.

Scope:	Design assignable
Type:	Logic
Initial value:	.T.

1.6.1.9.1.5 TMem:IWantReturn

If it is .T. the control will process the <Return> key, and it will create a line feed in the text when it is pressed.

Scope:	Design assignable
Type:	Logic

Initial value:	.T.
-----------------------	-----

1.6.1.9.1.6 TMemo:IWantTab

If it is .T. the control will process the <Tab> key, when it is pressed.

Scope:	Design assignable
Type:	Logic
Initial value:	.F.

1.6.1.9.1.7 TMemo:nColumn

Cursor column position.

Scope:	Run-time assignable
Type:	Numeric
Initial value:	0

1.6.1.9.1.8 TMemo:nHeight

Indicates the control height specified in pixels.

Scope:	Assignable
Type:	Numeric
Initial value:	60

1.6.1.9.1.9 TMemo:nLine

Cursor line position.

Scope:	Run-time assignable
Type:	Numeric
Initial value:	0

1.6.1.9.1.10 TMemo:nLineCount

Indicates the number of lines used by the text.

Scope:	Read only
Type:	Numeric
Initial value:	0

1.6.1.9.1.11 TMemo:nWidth

Indicates the control width in pixels.

Scope:	Assignable
Type:	Numeric
Initial value:	100

1.6.1.9.2 TMemo:Methods

■ Constructor ■ Standard ■ Only after Create()

Typ Name
■ Append
■ Clear
■ Replace

1.6.1.9.2.1 TMemo:Append

Adds text to the end of the control.

Type	Only after Create()
Parameters	<cText> Text to add
Return value	NIL

1.6.1.9.2.2 TMemo:Clear

Clears the control text.

Type	Standard
Parameters	None

Return value	NIL
---------------------	-----

1.6.1.9.2.3 TMemo:Replace

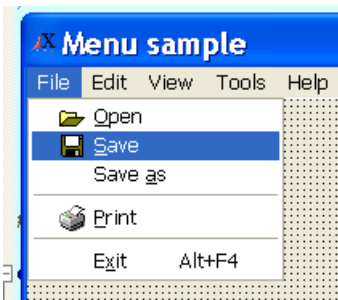
Replaces the selected text with new text.

Type	Only after Create()
Parameters	<cText> New text to add instead the selected text
Return value	NIL

1.6.1.10 TMenu

This class manages the Menu type objects from the Windows API. The Menu objects allow you to create the classic menus included in the main application windows in the upper part of the screen after the window title, used to select different tasks or options.

To create floating menus, you should use the **TPopupMenu** class that inherits directly from TMenu. However, this is the only difference to take into consideration when you use floating menus, due their properties and behavior are exactly the same. That's the reason why both classes have the same documentation.



Hierarchy	Inherits from TWinObject
File name	\source\Menu.prg

1.6.1.10.1 TMenu:Properties

■ read Only ■ Assignable ■ Design assignable ■ Run-time assignable

Scope	Name	Type	Initial value
■	altems	Array	{}
■	nStyle	Numeric	msDEFAULT
■	cMessage	Character	""
■	oImageList	Object	TImageList
■	oFont	Object	TFont

■	oParent	Object	NIL
---	---------	--------	-----

1.6.1.10.1.1 TMenu:altems

TMenuItem object array with the different menu elements.

Scope:	readOnly
Type:	Array
Initial value:	{}

This array is fed through the AddItem and AddSeparator methods.

1.6.1.10.1.2 TMenu:nStyle

Painting style.

Scope:	Assignable
Type:	Numeric
Initial value:	msDEFAULT
Possible values:	msDEFAULT, msOFFICEXP, msOFFICE2003, msOFFICE2007, msOFFICE2007EX

1.6.1.10.1.3 TMenu:cMessage

Text to be shown by the TStatusBar control from the same form where the menu belongs, when the cursor is located on it.

Scope:	Assignable
Type:	Character
Initial value:	""

1.6.1.10.1.4 TMenu:oImageList

TImageList object with all the images to be used by the control.

Scope:	Assignable
Type:	Object
Initial value:	TImageList

The TImageList is instantiated from the control creation with its New(0 constructor. The first image

that is included in the TImageList establishes the dimensions from the next images to be added. If the first image is an image that includes more than one bitmap, it is important to set the nHeight and nWidth properties from the TImageList before to add a bitmap.

1.6.1.10.1.5 TMenu:oParent

Control oParent object container. In this case, it is always a TForm object type.

Scope:	read Only
Type:	Object
Initial value:	NIL

This value is directly assigned in the Create method.

1.6.1.10.1.6 TMenu:oFont

TFont to be used for the text.

Scope:	Assignable
Type:	Object
Initial value:	NIL

Description:

This property allows to change the font to be used in the control when it is painted.

See also the TFont class for more information.

1.6.1.10.2 TMenu:Methods

■ Constructor ■ Standard ■ Only after Create()

Typ	Name
■	AddItem
■	AddSeparator
■	Create
■	Destroy
■	DoDefault
■	Refresh
■	SetMenu

1.6.1.10.2.1 TMenu:AddItem

Adds a new TMenuItem object to the TMenu object.

Type	Standard
Parameters	<p><cText> Text to be shown in the menu.</p> <p>[<cMessage>]: Message to show in the TStatusBar from its container. Default value: Blank.</p> <p>[<nImage>]: Image number to show in the olmageList object. Default: 0.</p> <p>[<IChecked>]: If .T. the menu element will show the marked status. Default: .F.</p> <p>[<IHilited>]: If it is .T. the menu element will be shown in bold. Default: .F.</p> <p>[<IDefault>]: If it is .T. the menu element will be the default itemd. Default: .F.</p> <p>[<IRight>]: If it is .T. the menu element will be shown aligned to the right. It is only possible in horizontal menus and when this is the last element from its container menu</p> <p>[<IDisabled>]: If it is .T. the menu element will be disabled. Default: .F.</p> <p>[<OnClick>]: Event to trigger when its selected the menu option.</p> <p>[<OnCheckState>]: Event that is triggered every time that the menu is displayed to establish if the menu option will be available. If this event returns a .F. value, it will be shown disabled.</p>
Return value	TMenuItem

1.6.1.10.2.2 TMenu:AddSeparator

Adds a new TMenuItem separator object type to the TMenu object.

Type	Standard
Parameters	None
Return value	TMenuItem

1.6.1.10.2.3 TMenu:Create

Creates a TMenu object with the passed parameters.

Type	Constructor
Parameters	<oParent> Menu container object form
Return value	Self

1.6.1.10.2.4 TMenu:Destroy

Destroys the object and releases the system resources.

Type	Standard
Parameters	None
Return value	NIL

1.6.1.10.2.5 TMenu:DoDefault

Executes the actions from the TmenuItem object marked by default through its IDefault property.

Type	Standard
Parameters	None
Return value	NIL

1.6.1.10.2.6 TMenu:Refresh

Redraws the menu. It must be called after adding or deleting menu items.

Type	Standard
Parameters	None
Return value	NIL

1.6.1.10.2.7 TMenu:SetMenu

Assigns the menu to its container form. It is needed to call this method when all the menus are already defined, to allow its container object to show them or update them.

Type	Standard
Parameters	None
Return value	NIL

Note: This method is only needed to use it in the TMenu objects type.

1.6.1.11 TMenuItem

This class allows to manage different existing elements in a TMenu or TPopupMenu control.

Hierarchy	Inherits from TWinObject
File name	\source\Menu.prg

1.6.1.11.1 TMenuItem:Properties

■ read Only ■ Assignable ■ Design assignable ■ Run-time assignable

Scope	Name	Type	Initial value
■	cMessage	Character	""
■	cText	Character	""
■	IChecked	Logic	.F.
■	IDefault	Logic	.F.
■	IEnabled	Logic	.T.
■	IHighlighted	Logic	.F.
■	IRight	Logic	.F.
■	ISeparator	Logic	.F.
■	nImage	Numeric	0
■	oSubMenu	Object	NIL

1.6.1.11.1.1 TMenuItem:cMessage

Text to be shown by the TStatusBar control from the same form where the menu is located when the mouse pointer is moved on it.

Scope:	Assignable
Type:	Character
Initial value:	""

1.6.1.11.1.2 TMenuItem:cText

Text to be shown in the control.

Scope:	Assignable
Type:	Character
Initial value:	""

1.6.1.11.1.3 TMenuItem:IChecked

If it is .T. the element will be shown with a check mark.

Scope:	Assignable
Type:	Logic
Initial value:	.F.

1.6.1.11.1.4 TMenuItem:IDefault

If it is .T. the menu element will be the default menu option. This property only makes sense when it is used with the DoDefault method from the TMenu class.

Scope:	Assignable
Type:	Logic
Initial value:	.F.

1.6.1.11.1.5 TMenuItem:IEnabled

If it is .F. the menu element will be shown disabled. You can use the OnCheckState event to disable the element in a more dynamic way.

Scope:	Assignable
Type:	Logic
Initial value:	.T.

1.6.1.11.1.6 TMenuItem:IHilited

If it is .T. the menu element will be shown in bold.

Scope:	Assignable
Type:	Logic
Initial value:	.F.

1.6.1.11.1.7 TMenuItem:IRight

If it is .T. the menu element will be shown aligned to the right.

This property only is effective when the menu is horizontal type -it is the firm form menu- and this is the last element from the menu.

Scope:	Assignable
Type:	Logic
Initial value:	.F.

1.6.1.11.1.8 TMenuItem:ISeparator

If it is .T. the menu element is a separator - a simple horizontal line.

Scope:	Assignable
Type:	Logic
Initial value:	.F.

1.6.1.11.1.9 TMenuItem:nImage

Image number from the oImageList of its main menu container to be shown in the menu element.

Scope:	Assignable
Type:	Numeric
Initial value:	0

1.6.1.11.1.10 TMenuItem:oSubMenu

TMenu object possible to show then the element is selected. This object is created automatically when the AddMenu method is executed.

Scope:	Assignable
Type:	Object
Initial value:	NIL

1.6.1.11.2 TMenuItem:Methods

■ Constructor ■ Standard ■ Only after Create()

Typ e	Name
■	AddMenu
■	Create
■	Destroy
■	GetMainMenu
■	oImageList

1.6.1.11.2.1 TMenuItem:AddMenu

Adds a submenu to the menu element.

Type	Standard
Parameters	[<oSubMenu>] Possible TMenu existing object. By default it will create a new TMenu
Return value	TMenu object

1.6.1.11.2.2 TMenuItem:Create

Creates an element in the menu with the parameters passed.

Type	Constructor
Parameters	<oParent> TMenu or TPopupMenu container object
Return value	Self

1.6.1.11.2.3 TMenuItem:Destroy

Destroys the object and releases the system resources.

Type	Standard
Parameters	None
Return value	NIL

1.6.1.11.2.4 TMenuItem:GetMainMenu

Returns a referent to the main TMenu object in the case of multiple nested menus using the oSubMenu property from their own TMenuItem.

Type	Standard
Parameters	None
Return value	TMenu object

1.6.1.11.2.5 TMenuItem:olmageList

Returns a reference to the TImagelist object form the main TMenu.

Type	Standard
Parameters	None
Return value	TImagelist object

1.6.1.11.3 TMenuItem:Events

Name
OnCheckState
OnClick

1.6.1.11.3.1 TMenuItem:OnCheckState

Event that is produced when is displayed the menu that allows to establish the element enable status.

Parameters	<oSender>:
:	Object that triggers the event

Return value: <IValue>
 If returns .F. the menu element will be shown disabled, if returns .T. the menu element will be shown enabled

1.6.1.11.3.2 TMenuItem:OnClick

Event that is produced when an element is selected.

Parameters : <oSender>
 Object that triggers the event
 <oMenu>
 Its TMenu container

Return value: NIL

1.6.1.12 TRadio

This class represents a 'Radio' Windows control type.



Hierarchy Inherits from TStdControl
See also TRadioMenu
File name \source\Radio.prg

1.6.1.12.1 TRadio:Properties

■ read Only ■ Assignable ■ Design assignable ■ Run-time assignable

Scope	Name	Type	Initial value
■	IChecked	Logic	.F.
■	IGroup	Logic	.F.
■	IMultiLine	Logic	.F.
■	IPushLike	Logic	.T.
■	ITransparent	Logic	.F.
■	nAlignment	Numeric	taRIGHT
■	nHeight	Numeric	18
■	nVAlignment	Numeric	vaCENTER
■	nWidth	Numeric	90

1.6.1.12.1.1 TRadio:IChecked

Indicates or establishes the control status.

Scope:	Assignable
Type:	Logic
Initial value:	.F.

1.6.1.12.1.2 TRadio:IGroup

Establishes the control as the first element of a 'Radio' control group.

Scope:	Design alignment
Type:	Logic
Initial value:	.F.

1.6.1.12.1.3 TRadio:IMultiLine

If it is .T. the control will be shown in more than one line.

Scope:	Design alignment
Type:	Logic
Initial value:	.F.

1.6.1.12.1.4 TRadio:IPushLike

If it is .T the control will show a button 'look and feel'.

Scope:	Design alignment
Type:	Logic
Initial value:	.F.

1.6.1.12.1.5 TRadio:ITransparent

If it is .T. the control background will be transparent. In that case the nClrPane property has no effect.

Scope:	Design alignment
Type:	Logic

Initial value:	.T. .
-----------------------	-------

1.6.1.12.1.6 TRadio:nAlignment

Indicates the horizontal button text alignment.

Scope:	Design alignment
Type:	Numeric
Initial value:	taCENTER
Possible values:	taLEFT, taCENTER, taRIGHT

1.6.1.12.1.7 TRadio:nHeight

Indicate the button height.

Scope:	Assignable
Type:	Numeric
Initial value:	25

1.6.1.12.1.8 TRadio:nVAlignment

Indicates the vertical button text alignment.

Scope:	Design alignment
Type:	Numeric
Initial value:	vaCENTER
Possible values:	vaTOP, vaCENTER, vaBOTTOM

1.6.1.12.1.9 TRadio:nWidth

Indicates the button width.

Scope:	Assignable
Type:	Numeric
Initial value:	80

1.6.1.12.2 TRadio:Methods

Constructor
 Standard
 Only after Create()

Typ	Name
<input checked="" type="checkbox"/>	Click

1.6.1.12.2.1 TRadio:Click

Changes the control status.

Type	Only after Create()
Parameters	None
Return value	The value returned by OnClick

1.6.1.12.3 TRadio:Events

Name
OnChange
OnClick

1.6.1.12.3.1 TRadio:OnChange

Event that is produced when the button changes its state.

Parameters	<oSender> :
:	Reference to the object that triggers the event
Return value:	NIL

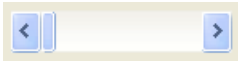
1.6.1.12.3.2 TRadio:OnClick

Event that is produced when the user clicks the mouse pointer on the button.

Parameters	<oSender> :
:	Reference to the object that triggers the event
Return value:	NIL

1.6.1.13 TScrollBar

This class represents a Windows scroll bar control.



Hierarchy Inherits from TStdControl
See also TTrackBar
File name \source\ScrollBar.prg

1.6.1.13.1 TScrollBar:Properties

■ read Only ■ Assignable ■ Design assignable ■ Run-time assignable

Scope	Name	Type	Initial value
■	IDisableNoScroll	Logic	.F.
■	nClrPane	Numeric	clScrollBar
■	nHeight	Numeric	20
■	nMax	Numeric	10
■	nMin	Numeric	0
■	nOrientation	Numeric	orHORIZONTAL
■	nPage	Numeric	1
■	nPos	Numeric	0
■	nWidth	Numeric	100

1.6.1.13.1.1 TScrollBar:IDisableNoScroll

If it is .T., the scroll bar will be visible even when the ranges or position are not valid. In that case, the scroll bar will be shown as disabled.

Scope:	Assignable
Type:	Logic
Initial value:	.T.

1.6.1.13.1.2 TScrollBar:nClrPane

Indicates the control background color.

Scope:	Assignable
Type:	Numeric
Initial value:	clScrollBar

(See the appendix to check the colors available)

1.6.1.13.1.3 TScrollBar:nHeight

Indicates the control height.

Scope:	Assignable
Type:	Numeric
Initial value:	100

1.6.1.13.1.4 TScrollBar:nMax

Indicates the maximum scroll value.

Scope:	Assignable
Type:	Numeric
Initial value:	10

1.6.1.13.1.5 TScrollBar:nMin

Indicates the minimum scroll value..

Scope:	Assignable
Type:	Numeric
Initial value:	0

1.6.1.13.1.6 TScrollBar:nOrientation

Indicates the scroll bar orientation.

Scope:	Assignable
Type:	Numeric
Initial value:	orHORIZONTAL
Possible values:	orHORIZONTAL, orVERTICAL

1.6.1.13.1.7 TScrollBar:nPage

Indicates the scroll value when the indicator is moved from page to page.

Scope:	Assignable
Type:	Numeric
Initial value:	1

1.6.1.13.1.8 TScrollBar:nPos

Indicates the position or value from the scroll bar.

Scope:	Assignable
Type:	Numeric
Initial value:	0

1.6.1.13.1.9 TScrollBar:nWidth

Indicates the control width.

Scope:	Assignable
Type:	Numeric
Initial value:	100

1.6.1.13.2 TScrollBar:Methods

■ Constructor ■ Standard ■ Only after Create()

Typ	Name
■	Dec
■	Inc

1.6.1.13.2.1 TScrollBar:Dec

Decreases the value for the control indicator.

Type	Only after Create()
Parameters	<nValue> Number of positions to decrease
Return value	NIL

1.6.1.13.2 TScrollBar:Inc

Increases the value for the control indicator.

Type	Only after Create()
Parameters	<nValue> Number of positions to increase
Return value	NIL

1.6.1.13.3 TScrollBar:Events

Name	
	OnChange
	OnDown
	OnPageDown
	OnPageUp
	OnThumbPos
	OnThumTrack
	OnUp

1.6.1.13.3.1 TScrollBar:OnChange

Event that is produced when the indicator changes its position.

Parameters	<oSender>: Reference to the object that triggers the event
:	<nPos>: New indicator position
	<nOldPos>: Old indicator position
Return value:	<IOk>: If .F. the Scroll bar position will not change

1.6.1.13.3.2 TScrollBar:OnDown

Event that is activated when the indicator is moved down.

Parameters	<oSender>: Reference to the object that triggers the event
:	<nPos>:

	New indicator position
Return value:	<nStep>: Scroll value. If it is NIL, it assumes 1

1.6.1.13.3.3 TScrollBar:OnPageDown

Event that is activated when the indicator is moved one page down.

Parameters	<oSender>: Reference to the object that triggers the event
:	<nPage>: New indicator position
Return value:	<nStep>: Scroll value. If it is NIL, it assumes the nPage value

1.6.1.13.3.4 TScrollBar:OnPageUp

Event that is activated when the indicator is moved one page up.

Parameters	<oSender>: Reference to the object that triggers the event
:	<nPage>: New indicator position
Return value:	<nStep>: Scroll value

1.6.1.13.3.5 TScrollBar:OnThumbPos

Event that is activated when the user stops to drag the indicator.

Parameters	<oSender>: Reference to the object that triggers the event
:	<nTrackPos>: New indicator position
Return value:	<nPos>: Scroll value. If it is NIL, it assigns the default value

1.6.1.13.3.6 TScrollBar:OnThumbTrack

Event that is activate when the indicator is dragged.

Parameters	<oSender> : Reference to the object that triggers the event
:	<nTrackPos> : New indicator position
Return value:	<nPos> : Scroll value. If it is NIL, it assigns the default value

1.6.1.13.3.7 TScrollBar:OnUp

Event that is activate when the indicator is moved up.

Parameters	<oSender> : Reference to the object that triggers the event
:	<nPos> : New indicator position
Return value:	<nStep> : Scroll value. If it is NIL, it assumes 1

1.6.2 Additional

1.6.2.1 TAnimatedGif

This class represents a control that displays an animated GIF image.

The animation is performed entirely in a second thread, without requiring any 'Timer'. For this reason, it is not necessary to call *ProcessMessages()* in any case.

Hierarchy TControl descendant
File \source\AnimatedGif.prg

1.6.2.1.1 TAnimatedGif:Properties

■ Read only ■ Assignable ■ Design assignable ■ Run-time assignable

Scope	Name	Type	Initial value
■	IPlay	Logical	.T.
■	ITransparent	Logical	.T.
■	nBorderStyle	Numeric	bvNONE
■	oPicture	Object	NIL

1.6.2.1.1.1 TAnimatedGif:IPlay

The control animation starts or stops.

Scope	Assignable
Type	Lógico
Initial value	.T.

1.6.2.1.1.2 TAnimatedGif:ITransparent

The control is drawn transparent on the form to which it belongs.

For this property to work correctly the GIF must have a transparent background. When it has a transparent background and **ITransparent** is set to false, the background will be painted with the color indicated in its **nClrPane** property.

GIF transparent	::ITransparent	Painting effect
No	No	It is painted completely opaque, and the parts of the control that do not cover the image are painted with the nClrPane color.
Yes	No	It is painted transparent on an nClrPane colored background.
Yes	Yes	Everything is painted transparent

Scope	Assignable
Type	Logical
Initial value	.T.

1.6.2.1.1.3 TAnimatedGif:nBorderStyle

Style with which the border is drawn.

Scope	Assignable
Type	Numeric
Initial value	bvNONE
Possible values	bvNONE, bvRAISED, bvSUNKEN, bvBUMP, bvETCHED, bvFLAT

1.6.2.1.1.4 TAnimatedGif:Picture

TPicture object with the image to be displayed on the control.

Scope	Assignable
Type	Object
Initial value	Nil

1.6.2.2 TBevel

This class represents a Bevel control, or frame, with two different line styles, that might contain other controls.

It supports a transparent mode, and allows to see the background image from the control or form that contains it, in Windows XP and Windows 98. It might have a background image.

Hierarchy	Inherits from TWinControl
See also	TPanel
File name	\source\Bevel.prg

1.6.2.2.1 TBevel:Properties

■ read Only ■ Assignable ■ Design assignable ■ Run-time assignable

Scope	Name	Type	Initial value
■	IOwnBorder	Logic	.F.
■	ITransparent	Logic	.F.
■	nBorderStyle	Numeric	bvETCHED
■	nHeigth	Numeric	100
■	nOpacity	Numeric	0
■	nPenColor	Numeric	clLightGray
■	nPenSize	Numeric	2
■	nPenStyle	Numeric	psSolid
■	nWidth	Numeric	100

1.6.2.2.1.1 TBevel:IOwnBorder

The control has its own border painting routine based on the properties: nPenColor, nPenSize and nPenStyle.

Scope	Assignable
Type	Logic
Initial value	.F.

1.6.2.2.1.2 TBevel:ITransparent

The control is painted transparent on the form it belongs to.

Scope	Assignable
Type	Logic
Initial value	.F.

1.6.2.2.1.3 TBevel:nBorderStyle

Indicates the border style.

Scope	Assignable
Type	Numeric
Initial value	bvETCHED
Possible values	bvNONE, bvRAISED, bvSUNKEN, bvBUMP, bvETCHED, bvFLAT

1.6.2.2.1.4 TBevel:nHeight

Indicates the control height.

Scope	Assignable
Type	Numeric
Initial value	100

1.6.2.2.1.5 TBevel:nOpacity

Control opacity in percentage when its property ITransparent is set to true.

Scope:	Assignable
Type:	Numeric
Initial value:	0
Possible values:	0-100

1.6.2.2.1.6 TBevel:nPenColor

The border color when the property IOwnBorder is set to .T.

Scope	Assignable
Type	Numeric
Initial value	clLightGray

Consult the appendix for the list of available colors

1.6.2.2.1.7 TBevel:nPenSize

The border size in pixels when the property IOwnBorder is set to .T.

Scope	Assignable
Type	Numeric
Initial value	2

1.6.2.2.1.8 TBevel:nPenStyle

The border styles when the property IOwnBorder is set to .T.

Scope	Assignable
Type	Numeric
Initial value	psSolid
Possible values	psSolid, psDash, psDot, psDashDot, psDashDotDot, psNull, psInsideFrame

1.6.2.2.1.9 TBevel:nWidth

Indicates the control width.

Scope	Assignable
Type	Numeric
Initial value	100

1.6.2.3 TBtnBmp

This class represents a Windows Button control, but with the possibility to show an image on it.



Hierarchy Inherits from TButton
See also TButton
File name \source\BtnBmp.prg

1.6.2.3.1 TBtnBmp:Properties

■ read Only ■ Assignable ■ Design assignable ■ Run-time assignable

Scope	Name	Type	Initial value
■	IFlat	Logic	.F.
■	IXPLook	Logic	.T.
■	nBmpHeight	Numeric	1
■	nBmpWidth	Numeric	1
■	nFlatStyle	Numeric	fsNONE
■	nMargin	Numeric	-1
■	nOrientation	Numeric	orLEFT
■	nWidth	Numeric	90
■	oBitmaps	Object	NIL

1.6.2.3.1.1 TBtnBmp:IFlat

If it is .T. the button will be shown as a flat button.

Scope:	Design assignable
Type:	Logic
Initial value:	.F.

1.6.2.3.1.2 TBtnBmp:IXPLook

If it is .T. the button will be shown with XP style themes.

Scope:	Design assignable
Type:	Logic
Initial value:	.T.

1.6.2.3.1.3 TBtnBmp:nBmpHeight

Indicates the bitmap height.

Scope:	Design assignable
Type:	Numeric
Initial value:	1

This property indicates the bitmap height that will be shown in the button. The default value 1 indicates that it will use the real height from the first bitmap indicated in the property oBitmaps. If indicates a lower height than the real bitmap height, the image will be shown incomplete.

1.6.2.3.1.4 TBtnBmp:nBmpWidth

Indicates the bitmap width.

Scope:	Design assignable
Type:	Numeric
Initial value:	1

This property indicates the bitmap width that will be shown in the button. The default value 1 indicates that it will use the real width from the first bitmap indicated in the oBitmaps property. If indicates a lower width than the real bitmap width, the image will be shown incomplete.

1.6.2.3.1.5 TBtnBmp:nFlatStyle

Sets the bitmap style when the property IFlat is set to true.

Scope:	Design assignable
Type:	Numeric
Initial value:	fsNONE
Possible values:	fsNONE, fsRECTANGLE, fsRAISED

See also the property IFlat.

1.6.2.3.1.6 TBtnBmp:nMargin

Indicates the space between the button and the image.

Scope:	Assignable
Type:	Numeric

Initial value: -1

The -1 value indicates that the image will be established automatically.

1.6.2.3.1.7 TBtnBmp:nOrientation

Indicates the bitmap orientation in relation to the text.

Scope:	Assignable
Type:	Numeric
Initial value:	orLEFT
Possible values:	orTOP, orBOTTOM, orLEFT, orRIGHT

1.6.2.3.1.8 TBtnBmp:nWidth

Indicates the button width.

Scope:	Assignable
Type:	Numeric
Initial value:	90

1.6.2.3.1.9 TBtnBmp:oBitmaps

TImageList object with all the possible images available for the button.

Scope:	Assignable
Type:	Object
Initial value:	90

The TBtnBmp button can show 4 different images depending of its state. They can be:

- Normal, it corresponds to the first TImageList image.
- With the mouse pointer over it, that corresponds to the second TImageList image.
- Pressed, and corresponds to the third TImageList image.
- Disable, that corresponds to the fourth TImageList image.

The button will use the first bitmap for all the states where the image has not been defined.

There are several ways to assign this property:

- Through a previously created TImageList object.
- Through a literal with the resource name of Bitmap type file.
- Through an 1, 2, 3 or 4 elements array with the resource names or the bitmap type files for

every possible button state.

For any of the last 2 cases the button will create a TImageList with the information provided.

1.6.2.4 TColorComboBox

This class represents a Windows ComboBox type specialized in the color selection.



Hierarchy Inherits from TComboBox
File name \source\ColorComboBox.prg

1.6.2.4.1 TColorComboBox:Properties

■ read Only ■ Assignable ■ Design assignable ■ Run-time assignable

Scope	Name	Type	Initial value
■	aColors	Array	{}
■	IAddColor	Logic	.T.
■	IShowText	Logic	.F.
■	nColor	Numeric	16
■	nColorBoxSize	Numeric	clBlack
■	nLineHeight	Numeric	16

1.6.2.4.1.1 TColorComboBox:aColors

List of colors to be displayed by the control.

Scope:	Assignable
Type:	Array
Initial value:	{}

By default, it will show the 16 basic colors:

CLR_WHITE, CLR_YELLOW, CLR_HMAGENTA, CLR_HRED, CLR_HCYAN, CLR_HGREEN,
 CLR_HBLUE, CLR_GRAY, CLR_HGRAY, CLR_BROWN, CLR_MAGENTA, CLR_RED,
 CLR_CYAN, CLR_GREEN, CLR_BLUE, CLR_BLACK

1.6.2.4.1.2 TColorComboBox:IAddColor

If it is .T., when a new color is selected on the property nColor is automatically included on aColors array.

Scope:	Assignable
Type:	Logic
Initial value:	.T.

1.6.2.4.1.3 TColorComboBox:IShowText

If it is .T., it will show together with the color, its value in the RGB(0-255, 0-255, 0-255) format.

Scope:	Assignable
Type:	Logic
Initial value:	.F.

1.6.2.4.1.4 TColorComboBox:nColorBoxSize

Indicates the square size where the color is shown.

Scope:	Assignable
Type:	Numeric
Initial value:	16

1.6.2.4.1.5 TColorComboBox:nColor

Indicates the current selected color.

Scope:	Assignable
Type:	Numeric
Initial value:	clBlack

1.6.2.4.1.6 TColorComboBox:nLineHeight

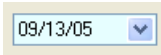
Indicates the size in pixels of every control line.

Scope:	Assignable
Type:	Numeric

Initial value: 16

1.6.2.5 TDateEdit

This class represents a TMaskEdit Windows control specialized to edit dates. The main difference with the TDatePicker control is that allows to manage blank dates without the need of an additional checkbox.



Hierarchy Inherits from TMaskEdit
File name \source\DateEdit.prg

1.6.2.5.1 TDateEdit:Propiedades

■ read Only ■ Assignable ■ Design assignable ■ Run-time Assignable

Scope	Name	Type	Initial value
■	nWidth	Numeric	82
■	Value	Date	Ctod("")

1.6.2.5.1.1 TDateEdit:nWidth

Indicates the control width specified in pixels.

Scope:	Assignable
Type:	Numeric
Initial value:	82

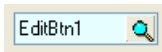
1.6.2.5.1.2 TDateEdit:Value

Indicates the control's date value

Scope:	Assignable
Type:	Date
Initial value:	Dtoc("")

1.6.2.6 TEditBtn

This class represents a TEdit control that may include a button.



Hierarchy Inherits from TEdit
File name \source\EditBtn.prg

1.6.2.6.1 TEditBtn:Properties

■ read Only ■ Assignable ■ Design assignable ■ Run-time assignable

Scope	Name	Type	Initial value
■	cPrompt	String	""
■	nStyle	Numeric	ebBITMAP
■	oBitmap	Object	NIL

1.6.2.6.1.1 TEditBtn:cPrompt

String to show when nStyle property is set to ebPROMPT.

Scope:	Assignable
Type:	String
Initial value:	""

1.6.2.6.1.2 TEditBtn:nStyle

Button style. By default shows the image oBitmap.

Scope:	Assignable
Type:	Numeric
Initial value:	ebBITMAP
Possible values:	ebBITMAP, ebDOTS, ebCOMBO, ebARROW, ebPROMPT

1.6.2.6.1.3 TEditBtn:oBitmap

TBitmap object type with the image that will be shown in the button.

Scope:	Assignable
---------------	------------

Type:	Object
Initial value:	NIL

1.6.2.6.2 TEditBtn:Events

Name
OnBtnClick

1.6.2.6.2.1 TEditBtn:OnBtnClick

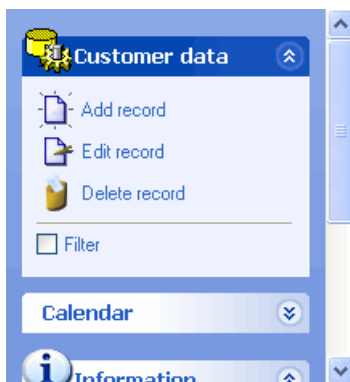
Event that is produced then the control changes its value.

Parameters	<oSender>: Reference to the objet that triggers the event
<Value>:	Control value passed by reference
Return value:	If it is not NIL, the Value control property is updated with the returned value

1.6.2.7 TExplorerBar

This class represents a explorer control bar. Its behavior is similar to a button bar, but it has more functionality because it allows to create groups and the possibility to collapse or expand any of them. Every TExplorerBar group is an object from the TExplorerGroup class that might contain any Xailer control. The control visual aspect when it is used with Windows XP themes, is much better than the classic button bar.

This control is commonly used in the Windows Explorer.



Hierarchy	Inherits from TScrollingWinControl
See also	TExplorerGroup
File name	\source\ExplorerBar.prg

1.6.2.7.1 TExplorerBar:Properties

■ read Only ■ Assignable ■ Design assignable ■ Run-time assignable

Scope	Name	Type	Initial value
■	altems	Array	{}
■	ITransparent	Logical	.F.
■	nAlign	Numeric	alLEFT
■	nClrPane	Numeric	clWindow
■	nOpacity	Numeric	0
■	nSpacing	Numeric	10
■	nWidth	Numeric	200
■	oPicture	Object	NIL

1.6.2.7.1.1 TExplorerBar:altems

List of TExplorerGroup object with all the groups to be displayed by the control. Every TExplorerGroup control might contain any Xailer control and not only buttons.

Scope:	Design Assignable
Type:	Array
Initial value:	{}

It is not common that you need to manipulate this property directly.

1.6.2.7.1.2 TExplorerBar:ITransparent

If true the background is transparent

Scope:	Assignable
Type:	Logical
Initial value:	.F.

1.6.2.7.1.3 TExplorerBar:nAlign

Indicates the control alignment in its oParent container object.

Scope:	Design Assignable
Type:	Numeric
Initial value:	alLEFT
Possible values:	alNONE, alLEFT, alTOP, alRIGHT, alBOTTOM, alCLIENT

Description:

This property allows to the controls to adjust its dimensions to the position of its oParent container object. The alignment can be:

- **None:** Default value
- **Left:** The control is aligned to the left of its oParent object and takes it height from the client area of its container object.
- **Top:** The control is aligned to the top of its oParent object and takes its width from the client area of its container object.
- **Right:** The control is aligned to the right of its oParent object and takes it height from the client area of its container object.
- **Bottom:** The control is aligned to the bottom of its oParent object and takes its width from the client area of its container object.
- **Client:** The control is aligned to all the client area of its oParent object adjusting its size to it.

1.6.2.7.1.4 TExplorerBar:nClrPane

Indicates the background color.

Scope:	Assignable
Type:	Numeric
Initial value:	clWindow

(See also the appendix to check the colors available)

1.6.2.7.1.5 TExplorerBar:nOpacity

Control opacity in percentage when its property ITransparent is set to true.

Scope:	Assignable
Type:	Numeric
Initial value:	0
Possible values:	0-100

1.6.2.7.1.6 TExplorerBar:nSpacing

Indicates the space among TExplorerGroup elements.

Scope:	Assignable
Type:	Numeric
Initial value:	10

1.6.2.7.1.7 TExplorerBar:nWidth

Indicates the control width.

Scope:	Assignable
Type:	Numeric
Initial value:	200

1.6.2.7.1.8 TExplorerBar:oPicture

Picture to show on the bottom right corner of the control.

Scope:	Assignable
Type:	Object
Initial value:	NIL

1.6.2.7.2 TExplorerBar:Methods

■ Constructor ■ Standard ■ Only after Create()

Typ	Name
■	AddItem
■	HitTest

1.6.2.7.2.1 TExplorerBar:AddItem

Adds a new TExplorerGroup group to the control.

Type	Standard
Parameters	<p><cltem>: Group description that will be shown as title</p> <p>[<olcon>]: TIcon object that will be shown in the group title. Default: no icon</p> <p>[<ISpecial>]: If it is .T. the group will show with a different color. Default: .F.</p> <p>[<IExpanded>]: If it is .T. the group will be shown expanded. Default: .T.</p> <p>[<IVisible>]: If it is .F. the group will not be shown. Default: .T.</p>
Return value	<oltem>: New TExplorerGroup created object

1.6.2.7.2 TExplorerBar:HitTest

Returns the TExplorerGroup object from an specific coordinate in the TExplorerBar control.

Type	Standard
Parameters	<nX> X Coordinate <nY> Y coordinate
Return value	TExplorerGroup object or NIL

1.6.2.7.3 TExplorerBar:Eventos

Name	OnItemExpand
-------------	--------------

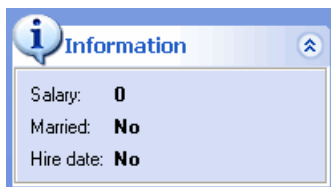
1.6.2.7.3.1 TExplorerBar:OnItemExpand

Evento fired when any ExplorerBar item is expanded.

Parameters	<oSender>: : Object that triggers the event. <oItem>: Item to be expanded.
Return value:	<IOk> NIL Only if it returns FALSE the expansions is canceled

1.6.2.8 TExplorerGroup

This class represents the different groups contained in a TExplorerBar.



Hierarchy	Inherits from TComponent
See also	TExplorerBar
File name	\source\ExplorerGroup.prg

1.6.2.8.1 TExplorerGroup:Properties

read Only
 Assignable
 Design assignable
 Run-time assignable

Scope	Name	Type	Initial value
<input type="checkbox"/>	cText	Character	""
<input type="checkbox"/>	IExpanded	Logic	.T.
<input type="checkbox"/>	ISpecial	Logic	.F.
<input type="checkbox"/>	IVisible	Logic	.T.
<input type="checkbox"/>	nItem	Numeric	0
<input type="checkbox"/>	oControl	Object	NIL
<input type="checkbox"/>	oIcon	Object	NIL
<input type="checkbox"/>	oParent	Object	NIL

1.6.2.8.1.1 TExplorerGroup:cText

Text to be shown as the ExplorerGroup title.

Scope	Assignable
Type	Character
Initial value	""

1.6.2.8.1.2 TExplorerGroup:IExpanded

If it is .T. the ExplorerGroup will be shown expanded.

Scope	Assignable
Type	Logic
Initial value	.T.

1.6.2.8.1.3 TExplorerGroup:ISpecial

If it is .T. the ExplorerGroup will be shown with a highlighted color.

Scope	Assignable
Type	Logic
Initial value	.F.

1.6.2.8.1.4 TExplorerGroup:IVisible

If it is .T. the ExplorerGroup will be visible.

Scope	Assignable
Type	Logic
Initial value	.T.

1.6.2.8.1.5 TExplorerGroup:nItem

Indicates the ExplorerGroup box creation order in its TExplorerBar container. This value corresponds to the `altems` array element from the TExplorerBar.

Scope	read Only
Type	Numeric
Initial value	0

1.6.2.8.1.6 TExplorerGroup:oControl

Control that is shown inside the ExplorerGroup. Only one control can be inside the ExplorerGroup, however, this control can be a control that admits more controls on it, like the TBevel control.

Scope	Assignable
Type	Object
Initial value	NIL

1.6.2.8.1.7 TExplorerGroup:olcon

Icon to be shown in the ExplorerGroup title.

Scope	Assignable
Type	Object
Initial value	NIL

1.6.2.8.1.8 TExplorerGroup:oParent

Reference to its TExplorerBar container object.

Scope	read Only
--------------	-----------

Type	Object
Initial value	NIL

1.6.2.8.2 TExplorerGroup:Methods

■ Constructor ■ Standard

Typ Name
■ Create
■ Refresh

1.6.2.8.2.1 TExplorerGroup:Create

Class constructor.

Type	Constructor
Parameters	<oParent> Control container TExplorerBar object type
Return value	<oItem> Reference to the new created ExplorerGroup

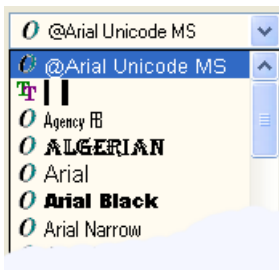
1.6.2.8.2.2 TExplorerGroup:Refresh

Refreshes the control.

Type	Standard
Parameters	None
Return value	NIL

1.6.2.9 TFontComboBox

This class represents a Windows ComboBox control type specialized in the font selection.



Hierarchy Inherits from TComboBox
File name \source\FontComboBox.prg

1.6.2.9.1 TFontComboBox:Properties

■ read Only ■ Assignable ■ Design assignable ■ Run-time assignable

Scope	Name	Type	Initial value
■	cSampleText	Character	"AbCdEfGhIj"
■	IComboFont	Logic	.T.
■	IEditBmp	Numeric	.F.
■	IFixedPitch	Numeric	.F.
■	nCharSet	Numeric	csDEFAULT
■	nFontHeight	Numeric	17
■	nFontType	Numeric	ftALL

1.6.2.9.1.1 TFontComboBox:cSampleText

Sample text to show on the combobox.

Scope:	Assignable
Type:	Character
Initial value:	"AbCdEfGhIj"

1.6.2.9.1.2 TFontComboBox:IComboFont

Enables/Disables the font type preview in every ComboBox element.

Scope:	Assignable
Type:	Logic
Initial value:	.T.

1.6.2.9.1.3 TFontComboBox:IEditBmp

Enables/Disables the bitmap use in the edit control.

Scope:	Assignable
Type:	Logic
Initial value:	.F.

1.6.2.9.1.4 TFontComboBox:IFixedPitch

If it is .T., it will show only the no proportional fonts.

Scope:	Design assignable
Type:	Logic
Initial value:	.F.

1.6.2.9.1.5 TFontComboBox:nFontHeight

Establishes the height of every ComboBox element.

Scope:	Assignable
Type:	Numeric
Initial value:	17

1.6.2.9.1.6 TFontComboBox:nCharSet

Selects the font char set. The most common values are:

- DEFAULT_CHARSET
- ANSI_CHARSET
- OEM_CHARSET
- SYMBOL_CHARSET

Scope:	Design assignable
Type:	Numeric
Initial value:	csDEFAULT
Possible values:	csANSI, csDEFAULT, csSYMBOL, csSHIFTJIS, csGB2312, csHANGEUL, csCHINESEBIG5, csOEM

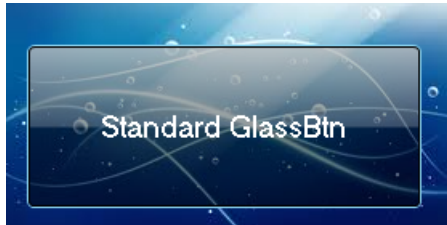
1.6.2.9.1.7 TFontComboBox:nFontType

Indicates the font type.

Scope:	Design assignable
Type:	Numeric
Initial value:	ftALL
Possible values:	ftDEVICE, ftRASTER, ftTRUETYPE, ftDEVICERASTER, ftDEVICETRUEYPE, ftRASTERTRUETYPE, ftALL

1.6.2.10 TGlassBtn

This class represents a Button control with a glass effect.



Hierarchy	TBtnBmp descendant
See also	TButton , TBtnBmp
File	\source\GlassBtn.prg
Sample	\samples\GlassBtn

1.6.2.10.1 TGlassBtn:Properties

■ Read only ■ Assignable ■ Design assignable ■ Run-time assignable

Scope	Name	Type	Initial Value
■	nClrGlow	Numeric	clSkyBlue
■	nClrInnerBorder	Numeric	clBlack
■	nClrOuterBorder	Numeric	clWhite
■	nClrPane	Numeric	clBlack
■	nClrShine	Numeric	clWhite
■	nClrText	Numeric	clWhite
■	nHeight	Numeric	32
■	nWidth	Numeric	120

1.6.2.10.1.1 TGlassBtn:nClrGlow

Brightness color that will be shown on the bottom half and border of the button.

Scope:	Asignable
Type:	Numeric
Initial Value:	clSkyBlue

1.6.2.10.1.2 TGlassBtn:nClrInnerBorder

Inner border color.

Scope:	Assignable
Type:	Numeric
Initial Value:	clBlack

1.6.2.10.1.3 TGlassBtn:nClrOuterBorder

Outer border color.

Scope:	Assignable
Type:	Numeric
Initial Value:	clWhite

1.6.2.10.1.4 TGlassBtn:nClrPane

Button background color.

Scope:	Assignable
Type:	Numeric
Initial Value:	clBlack

1.6.2.10.1.5 TGlassBtn:nClrShine

Brightness color that will be shown on the button top half.

Scope:	Assignable
Type:	Numeric
Initial Value:	clWhite

1.6.2.10.1.6 TGlassBtn:nClrText

Button text color.

Scope:	Asignable
Type:	Numeric
Initial Value:	clBlack

1.6.2.10.1.7 TGlassBtn:nHeight

Button height.

Scope:	Asignable
Type:	Numeric
Initial Value:	32

1.6.2.10.1.8 TGlassBtn:nWidth

Button width.

Scope:	Asignable
Type:	Numeric
Initial Value:	120

1.6.2.11 THyperLink

This class represents an HyperLink control.

[HyperLink1](#)

Description:

An hyperlink is a text label that is show with the same look and feel than the common web links, and when the user clicks on it, the link content is open.

Hierarchy	Inherits from TLabel
File name	\source\HyperLink.prg

1.6.2.11.1 THyperLink:Properties

■ read Only
 ■ Assignable
 ■ Design assignable
 ■ Run-time assignable

Scope	Name	Type	Initial value
■	cURL	Character	""
■	nClrOver	Numeric	clRed
■	nClrText	Numeric	clBlue
■	IUnderline	Logic	.T.

1.6.2.11.1.1 THyperLink:cURL

Indicates a web address. For example: '<http://www.xailer.com>'

Scope	Assignable
Type	Character
Initial value	""

1.6.2.11.1.2 THyperLink:nClrOver

Text color when the users moves the mouse pointer on the control.

Scope	Assignable
Type	Numeric
Initial value	clBlue

Consult the appendix for the list of available colors

1.6.2.11.1.3 THyperLink:nClrText

Indicates the text color.

Scope	Assignable
Type	Numeric
Initial value	clRed

Consult the appendix for the list of available colors

1.6.2.11.1.4 THyperLink:IUnderline

If it is .T., the control text will be shown underlined.

Scope	Assignable
--------------	------------

Type	Logic
Initial value	.T.

1.6.2.11.2 THyperLink:Events

Name	
OnClick	
OnRClick	

1.6.2.11.2.1 THyperLink:OnClick

When the user clicks on the text, the event is triggered, if it is assigned. Otherwise, it will open automatically the link that the cURL property is pointing.

Parameters	<oSender> Reference to te object that triggers the event <nFlags> Key combination pressed when the user clicked the right mouse button <nPosX> Mouse pointer X coordinate <nPosY> Mouse pointer Y coordinate
Return value	NIL

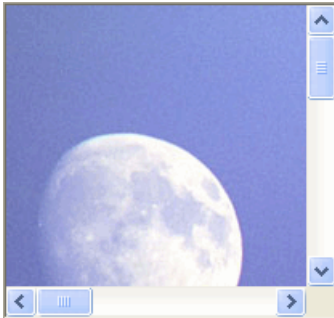
1.6.2.11.2.2 THyperLink:OnRClick

Event that is produced when the user right-clicks the mouse.

Parameters	<oSender> Reference to te object that triggers the event <nFlags> Key combination pressed when the user clicked the right mouse button <nPosX> Mouse pointer X coordinate <nPosY> Mouse pointer Y coordinate
Return value	NIL

1.6.2.12 TImage

This class represents an Image control. It is a control that shows and image. It supports several image formats: BMP, ICO, WMF, EMF, JPEG, GIF. Since Windows XP or higher it also natively supports the formats PNG and TIFF.



If the free library [FreeImage.DLL](#) is available on your system (application directory, Windows directory or Windows\System32 directory) it will also support the following formats: DDS, Dr. Halo, HDR, IFF, FBIG, JNG, KOALA, LBM, Kodak Photo CD, MNG, PCX, PBM, PGM, PNG, RPM, PhotoShop, Raw Fax G3, SGI, Sun RAS, TARGA, TIFF, WBMP, XBM and XPM.

When using FreeImage in your open source or commercial application, you are REQUIRED to :

- distribute the license ([GNU GPL](#) or [FIPL](#)) you choosed with your application (i.e. the TXT file)
- provide a suitable acknowledgement, either in the program's "About" box or in the user's manual (or both), for example :

This software uses the FreeImage open source image library. See <http://freeimage.sourceforge.net> for details.

FreeImage is used under the (GNU GPL or FIPL), version (licence version).

The TImage control does not support stretching (the image is adjusted to the control dimensions) directly. To get this effect you need to use a TBevel control type and you need to indicate in its oBkGnd property the background image establishing the nBkGndMode property to biSTRETCH.

Note: The GIF format is patented, so if you want to use it, you should pay to Unisys the corresponding license if applicable. More information in http://www.unisys.com/about__unisys/lzw

Hierarchy Inherits from TScrollingWinControl
See also TPicture
File name \source\Image.prg

1.6.2.12.1 TImage:Properties

■ read Only ■ Assignable ■ Design assignable ■ Run-time assignable

Scope	Name	Type	Initial value
■	IAutoFit	Logic	.F.
■	ISmooth	Logic	.F.
■	ITabStop	Logic	.T.
■	ITransparent	Logic	.F.

■	nBorderStyle	Numeric	bvSUNKEN
■	nClrPane	Numeric	clWhite
■	nHeight	Numeric	100
■	nWidth	Numeric	100
■	nZoom	Numeric	100
■	oPicture	Object	NIL

1.6.2.12.1.1 TImage:IAutoFit

When this property is true, the image will be scaled and centered within its client area. The property nZoom will be useless.

Scope	Assignable
Type	Logic
Initial value	.F.

1.6.2.12.1.2 TImage:ISmooth

When the image dimension is not the same than its display coordinates, some pixels maybe interpolated or eliminated to improve its display, but it loses some speed.

Scope	Assignable
Type	Logic
Initial value	.F.

1.6.2.12.1.3 TImage:ITabStop

The control receives the focus when the user presses the TAB key.

Scope	1
Type	Logic
Initial value	.T.

1.6.2.12.1.4 TImage:ITransparent

The control is drawn in transparent mode on its container.

Scope	Assignable
Type	Logic
Initial value	.F.

1.6.2.12.1.5 TImage:nBorderStyle

Indicates the border style.

Scope	Assignable
Type	Numeric
Initial value	bvSUNKEN
Possible values	bvNONE, bvRAISED, bvSUNKEN, bvBUMP, bvETCHED, bvFLAT

1.6.2.12.1.6 TImage:nClrPane

Indicates the control background color.

Scope:	Assignable
Type:	Numeric
Initial value:	clWhite

(See also the appendix to check the colors available)

1.6.2.12.1.7 TImage:nHeight

Indicates the control height.

Scope	Assignable
Type	Numeric
Initial value	100

1.6.2.12.1.8 TImage:nWidth

Indicates the control width.

Scope	Assignable
Type	Numeric
Initial value	100

1.6.2.12.1.9 TImage:nZoom

Percentage zoom to be applied to the image. A 100 value indicates the original value.

Scope	Assignable
Type	Numeric
Initial value	100

1.6.2.12.1.10 TImage:oPicture

TPicture object with the image to show in the control.

Scope	Assignable
Type	Object
Initial value	NIL

1.6.2.13 TImageEditor

This class represents a TImage control with advanced capabilities:

- Brightness
- Contrast
- Rotation
- Vertical and horizontal flip
- Crop

Hierarchy	TImage descendant
See also	TPicture, Scanner sample
File	Internal

1.6.2.13.1 TImageEditor:Properties

■ Read only ■ Assignable ■ Design assignable ■ Run-time assignable

Scope	Name	Type	Initial value
■	IShowHandlers	Logical	.T.
■	nBrightness	Numeric	1.0
■	nClrPane	Numeric	RGB(210,210,210)
■	nContrast	Numeric	1.0
■	nGamma	Numeric	0
■	nSaturation	Numeric	0
■	nTransparency	Numeric	1

1.6.2.13.1.1 TImageEditor:lShowHandlers

If true shows the image handlers.

Scope	Assignable
Type	Logical
Initial value	.T.

1.6.2.13.1.2 TImageEditor:nBrightness

Image brightness.

Scope	Assignable
Type	Numeric
Initial value	1.0
Possible values	-100 to +100

1.6.2.13.1.3 TImageEditor:nClrPane

Background color.

Scope	Assignable
Type	Numeric
Initial value	RGB(210,210,210)

1.6.2.13.1.4 TImageEditor:nContrast

Image contrast.

Scope	Assignable
Type	Numeric
Initial value	1.0
Possible values	-100 to +100

1.6.2.13.1.5 TImageEditor:nGamma

Image Gamma.

Scope	Assignable
Type	Numeric
Initial value	0

1.6.2.13.1.6 TImageEditor:nSaturation

Image saturation.

Scope	Assignable
Type	Numeric
Initial value	0
Possible values	0.00 to 1.00

1.6.2.13.1.7 TImageEditor:nTransparency

Image transparency.

Scope	Assignable
Type	Numeric
Initial value	1
Possible values	0.0 to 1.0

1.6.2.13.2 TImageEditor:Methods

■ Constructor ■ Standard

Type	Name
■	ApplyLevels
■	BlackWhite
■	Crop
■	FlipH
■	FlipV
■	GrayScale
■	Negative
■	Polaroid
■	RGBToBGR
■	Rotate

■	Save
■	SaveToStream
■	Sepia
■	SetColorMatrix

1.6.2.13.2.1 TImageEditor:ApplyLevels

Sets brightness and contrast on a single operation.

Type	Standard
Parameters	[<nBrightness>]: Brightness value. [<nContrast>]: Contrast value.
Return value	Nil

1.6.2.13.2.2 TImageEditor:BlackWhite

Converts image to black and white.

Type	Standard
Parameters	None
Return value	Nil

1.6.2.13.2.3 TImageEditor:Crop

Crops the image.

Type	Standard
Parameters	[<aRect>]: Rectangle array with the final image dimension. Optional. By default, actual rectangle set visually.
Return value	Nil

1.6.2.13.2.4 TImageEditor:FlipH

Flips the image horizontally.

Type	Standard
Parameters	None

Return value	Nil
---------------------	-----

1.6.2.13.2.5 TImageEditor:FlipV

Flips the image vertically.

Type	Standard
Parameters	None
Return value	Nil

1.6.2.13.2.6 TImageEditor:GrayScale

Turns the image to gray scale.

Type	Standard
Parameters	None
Return value	Nil

1.6.2.13.2.7 TImageEditor:Negative

Inverts image colors.

Type	Standard
Parameters	None
Return value	Nil

1.6.2.13.2.8 TImageEditor:Polaroid

Sets a polaroid filter to the image.

Type	Standard
Parameters	None
Return value	Nil

1.6.2.13.2.9 TImageEditor:RGBToBGR

Swaps red and blue colors on the image.

Type	Standard
Parameters	None
Return value	Nil

1.6.2.13.2.10 TImageEditor:Rotate

Rotates the image

Type	Standard
Parameters	<nAngle> Angle to rotate. Possible values: 90, 180 y 270
Return value	Nil

1.6.2.13.2.11 TImageEditor:Save

Saves the image to disk.

Type	Standard
Parameters	<cFileName> Image name [<nType>] Image type. Defined on the property TPicture.nImageType [<nQuality>] Image quality. Possible values from 0 to 100. By default 50. Only applicable to JPEG styles
Return value	<ISuccess> .T. if the file is saved successfully

This method tries to save the image with the following order:

1. Using the native API if it was used to load it and its saved using the same type
2. Using GDI+

1.6.2.13.2.12 TImageEditor:SaveToStream

Saves the image to a data stream. The conversion among different image types is not supported.

Type	Standard
-------------	----------

Parameters	[<nType> Image type. The type set by the property TPicture.nImageType [<nQuality> Image quality. Possible values from 0 to 100. By default 50. Only applicable to JPEG styles
Return value	<cStream> Data stream

This method tries to save the image with the following order:

1. Using the native API if it was used to load it and its saved using the same type
2. Using GDI+

1.6.2.13.2.13 TImageEditor:Sepia

Sets a sepia filter to the image.

Type	Standard
Parameters	None
Return value	Nil

1.6.2.13.2.14 TImageEditor:SetColorMatrix

Defines image colors and attributes to be set to the image.

Type	Standard
Parameters	<aMatrix> Numeric array with defined color and attributes of the image. Based on structure ColorMatrix of GDI+
Return value	Nil

1.6.2.13.3 TImageEditor:Events

Nombre	OnChangeRect
---------------	--------------

1.6.2.13.3.1 TImageEditor:OnChangeRect

Evento triggered when the crop rectangle dimensions change.

Parameters	<oSender>: A reference to the object that triggers the event.
-------------------	--

	<aRect>: New rectangle
Return value	Nil

1.6.2.14 TLabelBuddy

This class represents a TLabelBuddy control. A LabelBuddy is just a text label that its position depends on another control, commonly a TEdit control and it is normally used to describe the TEdit control.

The TEdit on all its inherited controls, on its contextual menu have an option to quickly create a TLabelBuddy control associated with itself which accelerates significantly the visual design process.

Hierarchy	Inherits from TLabel
File	\source\LabelBuddy.prg

1.6.2.14.1 TLabelBuddy:Properties

read Only
 Assignable
 Design assignable
 Run-time assignable

Scope	Name	Type	Initial value
<input type="checkbox"/>	nBuddyAling	Numeric	baLEFT
<input type="checkbox"/>	nBuddyOffset	Numeric	80
<input type="checkbox"/>	nVAlignment	Numeric	vaCENTER
<input type="checkbox"/>	oBuddy	Object	NIL

1.6.2.14.1.1 TLabelBuddy:nBuddyAlign

Control alignment respecto to its oBuddy control.

Scope	Assignable
Type	Numeric
Initial value	baLEFT
Possible values	baLEFT, baLEFTTOP, baLEFTBOTTOM, baRIGHT, baRIGHTTOP, baRIGHTBOTTOM, baBOTTOMLEFT, baBOTTOMCENTER, baBOTTOMRIGHT

1.6.2.14.1.2 TLabelBuddy:nBuddyOffset

Distance between the control and its oBuddy control. The distance is always measured from the initial coordinates of the control.

Scope	Assignable
Type	Numeric
Initial value	80

1.6.2.14.1.3 TLabelBuddy:nVAlignment

Vertical text alignment. Shows the text box aligned to the top, centered or aligned to the bottom in reference to the text.

Scope	Assignable
Type	Numeric
Initial value	vaTOP
Possible values	vaTOP, vaBOTTOM, vaCENTER

1.6.2.14.1.4 TLabelBuddy:oBuddy

Control from its position depends on.

Scope	Assignable
Type	Object
Initial value	NIL

1.6.2.15 TLabelEx

This class represents a text TLabel control with extended functionalities since it has some typical HTML extensions:

- Boldface usage with and tags
- Italic usage with <i> and </i> tags
- Underline usage with <u> and </u> tags
- Link usage with <a> tags
- Color usage with <#rrggbb> tags
- HTML tags <h1> to <h6>,
, <hr>, , and

Note: The <a> HTML tag does not include any link information. You must use the OnLinkClick event to control it.

You can consult a list of complete color values on this link:

[****]
<http://html-color-codes.info/>

The tag in order to work correctly must be included on a or structure. For example:

```
<ol>
  <li>one</li>
  <li>two</li>
</ol>
<ul>
  <li>one</li>
  <li>two</li>
</ul>
```

Consult the sample `\samples\LabelEx.xpj` for further information

Hierarchy TScrWinControl descendant
File \source\LabelEx.prg

1.6.2.15.1 TLabelEx:Properties

■ read Only ■ Assignable ■ Design assignable ■ Run-time assignable

Scope	Name	Type	Initial value
■	IAutoSize	Logic	.F.
■	IMultiLine	Logic	.T.
■	ITransparent	Logic	.T.
■	nBorderStyle	Numeric	bvNONE
■	nAlignment	Logic	taLEFT
■	nClrLink	Numeric	cIBlue
■	nLineSpacing	Numeric	100
■	nMargins	Numeric	0
■	nOpacity	Numeric	0
■	nScrollIncrement	Numeric	8
■	nVAlignment	Numeric	vaTOP

1.6.2.15.1.1 TLabelEx:IAutoSize

The vertical height of the control is adjusted automatically.

Scope	Assignable
Type	Logic
Initial value	.F.

1.6.2.15.1.2 TLabelEx:IMultiLine

Allows to the control to sow several lines of text.

Scope	Assignable
Type	Logic
Initial value	.F.

1.6.2.15.1.3 TLabelEx:ITransparent

The control is painted in transparent mode over the parent form.

Scope	Assignable
Type	Logic
Initial value	.T.

1.6.2.15.1.4 TLabelEx:nAlignment

Horizontal text alignment. The possible values are taRIGHT, taCENTER and taLEFT.

Scope	Assignable
Type	Numeric
Initial value	taLEFT
Possible values	taLEFT, taRIGHT, taCENTER

1.6.2.15.1.5 TLabelEx:nBorderStyle

Indicates the control border style.

Scope	Assignable
Type	Numeric
Initial value	bvNONE
Possible values	bvNONE, bvRAISED, bvSUNKEN, bvBUMP, bvETCHED, bvFLAT

1.6.2.15.1.6 TLabelEx:nClrLink

The control text color for link tags.

Scope	Assignable
Type	Numeric
Initial value	clBlue

1.6.2.15.1.7 TLabelEx:nLineSpacing

Sets line spacing percentage. Maximum value: 100.

Scope	Assignable
Type	Numeric
Initial value	100

1.6.2.15.1.8 TLabelEx:nMargins

Left, right and top margins

Scope	Assignable
Type	Numeric
Initial value	0

1.6.2.15.1.9 TLabelEx:nOpacity

Control opacity in percentage when its property ITransparent is set to true.

Scope:	Assignable
Type:	Numeric
Initial value:	0
Possible values:	0-100

1.6.2.15.1.10 TLabelEx:nScrollIncrement

Vertical control scroll increment specified in pixels.

Scope:	Assignable
---------------	------------

Type:	Numeric
Initial value:	8

1.6.2.15.1.11 TLabelEx:nVAlignment

Vertical text alignment. Shows the text box aligned to the top, centered or aligned to the button in reference to the text.

Scope	Assignable
Type	Numeric
Initial value	vaTOP
Possible values	vaTOP, vaBOTTOM, vaCENTER

1.6.2.15.2 TLabelEx:Events

Name
OnLinkClick
OnScrollDown
OnScrollPageDown
OnScrollPageUp
OnThumbPosV
OnThumbTrackV

1.6.2.15.2.1 TLabelEx:OnLinkClick

Event that is triggered when the user clicks on a link tag.

Parameters	<oSender> Reference to the object that triggers the event <cText> Link text <nLink> Ordinal link clicked, starting from one
Return value	NIL

1.6.2.15.2.2 TLabelEx:OnScrollDown

Event that is produced when the indicator is moved down.

Parameters:	<oSender>: Reference to the object that triggers the event
--------------------	--

	<nPos>: New indicator position
Return value:	<nPos>: If returns a value different than NIL, this must be the new indicator position. When the value is NIL the indicator increments automatically for the value indicated in the nScrollIncrement property.

1.6.2.15.2.3 TLabelEx:OnScrollPageDown

Event that is produced when the indicator is moved one page down.

Parameters:	<oSender>: Reference to the object that triggers the event <nPos>: New indicator position
Return value:	<nPos>: If returns a value different than NIL, this must be the new indicator position. When the value is NIL the indicator increments automatically one page

1.6.2.15.2.4 TLabelEx:OnScrollPageUp

Event that is produced when the indicator is moved one page up.

Parameters:	<oSender>: Reference to the object that triggers the event <nPos>: New indicator position
Return value:	<nPos>: If returns a value different than NIL, this must be the new indicator position. When the value is NIL the indicator decrements automatically one page

1.6.2.15.2.5 TLabelEx:OnScrollUp

Event that is produced when the indicator is moved up.

Parameters:	<oSender>: Reference to the object that triggers the event <nPos>: New indicator position
Return	<nPos>:

value:	If returns a value different than NIL, this must be the new indicator position. When the value is NIL the indicator decrements automatically for the value indicated in the nScrollIncrement property.
---------------	--

1.6.2.15.2.6 TLabelEx:OnThumbPosV

Event that is activated when the user stops to drag the vertical indicator.

Parameters:	<oSender>: Reference to the object that triggers the event <nPos>: New indicator position
Return value:	<nPos>: Scroll value to be applied. If it is NIL it assigns the default value

1.6.2.15.2.7 TLabelEx:OnThumbTrackV

Event that is activated when the vertical indicator is dragged.

Parameters:	<oSender>: Reference to the object that triggers the event <nPos>: New indicator position
Return value:	<nPos>: Scroll value to be applied. If it is NIL it assigns the default value

1.6.2.16 TMaskEdit

This class represents a TEdit specialized edit control with mask or templates.

The template format is the same used by [x]Harbour in the PICTURE clause in the GET command. The control also manage any basic [x]Harbour type: string, date, numeric or logic, instead of the TEdit control, that received any type of data converted to character type.



Hierarchy	Inherits from TEdit
File name	\source\MaskEdit.prg

1.6.2.16.1 TMaskEdit:Propiedades

read Only
 Assignable
 Design assignable
 Run-time assignable

Scope	Name	Type	Initial value
<input type="checkbox"/>	cPicture	Character	""
<input type="checkbox"/>	lTrimSpaces	Logic	.T.
<input type="checkbox"/>	lZeroFill	Logic	.F.

1.6.2.16.1.1 TMaskEdit:cPicture

Mask or template to be used in the edition. The template format is the same used in the [x]Harbour PICTURE clause in the GET command.

Scope:	Assignable
Type:	Character
Initial value:	""

For more information, see the [x]Harbour documentation about the GET command or the Transform() function, due they use the same type of templates.

The template characters supported by TMaskEdit are the following:

- Picture function symbols:

Character	Type	Description
A	C	Allows only alphabetic characters
B	N	Shows numbers with left alignment
C	N	Displays CR after the positive numbers
E	D,N	Displays dates with the day and month inverted, Regardless the current SET DATE, Numbers with European format (comma and decimal point inverted)
K	All	Clears all the GET clauses if the first key pressed is not a cursor key.
R	C	There are inserted characters that are not templates, but they are not stored in the GET variable
X	N	Shows DB after the negative numbers
Z	N	Shows blank spaces instead of zeros
(N	Shows negative numbers closed in parenthesis with spaces to the left
)	N	Shows negative numbers closed between parenthesis without spaces to the left
!	C	Transform the alphabetic characters in uppercase

- Picture template symbols:

Character	Description
A	Allows only alphabetic characters.

N	Allows only alphabetic characters and numbers.
X	Allows anything
Y	Allows Y, N, T or F (only available for logical values)
9	Allows digits for any type of data, including the sign for the numbers.
#	Allows digits, signs and spaces for any type of data.
L	Allows only T, F, Y or N.
!	Converts to uppercase any alphabet character.
\$	Shows a dollar sign instead a space to the left of the number.
*	Shows the asterisk instead a space to the left of the number.
.	Shows a decimal dot.
,	Shows a comma

1.6.2.16.1.2 TMaskEdit:ITrimSpaces

If it is .T. and the Value property is character type, all the spaces at the end of the value will be truncated.

Scope:	Assignable
Type:	Logic
Initial value:	.T.

1.6.2.16.1.3 TMaskEdit:IZeroFill

If it is .T. and the Value property is character type, this will be filled with zeros to the left when the control loose the control. For example, if the control has a maximum length of 5 and the user types the value "84", the value will be transformed to "00084".

Scope:	Assignable
Type:	Logic
Initial value:	.F.

1.6.2.17 TNoEdit

This class represents a TLabel label control with border.

Description:

A TNoEdit is a text label and looks like a TEdit edit control, but with the advantage that can use a mask through its cPicture property.

Hierarchy Inherits from TLabel
File name \source\NoEdit.prg

1.6.2.17.1 TNoEdit:Properties

■ read Only ■ Assignable ■ Design assignable ■ Run-time assignable

Scope	Name	Type	Initial value
■	cPicture	Character	""
■	nClrPane	Numeric	clWindow
■	nClrText	Numeric	clGray
■	nHeight	Numeric	20
■	nWidth	Numeric	90
■	Value	Character, Numeric, Date or Logic	""

1.6.2.17.1.1 TNoEdit:cPicture

Indicates the mask to be used.

Scope	Assignable
Type	Character
Initial value	""

For more information about the mask types, see the [x]Harbour documentation.

1.6.2.17.1.2 TNoEdit:nClrPane

Indicates the background color.

Scope	Assignable
Type	Numeric
Initial value	clGray

Consult the appendix for the list of available colors

1.6.2.17.1.3 TNoEdit:nClrText

Indicates the text color.

Scope	Assignable
Type	Numeric
Initial value	clGray

Consult the appendix for the list of available colors

1.6.2.17.1.4 TNoEdit:nHeight

Indicates the control height.

Scope:	Assignable
Type:	Numeric
Initial value:	20

1.6.2.17.1.5 TNoEdit:nWidth

Indicates the control width.

Scope:	Assignable
Type:	Numeric
Initial value:	90

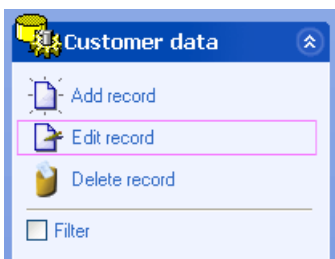
1.6.2.17.1.6 TNoEdit:Value

Indicates the control's value.

Scope:	Assignable
Type:	Character, Numeric, Date or Logic
Initial value:	""

1.6.2.18 TOptionItem

This class represents the item from the TOptionList control.



Hierarchy Inherits from TComponent
File name \source\OptionItem.prg

1.6.2.18.1 TOptionItem:Properties

■ read Only ■ Assignable ■ Design assignable ■ Run-time assignable

Scope	Name	Type	Initial value
■	altems	Array	{}
■	cText	Character	""
■	IChecked	Logic	.F.
■	ICollapsable	Logic	.F.
■	IEnabled	Logic	.T.
■	IExpanded	Logic	.T.
■	IMultiLine	Logic	.F.
■	nImage	Numeric	0
■	nImageDisabled	Numeric	0
■	nType	Numeric	otBUTTON
■	oOptionList	Object	NIL

1.6.2.18.1.1 TOptionItem:altems

Array with all the item that depends from the current one.

Scope	read Only
Type	Array
Initial value	{}

1.6.2.18.1.2 TOptionItem:cText

Text to be shown in the item.

Scope	Assignable
Type	Character
Initial value	""

1.6.2.18.1.3 TOptionItem:IChecked

If it is .T. it will show the item with the checkbox activated. The nType property must have the otCHECK value.

Scope	Assignable
Type	Logic
Initial value	.F.

1.6.2.18.1.4 TOptionItem:ICollapsible

If it is .T., the item can be expanded or collapsible.

Scope	Assignable
Type	Logic
Initial value	.F.

1.6.2.18.1.5 TOptionItem:IEnabled

If it is .T. the item is enabled.

Scope	Assignable
Type	Logic
Initial value	.T.

1.6.2.18.1.6 TOptionItem:IExpanded

If it is .T. the item will be expanded.

Scope	Assignable
Type	Logic
Initial value	.T.

1.6.2.18.1.7 TOptionItem:IMultiLine

If it is .T. the item may be multi line.

Scope	Assignable
Type	Logic
Initial value	.F.

1.6.2.18.1.8 TOptionItem:nImage

Indicates the image to be shown with the item. It corresponds to the image number in the olmagedList object from its TOptionList container object.

Scope	Assignable
Type	Numeric
Initial value	0

1.6.2.18.1.9 TOptionItem:nImageDisabled

Indicates the image to be shown with the item when is disabled. It corresponds to the image number in the olmagedList object from its TOptionList container object.

Scope	Assignable
Type	Numeric
Initial value	0

1.6.2.18.1.10 TOptionItem:nType

Indicates the item type.

Scope	Assignable
Type	Numeric
Initial value	otBUTTON
Possible values	otSEPARATOR, otBUTTON, otCHECK, otRADIO, otTEXT

1.6.2.18.1.11 TOptionItem:oOptionList

Reference to its TOptionList container object.

Scope	read Only
Type	Object
Initial value	NIL

1.6.2.18.2 TOptionItem:Methods

■ Constructor ■ Standard

Typ	Name
■	AddItem
■	AddSeparator
■	Create
■	Delete
■	InsertItem

1.6.2.18.2.1 TOptionItem:AddItem

Adds a new item in the current item branch.

Type	Standard
Parameters	<p><cltem> Item text</p> <p><xImage> Image to use in the item. It corresponds with the number of its olmageList object of its TOptionList container</p> <p><nType>: Indicates the item type. it corresponds to the nType property</p> <p><IMultiLine> If it is .T. the item will be multiline. It corresponds to the IMultiLine property</p> <p><IEnabled> If it is .T. the item will be enabled. It correspond to the IEnabled property IEnabled</p> <p><IChecked> If it is .T. the item will show a activated checkbox. nType must have the otCHECK value</p> <p><ICollapsible> If it is .T. the item can be collapsed. It corresponds to the ICollapsible property</p> <p><IExpanded> If it is .T. the item will be expanded. It corresponds to the IExpanded property</p> <p>The default values correspond to their equivalent properties</p>
Return value	<p><oltem> Reference to the new item created</p>

1.6.2.18.2.2 TOptionItem:AddSeparator

Adds a separator item.

Type	Standar
Parameters	None
Return value	<oItem> Reference to the new item created

1.6.2.18.2.3 TOptionItem:Create

Class constructor.

Type	Constructor
Parameters	<oParent> Container object for th item. It can be another item or the same TOptionList <nPos> : New item position <cItem> Item text <xImage> Image to use in the item. It corresponds with the number of its olmageList object of its TOptionList container <nType> : Indicates the item type. it corresponds to the nType property <IMultiLine> If it is .T. the item will be multiline. It corresponds to the IMultiLine property <IEnabled> If it is .T. the item will be enabled. It correspond to the IEnabled property IEnabled <IChecked> If it is .T. the item will show a activated checkbox. nType must have the otCHECK value <ICollapsable> If it is .T. the item can be collapsed. It corresponds to the ICollapsable property <IExpandede> If it is .T. the item will be expanded. It corresponds to the IExpanded property The default values correspond to their equivalent properties
Return value	<oItem> Reference to the new item created

1.6.2.18.2.4 TOptionItem:Delete

Deletes the current item. If the item has branches, all of them are deleted as well.

Type	Standard
Parameters	None
Return value	NIL

1.6.2.18.2.5 TOptionItem:InsertItem

Inserts a new item in the current item branch.

Type	Standard
Parameters	<p><nPos>: New item position</p> <p><cItem> Item text</p> <p><xImage> Image to use in the item. It corresponds with the number of its oImageList object of its TOptionList container</p> <p><nType>: Indicates the item type. it corresponds to the nType property</p> <p><IMultiLine> If it is .T. the item will be multiline. It corresponds to the IMultiLine property</p> <p><IEnabled> If it is .T. the item will be enabled. It correspond to the IEnabled property</p> <p><IChecked> If it is .T. the item will show a activated checkbox. nType must have the otCHECK value</p> <p><ICollapsible> If it is .T. the item can be collapsed. It corresponds to the ICollapsible property</p> <p><IExpanded> If it is .T. the item will be expanded. It corresponds to the IExpanded property</p> <p>The default values correspond to their equivalent properties</p>
Return value	<p><oItem> Reference to the new item created</p>

1.6.2.18.3 TOptionItem:Events

Name

OnClick

1.6.2.18.3.1 TOptionItem:OnClick

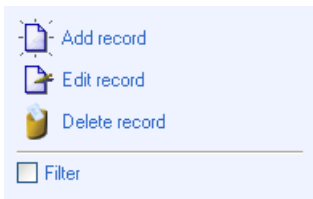
Event that is produced when the user clicks the item and the nType property is set to otBUTTON.

Parameters	<oSender>:
:	Reference to the object that triggers the event
Return value:	NIL

1.6.2.19 TOptionList

This class represents a control that shows a list of options, with the possibility to associate an action to every element of the list. Every element can behave as a button, a checkbox, a radio button, a simple text or a separator line. The list can have different nested levels and this allows to build list with tree structures.

Its use is very similar to the menus, but it offer a better visual look and allows to use it in any position in the form. Its use with TExplorerBar control types is very common.



Hierarchy Inherits from TStdControl
File name \source\OptionList.prg

1.6.2.19.1 TOptionList:Properties

■ read Only ■ Assignable ■ Design assignable ■ Run-time assignable

Scope	Name	Type	Initial value
■	altems	Array	{}
■	IAutoSize	Logic	.F.
■	ITabStop	Logic	.T.
■	ITransparent	Logic	.T.
■	nBorderStyle	Numeric	bvNONE
■	nClrPane	Numeric	clBtnFace
■	nClrText	Numeric	clHighLite

■	nClrTextHot	Numeric	clActiveCaption
■	nHeight	Numeric	100
■	nHotStyle	Numeric	hsUNDERLINE
■	nItemHeight	Numeric	16
■	nLeftMargin	Numeric	8
■	nMargin	Numeric	8
■	nScrollIncrement	Numeric	8
■	nTopMargin	Numeric	8
■	nWidth	Numeric	100
■	oImageList	Object	TImageList

1.6.2.19.1.1 TOptionList:Items

List of TOptionItem objects displayed by the control.

Scope:	Design assignable
Type:	Array
Initial value:	{}

1.6.2.19.1.2 TOptionList:IAutoSize

If it is `.T.` the control dimensions will be automatically adjusted based in its TOptionItem elements.

Scope:	Assignable
Type:	Logic
Initial value:	<code>.F.</code>

1.6.2.19.1.3 TOptionList:ITabStop

The control receives the focus when the user presses the TAB key.

Scope	Design assignable
Type	Logic
Initial value	<code>.T.</code>

1.6.2.19.1.4 TOptionList:ITransparent

The control is drawn in transparent mode on the form where it belongs.

Scope	Asignable
Type	Logic
Initial value	.F.

1.6.2.19.1.5 TOptionList:nBorderStyle

Indicates the border style.

Scope	Asignable
Type	Numeric
Initial value	bvNONE
Possible values	bvNONE, bvRAISED, bvSUNKEN, bvBUMP, bvETCHED, bvFLAT

1.6.2.19.1.6 TOptionList:nClrPane

Indicates the control background color.

Scope:	Asignable
Type:	Numeric
Initial value:	clBtnFace

(See also the appendix to check the color available)

1.6.2.19.1.7 TOptionList:nClrText

Indicates the control text color.

Scope:	Asignable
Type:	Numeric
Initial value:	clHighLite

(See also the appendix to check the color available)

1.6.2.19.1.8 TOptionList:nClrTextHot

Indicates the control text color when it is marked by the mouse cursor.

Scope:	Assignable
Type:	Numeric
Initial value:	clActiveCaption

(See also the appendix to check the color available)

1.6.2.19.1.9 TOptionList:nHeight

Indicates the control height.

Scope:	Assignable
Type:	Numeric
Initial value:	100

1.6.2.19.1.10 TOptionList:nHotStyle

Item painting style when the mouse is on top of it.

Scope	Assignable
Type	Numeric
Initial value	hsUNDERLINE
Possible values	hsUNDERLINE, hsMENU, hsMENUXP, hsMENU2003

1.6.2.19.1.11 TOptionList:nItemHeight

Indicates the control item height. The items height is established based on the bitmap and font height, and the value of this property.

Scope:	Assignable
Type:	Numeric
Initial value:	16

1.6.2.19.1.12 TOptionList:nLeftMargin

Indicates the left margin to display the control items.

Scope:	Assignable
Type:	Numeric
Initial value:	8

1.6.2.19.1.13 TOptionList:nMargin

Indicates the left and upper margin to display the control items.

Scope:	Assignable
Type:	Numeric
Initial value:	8

This property is obsolete, is maintained for backwards compatibility with previous versions. You should use the properties nTopMargin and nLeftMargin. Changing this value automatically changes the value of these same properties.

1.6.2.19.1.14 TOptionList:nScrollIncrement

Indicates the vertical scroll increment specified in pixels to move when the user presses the vertical scroll buttons .

Scope:	Assignable
Type:	Numeric
Initial value:	8

1.6.2.19.1.15 TOptionList:nTopMargin

Indicates the upper margin to display the control items.

Scope:	Assignable
Type:	Numeric
Initial value:	8

1.6.2.19.1.16 TOptionList:nWidth

Indicates the control width.

Scope:	Assignable
Type:	Numeric
Initial value:	100

1.6.2.19.1.17 TOptionList:olmageList

TImageList object with all the images to be used by the control.

Scope	Assignable
Type	Object
Initial value	TImageList()

The TImageList object is instantiated since the control creation with its New() constructor. The first image that is included in the TImageList establishes the dimension of the following images to be entered. If the first image includes more than one bitmap, it is important to establish the nHeight and nWidth TImageList properties before to add the bitmap to the TImageList object.

1.6.2.19.2 TOptionList:Methods

■ Constructor ■ Standard ■ Only after Create()

Typ	Name
■	AddItem
■	AddSeparator
■	GetItemByPos
■	GetItemCount
■	GetItemPos
■	HitTest
■	InsertItem

1.6.2.19.2.1 TOptionList:AddItem

Adds a new TOptionItem element to the control list of values.

Type	Standard
Parameters	<cltem> : Text to be shown by the new element [<nImage>] : Image number in its olmageList property [<nType>] : Item type. For more information, see also the

	<p>nType property from the TOptionItem class</p> <p>[<IMultiLine>]: If it is .T. it might show more than one line</p> <p>[<IEnabled>]: If it is .T. the item is enabled</p> <p>[<IChecked>]: If it is .T. and it is a checkbox style, the checkbox will be checked.</p> <p>[<ICollapsible>]: If it is .T. and the item has branches, these can be collapsed</p> <p>[<IExpanded>]: If it is .T. and the item has branches, this will be shown expanded</p> <p>[<nImageDisabled>]: Image number when the item is disabled in its olmageList property</p> <p>The default values are the same offered by default by the TOptionItem class.</p>
Return value	<oltem>: New TOptionItem object created

1.6.2.19.2.2 TOptionList:AddSeparator

Adds a new TOptionItem separator type element to the control list of values.

Type	Standard
Parameters	None
Return value	<oltem>: New TOptionItem object created

1.6.2.19.2.3 TOptionList:GetItemByPos

Returns the existing TOptionItem element in an specific position. That position is the line number if all the OptionList branches were expanded.

Type	Standard
Parameters	<nPos>: Element position
Return value	<oltem>: TOptionItem object

1.6.2.19.2.4 TOptionList:GetItemCount

Returns the total number of existing TOptionItem elements in the control.

Type	Standard
Parameters	None
Return value	<nTotal>: Total number of elements

1.6.2.19.2.5 TOptionList:GetItemPos

Returns the position from an specific TOptionItem element. That position is the line number if all the OptionList branches were expanded.

Type	Standard
Parameters	<oItem>: Object TOptionItem
Return value	<nPos>: Element position

1.6.2.19.2.6 TOptionList:HitTest

Returns the existing item in an specific coordinate.

Type	Only after Create()
Parameters	<nX>: X coordinate <nY>: Y coordinate
Return value	<oItem> Existing TOptionItem or NIL

1.6.2.19.2.7 TOptionList:InsertItem

Inserts a new TOptionItem element to the control list of values.

Type	Standard
Parameters	<nPos>: New element position <cItem>: Text to be shown by the new element

	<p>[<nImage>]: Image number in its olmageList property</p> <p>[<nType>]: Item type. For more information, see also the nType property from the TOptionItem class</p> <p>[<IMultiLine>]: If it is .T. it might show more than one line</p> <p>[<IEnabled>]: If it is .T. the item is enabled</p> <p>[<IChecked>]: If it is .T. and it is a checkbox style, the checkbox will be checked.</p> <p>[<ICollapsible>]: If it is .T. and the item has branches, these can be collapsed</p> <p>[<IExpanded>]: If it is .T. and the item has branches, this will be shown expanded</p> <p>[<nImageDisabled>]: Image number when the item is disabled in its olmageList property</p> <p>The default values are the same offered by default by the TOptionItem class.</p>
Return value	<p><oltem>: New TOptionItem object created</p>

1.6.2.19.3 TOptionList:Events

Name
OnScrollDown
OnScrollPageDown
OnScrollPageUp
OnScrollUp
OnThumbPosV
OnThumbTrackV

1.6.2.19.3.1 TOptionList:OnScrollDown

Event that is produced when there is a scroll down.

Parameters	<p><oSender>: Reference to the object that triggers the event</p> <p><nPos>: Scroll bar position</p>
Return value:	<p><nNewPos> If it returns a value different to NIL, the scroll bar will be moved to the returned value, otherwise it will</p>

be moved down to the value indicated by the `nScrollIncrement` property.

1.6.2.19.3.2 TOptionList:OnScrollPageDown

Event that is produced when there is a scroll page down.

Parameters	<oSender> : Reference to the object that triggers the event
:	<nPos> : Scroll bar position
Return value:	<nNewPos> If it returns a value different to NIL, the scroll bar will be moved to the returned value, otherwise it will be moved one page down.

1.6.2.19.3.3 TOptionList:OnScrollPageUp

Event that is produced when there is a scroll page up.

Parameters	<oSender> : Reference to the object that triggers the event
:	<nPos> : Scroll bar position
Return value:	<nNewPos> If it returns a value different to NIL, the scroll bar will be moved to the returned value, otherwise it will be moved one page up.

1.6.2.19.3.4 TOptionList:OnScrollUp

Event that is produced when there is a scroll up.

Parameters	<oSender> : Reference to the object that triggers the event
:	<nPos> : Scroll bar position
Return value:	<nNewPos> If it returns a value different to NIL, the scroll bar will be moved to the returned value, otherwise it will be moved up to the value indicated by the <code>nScrollIncrement</code> property.

1.6.2.19.3.5 TOptionList:OnThumbPosV

Event that is produced when the scroll bar stops to move.

Parameters	<oSender> : Reference to the object that triggers the event
:	<nPos> : New scroll bar position
Return value:	<nNewPos> Scroll value to apply. If it is NIL it assigns the default value

1.6.2.19.3.6 TOptionList:OnThumbTrackV

Event that is triggered when the vertical scroll bar is moved.

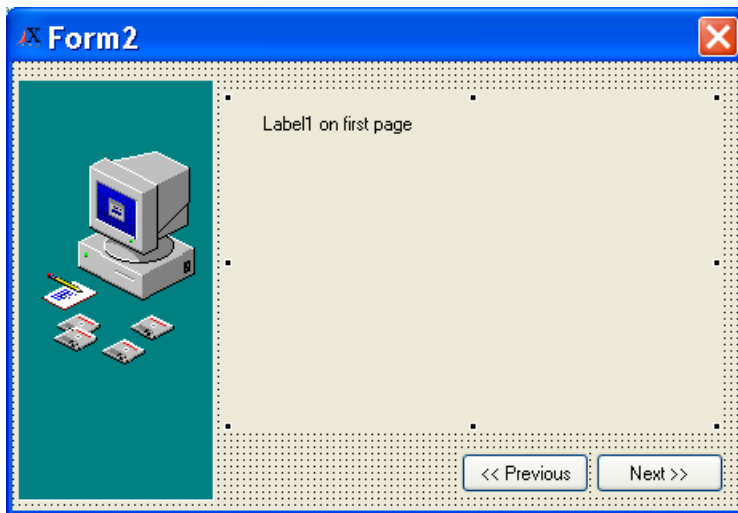
Parameters	<oSender> : Reference to the object that triggers the event
:	<nPos> : New scroll bar position
Return value:	<nNewPos> Scroll value to apply. If it is NIL it assigns the default value

1.6.2.20 TPage

This class represents a page of the TPages control.

You can not create a TPage control directly, but the TPages control creates automatically a **TPage** object for every page it has defined.

Supports a background image and automatic scrollbars when the controls do not fit on the client area.



Hierarchy TScrollingWinControl descendant
See also TPages
File \source\Page.prg

1.6.2.20.1 TPage:Properties

Read only assignable Design assignable Run-time assignable

Scope	Name	Type	Initial value
<input checked="" type="checkbox"/>	ITransparent	Logical	.T.
<input checked="" type="checkbox"/>	nItem	Numerical	0

1.6.2.20.1.1 TPage:ITransparent

The controls is transparent.

Scope	assignable
Type	Logical
Initial value	.T.

1.6.2.20.1.2 TPage:nItem

Index number on itsTPages container.

Scope	Read only
Type	Numerical
Initial value	0

1.6.2.20.2 TPage:Methods

■ Constructor ■ Standard ■ Only after Create()

Type	Name
■	Delete
■	Select

1.6.2.20.2.1 TPage:Delete

Deletes the page from its TPages container.

Type	Standard
Parameters	None
Return value	<IOk> True if success

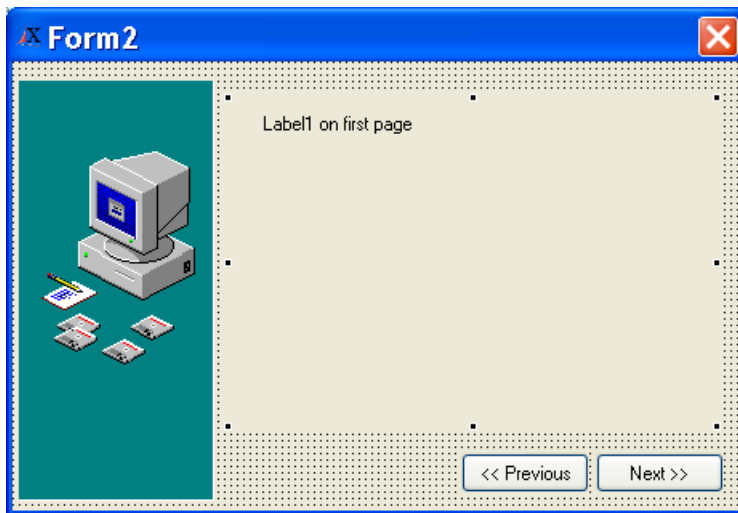
1.6.2.20.2.2 TPage:Select

Selects the TPage control on its owner TPages container.

Type	Standard
Parameters	None
Return value	NIL

1.6.2.21 TPages

This control lets you have multiple panels or pages that can be also other controls containers. The class gives you the functionality to navigate around the distinct pages. Its use is quite usual within the typical dialogs with the buttons 'Previous' and 'Next'. Every page of the control is a object of the TPage class.



Hierarchy TWinControl descendant
See also TPage
File \source\Pages.prg

1.6.2.21.1 TPages:Properties

Read only
 Assignable
 Design assignable
 Run-time assignable

Scope	Name	Type	Initial value
<input type="checkbox"/>	altems	Array	{}
<input type="checkbox"/>	ITabStop	Lógica	.F.
<input type="checkbox"/>	ITransparent	Lógica	.F.
<input type="checkbox"/>	nBorderStyle	Numeric	bvNONE
<input type="checkbox"/>	nHeight	Numeric	160
<input checked="" type="checkbox"/>	nIndex	Numeric	0
<input type="checkbox"/>	nWidth	Numeric	200

1.6.2.21.1.1 TPages:altems

List of TPage controls owned by the controls.

Scope	Design assignable
Type	Array
Initial value	{}

1.6.2.21.1.2 TPages:ITabStop

The control receives focus when navigating with the TAB key.

Scope	Design assignable
Type	Lógical
Initial value	.F.

1.6.2.21.1.3 TPages:ITransparent

Control background is transparent.

Scope	Design assignable
Type	Lógical
Initial value	.F.

1.6.2.21.1.4 TPages:nBorderStyle

Border style.

Scope	Assignable
Type	Numeric
Initial value	bvNONE
Possible values	bvNONE, bvRAISED, bvSUNKEN, bvBUMP, bvETCHED, bvFLAT

1.6.2.21.1.5 TPages:nHeight

Control height.

Scope	Assignable
Type	Numeric
Initial value	160

1.6.2.21.1.6 TPages:nIndex

Active page.

Scope	Assignable
--------------	------------

Type	Numeric
Initial value	0

1.6.2.21.1.7 TPages:nWidth

Control width.

Scope	Assignable
Type	Numeric
Initial value	200

1.6.2.21.2 TPages:Methods

Constructor Standard Only after Create()

Typ e	Name
<input type="checkbox"/>	AddItem
<input type="checkbox"/>	DeleteItem
<input type="checkbox"/>	DeleteItems
<input type="checkbox"/>	InsertItem
<input type="checkbox"/>	Next
<input type="checkbox"/>	Previous

1.6.2.21.2.1 TPages:AddItem

Adds a new page to the TPage control.

Type	Standard
Parameters	None
Return value	TPage object

1.6.2.21.2.2 TPages>DeleteItem

Deletes a specific control page.

Type	Standard
Parameters	<nItem>: Page number to delete
Return value	NIL

1.6.2.21.2.3 TPages:DeleteItems

Deletes all the control pages.

Type	Standard
Parameters	None
Return value	NIL

1.6.2.21.2.4 TPages:InsertItem

Inserts a new TPage object to the control.

Type	Standard
Parameters	<nPos>: New page position
Return value	TPage object

1.6.2.21.2.5 TPages:Next

Moves forward to next control page.

Type	Standard
Parameters	None
Return value	<nIndex> New page index

1.6.2.21.2.6 TPages:Previous

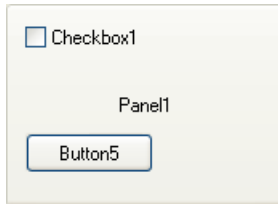
Moves backward to previous control page..

Type	Standard
Parameters	None
Return value	<nIndex> Índice de la nueva página

1.6.2.22 TPanel

This class represents a panel control, it means a control container.

It supports different border styles and a background image as well. It supports a Windows XP look, but in that case it is not possible to specify the style and the background image neither.



Hierarchy Inherits from TWinControl
See also TScrollBar
File name \source\Panel.prg

1.6.2.22.1 TPanel:Properties

■ read Only ■ Assignable ■ Design assignable ■ Run-time assignable

Scope	Name	Type	Initial value
■	IXPLook	Logic	.T.
■	nHeight	Numeric	90
■	nBevelInner	Numeric	bvNONE
■	nBevelOuter	Numeric	bvRAISED
■	nWidth	Numeric	120

1.6.2.22.1.1 TPanel:IXPLook

Shows the panel with Windows XP "look and feel".

Scope	Assignable
Type	Logic
Initial value	.T.

1.6.2.22.1.2 TPanel:nHeight

Indicates the control height.

Scope	Assignable
Type	Numeric
Initial value	90

1.6.2.22.1.3 TPanel:nBevelOuter

Indicates the external bevel style to be used

Scope	Assignable
Type	Numeric
Initial value	bvRAISED
Possible values	bvNONE, bvRAISED, bvSUNKEN

1.6.2.22.1.4 TPanel:nBevelInner

Indicates the internal bevel style to be used.

Scope	Assignable
Type	Numeric
Initial value	bvNONE
Possible values	bvNONE, bvRAISED, bvSUNKEN

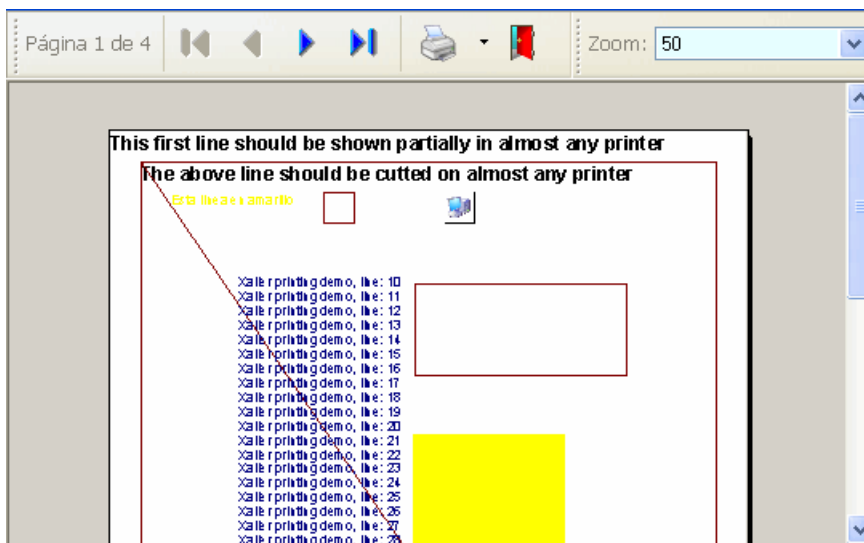
1.6.2.22.1.5 TPanel:nWidth

Indicates the control width.

Scope	Assignable
Type	Numeric
Initial value	120

1.6.2.23 TPreviewControl

This class represents the control used for Xailer to preview the print out.



As you can see it is not exactly the preview form but the control aligned to the client that is in it. This control has a button bar with some classic operations and a TPreviewPage control type aligned to the client where the page preview is shown.

This class is not exactly an inherited object from TForm due it offers the possibility to show the preview page in a MDI child window. In that case only you need to load the Preview method from the TPrinter class to use a MDI instead a normal form.

The Professional Xailer version offers the full source code for this class and you can modify it according to your own needs.

It is possible to export the preview to a PDF file and even send it by Email on PDF format as well. You will need the commercial library Image2PDF.dll to do the job. (Visit [Utility warrior](#) for further information) When that library is present on the system it will be possible to show two new buttons to export to PDF and send by Email, being necessary also to configure the TPrinter.oExportInfo property.

Hierarchy	Inherits from TScrollingWinControl
See also	TPreviewPage
File name	\\source\\PreviewControl.prg

1.6.2.23.1 TPreviewControl:Properties

■ read Only ■ Assignable ■ Design assignable ■ Run-time assignable

Scope	Name	Type	Initial value
■	nPage	Numeric	1
■	nZoomIndex	Numeric	1
■	oCombo	Object	TComboBox
■	oPreviewDC	Object	TPreviewDC

■	oPreviewPage	Object	TPreviewPage
■	oPrnMenu	Object	TMenu
■	oRebar	Object	TRebar
■	oRebarBand	Object	TRebarBand
■	oToolBar	Object	TToolBar

All read Only components reflects the controls that the control has in it Those controls are:

oCombo:	TComboBox control to establishes the zoom
oPreviewDC	Non documented Internal Xailer object that manages the EMF collection with the Preview operation
oPreviewPage:	TPreviewPage control where the preview for every page is shown
oPrnMenu:	TMenu control that depends from the Print button in the button bar
oRebar:	TRebar control that is shown aligned to alTOP
oRebarBand	TRebarBand control with the first band where the ComboBox control is located
oToolBar:	TToolBar control for the button bar

1.6.2.23.1.1 TPreviewControl:nPage

Indicates the page number that is being showed.

Scope:	Assignable
Type:	Numeric
Initial value:	1

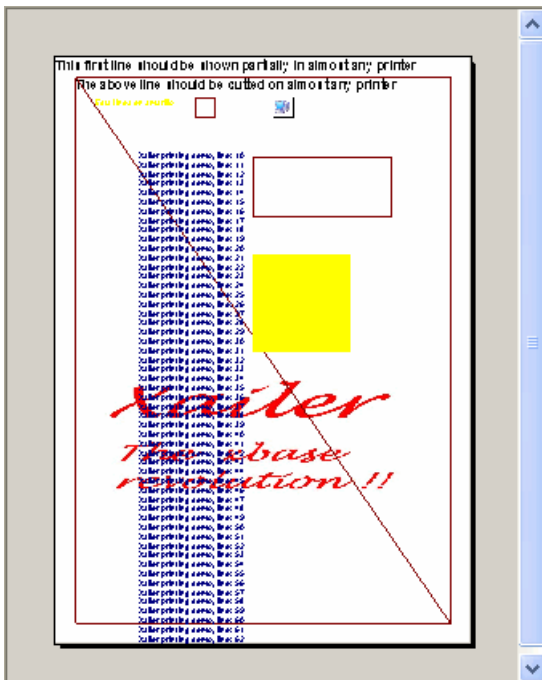
1.6.2.23.1.2 TPreviewControl:nZoomIndex

Zoom index to use based on **oCombo** Zoom values.

Scope:	Assignable
Type:	Numeric
Initial value:	1

1.6.2.24 TPreviewPage

This class represents a control to display EMF (Enhanced Meta file) files that is internally used by the TPreviewControl to show all the printer preview pages.



Hierarchy Inherits from TScrollingWinControl
See also TPreviewControl
File name \source\PreviewPage.prg

1.6.2.24.1 TPreviewPage:Properties

■ read Only ■ Assignable ■ Design Assignable ■ Run-time assignable

Scope	Name	Type	Initial value
■	nFitInPage	Numeric	pfNONE
■	nZoomFactor	Numeric	100
■	oEmf	Object	NIL
■	oPrinter	Object	NIL

1.6.2.24.1.1 TPreviewPage:nFitInPage

Indicates if the EMF file should be shown adjusted to the control coordinates.

Scope: Assignable
Type: Numeric
Initial value: pfNONE

Possible values:	pfNONE, pfALL, pfWIDTH, pfHEIGHT
-------------------------	----------------------------------

1.6.2.24.1.2 TPreviewPage:nZoomFactor

Zoom factor from the EMF in the control.

Scope:	Assignable
Type:	Numeric
Initial value:	100

1.6.2.24.1.3 TPreviewPage:oEmf

TEnhMetaFile object type to show.

Scope:	Assignable
Type:	Object
Initial value:	NIL

1.6.2.24.1.4 TPreviewPage:oPrinter

TPrinter object type (Dimension and page resolution) that will be uses as figure base for the oEmf control presentation.

Scope:	Assignable
Type:	Object
Initial value:	NIL

1.6.2.24.2 TPreviewPage:Methods

■ Constructor ■ Standard ■ Only after Create()

Typ Name
■ Create New

1.6.2.24.2.1 TPreviewPage:Create | New

Class constructor.

Type	Standard
Parameters	[<oEmf> TEnhMetaFile object to show [<oPrinter> TPrinter object type (Dimension and page resolution) that will be used as figure base for the oEmf control presentation
Return value	Self reference

1.6.2.24.3 TPreviewPage:Events

Name	
	OnFirstPage
	OnLastPage
	OnNextPage
	OnPreviousPage
	OnZoomChanged
	OnZoomIn
	OnZoomOut

1.6.2.24.3.1 TPreviewPage:OnFirstPage

Event that is produced when the user presses the 'Ctrl' + 'PgUp' or the 'home' keys.

Parameters	<oSender>:
:	Reference to the object that triggers the event
Return value:	NIL

1.6.2.24.3.2 TPreviewPage:OnLastPage

Event that is produced when the user presses the 'Ctrl' + 'PgDn' or the 'end' keys.

Parameters	<oSender>:
:	Reference to the object that triggers the event
Return value:	NIL

1.6.2.24.3.3 TPreviewPage:OnNextPage

Event that is produced when the user presses the 'PgDn' key.

Parameters	<oSender> :
:	Reference to the object that triggers the event
Return value:	NIL

1.6.2.24.3.4 TPreviewPage:OnPreviousPage

Event that is produced when the user presses the 'PgUp' key.

Parameters	<oSender> :
:	Reference to the object that triggers the event
Return value:	NIL

1.6.2.24.3.5 TPreviewPage:OnZoomChanged

Event that is produced when there is a change in the zoom level.

Parameters	<oSender> :
:	Reference to the object that triggers the event
	<nZoomFactor> :
	Zoom factor
Return value:	NIL

1.6.2.24.3.6 TPreviewPage:OnZoomIn

Event that is produced when the zoom level is increased.

Parameters	<oSender> :
:	Reference to the object that triggers the event
Return value:	NIL

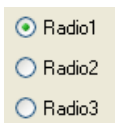
1.6.2.24.3.7 TPreviewPage:OnZoomOut

Event that is produced when the zoom level is reduced.

Parameters	<oSender>:
:	Reference to the object that triggers the event
Return value:	NIL

1.6.2.25 TRadioMenu

This class represents a 'Radio' button type control menu.



Hierarchy	Inherits from TStdControl
See also	TRadio
File name	\source\RadioMenu.prg

1.6.2.25.1 TRadio:Properties

■ read Only ■ Assignable ■ Design assignable ■ Run-time assignable

Scope	Name	Type	Initial value
■	altems	Array	{}
■	ITabStop	Logic	.F.
■	ITransparent	Logic	.F.
■	nBorderStyle	Numeric	bvNONE
■	nClrPane	Numeric	clBtnFace
■	nColumns	Numeric	0
■	nHeight	Numeric	100
■	nIndex	Numeric	1
■	nMargin	Numeric	2
■	nOrientation	Numeric	orVERTICAL
■	nSpacing	Numeric	10
■	nWidth	Numeric	100

1.6.2.25.1.1 TRadioMenu:altems

Array with the names of the different Radiomenu elements.

Scope:	Assignable
Type:	Array
Initial value:	{}

1.6.2.25.1.2 TRadioMenu:ITabStop

Indicates if the control receives the focus when the user press the TAB key.

Scope	Design assignable
Type	Logic
Initial value	.F.

1.6.2.25.1.3 TRadioMenu:ITransparent

If it is .T. the control's background will be transparent. In that case the nClrPane has any effect.

Scope:	Design assignable
Type:	Logic
Initial value:	.T. .

1.6.2.25.1.4 TRadioMenu:nBorderStyle

Indicates the border style.

Scope	Design assignable
Type	Numeric
Initial value	bvNONE
Possible values	bvNONE, bvRAISED, bvSUNKEN, bvBUMP, bvETCHED, bvFLAT

1.6.2.25.1.5 TRadioMenu:nClrPane

Indicates the control's background color.

Scope:	Assignable
---------------	------------

Type:	Numeric
Initial value:	clBtnFace

(See also the appendix to check the colors available)

1.6.2.25.1.6 TRadioMenu:nColumns

Indicates the number of columns to be shown by the control.

Scope:	Design assignable
Type:	Numeric
Initial value:	1

1.6.2.25.1.7 TRadioMenu:nHeight

Indicates the control height.

Scope:	Assignable
Type:	Numeric
Initial value:	100

1.6.2.25.1.8 TRadioMenu:nIndex

Element selected from the RadioMenu.

Scope:	Assignable
Type:	Numeric
Initial value:	1

1.6.2.25.1.9 TRadioMenu:nMargin

Margin from the control border to the Radiomenu elements.

Scope:	Design assignable
Type:	Numeric
Initial value:	2

1.6.2.25.1.10 TRadioMenu:nOrientation

Indicates the RadioMenu element's orientation.

Scope:	Design assignable
Type:	Numeric
Initial value:	orVERTICAL
Possible values:	orVERTICAL, orHORIZONTAL

1.6.2.25.1.11 TRadioMenu:nSpacing

Separation space among the different RadioMenu columns.

Scope:	Design assignable
Type:	Numeric
Initial value:	10

1.6.2.25.1.12 TRadioMenu:nWidth

Indicates the button width.

Scope:	Assignable
Type:	Numeric
Initial value:	100

1.6.2.25.2 TRadio:Events

Name	
	OnChange
	OnClick

1.6.2.25.2.1 TRadio:OnChange

Event that is produced when the control changes its value.

Parameters	<oSender>:
:	Reference to the object that triggers the event
	<nOldIndex>:
	Last item selected
Return	NIL

value:

1.6.2.25.2 TRadio:OnClick

Event that is produced when the user clicks the button.

Parameters	<oSender>:
:	Reference to the object that triggers the event
Return value:	NIL

1.6.2.26 TShiftPanel

This class represents a panel control with enhanced capabilities, like the ability to show a title with a small icon and a degraded background and the possibility to expand or contract the control showing only its title.



Hierarchy	TScrollingWinControl descendant
See also	TScrollBar
File	\source\ShiftPanel.prg

1.6.2.26.1 TShiftPanel:Properties

■ read Only ■ Assignable ■ Design assignable ■ Run-time assignable

Scope	Name	Type	Initial value
■	IOpened	Logic	.T.
■	nCaptionSize	Numeric	24
■	nClrTitleEnd	Numeric	clGradientInactiveCaption
■	nClrTitleStart	Numeric	clActiveCaption
■	nClrTitleText	Numeric	clCaptionText
■	nIconSize	Numeric	16
■	nOrientation	Numeric	orTOP
■	nVAlignment	Numeric	vaCENTER
■	oFontTitle	Objeto	NIL

1.6.2.26.1.1 TShiftPanel:IOpened

Open-close state.

Scope	Run-time assignable
Type	Logic
Initial value	.T.

1.6.2.26.1.2 TShiftPanel:nCaptionSize

Title size in pixels (width or height depending the orientation).

Scope	Assignable
Type	Numeric
Initial value	24

1.6.2.26.1.3 TShiftPanel:nClrTitleEnd

Final background title color.

Scope	Assignable
Type	Numeric
Initial value	clGradientInactiveCaption

Possible colors (Check appendix for possible values)

1.6.2.26.1.4 TShiftPanel:nClrTitleStart

Initial title background color.

Scope	Assignable
Type	Numeric
Initial value	clActiveCaption

Possible colors (Check appendix for possible values)

1.6.2.26.1.5 TShiftPanel:nClrTitleText

Text color.

Scope	Assignable
Type	Numeric
Initial value	clCaptionText

Possible colors (Check appendix for possible values)

1.6.2.26.1.6 TShiftPanel:nIconSize

Icon size to be drawn near the control title.

Scope	Assignable
Type	Numeric
Initial value	16

1.6.2.26.1.7 TShiftPanel:nOrientation

Control orientation.

Scope:	Design assignable
Type:	Numeric
Initial value:	orTOP
Valores posibles:	orTOP, orBOTTOM, orLEFT, orRIGHT

1.6.2.26.1.8 TShiftPanel:nVAlignment

Control vertical alignment.

Scope	Assignable
Type	Numeric
Initial value	vaTOP
Possible values	vaTOP, vaBOTTOM, vaCENTER

1.6.2.26.1.9 TShiftPanel:oFontTitle

Title font.

Scope	Assignable
Type	Objeto
Initial value	NIL

1.6.2.26.2 TShiftPanel:Events

Name
OnClose
OnClosed
OnOpen
OnOpened

1.6.2.26.2.1 TShiftPanel:OnClose

Event triggered when the control is closing.

Parameters:	<oSender> : Object that triggers the event.
Return value:	<NIL Logical> A false return value stops the closing process

1.6.2.26.2.2 TShiftPanel:OnClosed

Event triggered when the control is closed.

Parameters:	<oSender> : Object that triggers the event.
Return value:	<NIL>

1.6.2.26.2.3 TShiftPanel:OnOpen

Event triggered when the control is opening.

Parameters:	<oSender> : Object that triggers the event.
Return value:	<NIL Logical> A false return value stops the opening process

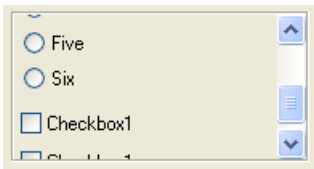
1.6.2.26.2.4 TShiftPanel:OnOpened

Event triggered when the control is opened.

Parameters:	<oSender> Object that triggers the event.
Return value:	<NIL>

1.6.2.27 TScrollBar

This class represents a container control that can show scroll bars in the case that the internal controls don't fit in it.



Hierarchy	Inherits from TScrollingWinControl
See also	TBevel, TPanel
File name	\source\ScrollBar.prg

1.6.2.27.1 TImage:Properties

■ read Only ■ Assignable ■ Design assignable ■ Run-time assignable

Scope	Name	Type	Initial value
■	ITransparent	Logic	.F.
■	nBorderStyle	Numeric	bvETCHED
■	nHeight	Numeric	100
■	nWidth	Numeric	100

1.6.2.27.1.1 TScrollBar:ITransparent

The control is drawn in transparent mode in the container where it belongs.

Scope	Assignable
Type	Logic
Initial value	.F.

1.6.2.27.1.2 TScrollBar:nBorderStyle

Indicates the border style.

Scope	Assignable
Type	Numeric
Initial value	bvETCHED
Possible values	bvNONE, bvRAISED, bvSUNKEN, bvBUMP, bvETCHED, bvFLAT

1.6.2.27.1.3 TScrollBar:nHeight

Indicates the control height.

Scope	Assignable
Type	Numeric
Initial value	100

1.6.2.27.1.4 TScrollBar:nWidth

Indicates the control width.

Scope	Assignable
Type	Numeric
Initial value	100

1.6.2.28 TSearchComboBox

This class represents a extended ComboBox with automatic search support.

Hierarchy	Inherits from TComboBox
File	\source\SearchComboBox.prg

1.6.2.28.1 TSearchComboBoxEx:Properties

■ read Only ■ Assignable ■ Design assignable ■ Run-time assignable

Scope	Name	Type	Initial value
■	IExact	Lógico	.F.
■	INoCase	Lógico	.T.

1.6.2.28.1.1 TSearchComboBox:IExact

if true, if the input text coincides with any altems element, its letter case will be changed to also fully coincide with it.

Scope:	Assignable
Type:	Logical
Initial value:	.F.

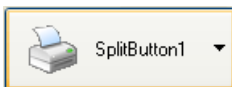
1.6.2.28.1.2 TSearchComboBox:INoCase

If true the search will not distinguish between upper and lower case.

Scope:	Assignable
Type:	Logical
Initial value:	.T.

1.6.2.29 TSplitButton

This class represents a TBtnBmp Button control, but with the possibility to show a menu with it.



Hierarchy	Inherits from TBtnBmp
See also	TBtnBmp
File name	\source\SplitButton.prg

1.6.2.29.1 TSplitButton:Properties

■ read Only ■ Assignable ■ Design assignable ■ Run-time assignable

Scope	Name	Type	Initial value
■	IMenuOnRight	Logic	.F.
■	oMenu	Object	NIL

1.6.2.29.1.1 TSplitButton:IMenuOnRight

If it is .T. the menu will be show on the right side of the control.

Scope:	Assignable
Type:	Logic

Initial value: .F.

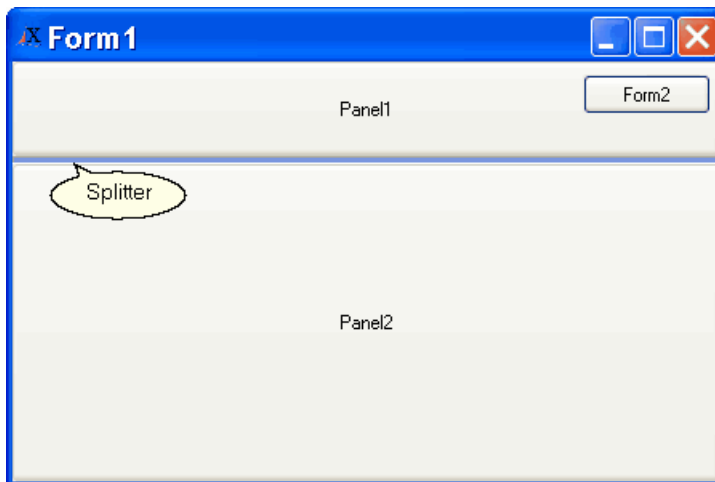
1.6.2.29.1.2 TSplitButton:oMenu

TMenu object to show.

Scope:	Assignable
Type:	Object
Initial value:	NIL

1.6.2.30 TSplitter

This class represents a splitter control. The splitter control is a bar type control, with horizontal or vertical orientation, that allows to modify the control dimensions, with an alignment, that can be before or after it, and inside the same container control, like a form or panel.



Description:

This control must be created just after other control that has a value different to aNONE and aCLIENT in the nAlign property, and it should have the same alignment.

Hierarchy	Inherits from TControl
File name	\source\Splitter.prg

1.6.2.30.1 TSplitter:Properties

read Only
 Assignable
 Design assignable
 Run-time assignable

Scope	Name	Type	Initial value
<input type="checkbox"/>	IGripper	Logic	.F.

■	nBorderStyle	Numeric	bvSUNKEN
■	nClrPane	Numeric	clInactiveCaption
■	nHeight	Numeric	5
■	nMaxPos	Numeric	0
■	nMinPos	Numeric	0
■	nMinSize	Numeric	20
■	nWidth	Numeric	5

1.6.2.30.1.1 TSplitter:IGripper

If true the control will show a gripper.

Scope	Assignable
Type	Logic
Initial value	.F.

1.6.2.30.1.2 TSplitter:nBorderStyle

Indicates the style used to draw the border.

Scope	Assignable
Type	Numeric
Initial value	bvSUNKEN
Possible values	bvNONE, bvRAISED, bvSUNKEN, bvBUMP, bvETCHED, bvFLAT

1.6.2.30.1.3 TSplitter:nClrPane

Indicates the background color.

Scope	Assignable
Type	Numeric
Initial value	clInactiveCaption

Consult the appendix for the list of available colors

1.6.2.30.1.4 TSplitter:nHeight

Indicates the control's height.

Scope	Assignable
--------------	------------

Type	Numeric
Initial value	5

1.6.2.30.1.5 TSplitter:nMaxPos

Maximum position that can move the control. A zero values means no limit.

Scope	Assignable
Type	Numeric
Initial value	0

1.6.2.30.1.6 TSplitter:nMinPos

Minimum position that can move the control.

Scope	Assignable
Type	Numeric
Initial value	0

1.6.2.30.1.7 TSplitter:nMinSize

Indicates the minimum adjusted space allowed.

Scope	Assignable
Type	Numeric
Initial value	20

1.6.2.30.1.8 TSplitter:nWidth

Indicates the control width.

Scope	Assignable
Type	Numeric
Initial value	5

1.6.2.30.2 TSplitter:Events

Name	OnMove
-------------	--------

1.6.2.30.2.1 TSplitter:OnMove

Event that is triggered when there is an adjustment in the related controls.

Type	Standard
Parameters	<oSender> Reference to the object that triggers the event
Return value	NIL

1.6.2.31 Browses

1.6.2.31.1 TBrowse

This class is the base class for all the Xailer Browses, like TArrayBrowse, TDbfBrowse and TDBBrowse. You might not need to instantiate directly a TBrowse class object.

FIRST	LAST	STREET	CITY	NPAGO
CLIFFORD	ABELSON	31831 WALNUT STREET	RIVER EDGE	2
BRUCE	ABELSON	22236 W WALNUT	LEES SUMMI	5
DOMINIC	ACKER	11414 GALLERIA BLVD	WILLIAMSTO'	3
DANIEL	ADIOS	2457 GREENWAY DR	VACAVILLE	1
KANDASAMY	AGUAYO	26220 CREEK CROSSING	BETHLEHEM	1
KANDASAMY	AGUAYO	13645 FOCHT AVENUE	MONTVILLE	1
LEN	AMARAL	1230 REYGER ST	KOWLOON B,	1
BARRY	AMMANN	5440 S.W. BLUE INN COU	SOLANA BEA	1
DAVID	BAILEY	9059 MAXFIELD CT	SCOTTSDALE	1

Hierarchy Inherits from TWinControl
See also TArrayBrowse, TDbfBrowse, TDBBrowse
File name \source\Browse.prg

1.6.2.31.1.1 TBrowse:Properties

read Only
 Assignable
 Design assignable
 Run-time assignable

Scope	Name	Type	Initial value
<input type="checkbox"/>	aCols	Array	{}
<input type="checkbox"/>	aSelected	Array	{}
<input type="checkbox"/>	IAllowAppend	Logic	.F.
<input type="checkbox"/>	IAllowColHiding	Logic	.T.

■	IAllowColSizing	Logic	.T.
■	IAllowRowSizing	Logic	.T.
■	IAllowColSwapping	Logic	.T.
■	IAllowDelete	Logic	.F.
■	IAllowEdit	Logic	.T.
■	IAllowInsert	Logic	.F.
■	IColDividerComplete	Logic	.T.
■	IAutoSave	Logic	.F.
■	IEditMode	Logic	.F.
■	IFastEdit	Logic	.F.
■	IFilterBar	Logic	.F.
■	IFooter	Logic	.F.
■	IHScroll	Logic	bsAUTO
■	IHeader	Logic	.T.
■	IRecordSelector	Logic	.T.
■	ITabStop	Logic	.T.
■	IVScroll	Logic	bsAUTO
■	nBorderStyle	Numeric	bvETCHED
■	nClrAltPane	Numeric	cWindow
■	nClrPane	Numeric	cWindow
■	nClrSelFocusPane	Numeric	cHighLight
■	nClrSelFocusText	Numeric	cWhite
■	nClrSelPane	Numeric	cInactiveCaptionText
■	nClrSelText	Numeric	cWindowText
■	nClrText	Numeric	cWindowText
■	nColDividerStyle	Numeric	biNOLINES
■	nColSel	Numeric	0
■	nDataLines	Numeric	1
■	nDataRows	Numeric	0
■	nEditMode	Numeric	beREAD
■	nFooterHeight	Numeric	24
■	nFreeze	Numeric	0
■	nHeaderHeight	Numeric	24
■	nHeight	Numeric	120
■	nKeyNo	Numeric	0
■	nLen	Numeric	0
■	nMarqueeStyle	Numeric	bmSOLIDCELL
■	nRowDividerStyle	Numeric	biNOLINES
■	nRowHeight	Numeric	0
■	nRowSel	Numeric	0
■	nWidth	Numeric	150
■	oHeader	Object	NIL
■	oFooter	Object	NIL
■	oImageList	Object	TImageList
■	oRecSel	Object	NIL
■	oSeek	Object	NIL

TBrwColumn object arrays with all the browse columns definitions.

Scope	Design assignable
Type	Array
Initial value	{}

Array with all the current selected rows from the browse. This property is used together with the nMarqueeStyle property setting its value to bmHIGHLROWMS. Every array element is returned by the OnBookMark event.

Scope	Assignable
Type	Array
Initial value	{}

If true the automatic append mode is permitted inside the browse by just pushing the down arrow cursor key beyond the last data line. You may also force the append mode state using the Append method.

Scope	Assignable
Type	Logic
Initial value	.F.

If it is .T., it is possible to hide specific columns during run-time. Clicking the right mouse button in any part of the header will show a context menu with all the available columns and then it is possible to hide any of them.

Scope	Assignable
Type	Logic
Initial value	.T.

If it is .T., it is possible to change the column width.

Scope	Assignable
Type	Logic

Initial value	.T.
----------------------	-----

If it is .T., it is possible to modify the browse row width. The IRecordSelector property must be .T..

Scope	Assignable
Type	Logic
Initial value	.T.

If it is .T., it is possible to change the columns position.

Scope	Assignable
Type	Logic
Initial value	.T.

If true the automatic deletion is permitted inside the browse by just pushing the key. You may also force the deletion using the Delete method.

Scope	Assignable
Type	Logic
Initial value	.F.

If true the automatic edition is permitted inside the browse by just pushing the <RETURN> key or double-clicking the mouse. You may also force the deletion using the Edit method.

Scope	Assignable
Type	Logic
Initial value	.T.

If true the automatic insert mode is permitted inside the browse by just pushing the <INSERT> key. You may also force the Insert mode state using the Insert method.

Scope	Assignable
Type	Logic
Initial value	.F.

If true, the append, insert, edit and delete processes are completely automatic. However you may modify the behaviour trapping the OnPostEdit event. Consult the documentation about this event for further information.

Ámbito	Asignable
Tipo	Lógico
Valor inicial	.F.

If it is .T. and there is an indicator to use vertical lines in the browse with the nColDividerStyle property, those lines will paint the browse from the header to the footer. Otherwise, they will be painted to the last browse row.

Scope	Asignable
Type	Logic
Initial value	.T.

True if the browse is in edit mode.

Scope	Read only
Type	Logic
Initial value	.F.

If it is .T., the browse enters automatically to edit mode when the user presses any key that is not a navigation key if the current active column allows the edition (TBrwColumn:nEditType) and its property IAllowEdit is set to true. That pressed key is introduced automatically to the edit field as the first character.

Scope	Asignable
Type	Logic
Initial value	.F.

If it is .T. the browse header will be shown with some additional fields to catch the values to filter the data. This property is used together with the OnFilterChange event.

Scope	Assignable
Type	Logic
Initial value	.F.

If it is .T., the browse will have a footer.

Scope	Assignable
Type	Logic
Initial value	.F.

Establishes the way to show the horizontal scroll bar.

Scope	Assignable
Type	Logic
Initial value	bsAUTO
Possible values	bsAUTO, bsYES, bsNO

If it is .T., the browse will have header.

Scope	Assignable
Type	Logic
Initial value	.T.

If it is .T., it shows in the browse an additional column to the left that allows to change the row width if IAllowRowSizing is .T. and it shows a small triangle indicating the current active row.

Scope	Assignable
Type	Logic
Initial value	.T.

The control receives the focus when the user press the TAB key.

Scope	Design assignable
--------------	-------------------

Type	Logic
Initial value	.T.

Establishes the way that will be shown the vertical scroll bar.

Scope	Assignable
Type	Logic
Initial value	bsAUTO
Possible values	bsAUTO, bsYES, bsNO

Indicates the style to use to draw the control border.

Scope	Assignable
Type	Numeric
Initial value	bvETCHED
Possible values	bvNONE, bvRAISED, bvSUNKEN, bvBUMP, bvETCHED, bvFLAT

Indicates the alternative color background.

Scope:	Assignable
Type:	Numeric
Initial value:	clWindow

The control background color is established as usual by the nClrPane. However the browse allows to establish an alternative color for the even lines to provide the effect of 'accounting paper' in these type of controls.

(See also the appendix, to check the colors available)

Indicates the control background color.

Scope:	Assignable
Type:	Numeric
Initial value:	clWindow

If you want to provide the 'accounting paper' effect, use the nClrAltPane property.

(See also the appendix, to check the colors available)

Indicates the standard background color for the current active row when it has the focus.

Scope:	Assignable
Type:	Numeric
Initial value:	clHighLight

(See also the appendix, to check the colors available)

Standard text color for the current active row when it has the focus.

Scope:	Assignable
Type:	Numeric
Initial value:	clWhite

(See also the appendix, to check the colors available)

Standard background color for the current active row.

Scope:	Assignable
Type:	Numeric
Initial value:	clInactiveCaptionText

(See also the appendix, to check the colors available)

Standard color text for the active row.

Scope:	Assignable
Type:	Numeric
Initial value:	clWindowText

(See also the appendix, to check the colors available)

Indicates the standard text color.

Scope:	Assignable
Type:	Numeric
Initial value:	clWindowText

(See also the appendix, to check the colors available)

Indicates the style to divide the columns.

Scope	Assignable
Type	Numeric
Initial value	bINOLINES
Possible values	bINOLINES, bBLACK, bDARKGRAY, bFORECOLOR, bLIGHTGRAY, bINSET, bRAISED

Indicates the column number for the current column in the Browse. This property provides the number from the current active column from the visible columns in the browse. To get the column object of that position, use the ColAtPos method.

Scope:	read Only
Type:	Numeric
Initial value:	0

Number of lines per row to be shown in the Browse. This property allows to indicate data rows that might have more than one text rows allowing to show labels that includes more than one line. If the nRowHeight property has not been assigned, the Browse will calculate automatically the row height based in this property and the control font.

When the value of this property is bigger than 1 it allows to visualize multi line text.

Scope:	Assignable
Type:	Numeric
Initial value:	1

Number of lines with data that is currently shown in the browse.

Scope:	read Only
Type:	Numeric
Initial value:	0

Actual edition state.

Scope:	Read only
---------------	-----------

Type:	Numeric
Initial value:	beREAD
Possible values:	beREAD beEDIT beAPPEND beINSERT

Indicates the footer height.

Scope	Assignable
Type	Numeric
Initial value	24

Indicates the number of columns to be frozen to the left of the browse.

Scope	Assignable
Type	Numeric
Initial value	0

Indicates the header height

Scope	Assignable
Type	Numeric
Initial value	24

Indicates the control height.

Scope	Assignable
Type	Numeric
Initial value	120

Indicates the relative row position from the total number of browse rows. This number is gotten internally by the browse calling to the KeyNo method.

Scope:	read Only
Type:	Numeric
Initial value:	0

Indicates the total number of rows in the browse. This number is gotten internally by the browse calling to the KeyCount method.

Scope:	read Only
Type:	Numeric
Initial value:	0

Indicates the style for the current row and selected column.

Scope	Assignable
Type	Numeric
Initial value	bmNOMARQUEE
Possible values	bmNOMARQUEE, bmDOTEDCELL, bmSOLIDCELL, bmHIGHLCCELL, bmHIGHLOWRC, bmHIGHLOW, bmHIGHLOWMS

Indicates the row divider style.

Scope	Assignable
Type	Numeric
Initial value	bINOLINES
Possible values	bINOLINES, bBLACK, bDARKGRAY, bFORECOLOR, bLIGHTGRAY, bINSET, bRAISED

Indicates the control row height. If the values is not assigned and keeps its zero initial value, the Browse will calculate automatically the needed height to show the row data based in the font used and the nDataLines property.

Scope	Assignable
Type	Numeric
Initial value	0

Current active row number in the browse. This property provides the current active row number from the visible rows in the browse.

Scope:	readOnly
Type:	Numeric
Initial value:	0

Indicates the control width.

Scope	Assignable
Type	Numeric
Initial value	150

THeader object that shows the control's header. This property only has a value after the Browse has been created and when the tHeader property is set to .T..

Scope	readOnly
Type	Object
Initial value	NIL

IFooter object that shows the control's footer. This property only has a value after the Browse has been created and when the IFooter property is set to .T..

Scope	readOnly
Type	Object
Initial value	NIL

TImageList object with all the images that will be used by the control.

Scope	Assignable
Type	Object
Initial value	TImageList()

The TImageList object is instantiated since the control creation through its New(0 constructor. The first image included in the TImageList establishes the image dimensions. If the first image is an image that includes more than one bitmap, is important that you establishes the TImageList nHeight and nWidth properties before to add a bitmap.

TBrwRecSel control object that shows the record select column. This property only has a value after the Browse creation and when the IRecordSelector property is set to .T..

Scope	read Only
Type	Object
Initial value	NIL

TLabel object type or other control that contains the cText property to show the current search value in the search incremental processes through the OnSeek event.

Scope	Assignable
Type	Object
Initial value	NIL

1.6.2.31.1.2 TBrowse:Methods

■ Constructor ■ Standard ■ Only after Create()

Type	Name
■	AddCol
■	AddImage
■	Adjust
■	AdjustCols
■	Append
■	Bookmark
■	CancelEdit
■	ClearAllFilters
■	ColAtIndex
■	ColAtPos
■	ColPos
■	ColWithHeader
■	DelCol
■	Delete
■	Edit
■	GoBottom
■	GoDown
■	GoHPos

■	GoLeft
■	GoLeftMost
■	GoPageDown
■	GoPageUp
■	GoRight
■	GoRightMost
■	Goto
■	GoTop
■	GoUp
■	InsCol
■	Insert
■	IsBof
■	IsEditable
■	IsEof
■	IsMultipleEdit
■	KeyCount
■	KeyNo
■	MouseColPos
■	MouseRowPos
■	MoveCol
■	Refill
■	Refresh
■	RefreshCurrent
■	Reset
■	RestoreState
■	SaveState
■	Seek
■	Select
■	SelectAll
■	SelectCellFromPoint
■	SelectCol
■	SelectNone
■	SelectedCol
■	StretchCols
■	SwapCols
■	ToExcel

Adds a new column to the Browse object.

Type	Standard
Parameters	<p>[<oCol>]: Reference to a TBrwColumn object. If it is not passed as parameter, the method will create one automatically</p> <p>[<IRefresh>]:</p>

	If it is .T. it will refresh the browse to show the new column. Default: .F.
Return value	<oCol> Reference to a TBrwColumn object.

Adds a new image to the olmageList control object.

Type	Standard
Parameters	<xImage>: Resource name, file name or bitmap handle. <IMasked>: If it is .T., the image will be converted to mask format, it means that its colors will be converted to gray tones
Return value	<nImage>: Image position number in the olmageList

Forces to recalculate all the internal Browse dimensions. This method is executed internally after the browse it is created but it can be used manually if there are important changes in the browse structure or properties.

Type	Only after Create()
Parameters	[<IForced>]: If it is .T. it forces to recalculate every column width in the browse and its Record Selector. Default: .F.
Return value	NIL

Adjust columns width to its desired width. Is the same effect that if you do doble-click on every column separator.

Type	Only after Create()
Parameters	None
Return value	NIL

Starts an append process.

Type	Only afterCreate()
Parameters	None
Return value	NIL

Retrieves or sets the current physical position.

Type:	Only after Create()
Parameters :	<nPos>: Physical position to move. If the parameter is not provided, it returns the current position.
Return value:	<nPos>: Current logical position

Cancel the edition (if active).

Type	Only after Create()
Parameters	[<IAbort>]: If it is .T. the edition is canceled and it is treated as the user presses the ESC key. Default: .F. [<nKey>]: indicates the last pressed key. If is zero, the users clicked the mouse pointer out of the edit zone. Default: 0
Return value	NIL

Clears any browse filter.

Type	Only after Create()
Parameters	None
Return value	NIL

Returns the column object that was created in the `nIndex` position. Take into consideration that it might not be the same as its current position in the `aCols` array.

Type	Only after Create()
Parameters	<nIndex>: Column order creation
Return value	<oCol> Reference to an TBrwColumn object

Returns the column object that is shown in an specific column in the Browse. Be careful that it might not be the same as its current position in the `aCols` array.

Type	Only after Create()
Parameters	<nDisplayPos>: Display position
Return value	<oCol> Reference to an TBrwColumn object

Returns the **oCol** column current position in the visible columns from the Browse. The column must be truly visible, it means that besides to have its `IVisible` property set to `.T.`, it must be visible in the Browse.

Type	Only after Create()
Parameters	<oCol>: TBrwColumn object where its needed to know its position
Return value	<nPos> Current position, zero if it is not visible

Returns the column object with an specific header text.

Type	Only after Create()
Parameters	<cHeader>: Header text
Return value	<oCol> Reference to an TBrwColumn object

Deletes an specific column from the Browse.

Type	Standard
Parameters	[<nIndex>]: Creation order for the column to be deleted [<IRefresh>]: If it is .T. the Browse will be refreshed to show the changes. Default: .F.
Return value	<ISuccess> .T. if the column has been deleted

Deletes de actual line.

Type	Only afterCreate()
Parameters	None
Return value	NIL

Starts the edition of all the editable visible columns that have their nEditMode property with a value different to beNONE.

Type	Only after Create()
Parameters	None
Return value	NIL

Moves the record pointer to the last row in the Browse.

Type	Only after Create()
Parameters	None
Return value	NIL

Moves the record pointer one unit down from the current Browse row.

Type	Only after Create()
Parameters	None

Return value	NIL
---------------------	-----

Moves the current active column to the position defined by **nPos**.

Type	Only after Create()
Parameters	<nPos> : New position
Return value	NIL

Moves the current active column one position to the left.

Type	Only after Create()
Parameters	[<IOffset>] : If it is .T., forces that the new active column is the first visible column in the Browse. Default: .F. [<IRefresh>] : .T. if it the Browse must be refreshed. Default: .T.
Return value	NIL

Moves the active column to the first column.

Type	Only after Create()
Parameters	None
Return value	NIL

Moves the record pointer one page down from the current Browse row.

Type	Only after Create()
Parameters	[<nLines>] : Number of lines to move down. Default: number of current visible lines in the Browse
Return value	NIL

Moves the record pointer one page up from the current Browse row.

Type	Only after Create()
Parameters	[<nLines>]: Number of lines to move up. Default: number of current visible lines in the Browse
Return value	NIL

Moves the current active column one position to the right.

Type	Only after Create()
Parameters	[<IOffset>]: If it is .T., forces that the new active column is the first visible column in the Browse. Default: .F. [<IRefresh>]: .T. if it the Browse must be refreshed. Default: .T.
Return value	NIL

Moves the active column to the last column.

Type	Only after Create()
Parameters	None
Return value	NIL

Moves the record pointer to the **nPos** Browse position.

Type	Only after Create()
Parameters	[<nPos>]: Position to move the pointer
Return value	NIL

Moves the record pointer to the first row in the Browse.

Type	Only after Create()
Parameters	None
Return value	NIL

Moves the record pointer one unit up from the current Browse row.

Type	Only after Create()
Parameters	None
Return value	NIL

Inserts a new column to the Browse object.

Type	Standard
Parameters	<nIndex> : New column position [<oCol>] : Reference to an TBrwColumn object. If it is not passes as parameter, the method will create it automatically [<IRefresh>] : If it is .T. it will refresh the Browse to show the new column. Default: .F.
Return value	<oCol> Reference to an TBrwColumn object

Starts an insert process.

Type	Only afterCreate()
Parameters	None
Return value	NIL

Returns true if the browse is at top.

Type	Only after Create()
Parameters	None
Return value	<IBof>: True if at BOF

Returns .T. if any of the visible columns are editable.

Type	Only after Create()
Parameters	None
Return value	<ISuccess> .T. if there are editable columns

Returns true if the browse is at bottom.

Type	Only after Create()
Parameters	None
Return value	<IEof>: True if at EOF

Returns .T. if the control is in edit mode in more than one column.

Type	Only after Create()
Parameters	None
Return value	<ISuccess> .T. if there is more than one column in edit mode

Returns the total number of rows in the Browse. This information is gotten evaluating the OnKeyCount method.

Type	Only after Create()
Parameters	None
Return value	<nTotal> Number of rows

Returns the relative position from the current row in the Browse. This information is gotten evaluating the OnKeyNo method.

Type	Only after Create()
Parameters	None
Return value	<nPos> Current row position

Returns the column number on a specific mouse coordinate.

Type	Only after Create()
Parameters	<nX>: Mouse X coordinate
Return value	<nColPos>: Column position

Returns the line number on a specific mouse coordinate.

Type	Only after Create()
Parameters	<nY>: Mouse X coordinate
Return value	<nLine>: Line number

Moves the **xCol** column to the **nPos** position.

Type	Standard
Parameters	<xCol>: Column object or its position <nPos>: Position to move [<IRefresh>]: .T. if the Browse will be refreshed. Default: .T.
Return value	NIL

Refreshes all the Browse rows and columns, re-reads all the data and establishes as the first Browse column the current active row. It is equivalent to call to the Refresh(.T.) method.

Type	Only after Create()
Parameters	None
Return value	NIL

Refreshes all the Browse rows and columns.

Type	Only after Create()
Parameters	[<IComplete>]: If it is .T. it will re-read all the data and it will establishes as the first Browse row the current active row
Return value	NIL

Refreshes the current Browse row.

Type	Only after Create()
Parameters	None
Return value	NIL

Resets the browse completely, deleting all its columns.

Type	Standard
Parameters	None
Return value	NIL

This method restores the complete Browse configuration: row height and visibility, column width and current position of every column from the string returned by method SaveState.

Type	Only after Create()
Parameters	<cState>

	String with the Browse configuration
Return value	<ISuccess> True if success

This method returns in a single string the complete Browse configuration: row height and visibility, column width and current position of every column.

Type	Only after Create()
Parameters	None
Return value	<cState> String with the current Browse configuration

Searches the **cSeek** string based on the OnSeek event definition.

Type	Only after Create()
Parameters	[<cSeek>]: String to search. If this parameter is not passed, it will reset the incremental search
Return value	NIL

Marks selected rows when the nMarqueeStyle property is set to bmHIGHLOWMS. This style allows to select a group of rows from the Browse that can be recovered through the aSelected array.

Type	Only after Create()
Parameters	<nOperation>: 0 Deletes all the selected rows 1 Adds the current row to the selected rows list 2 Changes the current state of the current row (CTRL-LeftClick) 3 Adds all the rows from the last selected to the current (SHIFT-LeftClick) 4 Selects all the rows
Return value	NIL

Selects all the rows in the browse when the nMarqueeStyle property is bmHIGHLROWMS. It is equivalent to the Select(4) method.

Type	Only after Create()
Parameters	None
Return value	NIL

Selects a row from the Browse according to the coordinate.

Type	Only after Create()
Parameters	<nPosX>: X Coordinate <nPosY>: Y Coordinate
Return value	<ISuccess> .T. if it was possible to select a row

Selects a Browse column according to its display order.

Type	Only after Create()
Parameters	<nCol>: Current column order.
Return value	NIL

De selects all the Browse columns when the nMarqueeStyle property is bmHIGHLROWMS. It is equivalent to the Select(0) method.

Type	Only after Create()
Parameters	None
Return value	NIL

Returns the current selected column.

Type	Only after Create()
-------------	---------------------

Parameters	None
Return value	<oCol> Current column

Reduces columns width proportionally so all the columns (from current column position) becomes visible in the browse.

Type	Only after Create()
Parameters	None
Return value	NIL

Interchanges the position for two columns.

Type	Only after Create()
Parameters	<xCol1>: First column object or its position <xCol2>: Second column object or its position [<IRefresh>]: .T. is the browse must be refreshed. Default: .T.
Return value	NIL

Send control data to Excel. Excel application must be present on the PC.

Type	Only after Create()
Parameters	<[cFile]> Name of the XLS file to create. If this parameter is left blank, the report is sent to Excel and when it is finished the spreadsheet is displayed. <[IXIsNumeric]> If true, all columns of numerical type will be transferred to Excel, also in numerical form. By default .T.
Return value	.T. if success

1.6.2.31.1.3 TBrowse:Events

Name
OnAppend
OnBof
OnBookMark
OnChange
OnChangeHeight
OnChangeSelection
OnClick
OnDbfClick
OnDelete
OnDrawRow
OnEdit
OnEof
OnFilterChange
OnGoBottom
OnInsert
OnGoTop
OnKeyCount
OnKeyNo
OnNewItem
OnPastEof
OnPostEdit
OnRClick
OnSeek
OnSkip

Event that is triggered before the browse row append operation.

Parameters	<oSender> : Reference to the object that triggers the event
Return value	<lSuccess> : If returns a .F. logic value, the append process is stopped

Event that is produced when the user asks to the object if it is at Bof. If defined, the default operation done by its inherited class will be superseded.

Parameters:	<oSender> : Reference to the object that triggers the event
Return value:	<lOnBof> : .T. if the record pointer is located at the first

data row.

Example:

```
oBrw:OnBof := { |o| Clients->( Bof() ) }
```

Event that is produced when there is a question about the current physical position or when there is a change in the current physical position. **xPos** does not need to be a number. If defined, the default operation done by its inherited class will be superseded.

Parameters:	<p><oSender>: Reference to the object that triggers the event</p> <p><xPos>: Physical position to jump to. If the parameter is not provided then it returns the current position.</p>
Return value:	<p><xPosActua>: Current physical position</p>

Example:

```
oBrw:OnBookMark := { |o, n| iif(n== nil, Recno(), DbGoto(n)) }
```

Event that is produced when there is change in the current browse position. It is triggered for the changes in the current row or column. In the case of operations like a mouse click that changes both values at the same time, this event can be triggered twice, one for the change in the row and other for the change in the column.

Parameters:	<p><oSender>: Reference to the object that triggers the event</p> <p><IBookmarkChange>: .T. if the event is produced by a change in the row</p>
Return value:	NIL

Event that is produced when there is a change in the row height in the browse.

Parameters:	<p><oSender>: Reference to the object that triggers the event</p> <p><nNewHeight>: New row height</p> <p><nOldHeight>: Old row height</p>
--------------------	--

Return value:	<ICancel> If it returns a .F. logic value, the operation to change the row height is aborted
----------------------	--

Event that is produced when there a change on the rows selected is produced. .This only happens on multi-select mode, which is when the property nMarqueeStyle is se to bmHIGHLROWMS.

Parameters:	<oSender> : Reference to the object that triggers the event <IBookmarkChange> : .T. if the event is produced by a change in the row
Return value:	NIL

Event that is produced when the user clicks the mouse pointer inside the control. This event is not triggered if has been previously processed by the column's OnDataClick event and has been returned a value different than NIL.

Parameters:	<oSender> : Reference to the object that triggers the event <nFlags> : Keyboard and/or mouse status. It can be a combination of the following values: MK_CONTROL CTRL key pressed MK_LBUTTON Left mouse button pressed MK_MBUTTON Central mouse button pressed MK_RBUTTON Right mouse button pressed MK_SHIFT Shift key pressed Use the IAnd() function to check status <nXPos> : Cursor X coordinate <nYPos> : Cursor Y coordinate
Return value:	NIL

Event that is triggered before the browse row delete operation.

Parameters:	<oSender> : Reference to the object that triggers the event
Return value:	<ISuccess> : If returns a .F. logic value, the delete process

is stopped

Event that is produced when the user double clicks the mouse pointer inside the control. The event is not triggered if previously has been processed by the column's OnDataDbClick event and has been returned a value different than NIL.

Parameters:	<oSender>: Reference to the object that triggers the event <nFlags>: Keyboard and/or mouse status. It can be a combination of the following values: MK_CONTROL CTRL key pressed MK_LBUTTON Left mouse button pressed MK_MBUTTON Central mouse button pressed MK_RBUTTON Right mouse button pressed MK_SHIFT Shift key pressed Use the IAnd() function to check status <nXPos>: Cursor X coordinate <nYPos>: Cursor Y coordinate
Return value:	NIL

Event that is produced before to paint a browse row.

Parameters:	<oSender>: Reference to the object that triggers the event <nRowPos>: Current row position from the row to be painted. It is the relative position inside the browse window
Return value:	NIL

Event that is triggered before the browse row edition.

Parameters:	<oSender>: Reference to the object that triggers the event
Return value:	<ISuccess>: If returns a .F. logic value, the edit process is stopped

Event that is produced when the user asks to the object if it is at Eof. If defined, the default operation done by its inherited class will be superseded.

Parameters:	<oSender>: Reference to the object that triggers the event
Return value:	<IOnEof>: .T. if the cursor is located at the end of the last row of data.

Example:

```
oBrw:OnEof := { |o| Clients->( Eof() ) }
```

Event that is produced when the user types a text in any of the filter fields in the oHeader object. This event is caught automatically by the lower hierarchy classes to adapt the data type to be shown.

Parameters:	<oSender>: Reference to the object that triggers the event <nColumn>: Column that triggers the event. When the filter bar is enabled/disabled, the event will be triggered and in that case this parameter has a zero value.
Return value:	NIL

Example:

```
oBrw:OnFilterChange := { |o,n| SetFilter( n ) }
```

Event that is produced when the user tries to move the cursor to the last row. If defined, the default operation done by its inherited class will be superseded.

Parameters:	<oSender>: Reference to the object that triggers the event
Return value:	NIL

Example:

```
oBrw:OnGoBottom := { |o, n| Clients->( DbGoBottom() ) }
```

Event that is produced when the user tries to move the cursor to the first row. If defined, the default operation done by its inherited class will be superseded.

Parameters:	<oSender>: Reference to the object that triggers the event
Return value:	NIL

Example:

```
oBrw:OnGoBottom := {|o, n| Clients->( DbGoTop() ) }
```

Event that is triggered before the browse row insertion.

Parameters:	<oSender>: Reference to the object that triggers the event
Return value:	<!Success>: If returns a .F. logic value, the insertion process is stopped

Event that is produced when the user asks for the total number of rows. If defined, the default operation done by its inherited class will be superseded.

Parameters:	<oSender>: Reference to the object that triggers the event
Return value:	<nTotal>: Total number of rows

Example:

```
oBrw:OnKeyCount := {|o, n| Clients->( OrdKeyCount() ) }
```

Event that is produced when the user asks for the current logical position or when the user wants to change the current logic position. If defined, the default operation done by its inherited class will be superseded.

Parameters:	<oSender>: Reference to the object that triggers the event
--------------------	--

	<nPos>: Logic position to move. If the parameter is not provided, it returns the current position.
Return value:	<nActual>: Current logical position

Example:

```
oBrw:OnKeyNo := {|o, n| iif(n== nil, OrdKeyNo(), OrdKeyGoto(n))}
```

Event that is produced when it is created a new column in the browse.

Parameters:	<oSender>: Reference to the object that triggers the event <oCol>: TBrwColumn object
Return value:	NIL

Event that is produced when the user tries to move beyond the last row with the keyboard.

Parameters:	<oSender>: Reference to the object that triggers the event
Return value:	<IValue>: IF it returns NIL and the property IAllowAppend is set to true an automatic append will be performed

Event that is produced when the edit process is finalized after the same event is produced in every columns that were edited.

Parameters:	<oSender>: Reference to the object that triggers the event <@ICanceled>: .T. if the user has canceled the edition with the ESC key <nKey>: Last key pressed. If the value is zero, the user ended the edition with a mouse click out of the edit zone
Return value:	NIL

When IAutoSave is set to true, through this event you may modify if the value should be saved or not modifying its ICanceled parameter.

If IAutoSave is set to false, this event will be the responsible to save all the changes that were produced in the edition. If you don't catch this method all the changes in the column will be lost, except if you have already save the information on the column OnPostEdit event.

Event that is produced when the user right clicks the mouse pointer inside the control This event is not triggered if has been processed before by the column's OnDataRClick event and this has returned a value different to NIL.

Parameters:	<p><oSender>: Reference to the object that triggers the event</p> <p><nFlags>: Keyboard and/or mouse status. It can be a combination of the following values:</p> <p>MK_CONTROL CTRL key pressed</p> <p>MK_LBUTTON Left mouse button pressed</p> <p>MK_MBUTTON Central mouse button pressed</p> <p>MK_RBUTTON Right mouse button pressed</p> <p>MK_SHIFT Shift key pressed</p> <p>Use the IAnd() function to check status</p> <p><nXPos>: Cursor X coordinate</p> <p><nYPos>: Cursor Y coordinate</p>
Return value:	NIL

Event that is produced when an incremental search is done inside the browse, for example, when the user presses an alphanumeric key. It is used together with the oSeek property.

Parameters:	<p><oSender>: Reference to the object that triggers the event</p> <p><cSeek>: String to search</p>
Return value:	<p><ISuccess>: True if the string is found</p>

Example:

```
oBrw:OnSeek := { |o,c| DbSeek( Upper( c ) ) }
```

Event that is produced when the user tries to modify the current row. This event is caught automatically by the lower hierarchy classes to adapt the data type to be shown.

Parameters:	<oSender>: Reference to the object that triggers the event <nSkip>: Number of rows to skip (it can be a negative number)
Return value:	<nSkipped>: Number of skipped rows

Example:

```
oBrw:OnSkip := { |o, n| DbSkipper(n) }
```

1.6.2.31.2 TArrayBrowse

This class is used to show and edit arrays in a browse control type.

	FIRST	LAST	STREET	CITY	NPAGO
▶	CLIFFORD	ABELSON	31831 WALNUT STREET	RIVER EDGE	2
	BRUCE	ABELSON	22236 W WALNUT	LEES SUMMI	5
	DOMINIC	ACKER	11414 GALLERIA BLVD	WILLIAMSTO'	3
	DANIEL	ADIOS	2457 GREENWAY DR	VACAVILLE	1
	KANDASAMY	AGUAYD	26220 CREEK CROSSING	BETHLEHEM	1
	KANDASAMY	AGUAYD	13645 FOCHT AVENUE	MONTVILLE	1
	LEN	AMARAL	1230 REYGER ST	KOWLOON B	1
	BARRY	AMMANN	5440 S.W. BLUE INN COU	SOLANA BEA	1
	DAVID	BAILEY	9059 MAXFIELD CT	SCOTTSDALE	1

Hierarchy Inherits from TBrowse
See also TDbfBrowse, TDBBrowse
File name \source\ArrayBrowse.prg

1.6.2.31.2.1 TArrayBrowse:Properties

■ read Only ■ Assignable ■ Design assignable ■ Run-time assignable

Scope	Name	Type	Initial value
■	aArrayData	Array	{}
■	IAutoOrder	Logic	.F.
■	nArrayAt	Numeric	1

Multidimensional array with the data to be shown in the browse.

Scope:	Assignable
Type:	Array
Initial value:	{}

The array format is the following:

```
{ {Row1Col1, Row1Col2, ..., Row1ColN }, ..., {RowNCol1, RowNCol2, ..., RowNColN } }
```

If TRUE the browse could be sorted by any column by just clicking on its header. A second click will change the sort from ascending to descending or viceversa.

Scope:	Assignable
Type:	Logic
Initial value:	.F.

Element from the active aArrayData array. it corresponds to the browse row that will be shown as active.

Scope:	Assignable
Type:	Numeric
Initial value:	1

1.6.2.31.2.2 TArrayBrowse:Methods

■ Constructor ■ Standard ■ Only after Create()

Typ Name	Legend
■ AddRow	Standard
■ DelRow	Standard
■ GetRow	Standard
■ GetValue	Standard
■ InsRow	Standard
■ IsFiltered	Standard
■ nArrayOrgAt	Standard
■ nArrayPos	Standard
■ SetArray	Standard
■ SetRow	Standard
■ SetValue	Standard

■ Zap

Adds a new row to the array.

Type	Standard
Parameters	<aRow> : Array with the new row information
Return value	NIL

The row format is as follows:

```
{ Col1, Row1Col2, ..., Row1ColN }
```

Nota: Take into consideration that this method will refresh the browse. If you are going to make massive row changes in the browse it is better to manage the aArrayData directly and then refresh the browse with the Refresh method.

Deletes a rows from the array.

Type	Standard
Parameters	[<nRow>] : Row number to delete. Default: the current position indicated by nArrayAt
Return value	NIL

Nota: Take into consideration that this method will refresh the browse. If you are going to make massive row changes in the browse it is better to manage the aArrayData directly and then refresh the browse with the Refresh method.

Returns an specific row from the array.

Type	Standard
Parameters	[<nRow>] : Row number to recover. Default: the current position indicated by nArrayAt
Return value	<aRow> Array with the row data

Returns a value from the column in the current row.

Type	Standard
Parameters	[<xCol>]: Column to recover. If it is a number, it corresponds to the array order. If is a label it refers to the column header. Default: current active row
Return value	<Value> Active column value

Nota: If the column is in edit mode the methods updates the value that is being edited.

Inserts a new row in the array.

Type	Standard
Parameters	<aRow>: Array with the new row information <nRow>: New row position
Return value	NIL

The row format is as follows:

```
{ Col1, Row1Col2, ..., Row1ColN }
```

Nota: Take into consideration that this method will refresh the browse. If you are going to make massive row changes in the browse it is better to manage the aArrayData directly and then refresh the browse with the Refresh method.

Returns true if the array is filtered.

Type	Standard
Parameters	None
Return value	<IFiltered>

Returns the actual row position in the original array if a filter is active or zero if is not.

Type	Standard
Parameters	None

Return value	<nRow 0> Current row position
---------------------	------------------------------------

Returns the actual row position in the original array, even if a filter is active.

Type	Standard
Parameters	None
Return value	<nRow> Current row position

Establishes the array data that will be shown by the browse.

Type	Standard
Parameters	<aData>: Data array [<aHeaders>]: Optional array with text header [<IAutoOrder>]: If it is .T. the browse will have an automatically column sort. Default: .F.
Return value	NIL

The array format is as follows:

```
{ {Row1Col1, Row1Col2, ..., Row1ColN }, ..., {RowNCol1, RowNCol2, ..., RowNColN } }
```

Establishes the value for a row in the browse.

Type	Standard
Parameters	<aRow>: Array with the row information <nRow>: Row position to modify
Return value	NIL

The row format is as follows:

```
{ Col1, Row1Col2, ..., Row1ColN }
```

Nota: Take into consideration that this method will refresh the browse. If you are going to

make massive row changes in the browse it is better to manage the aArrayData directly and then refresh the browse with the Refresh method.

Establishes the column value in the current row.

Type	Standard
Parameters	<p>[<xCol>]: Column to modify. If it is a number it corresponds to the array order. If it is a label it is the column header. Default: it refers to the current active column</p> <p><Value>: New value</p> <p>[<nAt>]: Array row to update. By default, current row</p>
Return value	<p><OldValue> Old value</p>

Nota: If the column is in edit mode and <nAt> coincides with the current row the method updates the value that is being edited.

Deletes all the array elements.

Type	Standard
Parameters	None
Return value	NIL

1.6.2.31.2.3 TArrayBrowse:Events

Name	
OnSort	

Event that is triggered when array sort is requested due a click on any column header when its property IAutoOrder is set to true. If this event is not assigned the sort will be made directly by the control itself.

Parameters	<p><oSender>: Reference to the object that triggers the event</p> <p><aData>: Arra to sort</p> <p><oCol>:</p>
-------------------	---

	TBrwColumn column object that triggered the event <nAt>: Ordinal column number to sort referred to the aData array <IDesc>: If true the sort should be of type descend
Return Value:	<aData>: Sorted array

1.6.2.31.3 TDbfBrowse

This class is used to show and edit DBF files in a Browse.

FIRST	LAST	STREET	CITY	NPAGO
CLIFFORD	ABELSON	31831 WALNUT STREET	RIVER EDGE	2
BRUCE	ABELSON	22236 W WALNUT	LEES SUMMI	5
DOMINIC	ACKER	11414 GALLERIA BLVD	WILLIAMSTO	3
DANIEL	ADIOS	2457 GREENWAY DR	VACAVILLE	1
KANDASAMY	AGUAYO	26220 CREEK CROSSING	BETHLEHEM	1
KANDASAMY	AGUAYO	13645 FOCHT AVENUE	MONTVILLE	1
LEN	AMARAL	1230 REYGER ST	KOWLOON B	1
BARRY	AMMANN	5440 S.W. BLUE INN COU	SOLANA BEA	1
DAVID	BAILEY	9059 MAXFIELD CT	SCOTTSDALE	1

Hierarchy	Inherits from TBrowse
See also	TArrayBrowse, TDBBrowse
File name	\source\DbfBrowse.prg

1.6.2.31.3.1 TDbfBrowse:Properties

Scope	Name	Type	Initial value
<input type="checkbox"/> read Only	cAlias	Character	""
<input type="checkbox"/> Assignable	cMsgNoFilter	Character	""

Indicates the Alias for the DBF file used in the Browse.

Scope:	read Only
Type:	Character
Initial value:	""

Nota: To assign the table to use in the Browse, you should use the SetDbf method.

Text message to be shown when there is not possible to filter an specific column. It is used together with the Browse IFilterBar property and the column object FilterEval property.

Scope:	Assignable
Type:	Character
Initial value:	""

The class allows to filter any column by default, however there are some cases where it makes no sense to do it, for example in a column that shows an image. In those cases you should assign the NIL value to the column object FilterEval property and indicate with this property the message that you want to show then the users tries to filter by a column that does not allow to do it. If you leave this property blank, the Browse simply will beep when the is not possible to filter by that specific column.

1.6.2.31.3.2 TDbfBrowse:Methods

■ Constructor ■ Standard ■ Only after Create()

Typ	Name
■	SetDbf

Establishes the DBF table to be shown in the browse.

Type	Standard
Parameters	<cAlias>: Indicates the Alias for the DBF file [<aHeaders>]: Optional array with the text for the column headers
Return value	NIL

1.6.2.31.4 TBrwColumn

This class allows to manage different TBrowse object columns or any of this inherited classes like TArrayBrowse, TDbfBrowse or TDBBrowse.

Every column of the TBrowse object is a TBrwColumn object class.

Besides the TbrwColumn, this class inherits from other three more classes : **TDbfBrwColumn**, **TArrayBrwColumn** and **TDbBrwColumn**.

The first, **TDbfBrwColumn**, is used in the TDbfBrowse columns and it adds the additional property, cFieldName that indicates the table's field to be shown and the property ICanFilter to indicate if is possible to filter on that column.

The second, **TArrayBrwColumn**, is used in the TArrayBrowse.

The third, **TDBBrwColumn** is used in the TDBBrowse columns and adds the additional property, oDataField that indicates the TDataField object to be shown, the property ICanFilter to indicate if it is possible to filter on that column and the property cSortEval to indicate the sort expression of the column when it has automatic sorting (TDBBrowse:IAutoOrder property).

Hierarchy Inherits from TComponent
File name \source\BrwCol.prg

1.6.2.31.4.1 TBrwColumn:Properties

■ read Only ■ Assignable ■ Design assignable ■ Run-time assignable

Scope	Name	Type	Initial value
■	aEditListBound	Array	{}
■	aEditListText	Array	{}
■	cFieldName	Character	""
■	cFilter	Character	""
■	cFooter	Character	""
■	cHeader	Character	""
■	cNullValue	Character	""
■	cPicture	Character	""
■	cSortEval	Character	""
■	cTooltip	Character	""
■	FilterEval	Character or Block	NIL
■	ICanFilter	Logic	.T.
■	IParentFont	Logic	.T.
■	IVisible	Logic	.T.
■	nAlignment	Numeric	taLEFT
■	nBmpAlignmen t	Numeric	taLEFT
■	nBmpMargin	Numeric	10
■	nClrPane	Numeric	oParent:nClrPane
■	nClrText	Numeric	oParent:nClrText
■	nEditLength	Numeric	-1
■	nEdityType	Numeric	beNONE
■	nFooterImage	Numeric	0
■	nHeaderImage	Numeric	0
■	nHeight	Numeric	0
■	nIndex	Numeric	0
■	nPos	Numeric	0
■	nSort	Numeric	bsNONE
■	nWidth	Numeric	0
■	oDataField	Object	NIL
■	oGridEdit	Object	NIL

■	oFont	Object	NIL
■	oFooter	Object	NIL
■	oHeader	Object	NIL
■	oParent	Object	NIL
■	uDefaultValue	Any	NIL

Array with a list of values related to the aEditListText array that will be used in the assignment in the OnPostEdit event.

Scope:	Assignable
Type:	Array
Initial value:	{}

Description:

When the column is editable with an nEditType style, beLISTBOX type or beGET_LISTBOX, the EditListText array takes the literal list with all the elements from the listbox. You can establish the aEditListBound array with any type of values that will be related to every aEditListText value.

Example:

```
WITH OBJECT oCol
  :nEditType      := beLISTBOX
  :aEditListText := { "Cass", "Check", "MoneyOrder" }
  :aEditListBound := { 1, 2, 3 }
END WITH
```

Array with a list of values to show in the listbox when the column be editable and with listbox.

Scope:	Assignable
Type:	Array
Initial value:	{}

Description:

When the column is editable with a nEditType style, beLISTBOX type or beGET_LISTBOX, the EditListText array takes the literal list with all the elements from the listbox. You can establish the aEditListBound array with any type of values that will be related to every aEditListText value.

Example:

```
WITH OBJECT oCol
  :nEditType      := beLISTBOX
  :aEditListText := { "Cass", "Check", "MoneyOrder" }
  :aEditListBound := { 1, 2, 3 }
END WITH
```

Field name in the DBF table associated with its TDbfBrowse container object.

Scope:	Assignable
Type:	Character
Initial value:	""

Nota: This property exists only in the columns created through the TDbfBrowse class. The columns that have this property have a lower inherit level, and its name comes from the **TDbfBrwColumn** class that inherits directly from TBrwColumn.

Filter text. Normally this property is assigned during run-time for the user when the IFilterBar browse property is .T.

Scope:	Assignable
Type:	Character
Initial value:	""

Description:

When the browse whose columns belongs to, has a IFilterBar = .T., the browse header will show an edition fields besides the text, where the user can establish a filter expression for that column.



The cFilter will take that value. This value will be used together with the FilterEval property to establish a filter for the records to be shown.

For TArrayBrowse and TDBBrowse browse you may use the operators '<' y '>' on date and numeric fields.

Footer text. This text only is visible in the case that the Browse has a footer. In other words, if its IFooter property is .T..

Scope:	Assignable
Type:	Character
Initial value:	""

Header text. This text only is visible in the case that the Browse has a header. In other words, if its IHeader property is .T..

Scope:	Assignable
Type:	Character
Initial value:	""

Text to be shown for null values. Only usable with TDBBrowse browse linked to SQL type datasets.

Scope:	Assignable
Type:	Character
Initial value:	""

Mask based in the Xbase standard picture for the oGridEdit control used in the edition.

If the **cPicture** property and the OnGetValue event are defined, it will be used to paint the column with the returned value and transformed by the **cPicture** mask, instead of the OnGetData assigned value.

Scope:	Assignable
Type:	Character
Initial value:	""

Description:

When the column is editable with a nEditType style beGET or beGET_LISTBOX or beGET_BUTTON type, it is possible to establish a mask or picture for the oGridEdit control used in the edition. The mask or picture follows the same CA-Clipper convention.

For more information about the masks/picture, please check the [x]Harbour or Clipper documentation.

Sort expression for the column. Usually there is no need to indicate this property since the sort expression is automatically set based on the column database field name. Anyhow, in some circumstances you may need to indicate that expression, like in case of calculated fields.

Scope:	Assignable
Type:	Character
Initial value:	""

Note: This property is only available with TDBBrowse browses.

Samples:

```
oCol:cSortEval := "first+last" // For DBF tables
oCol:cSortEval := "first ASC, last ASC" // For SQL queries
```

Tooltip to be shown.

Scope:	Assignable
Type:	Character
Initial value:	""

Description:

Pop-up window that shows the Tooltip text when the user keeps the mouse pointer over in the column header.



The Tooltip will be shown only when the column has its header active. In other words, when the IHeader data from its TBrowse container object is .T..

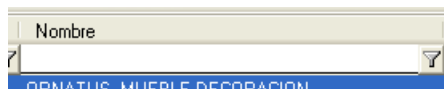
Filter expression to be used.

Scope:	Assignable
Type:	Character or Block
Initial value:	NIL

Note: Block type is only usable with TDbfBrowse objects.

Description:

When the browse that the column belongs to, has the IFilterBar = .T., the Browse header will show an edit field where the user will establish a filter expression for that column.



The cFilter will take that value. This value will be used together with the FilterEval property to establish a filter for the records to be shown.

This property **only makes sense** when it is used with the TDbfBrowse and TDBBrowse classes. The TArrayBrowse class has also a filter mechanism, however it does not need to use this property. The filter expression must be done according with the data type that is being used and

the Tbrowse class type. For example, to use it with the TDbfBrowse class, it's normal to use the field name to do a filter operation. This is a valid expression to filter all the records in the field NAME that contains the cFilter text typed by the user:

```
oCol:FilterEval := "'%' $ UPPER( NAME )" // Type Character
```

You need to take into consideration:

- The text typed by the user and taken by cFilter must be replaced by the % expression.
- Any literal that is incorporated with the expression must be enclosed in " or [].

Examples:

Here you can find some complex filter expression:

```
oCol:FilterEval := "Upper('%') $ Upper(Name)"
oCol:FilterEval := "Upper( '%' ) $ ( Family + ' ' + Upper(" +
::oDbfFam:Alias() + "->Name ) )"
oCol:FilterEval := "Upper( '%' ) $ ( Prov + ' ' + Upper(" +
::oDbfPrv:Alias() + "->Name ) )"
oCol:FilterEval := "Upper('%') = iif( Descatalog, 'Y', 'N' )"
```

If .F. the filter to this column will not be permitted. When you try to introduce a value on the filter field a sound will be played or a warning message with the oBrowse:cMsgNoFilter property will be shown.

Scope:	Assignable
Type:	Logic
Initial value:	.T.

.T. if the front from the Browse will be used.

Scope:	Assignable
Type:	Logic
Initial value:	.T.

If .T., it will show the column in the Browse.

Scope:	Assignable
Type:	Logic
Initial value:	.T.

Description:

This property allows to hide temporarily an specific Browse column. This property is internally used by the Browse to hide a column when the user right clicks the mouse pointer over the header and the oBrowse:AllowColHiding is .T..

The visibility column status is saved with the methods SaveState and RestoreState from its TBrowse container object as well.

Column alignment.

Scope:	Assignable
Type:	Numeric
Initial value:	taLEFT
Possible values:	taLEFT, taRIGHT, taCENTER, taLEFTHEADERRIGHT, taLEFTHEADERCENTER, taRIGHTHEADERLEFT, taRIGHTHEADERCENTER , taCENTERHEADERLEFT, taCENTERHEADERRIGHT

Description:

This property allows to align the column text, the header and the footer . The alignment can be: left, right or centered for the text and/or the header-footer.

Bitmap image column alignment.

Scope:	Assignable
Type:	Numeric
Initial value:	taLEFT
Possible values:	taLEFT, taRIGHT

Description:

This property allows to align an image in the column. The alignment can be left or right. When the text is empty the Bitmap is always aligned centered superseding the value of this property and the nBmpMargin.

For more information about images in the column, see also the `OnGetData` and `OnGetValue` events.

Bitmap image column margin.

Scope:	Assignable
Type:	Numeric
Initial value:	10

Description:

This property allows to establish the margin from the column's border to the image, indicated in pixels. Depending of the use alignment in the bitmap, the border can be the beginning of the column (left alignment) or the end of the column (right alignment). When the text is empty the Bitmap is always aligned centered superseding the value of this property and the `nBmpAlignment`

Column's background color.

Scope:	Assignable
Type:	Numeric
Initial value:	<code>oParent:nClrPane</code>

Consult the appendix for the list of available colors

Column's text color.

Scope:	Assignable
Type:	Numeric
Initial value:	<code>oParent:nClrText</code>

Consult the appendix for the list of available colors

Maximum column length for edition.

Scope:	Assignable
Type:	Numeric
Initial value:	-1

Description:

This property allows to indicate the maximum column length for edition through its `oGridEdit`

control. The -1 value means undetermined length. A value of 0 means that the maximum column length will be set automatically with the length of its initial value.

To edit any column, you should use the `nEditType` property to indicate the edit form and the `OnGetValue` event to indicate the initial value to show. If you want to use a Listbox to show a list of possible values, you need to fill the `aEditListTxt` array.

Edition column's type.

Scope:	Assignable
Type:	Numeric
Initial value:	beNONE
Possible values:	beNONE, beGET, beBUTTON, beLISTBOX, beGET_LISTBOX, beGET_BUTTON

Description:

This property allows to indicate the edit form in the column.

beGET	Edition with a TEdit control
beBUTTON	Edition with a TButton control
beLISTBOX	Edition with a TListbox
beGET_LISTBOX	Edition with a TEdit and a TListbox
beGET_BUTTON	Edition with a TEdit and a TButton

When a column has defined an edition type different than **beNONE** when I press <Enter>, double click over that column, or via code programming, call the `Edit` method to enter directly to the edit mode.

The edition object is a `TGridEdit` object type that you can access through the `oGridEdit` property. The `oGridEdit` object is available in the moment that the `Browse` is created and keeps hidden until enters to the edition mode with the `Edit` method. Then it can work with the events and properties of that control.

The edition is not modal: the program will not stop to wait that the user finishes to edit the cell. Is the programmer responsibility to catch the `OnPostEdit` event and make there all the operation needed. After the `OnPostEdit` event evaluation for all the columns that are in edit mode, the `Browse OnPostEdit` event is evaluated.

To edit any column you should use the `nEditType` property to indicate the edit form and the `OnGetValue` event to indicate the initial value to show. If besides that you want to use a listbox to show a list of possible values, you should fill the `aEditListTxt` array.

Image to show an image in the column's footer.

Scope:	Assignable
Type:	Numeric
Initial value:	0

Description:

This property allows to establish the image to be shown in the column's footer. The image will be shown only if the Browse footer is visible (IFooter = .T.).

The entered numeric value is related to the image order position in the oImageList (TImageList) control of its container browse oParent.

Image to show an image in the column's header.

Scope:	Assignable
Type:	Numeric
Initial value:	0

Description:

This property allows to establish the image to be shown in the column's header. The image will be shown only if the Browse header is visible (IHeader = .T.).

The entered numeric value is related to the image order position in the oImageList (TImageList) control of its container browse oParent.

Indicates the columns height in pixels.

Scope:	Assignable
Type:	Numeric
Initial value:	0

Description:

This property allows to establish the height of every browse column. Normally, this value is calculated automatically based in the used font and the bitmap size (if any). Later, the browse object container will establish the row height based in the maximum column height in the browse.

Once that the browse is created (Create() method), this property should be considered read only, and if you want to modify the height of every row in the browse, you should use the nRowHeight object from the TBrowse container object.

Number that indicates the order in which the column was created in its TBrowse container.

Scope:	read Only
Type:	Numeric
Initial value:	0

Description:

This property allows to know the column creation order in its container browse. As the column might change its position later in the aCols array, this data is the most appropriated to access to an specific column together with the ColAtIndex methods from the TBrowse class.

Current column's display order.

Scope:	read Only
Type:	Numeric
Initial value:	0

Description:

This property allows to know the order in which the columns will be shown. If the column is not visible (due its IVisible property is .F. or because currently is not displayed), the **nPos** value is zero.

Current column order status.

Scope:	Assignable
Type:	Numeric
Initial value:	bsNONE
Possible values:	bsNONE, bsASCENDING, bsDESCENDING

Description:

This property allows to establish a pictogram in the column header to indicate that the browse is sorted by this column.

A 0 value indicates that the column is not sorted, 1 means that has ascending sort and -1 means descending sort.

If you change the value of this property, it does not mean that the column will be sort. Is the programmer responsibility to make all the needed operations to reorder the Browse. The TArrayBrowse class use this property to sort any column when the **IAutSort** parameter is .T. in the SetArray method and it is made automatically.

This property only is available from the common API Windows control version 6.0 (Windows XP

or beyond).

Column width in pixels.

Scope:	Assignable
Type:	Numeric
Initial value:	0

Description:

This property allows to establish the width for every Browse column. Normally this value is automatically calculated based in the used font and the bitmap size (if any).

The column width can be modified even when the Browse is visible.

TDataField object linked to the column.

Scope:	Assignable
Type:	Object (TDataField)
Initial value:	NIL

Description:

This property is used for the specialized TDBBrowse Browse that use the Xailer 'Data Controls' to show and edit information that comes from the TDataSets.

A TDataField object type represents a field or column from a TDataSet object.

For more information, check the TDataField, TDataSet and TDataSource classes.

Note: This property only exist in the columns created through the TDBBrowse class. The columns with this property have a lower inherit level, and its name from the **TDBBrwColumn** class that inherits directly from TBrwColumn.

TGridedit object Linked to the column.

Scope:	Assignable
Type:	Object (TGridEdit)
Initial value:	NIL

Description:

The **oGridEdit** is responsible to edit the column when there is a edit type defined. This object is in the scope at the same time that its container column. In other words, it can read or modify any

oGridEdit property even when it is not in the edit mode.

The **oGridEdit** is instanced in the moment that the TBrwColumn **Create** method is execute. If you need to modify any **oGridEdit** data object you need to do it after the method Create is executed or in the OnGridEditCreate event.

For more information, check the TGridedit class.

THeaderItem object linked to the header column.

Scope:	Assignable
Type:	Object (THeaderItem)
Initial value:	NIL

Description:

The oHeader object is a THeaderItem object that belongs to the column container oHeader browse object.

Normally it will not be necessary to use this object because the TBrwColumn has properties that internally modify the associated THeaderItem. For example, what the cHeader property from the TBrwColumn does is modify the cText property from the associated THeaderItem.

This object only will be instantiated if its Browse container object has an active header.

For more information, check the THeaderItem class.

THeaderItem object linked to the footer column.

Scope:	Assignable
Type:	Object (THeaderItem)
Initial value:	NIL

Description:

The oFooter object is a THeaderItem object that belongs to the column container oFooter browse object.

Normally it will not be needed to use this object due the TBrwColumn class has properties that internally modify the associated THeaderItem. In this way, what the cFooter property from the TBrwColumn object really does is modify the cText property from the associated THeaderItem.

This object only will be instantiated if its Browse container object has an active footer.

For more information, check the THeaderItem class.

TFont text to be used.

Scope:	Assignable
Type:	Object
Initial value:	Nil

Description:

This property allows to change the font that will be used by the column. If the IParentFont property is set to .T. it will use the same font that its container Browse object.

For more information, check the TFont class.

oParent container object control.

Scope:	read Only
Type:	Object (Inherits from TBrowse)
Initial value:	Nil

Description:

This property returns the column container object from its Browse container.

Default column value for append and insert operations.

Scope:	Assignable
Type:	Any
Initial value:	Nil

Description:

This property permits to set the value of its oGridEdit object on append and insert operations. If this property is not assigned, then its value will be calculated based on the data type return value of its OnGetValue event.

1.6.2.31.4.2 TBrwColumnMethods

■ Constructor ■ Standard ■ Only after Create()

Typ Name	
■ Edit	
■ Select	
■ SetAlignment	
■ Value	

Method to enter to edit mode.

Type	Standard
Parameters	<nKey> Key to put in the edit field like a first typed key. <IMultiEdit> .T. if the edition can be done in more than one column. In other words, the edition is generated executing the Edit from the Browse.
Return value	<ISuccess> Key to put in the edit field like a first typed key.

Description:

This method is used to enter manually to edit mode. Normally the user will press the <Enter> key or will double-click the mouse pointer to enter to the edit mode.

Only will be possible to enter to the edit mode if the nEditType has been assigned with a different value than beNONE. The edition is not modal. It needs to catch the OnPostEdit event to make all the needed operations after the edition.

When enters to the edit mode, the oGridEdit object became visible and received the focus.

Selects the current column in its TBrowse container object.

Type	Standard
Parameters	<nKey>: Key to put in the edit field like a first typed key
Return value	<ISuccess> Key to put in the edit field like a first typed key

Description:

This methods allows to select the current column in the container Browse. It is needed that the column is not hidden through its IVisible property.

Sets column alignment. This method is internally used by the nAlignment property. But this method has a second paramter to avoid the complete refresh of the browse control.

Type	Standard
Parameters	<nAlign>: Alignment type. By default taLEFT. Possible values are: taLEFT, taRIGHT, taCENTER,

	taLEFTHEADERRIGHT, taLEFTHEADERCENTER, taRIGHTHEADERLEFT, taRIGHTHEADERCENTER, taCENTERHEADERLEFT, taCENTERHEADERRIGHT [<IRefresh>]: If TRUE the parent Browse control will be refreshed. By default .T.
Return value	NIL

Returns the actual column value.

Type	Standar
Parameters	None
Return value	<Value> Column actual value

Description:

Depending on the column state the returned value can change. If the column is on edit mode, inside its OnPostEdit event or inside its container browse OnPostEdit event, then the oGridEdit object value is returned. Otherwise its actual visible value is returned.

1.6.2.31.4.3 TBrwColumn:Events

Name
OnCreate
OnBtnClick
OnChangeWidth
OnDataClick
OnDataDbfClick
OnDrawCell
OnEdit
OnExit
OnFooterClick
OnFooterDbfClick
OnFooterRClick
OnGetData
OnGetValue
OnGridEditCreate
OnHeaderClick
OnHeaderDbfClick
OnHeaderRClick
OnPostEdit

Event that is triggered when the column is created..

Parameters	<oSender> :
:	Object that triggers the event.
Return value:	NIL

Description:

The event is produced when the column is created. Take into consideration that in that moment the Browse has not been created yet due it is created after to create all the columns.

Event that is produced when the oGridEdit column object is pressed.

Parameters	<oSender> :
:	Object that triggers the event(TGridEdit).
	<Value> : Current TGridEdit object value
Return value:	If different to NIL assign to oGridEdit:Value the returned value

Description:

The event is produced when in edit mode (beBUTTON or beGET_BUTTON type), the user press the button that incorporated the edit field.



The event can modify the current oGridEdit object value through the returned value. If it returns NIL it will not modify the **TGridEdit** object.

In the case that it returns a value if this is the only column that is being edited, the edit mode will be canceled.

Event that is produced when the column width is modified

Parameters	<oSender> :
:	Object that triggers the event (TBrwColumn).
	<@nNewWidth> : New column width
	<nOldWidth> : Old column width
Return value:	<IAbort> : If .T. the column width change is canceled

Description:

The event is produced when the column width is changed through code, or because the user modifies the column width during run-time. It receives as parameters the column object, the new column width and its old value.

The **nNewWidth** parameter is passed by reference, and allows that to modify the new column width from the event. The event should return a .T. logic value if you want to cancel the process and keep its old column width.

Event that is produced when the user clicks the mouse left button in the column data zone.

Parameters	<oSender> : Object that triggers the event <nFlags> : Virtual key status identifier. Parameter wParam from the WM_LBUTTONDOWN message. See the Windows API. <nPosX> : Mouse X coordinate. <nPosY> : Mouse Y coordinate.
Return value:	NIL

Description:

The event is produced immediately when the user clicks the mouse left button in the column data zone and receives the control and the mouse coordinates.

Event that is produced when the user double-clicks the mouse left button in the column data zone.

Parameters	<oSender> : Object that triggers the event <nFlags> : Virtual key status identifier. Parameter wParam from the WM_LBUTTONDOWN message. See the Windows API. <nPosX> : Mouse X coordinate. <nPosY> : Mouse Y coordinate.
Return value:	NIL

Description:

The event is produced immediately when the user clicks the mouse left button in the column data zone and receives the control and the mouse coordinates.

Event that is produced when the user clicks the mouse right button in the column data zone.

Parameters	<oSender>: Object that triggers the event <nFlags>: Virtual key status identifier. Parameter wParam from the WM_LBUTTONDOWN message. See the Windows API. <nPosX>: Mouse X coordinate. <nPosY>: Mouse Y coordinate.
Return value:	NIL

Description:

The event is produced immediately when the user clicks the mouse right button in the column data zone and receives the control and the mouse coordinates.

Event that is produced when the column cell is painted.

Parameters	<oSender>: Object that triggers the event (Self). <@cText>: Text to display. This parameter is received by reference. If is changed from the event the Browse will show its new value. <@nClrText>: Text color. This parameter is received by reference. If is changed from the event the text color will be modified only for that specific cell. <@nClrPane>: Background color. This parameter is received by reference. If it is changed from the event it will modify the background color only for that specific cell. <IHighlight>: .T. if the current cell is selected. When the property nMarqueeStyle of its container is cell type, like bmDOTEDCELL or bmSOLIDCELL this parameter will only be true when the cell is selected, otherwise it will be true for any column. <hDC>: Device context handle <aRect>: Cell coordinates
Return value:	<IOk> If it returns .F. the painting process will be interrupted

Description:

The event is produced every time that a column cell is painted and before to make any other painting operation.

It is useful to change the cell's background color or textcolor.

Event that is produced when the user enter to the column in edit mode.

Parameters	<oSender> : Object that triggers the event (Self). <Value> oGridEdit edit object value <IMultiEdit> : If .T. means the edition is of the full row
Return value:	NIL or .F.

Description:

This event is produced when the columns starts the edit mode and can be useful to modify any oGridEdit object property.

If the event returns .F., the edition is canceled.

To be able to edit any column you should use the nEditType property to indicate the edit type and the OnGetValue event to indicate the initial value to show. If you also want to use a listbox to show a possible value list, you should fill the aEditListTxt array.

Example:

```

WITH OBJECT oBrw := TDbfBrowse():New( oForm )
:nAlign          := alCLIENT
WITH OBJECT oCol := TBrwColumn():New( oBrw )
:cHeader         := "Code"
:OnGetData       := { |o, c| c:= Customer->Code }
:nEditType       := beEDIT
:OnGetValue      := { |o| Customer->Code }
:OnPostEdit     := { |o, v, l| iif( !l, Customer->Code := v, ) }
END
:AddCol( oCol )
WITH OBJECT oCol := TBrwColumn():New( oBrw )
:cHeader         := "Name"
:OnGetData       := { |o, c| c := Customer->Name }
:nEditType       := beEDIT
:OnGetValue      := { |o| Customer->Name }
END
:AddCol( oCol )
:Create()
END

```

Event that is produced when the TGridEdit edition object loses the focus.

Parameters	<oSender> : Object that triggers the event (Self). <oGridEdit> oGridEdit edition object <oNextCtl> : Next control that receives the focus
Return value:	<NIL> No action <.F.> Focus lose canceled

Description:

When the TBrwColumn objects go to edit mode, its oGridEdit object becomes visible and the edition starts. For convenience the OnExit event of the oGridEdit itself is routed to this event.

This event can be very handy when more than one column is edited at the same time, because in that case the OnPostEdit event is only triggered when the edition is finished, but not, when the TGridEdit controls of every column loses their focus.

Event that is produced when the user left-clicks the mouse pointer inside the column's footer.

Parameters	<oSender> : Object that trigger the event <nFlags> : State identifier from the virtual keys. Parameter wParam from the WM_LBUTTONDOWN message. See the Windows API. <nPosX> : Mouse X coordinate. <nPosY> Mouse Y coordinate.
Return value:	NIL

Description:

This event is produced after the user left-clicks the mouse pointer inside the column's footer and it receives as parameters the control and the mouse coordinates.

Event that is produced when the user double-clicks the left mouse button with the mouse pointer inside the column's footer.

Parameters	<oSender> : Object that trigger the event <nFlags> : State identifier from the virtual keys. Parameter wParam from the WM_LBUTTONDOWN message. See the Windows API. <nPosX> : Mouse X coordinate. <nPosY> : Mouse Y coordinate.
Return value:	NIL

Description:

This event is produced after the user double-clicks the left mouse button with the mouse pointer inside the column's footer and it receives as parameters the control and the mouse coordinates.

Event that is produced when the user right-clicks the mouse pointer inside the column's footer.

Parameters	<oSender> : Object that trigger the event <nFlags> : State identifier from the virtual keys. Parameter wParam from the WM_LBUTTONDOWN message. See the Windows API. <nPosX> : Mouse X coordinate. <nPosY> : Mouse Y coordinate.
Return value:	NIL

Description:

This event is produced after the user right-clicks the mouse pointer inside the column's footer and it receives as parameters the control and the mouse coordinates.

Event that is produced when the browse requires information or data for this column to be shown.

Parameters	<oSender> : Object that triggers the object (Self). <@Value> : Value with the information. This parameter is received by reference. <@nImage> : Image number to show according to the object
-------------------	--

	from the container Browse. This parameter is received by reference. Its default value zero (no image).
Return value:	NIL or value. If the returned value is not NIL, its value will be used to be shown by the container Browse even if the Value variable has been modified.

Description:

This event is produced every time that the Browse tries to recover information from the column to display it. You can define this event to change the information to be shown on the column.

Event that is produced every time that the column enters to edit mode. You can define this event to change the default value used for the edition.

Parameters	<oSender>: Object that trigger the event (Self)
:	<@Value>: Value with the information. This parameter is received by reference.
Return value:	NIL or value. If the returned value is not NIL, its value will be used for the editing even if the Value variable has been modified.

Description:

This event is produced every time that the column enters to edit mode and it is initialized with the oGridEdit object value.

Event that is produced when the oGridEdit object from the column is created.

Parameters	<oSender>: Object that triggers the event
:	
Return value:	NIL

Description:

The event is produced when the oGridEdit object is created, and is not necessarily at the same time that the column is created nor when the browse enters to edit mode. The oGridEdit object is created when the browse is created or when you add new columns to a browse previously created. This event can be very useful to establish a property or event in the oGridEdit object.

Event that is produced when the user left-clicks the mouse pointer inside the column's header.

Parameters	<oSender>: Object that trigger the event <nFlags>: State identifier from the virtual keys. Parameter wParam from the WM_LBUTTONDOWN message. See the Windows API. <nPosX>: Mouse X coordinate. <nPosY> Mouse Y coordinate.
Return value:	If it returns a logical value of .F. the possible sort operation on the browse will not be done

Description:

This event is produced after the user left-clicks the mouse pointer inside the column's header and it receives as parameters the control and the mouse coordinates.

Event that is produced when the user double-clicks the left mouse button with the mouse pointer inside the column's header.

Parameters	<oSender>: Object that trigger the event <nFlags>: State identifier from the virtual keys. Parameter wParam from the WM_LBUTTONDOWN message. See the Windows API. <nPosX>: Mouse X coordinate. <nPosY> Mouse Y coordinate.
Return value:	NIL

Description:

This event is produced after the user double-clicks the left mouse button with the mouse pointer inside the column's header and it receives as parameters the control and the mouse coordinates.

Event that is produced when the user right-clicks the mouse pointer inside the column's header.

Parameters	<oSender>: Object that trigger the event <nFlags>: State identifier from the virtual keys. Parameter
-------------------	---

	wParam from the WM_LBUTTONDOWN message. See the Windows API. <nPosX>: Mouse X coordinate. <nPosY> Mouse Y coordinate.
Return value:	NIL

Description:

This event is produced after the user right-clicks the mouse pointer inside the column's header and it receives as parameters the control and the mouse coordinates.

Event that is produced when the edit column mode is finalized.

Parameters	<oSender>: Object that triggers the event (Self) <@Value> oGridEdit object edit value. <@ICanceled> .F. if the edition has ended pressing the ESC key <nKey> Last key pressed. If edition is canceled clicking the mouse outside the edition area its value will be zero
Return value:	NIL

Description:

This event is produced when the column edition is finalized. This event permits to modify the value of its <Value> and <ICanceled> parameters since they are passed by reference or perform directly the save process. For the first option the property IAutoSave of its TBrowse container must be set to true.

When TBrowse:IAutoSave is set to true, through this event you may modify the value that will be saved definitely or even change if the value should be saved or not modifying its ICanceled parameter.

If TBrowse:IAutoSave is set to false, this event will be the responsible to save all the changes that were produced in the edition. If you don't catch this method all the changes in the column will be lost.

Be aware that when the edit process is of type multi-line, first the event **OnPostEdit** of every column is triggered and after, the event OnPostEdit of its OnPostEdit container.

To edit any column you should use the nEditType property to indicate the edit form, and the OnGetValue event to indicate the initial value to be shown.

Example:


```

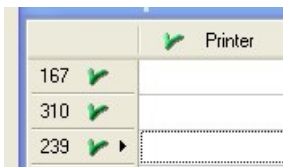
WITH OBJECT oBrw := TDbfBrowse():New( oForm )
:nAlign          := alCLIENT
WITH OBJECT oCol := TDbfBrwColumn():New( oBrw )
:cHeader        := "Code"
:OnGetData      := { |o, c| c:= Customer->Code }
:nEditType      := beEDIT
:OnGetValue     := { |o| Customer->Code }
:OnPostEdit    := { |o, v, l| iif( !l, Customer->Code := v, ) }
END
:AddCol( oCol )
:Create()
END

```

1.6.2.31.5 TBrwRecSel

This class allows to manage the RecordSelector column from a TBrowse object.

The RecordSelector object is only available if the IRecordSelector data from its Browse object is .T.



Hierarchy Inherits from TComponent
File Name \source\BrwRecSel.prg

1.6.2.31.5.1 TBrwRecSel:Properties

■ read Only ■ Assignable ■ Design assignable ■ Run-time assignable

Scope	Name	Type	Initial value
■	IParentFont	Logic	.T.
■	IRecordPointer	Logic	.T.
■	nAlignment	Numeric	taLEFT
■	nBmpAlignmen t	Numeric	taLEFT
■	nBmpMargin	Numeric	10
■	nClrPane	Numeric	oParent:nClrStdPane
■	nClrText	Numeric	oParent:nClrStdText
■	nHeight	Numeric	0
■	nWidth	Numeric	0
■	oFont	Object	NIL
■	oParent	Object	NIL

If it is .T. it will use the same font from the Browse.

Scope:	Assignable
Type:	Logic
Initial value:	.T.

Allows to see the record pointer.

Scope:	Assignable
Type:	Logic
Initial value:	.T.

Description:

This property allows to hide the record pointer that is shown in the Record Selector column.

Indicates the record selector alignment.

Scope:	Assignable
Type:	Numeric
Initial value:	taLEFT
Possible values:	taLEFT, taRIGHT, taCENTER

Description:

This property allows to align the Record Selector text. The alignment can be to the left, right or centered.

Image (bitmap) alignment from the Record Selector.

Scope:	Assignable
Type:	Numeric
Initial value:	taLEFT
Possible values:	taLEFT, taRIGHT

Description:

This property allows to align the image from the Record Selector. The alignment can be to the left

or to the right to the text.

For more information about the way to use images in the Record Selector, see the OnGetData event.

Image (bitmap) margin from the Record Selector.

Scope:	Assignable
Type:	Numeric
Initial value:	10

Description:

This property allows to establish the margins from the Record Selector border to the bitmap, indicated in pixels. Depending of the used alignment in the bitmap the limit could be the beginning of the Record Selector (left alignment) or the end of the Record Selector (right alignment).

Record Selector Background color.

Scope:	Assignable
Type:	Numeric
Initial value:	oParent:nClrStdPane

Consult the appendix for the list of available colors

Record Selector text color

Scope:	Assignable
Type:	Numeric
Initial value:	oParent:nClrStdText

Consult the appendix for the list of available colors

Record Selector Height specified in pixels.

Scope:	Assignable
Type:	Numeric
Initial value:	0

Description:

This property allows to establish the needed Record Selector height. Normally this value is

calculated automatically based in the used font and the bitmap size (if any). Later the Browse object container will establish the row height as the higher value from all the column heights and the Browse record selector.

Once the Browse is created (with the Create method) you should consider this data as read Only, and if you need to modify the row height for the Browse you should use the **oBrowse:nRowHeight** data.

Record Selector width specified in Pixels.

Scope:	Assignable
Type:	Numeric
Initial value:	0

Description:

This property allows to establish the Record Selector width in the browse. Normally this value is calculated automatically based in the font used and in the bitmap size (if any).

the Record Selector width can be modified even when the Browse is visible.

TFont object to be used for the text.

Scope:	Assignable
Type:	Object
Initial value:	NIL

Description:

This property allows to change the Font that will be used by the Record Selector. The default font that will be used will be the same Font used by its Browse container object.

See also the TFont class for more information.

oParent container control object.

Scope:	read Only
Type:	Object (Descending from TBrowse)
Initial value:	NIL

Description:

This property returns the container object from the column that is its container Browse.

1.6.2.31.5.2 TBrwRecSel:Events

Name
OnClick
OnDbClick
OnRClick
OnGetData

Event that is produced when the user clicks the left mouse button and the pointer is inside the Record Selector.

Parameters	<oSender>: Object that triggers the event <nFlags>: Virtual key identifier status. wParam parameter from the WM_LBUTTONDOWN message. For more information, check the Windows API. <nPosX>: X coordinate from the mouse pointer <nPosY> Y coordinate from the mouse pointer.
Return value:	NIL

Description:

The event is produced after the use clicks the left mouse button and the pointer is inside the Record Selector. It receives the control that triggers the event and the mouse pointer coordinates.

Event that is produced when the user double-clicks the left mouse button and the pointer is inside the Record Selector.

Parameters	<oSender>: Object that triggers the event <nFlags>: Virtual key identifier status. wParam parameter from the WM_LBUTTONDOWN message. For more information, check the Windows API. <nPosX>: X coordinate from the mouse pointer <nPosY> Y coordinate from the mouse pointer.
Return value:	NIL

Description:

The event is produced after the use double-clicks the left mouse button and the pointer is inside the Record Selector. It receives the control that triggers the event and the mouse pointer coordinates.

Event that is produced when the user right-clicks the mouse pointer inside the Record Selector.

Parameters	<oSender>: Object that triggers the event <nFlags>: Virtual key identifier status. wParam parameter from the WM_LBUTTONDOWN message. For more information, check the Windows API. <nPosX>: X coordinate from the mouse pointer <nPosY>: Y coordinate from the mouse pointer.
Return value:	NIL

Description:

The event is produced after the use right-clicks the mouse and the pointer is inside the Record Selector. It receives the control that triggers the event and the mouse pointer coordinates.

Event that is produced when the Browse request the information or data from the Record Selector to display it.

Parameters	<oSender>: Object that triggers the event (Self). <@cText>: Text with the information. This parameter is received by reference. Is the event responsibility to assign the cText variable due it will be used by the Browse to paint it. The cText value must be character type. <@nImage>: Image number to be shown according to the olmageList object from the container Browse. This parameter is received by reference. The default value is zero and it means no image. <ISelected>: Returns true if is the actual record selected.
Return value:	NIL or string. If the returned value is a string the value will be used by the container Browse regardless that the cText variable has been modified.

Description:

This event is produced every time that the browse tries to recover information from the Record Selector to display it.

Example:

```
WITH OBJECT oBrw := TDbfBrowse():New( oForm )
:nAlign          := alCLIENT
WITH OBJECT :oRecSel
:OnGetData := { |o, c| c:= Str( Recno() ) }
:nWidth     := 100
END
:Create()
END
```

1.6.2.31.6 TGridEdit

This class represents an specialized TMaskEdit control to edit a TBrowse control type. All the TBrowse columns (TBrwColumn) have a property from this class named oGridEdit to edit data on the fly. Several of this control properties are managed directly for the TBrwColumn object.



Hierarchy Inherits from TMaskEdit
File name \source\GridEdit.prg

1.6.2.31.6.1 TGridEdit:Properties

■ read Only ■ Assignable ■ Design assignable ■ Run-time assignable

Scope	Name	Type	Initial value
■	altems	Array	{}
■	altemsBound	Array	{}
■	IAutoSelect	Logic	asSELECTALL
■	IEditable	Logic	.T.
■	nEditMode	Numeric	emNORMAL(0)
■	oDataField	Object	NIL
■	oDataSet	Object	NIL

List with all the values that will show the control in its 'ListBox' when the style of the nEditType property from its container column is beLISTBOX or beGET_LISTBOX.
 This array corresponds completely with the aEditListText property of the same container column.

Scope: Assignable

Type:	Array
Initial value:	{}

List with all the values that corresponds to the `altems` property. This arrays corresponds completely with the `aEditListBound` property from its container column.

Scope:	Read only
Type:	Array
Initial value:	{}

Indicates the text selection when the control receives the focus.

Scope:	Design assignable
Type:	Logic o NIL
Initial value:	asSELECTALL
Possible values:	asDONTCARE, asSELECTALL, asSELECTNONE

If it is `.F.` the control can not be edited and it will be kept as read only. This property has only effect when it is used together with a `TDBBrowse` Browse type.

Scope:	Assignable
Type:	Logic
Initial value:	<code>.T.</code>

Indicates the control edit mode. To modify this property it is better to use the `nEditType` property from its container column.

Scope:	Assignable
Type:	Numeric
Initial value:	<code>emNORMAL</code> (0)
Possible values:	<code>emNORMAL</code> (0): Standard edition <code>emBUTTON</code> (1): Includes a button with 3 dots <code>emLIST</code> (2): Includes a ListBox

TDataField object type linked to the control. This property only has effect when it is used together with a TDBBrowse browse type.

Scope:	Assignable
Type:	Object
Initial value:	NIL

This property establishes all the display and edit mechanisms for the data linked to the control. If this property is not assigned the control will not show any information from the database and it is not possible to edit it neither.

This property can be assigned directly through a TDataField object or can be indicated with a text with the field name. In this case if its oDataSet has already assigned the control will look for that field in the oDataSet and if its found it will replace the text for the TDataField object found.

If this property is assigned directly with a TDataField object, the control automatically update the oDataSet property with the proprietary TDataSet object from the TDataField passed.

TDataSet object type linked to the control. This property only has effect when it is used together with a TDBBrowse Browse type.

Scope:	Assignable
Type:	Object
Initial value:	NIL

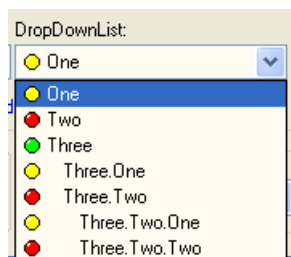
This property indicates that the TDataSet object (record set) will be linked to the control.

You must indicate its oDataField object to complete the link.

1.6.3 Win32

1.6.3.1 TComboBoxEx

This class represents an extended ComboBox that permits the use of images and item indentation.



Hierarchy File Inherits from TComboBox
 \source\ComboBoxEx.prg

1.6.3.1.1 TComboBoxEx:Propiedades

■ read Only ■ Assignable ■ Design assignable ■ Run-time assignable

Scope	Name	Type	Initial value
■	almages	Array	{}
■	altems	Array	{}
■	nIndentWidth	Numeric	10
■	oImageList	Object	TImageList

1.6.3.1.1.1 TComboBoxEx:almages

Numeric list with the image to show by each ComboBox element. The numeric index of each array element it corresponds with the ordinal of that image on its property oImageList.

Scope:	Assignable
Type:	Array
Initial value:	{}

For example, to set on a three elements list, that the first item will show the first image, the second item will show the second image and the third item will show the third image, the list will be like this:

```
oComboBox:aImages := { 1, 2, 3 }
```

1.6.3.1.1.2 TComboBoxEx:altems

List of elements displayed by the control. To achieve distinct indent level for every item of the list include as many spaces you want on the left side of every element. Each space it corresponds with one indentation level.

Scope:	Assignable
Type:	Array
Initial value	{}

Forexample:

```
oComboBoxEx:aItems := { "One", " Two", " Three" }
```

1.6.3.1.1.3 TComboBoxEx:nIndentWidth

Indentation width in pixels.

Scope:	Assignable
Type:	Numeric
Initial value:	10

1.6.3.1.1.4 TComboBoxEx:olmageList

TImageList object with all the images that will be used by the control.

Scope	Assignable
Type	Object
Initial value	TImageList()

The TImageList object is instantiated since the control is created with its New() constructor. The first image included in TImageList establishes the dimensions for the next images to be used. If the first image is an image that includes more than one bitmap it is important to establish the TImageList nHeight and nWidth properties before to add a bitmap.

1.6.3.2 TDatePicker

This class represents an standard windows date edit control.



Hierarchy	Inherits from TStdControl
See also	TDateEdit
File name	\source\DatePicker.prg

1.6.3.2.1 TDatePicker:Properties

■ read Only ■ Assignable ■ Design assignable ■ Run-time assignable

Scope	Name	Type	Initial value
■	cCustomForma	Character	"".

	t		
■	dMaxDate	Date	Ctod()
■	dMinDate	Date	Ctod()
■	dValue	Date	Date()
■	ICheckBox	Logic	.F.
■	IUpDown	Logic	.F.
■	nAlignment	Numeric	taLEFT
■	nClrMonthPane	Numeric	clWindow
■	nClrText	Numeric	clWindowText
■	nClrTitlePane	Numeric	clActiveCaption
■	nClrTitleText	Numeric	clCaptionText
■	nClrTrailingText	Numeric	clBtnShadow
■	nFormatType	Numeric	dfSHORTDATE
■	nHeight	Numeric	20
■	nWidth	Numeric	90

1.6.3.2.1.1 TDatePicker:cCustomFormat

User defined date format. It is needed to indicate in the nFormatType property the value dfCUSTOMFORMAT.

Scope:	Design assignable
Type:	Character
Initial value:	""

The date format is defined based in the following templates:

"d"	Day, with one or two digits.
"dd"	Day with two digits. Days with only one digit are preceded by a zero.
"ddd"	Day of the week specified with 3 characters.
"dddd"	Full day of the week.
"h"	Hour with one or two digits, in 12 hours format.
"hh"	Hour with two digits in 12 hours format. Hours with only one digit are preceding by a zero.
"H"	Hour with one or two digits in 24 hours format.
"HH"	Hour with two digits in 24 hours format. Hours with only one digit are preceding by a zero.
"m"	Minutes with one or two digits.
"mm"	Minutes with two digits. Minutes with only one digit are preceding by a zero.
"M"	Month number with one or two digits.
"MM"	Month number with two digits. Months with only one digit are preceding by a zero.
"MMM"	Month specified with 3 characters.
"MMMM"	Full month name.
"t"	AM/PM specified with one character (AM is shown as "A")

"tt"	AM/PM abbreviation.
"yy"	The last 2 digits from the year.
"yyyy"	Full year with 4 digits.

1.6.3.2.1.2 TDatePicker:dMaxDate

indicates the maximum date allowed by the control.

Scope:	Assignable
Type:	Date
Initial value:	Dtoc() (Blank Date)

1.6.3.2.1.3 TDatePicker:dMinDate

Indicates the minimum date allowed by the control.

Scope:	Assignable
Type:	Date
Initial value:	Dtoc() (Blank date)

1.6.3.2.1.4 TDatePicker:dValue

Indicates the control value.

Scope:	Assignable
Type:	Date
Initial value:	Date() (Date actual)

1.6.3.2.1.5 TDatePicker:ICheckBox

If it is .T., the control includes a checkbox to indicate blank dates. The only way to indicate a blank date in this control is using this property.

Scope:	Design assignable
Type:	Logic
Initial value:	.F.

1.6.3.2.1.6 TDatePicker:lUpDown

If it is .T. the control will show a Up/Down buttons that will allow to move through the dates, instead to show the calendar.

Scope:	Design assignable
Type:	Logic
Initial value:	.F.

1.6.3.2.1.7 TDatePicker:nAlignment

Indicates the calendar alignment that is displayed when the control button is pushed.

Scope:	Design assignable
Type:	Numeric
Initial value:	taLEFT
Possible values:	taLEFT, taRIGHT

1.6.3.2.1.8 TDatePicker:nClrMonthPane

Background color for the current month in the calendar.

Scope:	Design assignable
Type:	Numeric
Initial value:	clWindow

See also the appendix, to see the possible colors available

1.6.3.2.1.9 TDatePicker:nClrText

Indicates the control text color.

Scope:	Design assignable
Type:	Numeric
Initial value:	clWindowText

See also the appendix, to see the possible colors available

1.6.3.2.1.10 TDatePicker:nClrTitlePane

Background color for the calendar's title.

Scope:	Design assignable
Type:	Numeric
Initial value:	clActiveCaption

See also the appendix, to see the possible colors available

1.6.3.2.1.11 TDatePicker:nClrTitleText

Text color for the calendar's title.

Scope:	Design assignable
Type:	Numeric
Initial value:	clCaptionText

See also the appendix, to see the possible colors available

1.6.3.2.1.12 TDatePicker:nClrTrailingText

Indicates the color for the days that don't belong to the current month, but they are displayed anyway.

Scope:	Design assignable
Type:	Numeric
Initial value:	clBtnShadow

See also the appendix, to see the possible colors available

1.6.3.2.1.13 TDatePicker:nFormatType

Indicates the control format.

Scope:	Design assignable
Type:	Numeric
Initial value:	dfSHORTDATE
Possible values:	dfSHORTDATE dfSHORTDATECENTURY dfLONGDATEFORMAT dfCUSTOMFORMAT (You should also assign the cCustomFormat property)

1.6.3.2.1.14 TDatePicker:nHeight

Indicates the control height.

Scope:	Assignable
Type:	Numeric
Initial value:	20

1.6.3.2.1.15 TDatePicker:nWidth

Indicates the control width.

Scope:	Assignable
Type:	Numeric
Initial value:	90

1.6.3.2.2 TDatePicker:Events

Name
OnChange
OnCloseUp
OnDropDown

1.6.3.2.2.1 TDatePicker:OnChange

Event that is produced when the control's value is changed.

Parameters	<oSender>:
:	Reference to the object that triggers the event
	<dValue>:
	New date
Return value:	NIL

1.6.3.2.2.2 TDatePicker:OnCloseUp

Event that is produced when the calendar control is closed.

Parameters	<oSender>:
:	Reference to the object that triggers the event
Return	NIL

value:

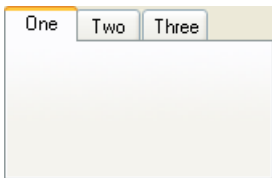
1.6.3.2.3 TDatePicker:OnDropDown

Event that is produced when the control calendar is displayed.

Parameters	<oSender>:
:	Reference to the object that triggers the event
Return value:	NIL

1.6.3.3 TFolder

This class represents a Windows Folder control. Every control tab represents an object from the TFolderPage class.



Any control that is inserted inside the TFolder object will be visible only in the active tab. If you need that all the tabs share the same panel, you should use the TTabCtrl control.

Hierarchy	Inherits from TTabCtrl
See also	TTabCtrl
File name	\source\Folder.prg

1.6.3.3.1 TTabCtrl:Properties

■ read Only ■ Assignable ■ Design assignable ■ Assignable en run-time

Scope	Name	Type	Initial value
■	altems	Array	{}

1.6.3.3.1.1 TTabCtrl:altems

List of control tabs represented by TFolderPage objects contained in the control.

Scope	Design assignable
Type	Array

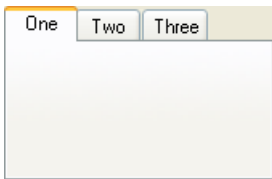
Initial value {}

1.6.3.4 TFolderPage

This class represents a TFolderPage, it means, every control that is created for every tab from a TFolder. It is at the same time, a control container.

It is not possible to create directly a TFolderPage. The TFolder control creates automatically a TFolderPage control for every tab defined.

It supports a background image and the automatic scroll bars when the control that contain don't fit in the client area.



Hierarchy Inherits from TScrollingWinControl
See also TFolder
File name \source\FolderPage.prg

1.6.3.4.1 TFolderPage:Properties

■ read Only ■ Assignable ■ Design assignable ■ Run-time assignable

Scope	Name	Type	Initial value
■	ITransparent	Logic	.F.
■	cText	Character	""
■	nClrText	Numeric	cWindowText
■	nImage	Numeric	0

1.6.3.4.1.1 TFolderPage:ITransparent

The control is drawn in transparent mode on its container object.

Scope	Assignable
Type	Logic
Initial value	.F.

1.6.3.4.1.2 TFolderPage:cText

ext to be shown in the control tab.

Scope	Assignable
Type	Character
Initial value	""

1.6.3.4.1.3 TFolderPage:nClrText

Standard text color.

Scope:	Assignable
Type:	Numeric
Initial value:	clWindowText

(See also the appendix to check the colors available)

1.6.3.4.1.4 TFolderPage:nImage

Image to shown in the tab. It is the image number in the olmageList object of its TFolder container.

Scope	Assignable
Type	Numeric
Initial value	0

1.6.3.4.2 TFolderPage:Methods

■ Constructor ■ Standard ■ Only after Create()

Typ	Name
■	Delete
■	Disable
■	Enable
■	Select

1.6.3.4.2.1 TFolderPage>Delete

Eliminates the FolderPage in its TFolder container object.

Type	Standard
-------------	----------

Parameters	None
Return value	<ISuccess> .T. if the operation is successful

1.6.3.4.2.2 TFolderPage:Disable

Disables the FolderPage in its TFolder container object.

Type	Standard
Parameters	None
Return value	NIL

1.6.3.4.2.3 TFolderPage:Enable

Enables the FolderPage in its TFolder container object.

Type	Standard
Parameters	None
Return value	NIL

1.6.3.4.2.4 TFolderPage:Select

Selects the FolderPage in its TFolder container object.

Type	Standard
Parameters	None
Return value	NIL

1.6.3.5 THeader

This class represents an standard windows header control. It is used to show the headers of any kind of columned information. You most probably will not use it directly, however, this control is used internally in all the browse controls type.



Hierarchy Inherits from TStdControl
See also TBrowse
File name \source\Header.prg

1.6.3.5.1 THeader:Properties

■ read Only ■ Assignable ■ Design assignable ■ Run-time assignable

Scope	Name	Type	Initial value
■	altems	Array	{}
■	IBorder	Logic	.F.
■	IButtons	Logic	.F.
■	ICancelDrag	Logic	.F.
■	ICancelTrack	Logic	.F.
■	IDragDrop	Logic	.T.
■	IFilterBar	Logic	.F.
■	IFlat	Logic	.F.
■	IFullDrag	Logic	.T.
■	IHotTrack	Logic	.T.
■	IResize	Logic	.T.
■	ITabStop	Logic	.F.
■	IWordBreak	Logic	.F.
■	nAlign	Numeric	atTOP
■	nHeight	Numeric	30
■	nWidth	Numeric	100
■	oEditFilter	Object	NIL
■	oimageList	Object	NIL

1.6.3.5.1.1 THeader:altems

THeaderItem object array with the different control's columns.

Scope:	Assignable
Type:	Array
Initial value:	{}

1.6.3.5.1.2 THeader:IBorder

If it is .T. the control will show a border.

Scope:	Design assignable
Type:	Logic
Initial value:	.F.

1.6.3.5.1.3 THeader:IButtons

If it is .T. the control will show the columns with a button 'look'.

Scope:	Design assignable
Type:	Logic
Initial value:	.F.

1.6.3.5.1.4 THeader:ICancelDrag

If it is .T. it will cancel any Drag operation in process.

Scope:	Run-time assignable
Type:	Logic
Initial value:	.F.

When any Drag & drop operation is executed, the event OnBeginDrag is triggered and later OnEndDrag. Is in this last event where you can set this property to .T. to cancel the operation.

1.6.3.5.1.5 THeader:ICancelTrack

If it is .T. it will cancel any Track operation in process.

Scope:	Run-time assignable
Type:	Logic
Initial value:	.F.

When any tracking operation is executed, the event OnBeginTrack is triggered and later OnEndTrack. Is in this last event where you can set this property to .T. to cancel the operation.

1.6.3.5.1.6 THeader:IDragDrop

If it is .T. allows Drag & Drop operations among the columns.

Scope:	Design assignable
Type:	Logic
Initial value:	.T.

1.6.3.5.1.7 THeader:IFilterBar

If it is .T. the control will show edit fields in every column to allow to make filter operations through them.

Scope:	Assignable
Type:	Logic
Initial value:	.F.

1.6.3.5.1.8 THeader:IFlat

If it is .T., the control will show a flat 'look'.

Scope:	Design assignable
Type:	Logic
Initial value:	.F.

1.6.3.5.1.9 THeader:IFullDrag

If it is .T. it will show the column's content moving in Drag & Drop operations.

Scope:	Design assignable
Type:	Logic
Initial value:	.T.

1.6.3.5.1.10 THeader:IHotTrack

If it is .T. the control will support 'Hot tracking'.

Scope:	Design assignable
Type:	Logic
Initial value:	.T.

1.6.3.5.1.11 THeader:IResize

If it is .T. any column in the control can be resize.

Scope:	Assignable
Type:	Logic

Initial value: .T.

1.6.3.5.1.12 THeader:ITabStop

If it is .T., the control will receive the focus when the user press the TAB key.

Scope	Design assignable
Type	Logic
Initial value	.F.

1.6.3.5.1.13 THeader:IWordBreak

If it is .T., if any item text does not fit an automatic word break will be done on that item.

You should increase the height of the control to show all the text when a word break is done.

Scope	Design
Type	Logic
Initial value	.F.

1.6.3.5.1.14 THeader:nAlign

Control alignment in its oParent container object.

Scope:	Assignable
Type:	Numeric
Initial value:	aITOP
Possible values:	aINONE, alLEFT, alTOP, alRIGHT, alBOTTOM, alCLIENT

Description:

This property allows to align the control dimensions and positions to its oParent container object. The alignment can be:

- **None:** Default value
- **Left:** The control is aligned to the left of its oParent control and takes the height from its client container area.
- **Top:** The control is aligned to the upper part of its oParent control and takes its width from its client container area.
- **Right:** The control aligns to the left of its oParent control and takes its height from tits client container area.
- **Bottom:** The control aligns to the lower part of its oParent control and takes its width from its

client container area.

- **Client:** The control aligns to all the area of its oParent client control adjusting its size to this area.

1.6.3.5.1.15 THeader:nHeight

Indicates the control's height.

Scope:	Assignable
Type:	Numeric
Initial value:	30

1.6.3.5.1.16 THeader:nWidth

Indicates the control width.

Scope:	Assignable
Type:	Numeric
Initial value:	100

1.6.3.5.1.17 THeader:oEditFilter

TInPlaceEdit object control with the actual filter. This control always points to the edit control with focus. If there is no control with focus its value will be NIL.

Scope:	Assignable
Type:	Object
Initial value:	NIL

1.6.3.5.1.18 THeader:oImageList

TImageList object with all the images for the columns.

Scope:	Assignable
Type:	Object
Initial value:	NIL

1.6.3.5.2 THeader:Methods

■ Constructor ■ Standard ■ only after Create()

Typ	Name
■	AddItem
■	Deleteltem
■	Deleteltems
■	EditFilter
■	HitTest
■	InsertItem

1.6.3.5.2.1 THeader:AddItem

Adds a new column to the control.

Type	Standard
Parameters	<p><cItem> Text to show</p> <p><nAlignment> Column alignment: taLEFT, taRIGHT, taCENTER. Default value: taLEFT</p> <p><nWidth> Column width. Default value: 50</p> <p><nImage> Image number inside the olmageList object. Default value: 0</p> <p><cTooltip> Column's Tooltip</p> <p><nSort> Sort pictogram: hsNONE, hsASCENDING, hsDESCENDING. Default value: hsNONE</p> <p><cFilter> Edit text when the IFilterBar property is active</p>
Return value	Object THeaderItem

1.6.3.5.2.2 THeader:Deleteltem

Deletes a column from the control.

Type	Standard
Parámetros	<p><nItem> Column number to be deleted</p>
Return value	<p><ISuccess> .T. if the operation is successful</p>

1.6.3.5.2.3 THeader:DeleteItems

Deletes all the control's columns.

Type	Standard
Parámetros	None
Return value	<ISuccess> .T. if the operation is successful

1.6.3.5.2.4 THeader:EditFilter

Set the focus to a column filter control.

Type	Standard
Parámetros	<nCol> Column to set the focus [<IDiscard>] If true and the user is currently editing the control, all its changes are lost. By default true.
Return value	<nResult> Non zero value if success

1.6.3.5.2.5 THeader:HitTest

Returns the THeaderItem object from an specific coordinate inside the THeader control.

Type	Standard
Parámetros	<nX> X coordinate <nY> Y coordinate
Return value	Object THeaderItem o NIL

1.6.3.5.2.6 THeader:InsertItem

Inserts a new column to the control.

Type	Standard
Parameters	<nIndex> New column position <cItem> Text to show

	<nAlignment> Column alignment: taLEFT, taRIGHT, taCENTER. Default value: taLEFT <nWidth> Column width. Default value: 50 <nImage> Image number inside the olmageList object. Default value: 0 <cTooltip> Column's Tooltip <nSort> Sort pictogram: hsNONE, hsASCENDING, hsDESCENDING. Default value: hsNONE <cFilter> Edit text when the IFilterBar property is active
Return value	Object THeaderItem

1.6.3.5.3 THeader:Events

Name
OnBeginDrag
OnBeginTrack
OnClick
OnDbClick
OnDividerDbClick
OnEndDrag
OnEndTrack
OnFilterBtnClick
OnFilterChange
OnFilterEdit
OnItemChanged
OnItemChanging
OnRClick
OnTrack

1.6.3.5.3.1 THeader:OnBeginDrag

Event that is produced when the user starts a Drag & Drop operation.

Parameters	<oSender>
:	Reference to the object that triggers the event
Return value:	None

1.6.3.5.3.2 THeader:OnBeginTrack

Event that is produced when the user starts a tracking operations or when a resize column operation is executed.

Parameters	<oSender>:
:	Reference to the object that triggers the event
Return value:	None

1.6.3.5.3.3 THeader:OnClick

Event that is produced when the user clicks the mouse pointer in any column in the control.

Parameters	<oSender>:
:	Reference to the object that triggers the event
	<nX>:
	X mouse coordinate
	<nY>:
	Y mouse coordinate
	<nItem>:
	Column number where the click was made
Return value:	None

1.6.3.5.3.4 THeader:OnDbClick

Event that is produced when the user double clicks the mouse pointer in any control column.

Parameters	<oSender>:
:	Reference to the object that triggers the event
	<nX>:
	X mouse coordinate
	<nY>:
	Y mouse coordinate
	<nItem>:
	Column number where the click was made
Return value:	None

1.6.3.5.3.5 THeader:OnDividerDbClick

Event that is produced when the user double clicks the mouse between 2 columns in the control.

Parameters	<oSender>:
:	Reference to the object that triggers the event
	<nX>:
	X mouse coordinate
	<nY>:
	Y mouse coordinate
	<nItem>:
	Column number (right) where the click was made
Return value:	None

1.6.3.5.3.6 THeader:OnEndDrag

Event that is produced at the end of a Drag & Drop operation. The Drag & Drop operation can be canceled if in this event the ICancelDrag property is set to .T..

Parameters	<oSender>:
:	Reference to the object that triggers the event
Return value:	None

1.6.3.5.3.7 THeader:OnEndTrack

Event that is produced when the tracking operation ends or when there is a resize operation in a column. the tracking operation can be canceled if in this event you set the ICancelTrack property to .T..

Parameters	<oSender>:
:	Reference to the object that triggers the event
Return value:	None

1.6.3.5.3.8 THeader:OnFilterBtnClick

Event that is produced when the user clicks the filter button and the IFilterBar property is active.

Parameters	<oSender>:
:	Reference to the object that triggers the event

	<nItem>: Column number where the click was made
Return value:	None

1.6.3.5.3.9 THeader:OnFilterChange

Event that is produced when the filter text is changed and the IFilterBar property is active.

Parameters	<oSender>: Reference to the object that triggers the event
:	<nItem>: Column number where the click was made
Return value:	None

1.6.3.5.3.10 THeader:OnFilterEdit

Event that is produced when the edit control receives focus and the IFilterBar property is active.

Parameters	<oSender>: Reference to the object that triggers the event
:	<nItem>: Column number where the click was made
Return value:	If it returns a logical .F. value the edit is canceled

1.6.3.5.3.11 THeader:OnItemChanged

Event that is produced after there is a change in any column.

Parameters	<oSender>: Reference to the object that triggers the event
:	
Return value:	None

1.6.3.5.3.12 THeader:OnItemChanging

Event that is produced when there is a change in any column.

Parameters	<oSender>:
:	Reference to the object that triggers the event
Return value:	None

1.6.3.5.3.13 THeader:OnRClick

Event that is produced when the user right-clicks the mouse pointer in any column in the control.

Parameters	<oSender>:
:	Reference to the object that triggers the event
	<nX>:
	X mouse coordinate
	<nY>:
	Y mouse coordinate
	<nItem>:
	Column number where the click was made
Return value:	None

1.6.3.5.3.14 THeader:OnTrack

Event that is produced when there is a tracking operation or when there is a resize column operation.

Parameters	<oSender>:
:	Reference to the object that triggers the event
Return value:	None

1.6.3.6 THeaderItem

This class represents an element in the Header Windows control

Description:

Every column from the THeader class is an object from the THeaderItem class. To establish any property in any header column, you need to use the properties described in this class.

Hierarchy Inherits from TComponent
File name \source\HeaderItem.prg

1.6.3.6.1 THeaderItem:Properties

■ read Only ■ Assignable ■ Design assignable ■ Run-time assignable

Scope	Name	Type	Initial value
■	cFilter	Character	""
■	cText	Character	""
■	cToolTip	Character	""
■	nAlignment	Numeric	taLEFT
■	nImage	Numeric	0
■	nIndex	Numeric	0
■	nOrder	Numeric	0
■	nSort	Numeric	hsNONE
■	nWidth	Numeric	50

1.6.3.6.1.1 THeaderItem:cFilter

Current filter expression for the column.

Scope	Assignable
Type	Character
Initial value	""

1.6.3.6.1.2 THeaderItem:cText

Text to be shown in the column.

Scope	Assignable
Type	Character
Initial value	""

1.6.3.6.1.3 THeaderItem:cToolTip

Tooltip to be shown in the column.

Scope	Assignable
Type	Character
Initial value	""

1.6.3.6.1.4 THeaderItem:nAlignment

Column alignment.

Scope	Assignable
Type	Numeric
Initial value	taLEFT
Possible values	taLEFT, taRIGHT, taCENTER

1.6.3.6.1.5 THeaderItem:nImage

Image number to be shown in the column based in the olmageList object of its THeader container.

Scope	Assignable
Type	Numeric
Initial value	0

1.6.3.6.1.6 THeaderItem:nIndex

Indicates the order sequence number in which it has been created in its THeader container.

Scope	read Only
Type	Numeric
Initial value	0

1.6.3.6.1.7 THeaderItem:nOrder

Indicates the order from the left to the right column.

Scope	Assignable
Type	Numeric
Initial value	0

1.6.3.6.1.8 THeaderItem:nSort

Image with the current sort column.

Scope	Assignable
Type	Numeric
Initial value	shNONE
Possible values	hsNONE, hsASCENDING, hsDESCENDING

1.6.3.6.1.9 THeaderItem:nWidth

Column's width indicated in pixels.

Scope	Assignable
Type	Numeric
Initial value	50

1.6.3.6.2 THeaderItem:Events

Name	
	OnClick
	OnDbClick
	OnDividerDbClick
	OnRClick

1.6.3.6.2.1 THeaderItem:OnClick

Event that is triggered when the users clicks the mouse pointer on the column.

Type	Standard
Parameters	<oSender> Reference to the object that produces the event <nX> X coordinate where the click is produced <nY> Y coordinate where the click is produced
Return value	NIL

1.6.3.6.2.2 THeaderItem:OnDbClick

Event that is triggered when the users double clicks the mouse pointer on the column.

Type	Standard
Parameters	<oSender> Reference to the object that produces the event <nX> X coordinate where the click is produced <nY> Y coordinate where the click is produced
Return value	NIL

1.6.3.6.2.3 THeaderItem:OnDividerDbClick

Event that is triggered when the users double clicks the mouse pointer between two the columns.

Type	Standard
Parameters	<oSender> Reference to the object that produces the event
Return value	NIL

1.6.3.6.2.4 THeaderItem:OnRClick

Event that is triggered when the users right clicks the mouse pointer on the column

Type	Standard
Parameters	<oSender> Reference to the object that produces the event <nX> X coordinate where the click is produced <nY> Y coordinate where the click is produced
Return value	NIL

1.6.3.7 THotKey

This class represents an special Windows API control to capture special keyboard keys pressed, like the combination of Control, Alt, Uppercase or several at the same time.

CTRL + ALT + J|

Hierarchy Inherits from TStdControl
File name \source\HotKey.prg

1.6.3.7.1 THotKey:Properties

■ read Only ■ Assignable ■ Design assignable ■ Run-time assignable

Scope	Name	Type	Initial value
■	IAlt	Logic	.F.
■	IBorder	Logic	.T.
■	ICtrl	Logic	.F.
■	IShift	Logic	.F.
■	IWinKey	Logic	.F.
■	nClrPane	Numeric	clWindow
■	nHeight	Numeric	20
■	nKey	Numeric	0
■	nWidth	Numeric	70

1.6.3.7.1.1 THotKey:IAlt

Indicates if the Alt key is pressed. In read operation indicates if the Alt key has been pressed. In write operation will show the Alt key as pressed in the control.

Scope:	Assignable
Type:	Logic
Initial value:	.F.

1.6.3.7.1.2 THotKey:IBorder

If it is .T. the control will be shown with a border.

Scope:	Design assignable
Type:	Logic
Initial value:	.T.

1.6.3.7.1.3 THotKey:lCtrl

Indicates if the Ctrl key is pressed. In read operation indicates the Ctrl key pressed. In write operation will show the Ctrl key as pressed in the control.

Scope:	Assignable
Type:	Logic
Initial value:	.F.

1.6.3.7.1.4 THotKey:lShift

Indicates if the Shift key is pressed. In read operation indicates the Shift key pressed. In write operation will show the Shift key as pressed in the control.

Scope:	Assignable
Type:	Logic
Initial value:	.F.

1.6.3.7.1.5 THotKey:lWinKey

Indicates if the windows key is pressed. In read operation indicates the Windows key pressed. In write operation will show the Windows key as pressed in the control.

Scope:	Assignable
Type:	Logic
Initial value:	.F.

1.6.3.7.1.6 THotKey:nClrPane

Indicates the background color.

Scope:	Assignable
Type:	Numeric
Initial value:	clWindow

(See also the appendix to check the colors available)

1.6.3.7.1.7 THotKey:nHeight

Indicates the control height specified in pixels.

Scope:	Assignable
Type:	Numeric
Initial value:	20

1.6.3.7.1.8 THotKey:nKey

Indicates the key pressed. In read operation indicates the key pressed. In write operation will show the key as pressed in the control.

Scope:	Assignable
Type:	Numeric
Initial value:	0

1.6.3.7.1.9 THotKey:nWidth

Indicates the control width specified in pixels.

Scope:	Assignable
Type:	Numeric
Initial value:	70

1.6.3.7.2 THotKey:Events

Name	
OnChange	

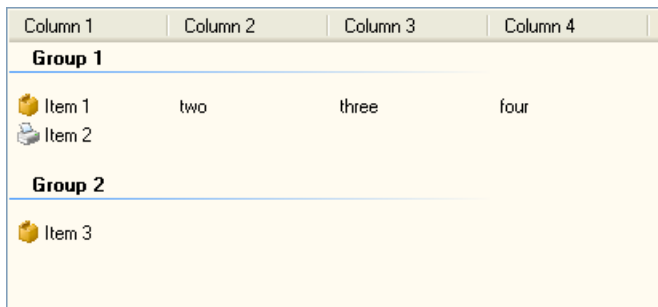
1.6.3.7.2.1 THotKey:OnChange

Event that is produced when the control changes its value.

Parameters	<oSender>:
:	Reference to the object that triggers the event
Return value:	NIL

1.6.3.8 TListView

This class represents a Windows API ListView controls. This class shows a list of elements with different views, as a icon collection, with large or small icons, a list or a grid showing additional information about each element.



Important note:

The property `items` holds all the items of the control. All the items have a property `index` that holds its creation position on the control TListView that can not be the same as its view order position.

In order to achieve the creation order criteria and to easily access to any control item through its array `items`, in case a deletion of any element is produced the array `items` is not reduced, instead, its element position is turned into a NIL value. For this reason, is very important that in your routines that go across the `items` array you take care that some elements may be NIL.

Hierarchy	Inherits from TStdControl
File name	\source\ListView.prg

1.6.3.8.1 TListView:Properties

■ read Only ■ Assignable ■ Design assignable ■ Run-time assignable

Scope	Name	Type	Initial Value
■	aColumns	Array	{}
■	aGroups	Array	{}
■	items	Array	{}
■	IAlignLeft	Logic	.F.
■	IAutoarrange	Logic	.T.
■	IBorder	Logic	.T.
■	ICheckBoxes	Logic	.F.
■	IDoubleBuffer	Logic	.F.
■	IEditLabels	Logic	.F.
■	IFullRowSelect	Logic	.F.
■	IGridLines	Logic	.F.
■	IGroupView	Logic	.F.
■	IHeaderDragDrop	Logic	.F.
■	IHotTrackSelect	Logic	.F.
■	ILabelTip	Logic	.F.
■	ILabelWrap	Logic	.T.

❑	InoColumnHeader	Logic	.F.
❑	InoSortHeader	Logic	.F.
❑	IShowSelAlways	Logic	.T.
❑	ISingleSel	Logic	.F.
❑	ITooltips	Logic	.T.
■	nClrPane	Numeric	cIWindow
■	nClrText	Numeric	cBtnText
■	nHeight	Numeric	90
■	nView	Numeric	MCON
■	nWidth	Numeric	120
■	oImageListHeader	Object	TImageList()
■	oImageListLarge	Object	TImageList()
■	oImageListSmall	Object	TImageList()

1.6.3.8.1.1 TListView:aColumns

List of objects TListViewColumn that shows the control.

Scope:	Assignable
Type:	Array
Initial Value:	{}

Is not very usual that you need to directly modify this property, you will normaly us the methods AddColumn and DelColumn, for example.

1.6.3.8.1.2 TListView:aGroups

List of objects TListViewGroup that shows the control. When a TListView has groups each item of the control must be assigned to a previous defined group. The elements will be shown ordered by group when the property IGroupView is used.

Scope:	Assignable
Type:	Array
Initial Value:	{}

Is not very usual that you need to directly modify this property, you will normaly us the methods AddGroup and DelGroup, for example.

1.6.3.8.1.3 TListView:altems

List of objects TListViewItem that shows the control.

Scope:	Assignable
---------------	------------

Type:	Array
Initial Value:	{}

Is not very usual that you need to directly modify this property, you will normaly us the methods AddItem for example.

1.6.3.8.1.4 TListView:IBorder

The control has border.

Scope	Design assignable
Type	Logic
Initial Value	.T.

1.6.3.8.1.5 TListView:IAAlignLeft

If true the control elements are left aligned when in icon view mode. See also the property nView.

Scope	Design assignable
Type	Logic
Initial Value	.F.

1.6.3.8.1.6 TListView:IAutoArrange

If true the control items are adjuted automatically.

Scope	Design assignable
Type	Logic
Initial Value	.T.

1.6.3.8.1.7 TListView:ICheckBoxes

If true, each control item will include a checkbox that can be checked or unchecked.

Scope	Design assignable
Type	Logic
Initial Value	.F.

1.6.3.8.1.8 TListView:IDoubleBuffer

If true, paints via double-buffering, which reduces flicker.

Scope	Design assignable
Type	Logic
Initial Value	.F.

1.6.3.8.1.9 TListView:IEditLabels

If true is possible to edit the control items through the Edit method of its TListViewItem.

Scope	Design assignable
Type	Logic
Initial Value	.F.

1.6.3.8.1.10 TListView:IFullRowSelect

If true, a complete line will be show with the selected item, otherwise, only the main text will be selected.

Scope	Design assignable
Type	Logic
Initial Value	.F.

1.6.3.8.1.11 TListView:IGridLines

If true, the control will show grid lines when in detail mode detalle. See also the property nView.

Scope	Design assignable
Type	Logic
Initial Value	.F.

1.6.3.8.1.12 TListView:IGroupView

If true the control wil show all the elements grouped by its group number. Se also the property aGroups.

Scope	Design assignable
--------------	-------------------

Type	Logic
Initial Value	.F.

1.6.3.8.1.13 TListView:IHeaderDragDrop

If true, when in details view, the column headers position can be changed by the user with a drag & drop operation. See also the property nView.

Scope	Design assignable
Type	Logic
Initial Value	.F.

1.6.3.8.1.14 TListView:IHotTrackSelect

If true, the element where the mouse cursor is will be automatically select after a second without moving the mouse cursor.

Scope	Design assignable
Type	Logic
Initial Value	.F.

1.6.3.8.1.15 TListView:ILabelTip

If true, If a partially hidden label in any list view mode lacks tooltip text, the list-view control will unfold the label.

Scope	Design assignable
Type	Logic
Initial Value	.F.

1.6.3.8.1.16 TListView:ILabelWrap

If true, the elements text can be wrapped.

Scope	Design assignable
Type	Logic
Initial Value	.T.

1.6.3.8.1.17 TListView:INoColumnHeader

If true, column headers are not displayed in report view.

Scope	Design assignable
Type	Logic
Initial Value	.F.

1.6.3.8.1.18 TListView:INoSortHeader

If true, Column headers do not work like buttons.

Scope	Design assignable
Type	Logic
Initial Value	.F.

1.6.3.8.1.19 TListView:IShowSelAlways

If true, the item or items selected are always visible.

Scope	Design assignable
Type	Logic
Initial Value	.T.

1.6.3.8.1.20 TListView:ISingleSel

By default, multiple items may be selected. If true, only one item can be selected.

Scope	Design assignable
Type	Logic
Initial Value	.F.

1.6.3.8.1.21 TListView:ITootltips

If true a tooltip will be shown when the mouse cursor is retained a certain time on top of one item.

Scope	Design assignable
Type	Logic
Initial Value	.T.

1.6.3.8.1.22 TListView:nClrPane

Indicates the control's background color.

Scope	Assignable
Type	Numeric
Initial value	clWindow

(See the appendix to check the colors available)

1.6.3.8.1.23 TListView:nClrText

Indicates the text color.

Scope	Assignable
Type	Numeric
Initial value	clBtnText

(See the appendix to check the colors available)

1.6.3.8.1.24 TListView:nHeight

Indicates the control's height.

Scope	Assignable
Type	Numeric
Initial Value	90

1.6.3.8.1.25 TListView:nView

View mode of the control.

Scope	Assignable
Type	Numeric
Initial Value	MCON
Valores posibles	MCON, IvSMALLICON, MLIST, IvTILE, IvDETAILS

1.6.3.8.1.26 TListView:nWidth

Indicates the control's width.

Scope	Assignable
Type	Numeric
Initial value	120

1.6.3.8.1.27 TListView:olmImageListHeader

TImageList object with all the images that will be used by the control columns headers when the view mode is deatils. Consult property nView for further information..

Scope	Assignable
Type	Object
Initial value	TImageList()

The TImageList object is instantiated since the control is created with its New() constructor. The first image included in TImageList establishes the dimensions for the next images to be used. If the first image is an image that includes more than one bitmap it is important to establish the TImageList nHeight and nWidth properties before to add a bitmap.

1.6.3.8.1.28 TListView:olmImageListLarge

TImageList object with all the images that will be used by the control columns headers when the view mode is lvICON. Consult property nView for further information..

Scope	Assignable
Type	Object
Initial value	TImageList()

The TImageList object is instantiated since the control is created with its New() constructor. The first image included in TImageList establishes the dimensions for the next images to be used. If the first image is an image that includes more than one bitmap it is important to establish the TImageList nHeight and nWidth properties before to add a bitmap.

1.6.3.8.1.29 TListView:olmImageListSmall

TImageList object with all the images that will be used by the control columns headers when the view mode is lvSMALLICON. Consult property nView for further information..

Scope	Assignable
Type	Object

Initial value TImageList()

The TImageList object is instantiated since the control is created with its New() constructor. The first image included in TImageList establishes the dimensions for the next images to be used. If the first image is an image that includes more than one bitmap it is important to establish the TImageList nHeight and nWidth properties before to add a bitmap.

1.6.3.8.2 TListView:Methods

■ Constructor ■ Standard ■ Only after Create()

Type	Name
■	AddColumn
■	AddGroup
■	AddItem
■	Arrange
■	CancelEdit
■	DelColumn
■	DeleteAllItems
■	DelGroup
■	GetFocused
■	GetSelection
■	GetVisibleCount
■	HitTest
■	InsertItem
■	IsEmpty
■	SetBkImage
■	Sort

1.6.3.8.2.1 TListView:AddColumn

Adds a new column to the control. The columns are only visible on 'details' view mode. Consult the property nView for further information.

Type	Standard
Parameters	<cText> : Text to show on the header of the column [<nWidth>] : Column width in pixels [<nAlignment>] : Column alignment. Possible values: taLEFT, taRIGHT and taCENTER. By default taLEFT [<nImage>] : Image number on the olmageListHeader. By default 0, which means without image
Return value	<oColumn> : TListViewColumn object created

1.6.3.8.2.2 TListView:AddGroup

Adds a new group to the control. The groups permit that the items can be displayed grouped based on the value of its nGroup value. The property IGroupView must be active to display the groups.

Type	Standard
Parameters	<cHeader> : Text header [<cFooter>] : Text footer [<nHAlignment>] : Header alignment. Possible values: taLEFT, taRIGHT and taCENTER. By default taLEFT [<nFAlignment>] : Footer alignment. Possible values: taLEFT, taRIGHT and taCENTER. By default taLEFT [<nState>] : Group state. Possible values: lsNORMAL, lsCOLLAPSED, lsHIDDEN. Default value lsNORMAL
Return value	<oGroup> : TListViewGroup object created

1.6.3.8.2.3 TListView:AddItem

Adds a new item to the control.

Type	Standard
Parameters	<cText> : Text to display [<nImage>] : Image number on olmageListLarge and olmageListSmall. By default 0, which means without image [<nGroup>] : Group number. The group number should be created first with the method AddGroup [<nIndent>] : Indent in pixels of the element in 'details' mode [<aData>] : Text array with the detail information. The lengthh of the array should be equal to the number of columns of the control [<IChecked>] : If true the checkbox of the item will be checked. Is necessary to use the style ICheckBoxes

Return value	<oltem>: TListViewItem object created
---------------------	--

1.6.3.8.2.4 TListView:Arrange

Adjusts the items position in the control.

Type	Only after create
Parameters	None
Return value	NIL

1.6.3.8.2.5 TListView:CancelEdit

Cancels current edit operation.

Type	Only after create
Parameters	None
Return value	NIL

1.6.3.8.2.6 TListView:DelColumn

Deletes a existing column.

Type	Standard
Parameters	<nCol>: Column number to delete
Return value	<ISuccess> True if success

If a column is deleted, the corresponding array element on the property **aData** of all its items will be also deleted and afterwards a complete recreation of the control will be done. If the control has a lot of items this may take some time.

1.6.3.8.2.7 TListView>DeleteAllItems

Deletes all the items of the controll.

Type	Standard
-------------	----------

Parameters	None
Return value	NIL

1.6.3.8.2.8 TListView:DelGroup

Deletes a existing group.

Type	Standard
Parameters	<nGroup>: Group number to delete
Return value	<ISuccess> True if success

If a group is deleted, all the items of that group will be moved to group 0 (no group).

1.6.3.8.2.9 TListView:GetFocused

Returns the current active item.

Type	Only after create
Parameters	None
Return value	<oItem> Active item

.

1.6.3.8.2.10 TListView:GetSelection

Returns an array with all the items selected.

Type	Only after create
Parameters	None
Return value	<aItems> Array of selected items

.

1.6.3.8.2.11 TListView:GetVisibleCount

Calculates the number of items that can fit vertically in the visible area of a list-view control when in list or report view.

Type	Only after create
Parameters	None
Return value	<nItems>

1.6.3.8.2.12 TListView:HitTest

Returns the item displayed on a specific position.

Type	Only after create
Parameters	<nX>: X coordinate <nY>: Y coordinate
Return value	<oItem> TListViewItem item or NIL

1.6.3.8.2.13 TListView:InsertItem

Inserts a new item on the control.

Type	Standard
Parameters	<cText>: Text to display [<nImage>]: Image number on olmageListLarge and olmageListSmall. By default 0, which means without image [<nGroup>]: Group number. The group number should be created first with the method AddGroup [<nIndent>]: Indent in pixels of the element in 'details' mode [<aData>]: Text array with the detail information. The lengthh of the array should be equal to the number of columns of the control [<IChecked>]: If true the checkbox of the item will be checked. Is necessary to use the style ICheckBoxes [<nPos>]: Ordinal position of the new element. By default it will be inserted at the end.
Return value	<oItem> TListViewItem object created

1.6.3.8.2.14 TListView:IsEmpty

Returns true if the control has no items.

Type	Standard
Parameters	None
Return value	<IEmpty> True if empty

1.6.3.8.2.15 TListView:SetBkImage

Sets a image background.

Type	Only after create
Parameters	<clmageURL>: Image path [<ITile>]: Show image on tile mode. By default false [<nXOffset>]: Horizontal offset respect its left corner. By default 0 [<nYOffset>]: Vertical offset respect its top corner. By default 0
Return value	<ISuccess> True if success

1.6.3.8.2.16 TListView:Sort

Sorts the items on the control. Can only be used in 'details' mode. For further information consult the property nView.

Type	Only after create
Parameters	<nColumn>: Column number to sort
Return value	<ISuccess> True if success

1.6.3.8.3 TListView:Events

Name	
	OnBeginLabelEdit

OnCheckStateChanged
OnClick
OnColumnClick
OnDeleteItem
OnDrawCell
OnItemChanged
OnEndLabelEdit
OnKeyDown
OnRClick

1.6.3.8.3.1 TListView:OnBeginLabelEdit

Event that is produced when a item edit operation starts.

Parameters:	<oSender>: Reference to the object that triggers the event
	<oItem>: TListViewItem object in edit mode
Return value:	NIL

1.6.3.8.3.2 TListView:OnCheckStateChanged

Event that is produced when the state of a item checkbox is changed.

Parameters:	<oSender>: Reference to the object that triggers the event
	<oItem>: TListViewItem object that has change its checkbox state
Return value:	NIL

1.6.3.8.3.3 TListView:OnClick

Event that is produced when the user clicks on the control.

Parameters:	<oSender>: Reference to the object that triggers the event
	<nPosX>: X mouse position
	<nPosY>: Y mouse position
Return value:	NIL

Nota: Use the HitTest method to get the item in the X,Y position.

1.6.3.8.3.4 TListView:OnColumnClick

Event that is produced when the user clicks on any column header.

Parameters:	<oSender>: Reference to the object that triggers the event <nColumn>: Column number that fired the event
Return value:	<IValue> If it returns false the automatic sort of the column will not be done

1.6.3.8.3.5 TListView:OnDeleteItem

Event that is produced when a control item is deleted.

Parameters:	<oSender>: Reference to the object that triggers the event <oItem>: TListViewItem object that is going to be deleted
Return value:	NIL

1.6.3.8.3.6 TListView:OnDrawCell

Event that is produced when a control item is painted.

Parameters:	<oSender>: Reference to the object that triggers the event <oItem>: TListViewItem object to be painted <nColumn>: Column to paint. Only when the property nView is LVDETAILS. Otherwise its value is zero <@nClrText>: Text color passed by reference in order to be modified if necessary. <@nClrPane>: Background color passed by reference in order to be modified if necessary <hDC> Handle to device context
--------------------	--

	<aRect> Printing array
Return value:	<nValue> : 0 : Notifies that all the item columns should receive also this event 1 : Notifies a change of color 2 : Notifies a owner draw process avoiding any default painting NIL : Default behaviour

Note: When nView mode is not LvDETAILS, this event is only received onces for every item, having the *nColumn* parameter the value zero. When the mode is LvDETAILS, is necessary to return a value of zero when the parameter *nColumn* is also zero in order to receive the event for every column.

1.6.3.8.3.7 TListView:OnItemChanged

Event that is produced when the users changes to the selected item.

Parameters:	<oSender> : Reference to the object that triggers the event <oItem> : TListViewItem object that has changed
Return value:	NIL

1.6.3.8.3.8 TListView:OnEndLabelEdit

Event that is produced when a item edit operation ends.

Parameters:	<oSender> : Reference to the object that triggers the event <oItem> : TListViewItem object that ends the edit operation
Return value:	NIL

1.6.3.8.3.9 TListView:OnKeyDown

Event that is produced when the user presses a key.

Parameters:	<oSender> : Reference to the object that triggers the event <nKey> : Key pressed
Return value:	NIL

1.6.3.8.3.10 TListView:OnRClick

Event that is produced when the user right-clicks in the control.

Parameters:	<oSender>: Reference to the object that triggers the event <nPoxX>: X mouse position <nPosY>: Y mouse position
Return value:	NIL

Nota: Use the HitTest method to get the item in the X,Y position.

1.6.3.9 TListViewColumn

This class represents the columns of the TListView control that are only displayed in details mode. For further information consult the property nView of TListView.

Hierarchy	Inherits from TComponent
File name	\source\ListViewColumn.prg

1.6.3.9.1 TListViewColumn:Properties

read Only
 Assignable
 Design assignable
 Run-time assignable

Scope	Name	Type	Initial value
<input type="checkbox"/>	cText	Character	""
<input type="checkbox"/>	nAlignment	Numeric	taLEFT
<input type="checkbox"/>	nImage	Numeric	0
<input type="checkbox"/>	nIndex	Numeric	0
<input type="checkbox"/>	nWidth	Numeric	100
<input type="checkbox"/>	oParent	Object	NIL

1.6.3.9.1.1 TListViewColumn:cText

Text to show on the column header.

Scope	Assignable
Type	Character
Initial value	""

1.6.3.9.1.2 TListViewColumn:nAlignment

Column alignment.

Scope	Assignable
Type	Numeric
Initial value	taLEFT
Possible values	taLEFT, taCENTER, taRIGHT

1.6.3.9.1.3 TListViewColumn:nImage

Image to show on the column. It corresponds with the image number in the olmageListHeader object of its container TListView.

Scope	Assignable
Type	Numeric
Initial value	0

1.6.3.9.1.4 TListViewColumn:nIndex

Creation order in its TListView container.

Scope	Read only
Type	Numeric
Initial value	0

1.6.3.9.1.5 TListViewColumn:nWidth

Column width in pixels.

Scope	Assignable
Type	Numeric
Initial value	100

1.6.3.9.1.6 TListViewColumn:oParent

TListView Container object from the current item

Scope	Read only
Type	Object
Initial value	NIL

1.6.3.9.2 TListViewColumn:Methods

■ Constructor ■ Standard

Type	Name
■	Create

1.6.3.9.2.1 TListViewColumn:Create

Class constructor. It receives as parameters its container TListView object.

Type	Constructor
Parameters	oParent
Return value	Self reference (Self)

1.6.3.10 TListViewGroup

This class represents the groups of a TListView control.

Hierarchy	Inherits from TComponent
File name	\source\ListViewGroup.prg

1.6.3.10.1 TListViewGroupProperties

■ read Only ■ Assignable ■ Design assignable ■ Run-time assignable

Scope	Name	Type	Initial value
■	cFooter	Character	""
■	cHeader	Character	""
■	nFooterAlign	Numeric	taLEFT
■	nHeaderAlign	Numeric	taLEFT
■	nIndex	Numeric	0
■	nState	Numeric	100
■	oParent	Object	IsNORMAL

1.6.3.10.1.1 TListViewGroup:cFooter

Text to show on the group footer.

Scope	Assignable
Type	Character
Initial value	""

1.6.3.10.1.2 TListViewGroup:cHeader

Text to show on the group header.

Scope	Assignable
Type	Character
Initial value	""

1.6.3.10.1.3 TListViewGroup:nFooterAlign

Footer alignment

Scope	Assignable
Type	Numeric
Initial value	taLEFT
Possible values	taLEFT, taCENTER, taRIGHT

1.6.3.10.1.4 TListViewGroup:nHeaderAlign

Header alignment.

Scope	Assignable
Type	Numeric
Initial value	taLEFT
Possible values	taLEFT, taCENTER, taRIGHT

1.6.3.10.1.5 TListViewGroup:nIndex

Group number identifier.

Scope	Read only
Type	Numeric
Initial value	0

1.6.3.10.1.6 TListViewGroup:nState

Group state.

Scope	Assignable
Type	Numeric
Initial value	IsNORMAL
Possible values	IsNORMAL, IsCOLLAPSED, IsHIDE

1.6.3.10.1.7 TListViewGroup:oParent

TListView Container object from the current item

Scope	Read only
Type	Object
Initial value	NIL

1.6.3.10.2 TListViewGroup:Methods

■ Constructor ■ Estándar

Typ e	Name
■	Create

1.6.3.10.2.1 TListViewGroup:Create

Class constructor. It receives as parameters its container TListView object.

Type	Constructor
Parameters	oParent
Return value	Self reference (Self)

1.6.3.11 TListViewItem

This class represents a TListView control items

Hierarchy Inherits from TComponent
File name \source\ListViewItem.prg

1.6.3.11.1 TListViewItem:Properties

■ read Only ■ Assignable ■ Design assignable ■ Run-time assignable

Scope	Name	Type	Initial value
■	aData	Array	{}
■	cText	Character	""
■	cTooltip	Character	""
■	lChecked	Logic	.F.
■	nGroup	Numeric	0
■	nImage	Numeric	0
■	nIndent	Numeric	0
■	nIndex	Numeric	0
■	oParent	Object	NIL

1.6.3.11.1.1 TListViewItem:aData

String array with the text to show when the control TListView is in details mode. The length of the array must be equal with the number of columns of the control. For further information consult the property nView of TListView.

Scope	Assignable
Type	Array
Initial value	{}

1.6.3.11.1.2 TListViewItem:cText

Text to display of the item.

Scope	Assignable
Type	Character
Initial value	""

1.6.3.11.1.3 TListItem:cTooltip

Tooltip to show when the mouse cursor is on top of an item for a small period of time.

Scope	Assignable
Type	Character
Initial value	""

1.6.3.11.1.4 TListItem:IChecked

If true the item will show a checked checkbox. (The property ICheckBoxes of its container TListView must be also active).

Scope	Assignable
Type	Logic
Initial value	.F.

1.6.3.11.1.5 TListItem:nGroup

Group who belongs in its container. Is important that the group is created before assigning this property to any item. For further information consult the property aGroups of its container TListView.

Scope	Assignable
Type	Numeric
Initial value	0

1.6.3.11.1.6 TListItem:nImage

Image to show with the item. It corresponds with the image number in the olmageListLarge and olmageListSmall objects from its container TListView.

Scope	Assignable
Type	Numeric
Initial value	0

1.6.3.11.1.7 TListViewItem:nIndent

Item Indentation when in details view mode. For further information consult the property nView of TListView.

Scope	Assignable
Type	Numeric
Initial value	0

1.6.3.11.1.8 TListViewItem:nIndex

Creation order in its TListView container.

Scope	Read only
Type	Numeric
Initial value	0

1.6.3.11.1.9 TListViewItem:oParent

TListView Container object from the current item

Scope	Read only
Type	Object
Initial value	NIL

1.6.3.11.2 TListViewItem:Methods

■ Constructor ■ Standard

Typ	Name
■	Create
■	Delete
■	Edit
■	EnsureVisible
■	Select

1.6.3.11.2.1 TListViewItem:Create

Class constructor. It receives as parameters its container TListView object.

Type	Constructor
-------------	-------------

Parameters	oParent
Return value	Self reference (Self)

1.6.3.11.2.2 TListViewItem:Delete

Deletes actual item.

Type	Standard
Parameters	None
Return value	<ISuccess> True if success

1.6.3.11.2.3 TListViewItem:Edit

Edits the item text. The property IEditLabels of its container must be set to true..

Type	Standard
Parameters	None
Return value	<ISuccess> Verdadero si éxito

1.6.3.11.2.4 TListViewItem:EnsureVisible

Forces de Item to be visible.

Type	Standard
Parameters	None
Return value	<ISuccess> .T. if the operation is successful

1.6.3.11.2.5 TListViewItem:Select

Selects the current item.

Type	Standard
Parameters	None
Return value	<ISuccess> .T. if the operation is successful

1.6.3.12 TMonthCalendar

This class represents a calendar control type.



Hierarchy Inherits from TStdControl
See also TDateEdit, TDatePicker
File name \source\MonthCalendar.prg

1.6.3.12.1 TMonthCalendar:Properties

■ read Only ■ Assignable ■ Design assignable ■ Run-time assignable

Scope	Name	Type	Initial value
■	aHolidays	Array	{}
■	dMaxDate	Date	Ctod()
■	dMinDate	Date	Ctod()
■	dToday	Date	Date()
■	dValue	Date	Date()
■	lBorder	Logic	.T.
■	lMultiSelect	Logic	.F.
■	lNoToday	Logic	.F.
■	lNoTodayCircle	Logic	.F.
■	lWeekNumbers	Logic	.F.
■	nClrMonthPane	Numeric	clWindow
■	nClrPane	Numeric	clWindow
■	nClrText	Numeric	clWindowText
■	nClrTitlePane	Numeric	clActiveCaption
■	nClrTitleText	Numeric	clCaptionText
■	nClrTrailingText	Numeric	clBtnShadow
■	nHeight	Numeric	158
■	nMaxSelCount	Numeric	7
■	nWidth	Numeric	200

1.6.3.12.1.1 TMonthCalendar:aHolidays

Array of dates that should be shown boldface.

Scope:	Assignable
Type:	Array
Initial value:	{}

1.6.3.12.1.2 TMonthCalendar:dMaxDate

Maximum date allowed for the control. If the IMultiSelect is .T. this property will indicate the upper range for the date to be selected.

Scope:	Assignable
Type:	Date
Initial value:	Dtoc() (Blank date)

1.6.3.12.1.3 TMonthCalendar:dMinDate

Minimum date allowed for the control. If the IMultiSelect is .T. this property will indicate the lower range for the date to be selected.

Scope:	Assignable
Type:	Date
Initial value:	Dtoc() (Blank date)

1.6.3.12.1.4 TMonthCalendar:dToday

Date that will be show as the current day.

Scope:	Assignable
Type:	Date
Initial value:	Date() (Date actual)

1.6.3.12.1.5 TMonthCalendar:dValue

Indicates the control value.

Scope:	Assignable
---------------	------------

Type:	Date
Initial value:	Date() (Current date)

1.6.3.12.1.6 TMonthCalendar:IBorder

If it is .T., the control will show a border.

Scope:	Design assignable
Type:	Logic
Initial value:	.T.

1.6.3.12.1.7 TMonthCalendar:IMultiSelect

If it is .T., it allows to select a date range. The maximum number of dates to be selected is established by the nMaxSelCount property. The dMinDate and dMaxDate properties indicate the minimum and maximum selection values.

Scope:	Design assignable
Type:	Logic
Initial value:	.F.

1.6.3.12.1.8 TMonthCalendar:INoToday

If it is .T. it will not show the current day.

Scope:	Design assignable
Type:	Logic
Initial value:	.F.

1.6.3.12.1.9 TMonthCalendar:INoTodayCircle

If it is .T. it will not show the current day marked.

Scope:	Design assignable
Type:	Logic
Initial value:	.F.

1.6.3.12.1.10 TMonthCalendar:IWeekNumbers

If it is .T. it will show the number of weeks.

Scope:	Design assignable
Type:	Logic
Initial value:	.F.

1.6.3.12.1.11 TMonthCalendar:nClrMonthPane

Indicates the background color for the month in the calendar.

Scope:	Assignable
Type:	Numeric
Initial value:	clWindow

See also the appendix to check the possible colors available

1.6.3.12.1.12 TMonthCalendar:nClrPane

Indicates the background color for the control

Scope:	Assignable
Type:	Numeric
Initial value:	clWindow

See also the appendix to check the possible colors available

1.6.3.12.1.13 TMonthCalendar:nClrText

Indicates the text color for the control.

Scope:	Assignable
Type:	Numeric
Initial value:	clWindowText

See also the appendix to check the possible colors available

1.6.3.12.1.14 TMonthCalendar:nClrTitlePane

Indicates the background color for the title.

Scope:	Assignable
Type:	Numeric
Initial value:	clActiveCaption

See also the appendix to check the possible colors available

1.6.3.12.1.15 TMonthCalendar:nClrTitleText

Indicates the text color for the title.

Scope:	Assignable
Type:	Numeric
Initial value:	clCaptionText

See also the appendix to check the possible colors available

1.6.3.12.1.16 TMonthCalendar:nClrTrailingText

Indicates the text color for the days that don't belong to the current month.

Scope:	Assignable
Type:	Numeric
Initial value:	clBtnShadow

See also the appendix to check the possible colors available

1.6.3.12.1.17 TMonthCalendar:nHeight

Indicates the control height.

Scope:	Assignable
Type:	Numeric
Initial value:	158

1.6.3.12.1.18 TMonthCalendar:nMaxSelCount

Indicates the maximum number of selected dates when the IMultiSelect property is used.

Scope:	Assignable
Type:	Numeric
Initial value:	7

1.6.3.12.1.19 TMonthCalendar:nWidth

Indicates the control width.

Scope:	Assignable
Type:	Numeric
Initial value:	90

1.6.3.12.2 TMonthCalendar:Events

Name	
OnChange	
OnSelect	

1.6.3.12.2.1 TMonthCalendar:OnChange

Event that is produced when the control changes its value.

Parameters	<oSender>:
:	Reference to the object that triggers the event
	<dValue>:
	New Date
Return value:	NIL

1.6.3.12.2.2 TMonthCalendar:OnSelect

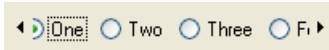
Event that is produced when the user selects a date range in the control.

Parameters	<oSender>:
:	Reference to the object that triggers the event
	<dFirst>:
	Lower selected date

	<dLast> : Upper selected date
Return value:	NIL

1.6.3.13 TPager

This class represents a Pager Windows control, that is a container control to use with other control that has not enough space to show its content.



Hierarchy	Inherits from TWinControl
See also	TScrollBar
Filename	\source\Pager.prg

1.6.3.13.1 TPager:Properties

read Only
 Assignable
 Design assignable
 Run-time assignable

Scope	Name	Type	Initial value
<input type="checkbox"/>	IAutoScroll	Logic	.F.
<input type="checkbox"/>	nHeigth	Numeric	30
<input type="checkbox"/>	nOrientation	Numeric	orHORIZONTAL
<input type="checkbox"/>	nPosition	Numeric	0
<input type="checkbox"/>	nScrollIncrement	Numeric	8
<input type="checkbox"/>	nWidth	Numeric	120

1.6.3.13.1.1 TPager:IAutoScroll

The control scrolls automatically then the mouse pointer is moved on the scroll arrow.

Scope	Assignable
Type	Logic
Initial value	.F.

1.6.3.13.1.2 TPager:nHeight

Indicates the control height.

Scope	Assignable
Type	Numeric
Initial value	30

1.6.3.13.1.3 TPager:nOrientation

Indicates if the control will be shown with vertical or horizontal orientation.

Scope	Design assignable
Type	Numeric
Initial value	orHORIZONTAL
Possible values	orHORIZONTAL, orVERTICAL

1.6.3.13.1.4 TPager:nPosition

Indicates the current scroll position.

Scope	Assignable
Type	Numeric
Initial value	0

1.6.3.13.1.5 TPager:nScrollIncrement

Indicates the scroll increment by default.

Scope	Assignable
Type	Numeric
Initial value	8

1.6.3.13.1.6 TPager:nWidth

Indicates the control width.

Scope	Assignable
--------------	------------

Type	Numeric
Initial value	120

1.6.3.14 TProgressBar

This class represents a Windows ProgressBar control.



Description:

The TProgressBar class allows to manage the standard Windows ProgressBar control.

Remarks:

The property ISmooth and the methods SetBarColor and SetBkColor are only supported in the Windows classic theme.

Hierarchy	Inherits from TControl
File name	\source\ProgressBar.prg

1.6.3.14.1 TProgressBar:Properties

■ read Only ■ Assignable ■ Design assignable ■ Runtime-assignable

Scope	Name	Type	Initial value
■	IMarquee	Logic	.F.
■	ISmooth	Logic	.F.
■	nHeight	Numeric	25
■	nMax	Numeric	100
■	nMarqueeSpeed	Numeric	100
■	nMin	Numeric	0
■	nOrientation	Numeric	orHORIZONTAL
■	nStep	Numeric	10
■	nValue	Numeric	0
■	nWidth	Numeric	120

1.6.3.14.1.1 TProgressBar:IMarquee

The progress indicator does not grow in size but instead moves repeatedly along the length of the bar, indicating activity without specifying what proportion of the progress is complete.

Scope	Design assignable
Type	Logic

Initial value	.F.
See also	nMarqueeSpeed

1.6.3.14.1.2 TProgressBar:ISmooth

Indicate the style to be used to fill the Progress Bar. If it is active, it will show the Progress Indicator like a filled rectangle, otherwise, it will show it segmented.

Scope	Design assignable
Type	Logic
Initial value	.F.

Remarks:

This property is only supported in the Windows Classic theme.

1.6.3.14.1.3 TProgressBar:nHeight

Indicates the control height (vertical size).

Scope	Assignable
Type	Numeric
Initial value	25

1.6.3.14.1.4 TProgressBar:nMax

Indicates the maximum value in the range indicated by the control. If it is not established, the operating system assigns 100.

Scope	Assignable
Type	Numeric
Initial value	100

1.6.3.14.1.5 TProgressBar:nMarqueeSpeed

Speed in milliseconds of animation.

Scope	Assignable
Type	Numeric

Initial value	100
----------------------	-----

See also	IMarquee
-----------------	----------

1.6.3.14.1.6 TProgressBar:nMin

Indicates the minimum value in the range indicated by the control. If it is not established, the operating system assigns 0.

Scope	Assignable
--------------	------------

Type	Numeric
-------------	---------

Initial value	0
----------------------	---

1.6.3.14.1.7 TProgressBar:nOrientation

indicates if the Progress Bar will be shown in horizontal or vertical position.

Scope	Design assignable
--------------	-------------------

Type	Numeric
-------------	---------

Initial value	orHORIZONTAL
----------------------	--------------

Possible values	orHORIZONTAL, orVERTICAL
------------------------	--------------------------

1.6.3.14.1.8 TProgressBar:nStep

Indicates the Progress Bar increment.

Scope	Assignable
--------------	------------

Type	Numeric
-------------	---------

Initial value	10
----------------------	----

1.6.3.14.1.9 TProgressBar:nValue

Indicates the current position in the Progress Bar.

Scope	Assignable
--------------	------------

Type	Numeric
-------------	---------

Initial value	0
----------------------	---

1.6.3.14.1.10 TProgressBar:nWidth

Indicates the control width (horizontal size).

Scope	Assignable
Type	Numeric
Initial value	120

1.6.3.14.2 TProgressBar:Methods

■ Constructor ■ Standard

Type	Name
■	SetBarColor
■	SetBkColor
■	StepIt

1.6.3.14.2.1 TProgressBar:SetBarColor

Sets the bar color.

Type	Standard
Parameters	<nColor> New bar color
Return value	NIL

Remarks:

This property is only supported in the Windows Classic theme.

Consult the appendix for the list of available colors

1.6.3.14.2.2 TProgressBar:SetBkColor

Sets the background color.

Type	Standard
Parameters	<nColor> New background color
Return value	NIL

Remarks:

This property is only supported in the Windows Classic theme.

Consult the appendix for the list of available colors

1.6.3.14.2.3 TProgressBar:StepIt

Increases the progress bar in the amount assigned in the nStep property.

Type	Standard
Parameters	None
Return value	<nPrevPos> The previous position

1.6.3.14.3 TProgressBar:Events

Name	OnChange
-------------	----------

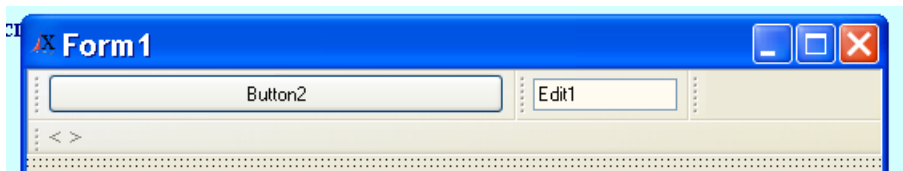
1.6.3.14.3.1 TProgressBar:OnChange

Event that is produced when the control changes its state.

Parameters	<oSender>: Reference to the object that triggers the event
:	<nPos>: New Position
Return value:	NIL

1.6.3.15 TRebar

This class manage the standard Rebar Windows control, that manages one or more bars treated as a unique control and normally are aligned to the top of its container form. Every TRebar object is a TRebarBand object.



Hierarchy	Inherits from TWinControl
See also	TRebarBand
File name	\source\Rebar.prg

1.6.3.15.1 TRebar:Properties

■ read Only ■ Assignable ■ Design assignable ■ Run-time assignable

Scope	Name	Type	Initial value
■	altems	Array	{}
■	IBandBorders	Logic	.T.
■	IBorder	Logic	.T.
■	IDblClkToggle	Logic	.F.
■	IFixedOrder	Logic	.F.
■	IVarHeight	Logic	.T.
■	nAlign	Numeric	a!TOP
■	nHeight	Numeric	30
■	nWidth	Numeric	120
■	oImageList	Object	TImageList

1.6.3.15.1.1 TRebar:altems

Array with all the TRebarBand objects included in the control. The TRebarBand bands are added through the AddItem method.

Scope	read Only
Type	Array
Initial value	{}

1.6.3.15.1.2 TRebar:IBandBorders

If it is .T. the control will show narrow bands to separate every contiguous band.

Scope	Design assignable
Type	Logic
Initial value	.T.

1.6.3.15.1.3 TRebar:IBorder

If it is .T. the control will show a border.

Scope	Design assignable
Type	Logic
Initial value	.T.

1.6.3.15.1.4 TRebar:IDbClkToggle

If it is .T. the bands will change its state (maximizing or minimizing) through a double click in the mouse. If it is .F. the state change will be done with a simple mouse click.

Scope	Design assignable
Type	Logic
Initial value	.F.

1.6.3.15.1.5 TRebar:IFixedOrder

If it is .T. the bands will be shown always in the same order.

Scope	Design assignable
Type	Logic
Initial value	.F.

1.6.3.15.1.6 TRebar:nAlign

Control alignment in its oParent object container.

Scope:	Design assignable
Type:	Numeric
Initial value:	aITOP
Possible values:	aINONE, aILEFT, aITOP, aIRIGHT, aIBOTTOM, aIClient

Description:

This property allows to adjust the control dimensions and position to its oParent objet container. The alignment can be:

- **None:** Default value
- **Left:** The control is aligned to the left of its oParent control and takes the height from the area of its container client.
- **Top:** The control is aligned to the top of its oParent control and takes the width from the area of its container client.
- **Right:** The control is aligned to the right of its oParent control and takes the height from the area of its container client.
- **Bottom:** The control is aligned to the bottom of its oParent control and takes the width from the area of its container client
- **Client:** The control is aligned to all the client area of its oParent control adjusting its size to the control area.

1.6.3.15.1.7 TRebar:IVarHeight

If it is .T. the bands will be shown with the minimum needed height. If it is .F. all the control bands will be shown with the same height, using the maximum height from all the visible bands.

Scope	Design assignable
Type	Logic
Initial value	.T.

1.6.3.15.1.8 TRebar:nHeight

Indicates the control height.

Scope:	Design assignable
Type:	Numeric
Initial value:	30

1.6.3.15.1.9 TRebar:nWidth

Indicates the control width.

Scope:	Design assignable
Type:	Numeric
Initial value:	120

1.6.3.15.1.10 TRebar:olmagedList

TImageList object with all the images to be used by the control.

Scope	Design assignable
Type	Object
Initial value	TImageList()

The TImageList is instantiated in the control creation with its New() constructor. The first image included in the TImageList indicates the dimension for the following images to be added. If the first image includes more than one bitmap, it is important to establish the nHeight and nWidth properties from the TImageList before to add any bitmap.

1.6.3.15.2 TRebar:Methods

■ Constructor
 ■ Standard
 ■ Only after Create()

Type	Name
■	AddItem

1.6.3.15.2.1 TRebar:Additem

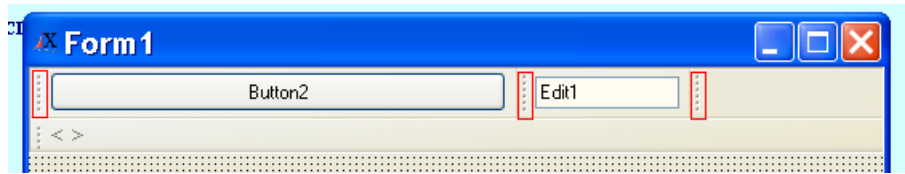
Creates a new TRebarBand band in the control.

Type	Standard
Parameters	<p>[<cText>] Text to be shown in the band. Default: no text.</p> <p>[<nImage>] image number from its olmageList property. Default value: 0.</p> <p>[<oBkgnd>] TPicture object to use as a background image. Default: no image.</p> <p>[<IBreak>] If it is .T. the band will use a new line. Default: .T.</p> <p>[<IChildEdge>] If it is .T. the band will have a border below its daughter window. Default: .F.</p> <p>[<IFixedBmp>] The oBkgnd object will not move when the band is resized. Default: .F.</p> <p>[<IFixedSize>] If it is .T. the band can not be resized and it will not show the resize handle. Default: .F.</p> <p>[<IGripper>] If it is .T. the band will show the resize handle even when it is the only existing band. Default: rgDEFAULT.</p> <p>[<IVisible>] If it is .T. the band will be visible. Default; .T.</p>
Return value	Object TRebarBand

For more information see also the TRebarBand class.

1.6.3.16 TRebarBand

Class to manage every band from the standard TReBar Windows control.



The zones marked with a red rectangle allow to select the object with a mouse click from the IDE.

Hierarchy	Inherits from TComponent
See also	TReBar
File name	\source\RebarBand.prg

1.6.3.16.1 TRebarBand:Properties

■ read Only ■ Assignable ■ Design assignable ■ Run-time assignable

Scope	Name	Type	Initial value
■	cText	Character	""
■	IBreak	Logic	.F.
■	IChildEdge	Logic	.F.
■	IFixedBmp	Logic	.F.
■	IFixedSize	Logic	.F.
■	IGripper	Logic	rgDEFAULT
■	IVisible	Logic	.T.
■	nImage	Numeric	0
■	nIndex	Numeric	0
■	oBkgnd	Object	NIL
■	oControl	Object	NIL

1.6.3.16.1.1 TRebarBand:cText

Text to be shown in the band. Optional

Scope	Assignable
Type	Character
Initial value	""

1.6.3.16.1.2 TRebarBand:IBreak

If it is .T. the band will be shown in a new line.

Scope	Assignable
Type	Logic
Initial value	.F.

1.6.3.16.1.3 TRebarBand:IChildEdge

If it is .T. the band will have a border in the top and bottom of its daughter window.

Scope	Assignable
Type	Logic
Initial value	.F.

1.6.3.16.1.4 TRebarBand:IFixedBmp

If it is .T. the oBkgnd object will not move when the band is resized.

Scope	Assignable
Type	Logic
Initial value	.F.

1.6.3.16.1.5 TRebarBand:IFixedSize

If it is .T. the band can not be resized and the size handler will not be shown.

Scope	Assignable
Type	Logic
Initial value	.F.

Note: This property was named **IFixed** before.

1.6.3.16.1.6 TRebarBand:IGripper

If it is .T. the band will show the size handler even when it is the only existing band.

Scope	Assignable
--------------	------------

Type	Logic
Initial value	rgDEFAULT
Possible values	<ul style="list-style-type: none"> • rgDEFAULT: The size handler is shown when there is more than one band. • rgALWAYS: The size handles is always shown • rgNEVER: The size handles will not be shown

Note: This property was named **IFixed** before.

1.6.3.16.1.7 TRebarBand:IVisible

If it is .T. the band will be visible.

Scope	Assignable
Type	Logic
Initial value	.T.

1.6.3.16.1.8 TRebarBand:nImage

Image number from the oImageList object from its TRebar container object.

Scope	Assignable
Type	Numeric
Initial value	0

1.6.3.16.1.9 TRebarBand:nIndex

Indicates the bar creation order in its TRebar object container.

Scope	readOnly
Type	Numeric
Initial value	0

1.6.3.16.1.10 TRebarBand:oBkgnd

TPicture object to be shown as a band background.

Scope	Standard
Type	Object

Initial value	NIL
----------------------	-----

1.6.3.16.1.11 TRebarBand:oControl

Control to place in the band. You can set any control type that inherits from TStdControl.

Scope	Standard
Type	Object
Initial value	NIL

Note:The **nWidth** property of this control will indicates the minimum width of the bar when it is minimized.

1.6.3.16.2 TRebarBand:Methods

■ Constructor ■ Standard

Typ	Name
■	Create
■	Delete

1.6.3.16.2.1 TRebarBand:Create

Creates a TRebarBand objet.

Type	Constructor
Parameters	<p><oParent> TRebar container object</p> <p>[<cText>] Text to be shown in the band. Default: no Text</p> <p>[<nImage>] Image number from its olmageList property. Default: 0</p> <p>[<oBkgnd>] TPicture object to be used as background image. Default: No image.</p> <p>[<IBreak>] If it is .T. the band will use a new line. Default: .T.</p> <p>[<IChildEdge>] If it is .T. the band will have a border in the top and bottom of its daughter window.</p> <p>[<IFixedBmp>] The oBkgnd object will not move when the band is resized. Default: .F.</p> <p>[<IFixedSize>] If it is .T. the band can not be resized and the</p>

	resize handle will not be shown either. Default: .F. [<IGripper>] If it is .T. the band will show the resize handle, even if it is the only existing band. Default: rgDEFAULT. [<IVisible>] If it is .T. the band will be visible. Default: .T.
Return value	<oRebarBand> TRebarBand object

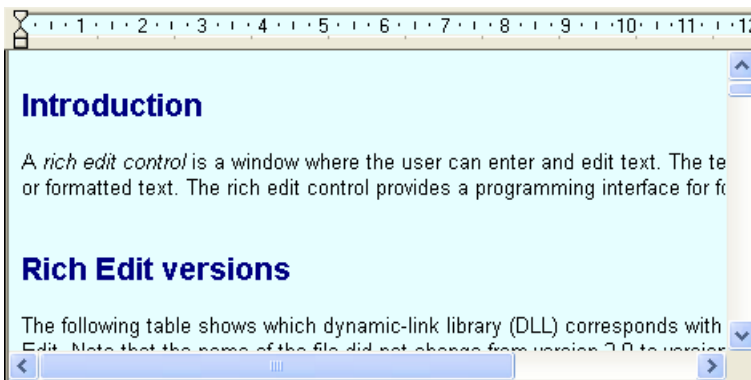
1.6.3.16.2 TRebarBand:Delete

Removes a band from its TRebar container object.

Type	Standard
Parameters	None
Return value	NIL

1.6.3.17 TRichEdit

This class represents a Windows RichEdit control to edit text. It offers more features than the basic TMemo control.



Hierarchy	Inherits from TStdControl
See also	TMemo
File name	\source\RichEdit.prg

1.6.3.17.1 TRichEdit:Properties

read Only
 Assignable
 Design assignable
 Run-time assignable

Scope	Name	Type	Initial value
<input type="checkbox"/>	aMargins	Array	{ 2.5, 2.5, 2.5, 2.5 }

■	cFileName	Character	""
■	cFindText	Character	""
■	cFontName	Character	"Arial"
■	cReplace	Character	""
■	cText	Character	""
■	IBold	Logic	.F.
■	IBorder	Logic	.T.
■	IDisableNoScroll	Logic	.F.
■	IFindCase	Logic	.F.
■	IFindDown	Logic	.T.
■	IFindWord	Logic	.F.
■	IHScroll	Logic	.T.
■	IHot	Logic	.F.
■	IHTML	Logic	.F.
■	IItalic	Logic	.F.
■	IPDF	Logic	.F.
■	IProtected	Logic	.F.
■	IRE30	Logic	.F.
■	IRE40	Logic	.F.
■	IRE50	Logic	.F.
■	IRE60	Logic	.F.
■	IRE70	Logic	.F.
■	IRE80	Logic	.F.
■	IReadOnly	Logic	.F.
■	IRule	Logic	.F.
■	IShowMessages	Logic	.F.
■	ISmallCaps	Logic	.F.
■	IStrikeOut	Logic	.F.
■	ISubScript	Logic	.F.
■	ISupScript	Logic	.F.
■	ITransparent	Logic	.F.
■	IURLDetect	Logic	.T.
■	IUnderline	Logic	.F.
■	IVScroll	Logic	.T.
■	IWord	Logic	.F.
■	IWordEx	Logic	.F.
■	IWordX	Logic	.F.
■	IWrite	Logic	.F.
■	IWordSel	Logic	.T.
■	IWordWrap	Logic	.T.
■	nAlignment	Numeric	rpLEFT
■	nClrPane	Numeric	clWindow
■	nClrText	Numeric	clWindowText
■	nConverter	Numeric	rcNONE
■	nFontColor	Numeric	0
■	nFontSize	Numeric	10

■	nFormat	Numeric	rfRTF
■	nHeight	Numeric	90
■	nHighLight	Numeric	clWhite
■	nLineSpacing	Numeric	rISINGLE
■	nMaxLength	Numeric	-1
■	nMaxUndo	Numeric	10
■	nNumStyle	Numeric	rpPAREN
■	nNumbering	Numeric	rpARABIC
■	nPageWidth	Numeric	16
■	nUlineColor	Numeric	rcBLACK
■	nUlineType	Numeric	ruLINE
■	nUnits	Numeric	ruCENTIMETERS
■	nWidth	Numeric	120
■	oPopup	Object	NIL

1.6.3.17.1.1 TRichEdit:aMargins

Establishes the page margins, according with the nUnits value. The positions for this array are:

1. Left
2. Upper
3. Right
4. button

Scope:	Assignable
Type:	Array
Initial value:	{ 2.5, 2.5, 2.5, 2.5 }

1.6.3.17.1.2 TRichEdit:cFileName

File name that contains the text in rfTEXT o rfRTF formats.

Scope:	Assignable
Type:	Character
Initial value:	""

1.6.3.17.1.3 TRichEdit:cFindText

Initial text to be found.

Scope:	Assignable
Type:	Character
Initial value:	""

1.6.3.17.1.4 TRichEdit:cFontName

Indicates the font name.

Scope:	Assignable
Type:	Character
Initial value:	"Arial"

1.6.3.17.1.5 TRichEdit:cReplace

Indicates the initial text to be replaced.

Scope:	Assignable
Type:	Character
Initial value:	""

1.6.3.17.1.6 TRichEdit:cText

Text to be shown in the control. If the cFileName property is empty, it is set in the control the cText content.

Scope:	Assignable
Type:	Character
Initial value:	""

Note: The returned value of cText is in RTF format. To get the value on plain text format use the method GetText.

1.6.3.17.1.7 TRichEdit:lBold

If it is .T. the font will be formatted in bold.

Scope:	Assignable
Type:	Logic
Initial value:	.F.

1.6.3.17.1.8 TRichEdit:IBorder

If it is .T. the control will show a border.

Scope:	Design assignable
Type:	Logic
Initial value:	.T.

1.6.3.17.1.9 TRichEdit:IDisableNoScroll

Deactivates scrollbars, instead of hidden them when not necessary.

Scope:	Assignable
Type:	Logic
Initial value:	.F.

1.6.3.17.1.10 TRichEdit:IFindCase

Indicates case sensitivity.

Scope:	Assignable
Type:	Logic
Initial value:	.F.

1.6.3.17.1.11 TRichEdit:IFindDown

Indicates the initial search direction. It can be:

- .T.: to the end
- .F.: to the beginning

Scope:	Assignable
Type:	Logic
Initial value:	.T.

1.6.3.17.1.12 TRichEdit:IFindWord

Searches only complete words.

Scope:	Assignable
Type:	Logic
Initial value:	.F.

1.6.3.17.1.13 TRichEdit:IHScroll

If it is .T. it will show an horizontal scroll bar.

Scope:	Assignable
Type:	Logic
Initial value:	.F.

1.6.3.17.1.14 TRichEdit:IHot

If it is .T. the border will change its color when it receives focus or the mouse moves on top of it.

Scope:	Assignable
Type:	Logic
Initial value:	.F.

1.6.3.17.1.15 TRichEdit:IHTML

If it is .T. indicates that the computer has installed the library distributed by Microsoft Office to convert a file in HTML format to RTF. This library is HTML32.cnv.

Scope:	read Only
Type:	Logic
Initial value:	.F.
See also:	RTFtoHTML, HTMLtoRTF

1.6.3.17.1.16 TRichEdit:Italic

If it is .T. the font will be formatted in italics.

Scope:	Assignable
---------------	------------

Type:	Logic
Initial value:	.F.

1.6.3.17.1.17 TRichEdit:IPDF

If it is .T. the dynamic link library Image2Pdf.dll, is present so it is possible to save the documents as PDF files.

Scope:	readOnly
Type:	Logic
Initial value:	.F.
See also:	RTFToPDF

1.6.3.17.1.18 TRichEdit:IProtected

If TRUE the font has a protected format.

Scope:	Assignable
Type:	Logic
Initial value:	.F.

1.6.3.17.1.19 TRichEdit:IRE30

If it is .T. the dynamic link library riched20.dll, encapsulates the rich edit control 3.0 version or beyond, and allows to access some particular features like the numbering, text justification and zoom.

Scope:	readOnly
Type:	Logic
Initial value:	.F.

1.6.3.17.1.20 TRichEdit:IRE40

If it is .T. the dynamic link library riched20.dll, encapsulates the rich edit control 4.0 version or beyond, and allows to access some particular features like Table creation. See method InsertTable for further information.

Scope:	readOnly
Type:	Logic

Initial value:	.F.
-----------------------	-----

1.6.3.17.1.21 TRichEdit:IRE50

If it is .T. the dynamic link library riched20.dll, encapsulates the rich edit control 5.0 version or beyond, and allows to access some particular features like SetHyperLinkTooltips. See method SetHyperLinkTooltips for further information.

Scope:	readOnly
Type:	Logic
Initial value:	.F.

1.6.3.17.1.22 TRicheEdit:IRE60

If it is .T. the dynamic link library riched20.dll, encapsulates the rich edit control 6.0 version or beyond, and allows to access some particular features like table management.

Scope:	readOnly
Type:	Logic
Initial value:	.F.

1.6.3.17.1.23 TRichEdit:IRE70

If it is .T. the dynamic link library riched20.dll, encapsulates the rich edit control 7.0 version or beyond.

Scope:	readOnly
Type:	Logic
Initial value:	.F.

1.6.3.17.1.24 TRichEdit:IRE80

If it is .T. the dynamic link library riched20.dll, encapsulates the rich edit control 8.0 version or beyond.

Scope:	readOnly
Type:	Logic
Initial value:	.F.

Note: With this version underline color can be modified

1.6.3.17.1.25 TRichEdit:IRedOnly

If it is .T. the control will be treated as read only.

Scope:	Design assignable
Type:	Logic
Initial value:	.F.

1.6.3.17.1.26 TRichEdit:IRule

If it is .T. it will show the rule indicator in the control.

Scope:	Assignable
Type:	Logic
Initial value:	.F.

1.6.3.17.1.27 TRichEdit:IShowMessages

If it is .T. the methods FindText and ReplaceAll will show information results.

Scope:	Assignable
Type:	Logic
Initial value:	.F.

1.6.3.17.1.28 TRichEdit:ISmallCaps

If TRUE the font has SmallCaps format.

Scope:	Assignable
Type:	Logic
Initial value:	.F.

1.6.3.17.1.29 TRichEdit:IStrikeOut

If it is .T. the font will be formatted in strike out format.

It only works for version 5.0 or above of RichEd20.dll

Scope:	Assignable
Type:	Logic
Initial value:	.F.

1.6.3.17.1.30 TRichEdit:ISubScript

If it is .T. the font will be formatted in subscript format.

Scope:	Assignable
Type:	Logic
Initial value:	.F.

1.6.3.17.1.31 TRichEdit:ISupScript

If it is .T. the font will be formatted in super script format.

Scope:	Assignable
Type:	Logic
Initial value:	.F.

1.6.3.17.1.32 TRichEdit:ITransparent

If true, the control will have a transparent background.

Scope:	Assignable
Type:	Logic
Initial value:	.F.

Note: This property has some limitations:

- Only one **TRichEdit** control can have this property to .T. on the same form
- Its container must be the form itself
- Can only be assigned on its OnCreate event

Otherwise, the edge and the rule that the control can show, it is very possible that it works erratically. If you are not going to show the control rule, you have the option to show the control without border and use the border of a container control, such as a TBevel.

1.6.3.17.1.33 TRichEdit:IURLDetect

If it is .T. the control automatically detects hyperlinks that start with the following prefixes:

- http:
- file:
- mailto:
- ftp:
- https:
- gopher:
- nntp:
- prospero:
- telnet:
- news:
- wais:

Scope:	Assignable
Type:	Logic
Initial value:	.T.

1.6.3.17.1.34 TRichEdit:IUnderline

If it is .T. the font will be formatted underlined.

Scope:	Assignable
Type:	Logic
Initial value:	.F.

1.6.3.17.1.35 TRichEdit:IVScroll

If it is .T. the control will show a vertical scroll bar.

Scope:	Assignable
Type:	Logic
Initial value:	.T.

1.6.3.17.1.36 TRichEdit:IWord

If it is .T. indicates that the computer has installed the libraries distributed by Microsoft Office to convert Word 6.0 and word97 formats to RTF. Those libraries are MsWrd632.wpc and MsWrd832.cnv.

Scope:	read Only
Type:	Logic
Initial value:	.F.
See also:	WordToRTF

1.6.3.17.1.37 TRichEdit:IWordEx

If it is .T. indicates the system has installed the Microsoft library to convert RTF documents to Microsoft Word and Microsoft Word extended format.

Scope:	read Only
Type:	Logic
Initial value:	.F.
See also:	RTFToWord, RTFToWordX

1.6.3.17.1.38 TRichEdit:IWordX

If it is .T. indicates the system has installed the Microsoft library to convert Microsoft Word extended documents to RTF. This library is Wordcnvpxy.cnv.

Scope:	read Only
Type:	Logic
Initial value:	.F.
See also:	WordXToRTF

1.6.3.17.1.39 TRichEdit:IWrite

If it is .T. indicates that the computer has installed the library distributed by Microsoft Office to convert Write file format to RTF. This library is Write32.wpc.

Scope:	read Only
Type:	Logic
Initial value:	.F.
See also:	WriteToRTF

1.6.3.17.1.40 TRichEdit:IWordSel

If it is .T. activates the automatic word selection. It means that if you start selecting text in the middle of a word and moves the selection to the next word, the begin of the selection will be expanded to the left to include the whole word.

Scope:	Assignable
Type:	Logic
Initial value:	.T.

1.6.3.17.1.41 TRichEdit:IWordWrap

If it is .T. activates word wrap.

Scope:	Assignable
Type:	Logic
Initial value:	.T.

1.6.3.17.1.42 TRichEdit:nAlignment

Indicates the text alignment.

Scope:	Assignable
Type:	Numeric
Initial value:	rpLEFT
Possible values:	rpLEFT, rpRIGHT, rpCENTER, rpJUSTIFY

1.6.3.17.1.43 TRichEdit:nClrPane

Indicates the background color.

Scope:	Assignable
Type:	Numeric
Initial value:	clWindow

(See also the appendix to check colors available)

1.6.3.17.1.44 TRichEdit:nClrText

Indicates the text color in the control.

Scope:	Assignable
Type:	Numeric
Initial value:	clWindowText

(See also the appendix to check colors available)

1.6.3.17.1.45 TRichEdit:nConverter

Indicates the active converter type. It will be reestablished internally after the HTML, Write or World file conversion to RTF.

Scope:	Assignable
Type:	Numeric
Initial value:	rcNONE
Possible values:	rcNONE, rcHTML, rcWRITE, rcWORD, rcWORDX

1.6.3.17.1.46 TRichEdit:nFontColor

Indicates the font color. The value of this property is in RGB(Red, Green, Blue) format.

Scope:	Assignable
Type:	Numeric
Initial value:	0 (Black)

1.6.3.17.1.47 TRichEdit:nFontSize

Indicates the font size.

Scope:	Assignable
Type:	Numeric
Initial value:	10

1.6.3.17.1.48 TRichEdit:nFormat

Indicates the control's content format.

Scope:	Assignable
Type:	Numeric
Initial value:	rfRTF
Possible values:	rfRTF, rfTEXT

1.6.3.17.1.49 TRichEdit:nHeight

Indicates the control height.

Scope:	Assignable
Type:	Numeric
Initial value:	90

1.6.3.17.1.50 TRichEdit:nHighLight

Indicates the highlight font color. The value of this property is in RGB(Red, Green, Blue) format.

Scope:	Assignable
Type:	Numeric
Initial value:	clWhite

1.6.3.17.1.51 TRichEdit:nLineSpacing

Indicates the space among lines.

Scope:	Assignable
Type:	Numeric
Initial value:	rSINGLE
Possible values:	rSINGLE, rONEHALF, rDOUBLE, rTWOHALF, rTRIPLE, rEXACT

1.6.3.17.1.52 TRichEdit:nMaxLength

Indicates the maximum amount of text that can be introduced in the control. This property is zero, the maximum is established in 64ks (65.536 characters). However, the RichEdit control can manage to 1 Gb ((1.048.578 characters). By default, -1, to establish the maximum allowed.

Scope:	Assignable
Type:	Numeric
Initial value:	-1

1.6.3.17.1.53 TRichEdit:nMaxUndo

Establishes the maximum number of actions that can be saved in the "Undo" queue.

Scope:	Assignable
Type:	Numeric
Initial value:	10

NOTE: We don't recommend to increase this number more than 100. In that case, it will use a lot of memory.

1.6.3.17.1.54 TRichEdit:nNumStyle

Indicates the numbering style.

Scope:	Assignable
Type:	Numeric
Initial value:	rpPAREN
Possible values:	rpPAREN, rpPARENS, rpPERIOD, rpPLAIN

1.6.3.17.1.55 TRichEdit:nNumbering

Indicates the numbering type.

Scope:	Assignable
Type:	Numeric
Initial value:	rpARABIC
Possible values:	rpNONE, rpARABIC, rpLCLETTER, rpUCLETTER, rpLCROMAN, rpUCROMAN

1.6.3.17.1.56 TRichEdit:nPageWidth

Indicates the page width. This property is read only and is calculated taking into consideration the aMargins parameters. For example, if the page is DIN4 and the left and right margins are 2.5 cm, this property will have the value of 16 cm.

Scope:	read Only
Type:	Numeric
Initial value:	16

1.6.3.17.1.57 TRichEdit:nUlineColor

Indicates the underline color. Only available on version 8. See property IRE80.

Scope:	Assignable
Type:	Numeric
Initial value:	rcBLACK

1.6.3.17.1.58 TRichEdit:nUlineType

Indicates the underline format.

Scope:	Assignable
Type:	Numeric
Initial value:	ruLINE
Possible values:	ruNONE, ruLINE, ruWORD, ruDOUBLE, ruDOTTED, ruDASH, ruDASHDOT, ruDASHDOTDOT, ruWAVE, ruTHICK

1.6.3.17.1.59 TRichEdit:nUnits

Indicates measure unit.

Scope:	Assignable
Type:	Numeric
Initial value:	ruCENTIMETERS
Possible values:	ruTWIPS, ruCENTIMETERS, ruINCHES, ruPIXELS

1.6.3.17.1.60 TRichEdit:nWidth

Indicates the control width.

Scope:	Assignable
Type:	Numeric
Initial value:	120

1.6.3.17.1.61 TRichEdit:oPopup

Pop-up menu object that is activates when the OnRButtonDown event is triggered. This object is created during run-time by the CreateMenuPopup method. However, it can be assigned by the OnContextMenu event.

Scope:	read Only
Type:	Object
Initial value:	NIL

1.6.3.17.2 TRichEdit:Methods

■ Constructor ■ Standard ■ Only after Create()

Typ	Name
■	CanClear
■	CanCopy
■	CanCut
■	CanPaste
■	CanRedo
■	CanUndo
■	CharFromPos
■	Clear
■	ColorDlg
■	Copy
■	Cut
■	EmptyUndoBuffer
■	FindDlg
■	FindText
■	FontDlg
■	GetBullet
■	GetCol
■	GetEventMask
■	GetFirstVisibleLine
■	GetLine
■	GetLineCount

<input type="checkbox"/>	GetLineFromChar
<input type="checkbox"/>	GetLineIndex
<input type="checkbox"/>	GetIndent
<input type="checkbox"/>	GetModify
<input type="checkbox"/>	GetNumbering
<input type="checkbox"/>	GetOptions
<input type="checkbox"/>	GetParagraphSpacing
<input type="checkbox"/>	GetPos
<input type="checkbox"/>	GetRedoName
<input type="checkbox"/>	GetPrintHeight
<input type="checkbox"/>	GetRow
<input type="checkbox"/>	GetRTF
<input type="checkbox"/>	GetRTFRange
<input type="checkbox"/>	GetSel
<input type="checkbox"/>	GetSelection
<input type="checkbox"/>	GetSelectionType
<input type="checkbox"/>	GetText
<input type="checkbox"/>	GetTextMode
<input type="checkbox"/>	GetTextRange
<input type="checkbox"/>	GetTypographyOptions
<input type="checkbox"/>	GetUndoName
<input type="checkbox"/>	GetZoom
<input type="checkbox"/>	GotoLine
<input type="checkbox"/>	HasText
<input type="checkbox"/>	HTMLToRTF
<input type="checkbox"/>	IsSelection
<input type="checkbox"/>	InsertBitmap
<input type="checkbox"/>	InsertFile
<input type="checkbox"/>	InsertObject
<input type="checkbox"/>	InsertObjectFromFile
<input type="checkbox"/>	InsertPageBreak
<input type="checkbox"/>	InsertPicture
<input type="checkbox"/>	InsertRTF
<input type="checkbox"/>	InsertTable
<input type="checkbox"/>	InsertText
<input type="checkbox"/>	IsRTF
<input type="checkbox"/>	Len
<input type="checkbox"/>	LoadFile
<input type="checkbox"/>	LoadRTF
<input type="checkbox"/>	MovePos
<input type="checkbox"/>	PageDlg
<input type="checkbox"/>	Paste
<input type="checkbox"/>	PasteSpecial
<input type="checkbox"/>	PictureDlg
<input type="checkbox"/>	PosFromChar
<input type="checkbox"/>	Preview

<input type="checkbox"/>	Print
<input type="checkbox"/>	PrintBox
<input type="checkbox"/>	PrintDlg
<input type="checkbox"/>	PrintSetup
<input type="checkbox"/>	Redo
<input type="checkbox"/>	Replace
<input type="checkbox"/>	ReplaceAll
<input type="checkbox"/>	ReplaceDlg
<input type="checkbox"/>	RTFToHTML
<input type="checkbox"/>	RTFToPDF
<input type="checkbox"/>	RTFToWord
<input type="checkbox"/>	RTFToWordX
<input type="checkbox"/>	RTFToWrite
<input type="checkbox"/>	SaveFile
<input type="checkbox"/>	SelectAll
<input type="checkbox"/>	SelectNone
<input type="checkbox"/>	SetBullet
<input type="checkbox"/>	SetEventMask
<input type="checkbox"/>	SetHideSelection
<input type="checkbox"/>	SetHyperLinkTooltips
<input type="checkbox"/>	SetIndent
<input type="checkbox"/>	SetModify
<input type="checkbox"/>	SetNumbering
<input type="checkbox"/>	SetOptions
<input type="checkbox"/>	SetParagraphSpacing
<input type="checkbox"/>	SetPos
<input type="checkbox"/>	SetSel
<input type="checkbox"/>	SetText
<input type="checkbox"/>	SetTextMode
<input type="checkbox"/>	SetTypographicOptions
<input type="checkbox"/>	SetZoom
<input type="checkbox"/>	StopGroupTyping
<input type="checkbox"/>	Undo
<input type="checkbox"/>	WordToRTF
<input type="checkbox"/>	WordXToRTF
<input type="checkbox"/>	WriteToRTF

1.6.3.17.2.1 TRichEdit:CanClear

Checks if the control can clear the current selection without copying it to the clipboard.

Type	Only after Create()
Parameters	None
Return value	<ICan> .T. if it is possible

1.6.3.17.2.2 TRichEdit:CanCopy

Checks if the control can copy the current selection to the clipboard.

Type	Only after Create()
Parameters	None
Return value	<ICan> .T. if it is possible

1.6.3.17.2.3 TRichEdit:CanCut

Checks if the control can cut the current selection copying it to the clipboard.

Type	Only after Create()
Parameters	None
Return value	<ICan> .T. if it is possible

1.6.3.17.2.4 TRichEdit:CanPaste

Checks if the control can paste the current selection from the clipboard.

Type	Only after Create()
Parameters	None
Return value	<ICan> .T. if it is possible

1.6.3.17.2.5 TRichEdit:CanRedo

Checks if the control can ReDo an operation.

Type	Only after Create()
Parameters	None
Return value	<ICan> .T. if it is possible

1.6.3.17.2.6 TRichEdit:CanUndo

Checks if the control can Undo an operation.

Type	Only after Create()
Parameters	None
Return value	<ICan> .T. if it is possible

1.6.3.17.2.7 TRichEdit:CharFromPos

Returns the closest character index based on the parameters given. The index is the number of characters from the beginning of the control

Type	Only after Create()
Parameters	<nCol> Column <nRow> Row
Return value	<nChar> Character index

1.6.3.17.2.8 TRichEdit:Clear

Clears the current selection without copying it to the clipboard.

Type	Only after Create()
Parameters	None
Return value	NIL

1.6.3.17.2.9 TRichEdit:ColorDlg

Executes the native ChooseColor Windows dialog using the TChooseColorDlg class.

Type	Only after Create()
Parameters	<IHighlight> If true the property nHighlight is set, otherwise nFontColor
Return value	NIL

1.6.3.17.2.10 TRichEdit:Copy

Copy the current selection to the clipboard.

Type	Only after Create()
Parameters	None
Return value	NIL

1.6.3.17.2.11 TRichEdit:Cut

Cuts the current selections to the clipboard.

Type	Only after Create()
Parameters	None
Return value	NIL

1.6.3.17.2.12 TRichEdit:EmptyUndoBuffer

Clears Undo and Redo buffers.

Type	Only after Create()
Parameters	None
Return value	NIL

1.6.3.17.2.13 TRichEdit:FindDlg

Executes the operating system native dialog Find, through the TFindDlg class. Uses the cFindText, IFindDown, IFindWord and IFindCase properties.

Type	Only after Create()
Parameters	None
Return value	NIL

1.6.3.17.2.14 TRichEdit:FindText

Starts a search process. If the text is found, the text is selected. If the property IShowMessages is set to true a result information will be shown.

Type	Only after Create()
Parameters	<cText> : text to search <IDown> : Initial search direction. Default: IFindDown value. <IWord> : Searches only complete words. Default: IFindWord value. <ICase> : Case sensitivity. Default: IFindCase value
Return value	NIL

1.6.3.17.2.15 TRichEdit:FontDlg

Executes the operating system native dialog ChooseFont, through the TChooseFontDlg class. It receives all the parameters from the current selection. If the dialog is closed with "OK", all the selection values are updated.

Type	Only after Create()
Parameters	None
Return value	NIL

1.6.3.17.2.16 TRichEdit:GetBullet

Checks if paragraph has bullets.

Type	Only after Create()
Parameters	None
Return value	<IOk> .T. if true

1.6.3.17.2.17 TRichEdit:GetCol

Returns the current cursor column position.

Type	Only after Create()
-------------	---------------------

Parameters	None
Return value	<nCol> Column number

1.6.3.17.2.18 TRichEdit:GetEventMask

Returns the notification code mask sent to its parent Window.

Type	Only after Create()
Parameters	None
Return value	<nMask> Notification mask

1.6.3.17.2.19 TRichEdit:GetFirstVisibleLine

Returns the number of the first visible line in the control.

Type	Only after Create()
Parameters	None
Return value	<nLine> Line number

1.6.3.17.2.20 TRichEdit:GetLine

Returns a text line.

Type	Only after Create()
Parameters	[<nLine>] Line number. By default actual line.
Return value	<cText> Text line

1.6.3.17.2.21 TRichEdit:GetLineCount

Returns the number of lines of the control.

Type	Only after Create()
Parameters	None
Return value	<nLines> Number of lines

1.6.3.17.2.22 TRichEdit:GetLineFromChar

Retrieves the line that contains a specific character.

Type	Only after Create()
Parameters	<nChar> Character to search
Return value	<nLine> Line number or zero

1.6.3.17.2.23 TRichEdit:GetLineIndex

Returns the index of the first character on the specified line. The index is the number of characters from the beginning of the control.

Type	Only after Create()
Parameters	[<nLine>] Line number. By default actual line
Return value	<nIndex> Character index

1.6.3.17.2.24 TRichEdit:GetIndent

Return an array with left, right and first indentation of actual paragraph.

Type	Only after Create()
Parameters	<nUnits> Measure unit
Return value	<aData> { Left indent, right indent, first line indent }

1.6.3.17.2.25 TRichEdit:GetModify

Return true if the control content has been modified.

Type	Only after Create()
Parameters	None
Return value	<IOk> .T. if true

1.6.3.17.2.26 TRichEdit:GetNumbering

Checks if paragraph is enumerated.

Type	Only after Create()
Parameters	None
Return value	<IOk> .T. if true

1.6.3.17.2.27 TRichEdit:GetOptions

Return a numerical value with the combinations of the control options. See the API message EM_SETOPTIONS for further information.

Type	Only after Create()
Parameters	None
Return value	<nOptions> Numerical value

1.6.3.17.2.28 TRichEdit:GetParagraphSpacing

Return an array with the previous and next spacing from the actual or selected paragraph.

Type	Only after Create()
Parameters	<nUnits> Measure unit
Return value	<aData> { previous spacing, next spacing }

1.6.3.17.2.29 TRichEdit:GetPos

Returns the current cursor position.

Type	Only after Create()
Parameters	None
Return value	<nPos>

1.6.3.17.2.30 TRichEdit:GetRedoName

Returns the type of next action to undo.

Type	Only after Create()
Parameters	None
Return value	<nType> See API message EM_GETREDONAME for further information

1.6.3.17.2.31 TRichEdit:GetRow

Returns the current cursor row position.

Type	Only after Create()
Parameters	None
Return value	<nRow> Row number

1.6.3.17.2.32 TRichEdit:GetPrintHeight

Retrieves the control text height giving a specific width.

Type	Only after Create()
Parameters	[<nWidth>] Text width. By default value calculated with current margins [<nUnits>] Measure unit. See property nUnits for further information
Return value	<nHeight> Text height

1.6.3.17.2.33 TRichEdit:GetRTF

Retrieves the text of the control in RTF format. If IPlainText parameter is not passed returns rich text or plain text depending on the value of nFormat.

Type	Only after Create()
Parameters	[<ISelected>] If true, it returns only the selected text. By default false [<IPlainText>]

	If true returns plain text. By default NIL
Return value	<cRTFText> Control text
See also	GetText, cText, GetRTFRange

1.6.3.17.2.34 TRichEdit:GetRTFRange

Retrieves a text region of the control in RTF format with no need to select any region.

Type	Only after Create()
Parameters	<nStart> Initial position <nEnd> Final position
Return value	<cRTFText> Control text
See also	GetText, cText, GetRTF

1.6.3.17.2.35 TRichEdit:GetSel

Return an array with the start and end character position of the selected text. If both values are equal indicates that there is no selection.

Type	Only after Create()
Parameters	None
Return value	<aData> { start position, end position }

1.6.3.17.2.36 TRichEdit:GetSelection

Return the selected text.

Type	Only after Create()
Parameters	None
Return value	<cText> Selected text
See also	GetSel

1.6.3.17.2.37 TRichEdit:GetSelectionType

Return the selection type.

Type	Only after Create()
Parameters	None
Return value	<nType> See the API message EM_SELECTIONTYPE for further information

1.6.3.17.2.38 TRichEdit:GetText

Retrieves the text of the control in plain text format.

Type	Only after Create()
Parameters	None
Return value	<cText> Control text
See also	GetRTF, cText

1.6.3.17.2.39 TRichEdit:GetTextMode

Retrieves text mode of the control.

Type	Only after Create()
Parameters	None
Return value	<nMode> See API message EM_GETTEXTMODE for further information

1.6.3.17.2.40 TRichEdit:GetTextRange

Retrieves the plain text included on a specific region.

Type	Only after Create()
Parameters	<nStart> Initial position <nEnd> End position
Return value	<cText> Text value

1.6.3.17.2.41 TRichEdit:GetTypographicOptions

Return the typographic options of the control.

Type	Only after Create()
Parameters	None
Return value	<nOptions> See the API message EM_GETTYPOGRAPHICOPTIONS for further information. It only works with RichEd20.dll version 3.0 and above

1.6.3.17.2.42 TRichEdit:GetUndoName

Return the type of next undo operation.

Type	Only after Create()
Parameters	None
Return value	<nType> See the API message EM_GETUNDONAME for further information

1.6.3.17.2.43 TRichEdit:GetZoom

Return an array with the zoom ratio. Value can be between 1/64 and 64.

Type	Only after Create()
Parameters	None
Return value	<aData> { numerator, denominator }
Sample	3/2 (150%), 5/4 (125%), 1/1 (100%), 3/4 (75%), 1/2 (50%),...

1.6.3.17.2.44 TRichEdit:GotoLine

Moves the cursor to the indicated line.

Type	Only after Create()
Parameters	<nLine>: Linea to jump

Return value	NIL
---------------------	-----

1.6.3.17.2.45 TRichEdit:HasText

Checks if control has text.

Type	Only after Create()
Parameters	None
Return value	<IOk> .T. if true

1.6.3.17.2.46 TRichEdit:HTMLToRTF

Converts the HTML file to RFT and inserts it to the control.

Type	Only after Create()
Parameters	<cFile>: File name <IInsert>: If true the text file is inserted.
Return value	<ISuccess>: .T. if the operation is successful

1.6.3.17.2.47 TRichEdit:InsertBitmap

Insert a bitmap through its handle.

Type	Only after Create()
Parameters	<hBitmap>: Bitmap handle [<nSizeX>]: Bitmap width. Default: original width. {<nSizeY>]: Bitmap height . Default: original height.
Return value	<ISuccess>: .T. if the operation is successful

1.6.3.17.2.48 TRichEdit.InsertFile

Inserts a file.

Type	Only after Create()
Parameters	<cFile> File name
Return value	NIL

1.6.3.17.2.49 TRichEdit.InsertObject

Inserts a embedded object.

Type	Only after Create()
Parameters	TOleAutoobject
Return value	NIL

1.6.3.17.2.50 TRichEdit.InsertObjectFromFile

Inserts a embedded object from file.

Type	Only after Create()
Parameters	<cFile> File object
Return value	NIL

1.6.3.17.2.51 TRichEdit.InsertPageBreak

Inserts a page break on the document.

Type	Only after Create()
Parameters	None
Return value	NIL

1.6.3.17.2.52 TRichEdit:InsertPicture

Inserts a image from a file.

Type	Only after Create()
Parameters	<cFile> : File name [<nSizeX>] : Bitmap width. Default: original width. {<nSizeY>} : Bitmap height . Default: original height.
Return value	<ISuccess> : .T. if the operation is successful

1.6.3.17.2.53 TRichEdit:InsertRTF

Inserts a text string with RTF codes.

Type	Only after Create()
Parameters	<cText> : Text string in RTF format
Return value	NIL

1.6.3.17.2.54 TRichEdit:InsertTable

Inserts a table in the document. The dynamic link library riched20.dll should be version 4.0 or beyond. See property IRE40.

Type	Only after Create()
Parameters	[<nRows>] : Number of rows to insert. By default, 1 [<nCols>] : Number of columns to insert. By default, 1 [<nHeigh>] : Row height. By default 0.5 cm. [<nWidth>] : Row width. By default 1 cm. [<nAlignment>] : Table alignment related to the document. Possible values: rpLEFT, rpCENTER y rpRIGHT [<nBackColor>] : Table background color. Can be any of 16 basic palette colors. By default, clWhite [<nUnits>] : Measure unit. Possible values: ruTWIPS, ruCENTIMETERS, ruINCHES. By default, ruCENTIMETERS

Return value	NIL
---------------------	-----

Important note: This feature does not work correctly under Windows 98

1.6.3.17.2.55 TRichEdit:InsertText

Inserts a text string without format.

Type	Only after Create()
Parameters	<cText>: Text string
Return value	NIL

1.6.3.17.2.56 TRichEdit:IsRTF

Checks if the file has text with RTF code.

Type	Only after Create()
Parameters	<cFile>: File name
Return value	<ISuccess>: .T. if it is RTF

1.6.3.17.2.57 TRichEdit:IsSelection

Checks if control has selected text.

Type	Only after Create()
Parameters	None
Return value	<IOk> .T. if true

1.6.3.17.2.58 TRichEdit:Len

Retrieves current text length.

Type	Only after Create()
Parameters	None

Return value	<nLen> Number of characters
---------------------	--------------------------------

1.6.3.17.2.59 TRichEdit:LoadFile

Replaces the control's content with the text string stored in **cFile**.

Type	Only after Create()
Parameters	<cFile>: File name <nFormat>: File format, rfTEXT or rfRTF <Insert>: Inserts the file
Return value	NIL

1.6.3.17.2.60 TRichEdit:LoadRTF

Replaces the control's content with the text string with RFT code store in **cFile**.

Type	Only after Create()
Parameters	<cFile>: File name <Insert>: If true the text file is inserted.
Return value	NIL

1.6.3.17.2.61 TRichEdit:MovePos

Moves the control cursor.

Type	Only after Create()
Parameters	[<nTimes>] Number of times [<nType>] Movement type: <ul style="list-style-type: none"> • rmCHARACTER • rmWORD • rmSENTENCE • rmPARAGRAPH • rmLINE

Return value	NIL
---------------------	-----

1.6.3.17.2.62 TRichEdit:PageDlg

Executes the operating system native dialog PageSetupDlg, through the TPageSetupDlg class.

Type	Only after Create()
Parameters	None
Return value	NIL

1.6.3.17.2.63 TRichEdit:Paste

Pastes the current selections to the clipboard.

Type	Only after Create()
Parameters	None
Return value	NIL

1.6.3.17.2.64 TRichEdit:PasteSpecial

Executes the operating system native dialog Paste Special, allowing to the user to choose the text format for clipboard text to be pasted.

Type	Only after Create()
Parameters	None
Return value	NIL

1.6.3.17.2.65 TRichEdit:PictureDlg

Executes the operating system native dialog FileOpenDlg, through the class, allowing to the use to choose the image file that wants to insert in the RichEdit Control.

Type	Only after Create()
Parameters	None
Return value	NIL

1.6.3.17.2.66 TRichEdit:PosFromChar

Returns the coordiantes of a specific character position.

Type	Only after Create()
Parameters	<nChar> Character index
Return value	<aCoors> { Column, Row }

1.6.3.17.2.67 TRichEdit:Preview

Executes the preliminary control view.

Type	Only after Create()
Parameters	None
Return value	NIL

1.6.3.17.2.68 TRichEdit:Print

Prints the control content or the selected text.

Type	Only after Create()
Parameters	[<cName>]: Document name; Default: "Xailer RichEdit file" [<hDC>]: Device handle ("printer") [<IPageNums>]: Select pages; Default: .F. [<ISelection>]: Prints only the selection. Default: .F. [<ICollate>]: Copy collate. Default: .F. [<nFromPage>]: Initial page. Default: 1 [<nToPage>]: End page. Default: 1 [<nCopies>]: Number of copies; Default: 1 [<aMargins>]: Margins; Default: the established in PageDlg
Return	NIL

value	
--------------	--

NOTE: Instead to use this method, we recommend to use the PrintDlg method.

1.6.3.17.2.69 TRichEdit:PrintBox

Prints the control content in a rectangle in the printing device.

Type	Only after Create()
Parameters	<hDC>: Device handle ("printer") <aRect>: Rectangle printing area in format {nLeft, nTop, nRight, nBottom} . On return nBottom value is replaced by the last printed line position. <nFrom>: Initial page <nUnits>: measure unit: ruTWIPS, ruCENTIMETERS, ruINCHES, ruPIXELS
Return value	<nTo>: Last printer position. Return -1 if the complete text has been printed.

NOTE: Instead to use this method, we recommend to use the PrintDlg method.

1.6.3.17.2.70 TRichEdit:PrintDlg

Executes the PrintDlg operating system native dialog through the TPrintDlg class.

Type	Only after Create()
Parameters	[<cPrintName>]: Printer name. Optional
Return value	NIL

1.6.3.17.2.71 TRichEdit:PrintSetup

Executes the PrintSetup Operating system native dialog, through the TPrintDlg class.

Type	Only after Create()
Parameters	None
Return value	NIL

1.6.3.17.2.72 TRichEdit:Redo

Redoes the last undo action

Type	Only after Create()
Parameters	None
Return value	NIL

1.6.3.17.2.73 TRichEdit:Replace

Replaces the selected text by **cReplace**.

Type	Only after Create()
Parameters	<cReplace> : Text to replace the selected text
Return value	NIL

1.6.3.17.2.74 TRichEdit:ReplaceAll

Replaces all the **cText** coincidences with **cReplace**. If the property IShowMessages is set to true a result information will be shown.

Type	Only after Create()
Parameters	<cText> : Text to search <cReplace> : text to replace <IDown> : Start search direction. Default: the IFindDown value <IWord> : Search only whole words. Default: IFindWord value <ICase> : Case sensitive. Default: IFindCase value
Return value	NIL

1.6.3.17.2.75 TRichEdit:ReplaceDlg

Executes the Replace operating system native dialog, through the TReplaceDlg class. It uses the cFindText, cReplace, IFindDown, IFindWord and IFindCase properties.

Type	Only after Create()
Parameters	None
Return value	NIL

1.6.3.17.2.76 TRichEdit:RTFtoHTML

Saves current RTF text in HTML format.

Type	Only after Create()
Parameters	<cFile>: File name
Return value	<ISuccess>: .T. if the operation is successful

1.6.3.17.2.77 TRichEdit:RTFtoPDF

Saves current RTF text in PDF format.

You will need the commercial library Image2PDF.dll to do the job. (Visit [Utilitywarrior](#) for further information)

Type	Only after Create()
Parameters	<cFile>: File name
Return value	<ISuccess>: .T. if the operation is successful

1.6.3.17.2.78 TRichEdit:RTFtoWord

Saves current RTF text in Microsoft Word format.

Type	Only after Create()
Parameters	<cFile>: File name
Return value	<ISuccess>: .T. if the operation is successful

1.6.3.17.2.79 TRichEdit:RTFToWordX

Saves current RTF text in Microsoft Word extended format.

Type	Only after Create()
Parameters	<cFile>: File name
Return value	<ISuccess>: .T. if the operation is successful

1.6.3.17.2.80 TRichEdit:RTFToWrite

Saves current RTF text in Microsoft Write format.

Type	Only after Create()
Parameters	<cFile>: File name
Return value	<ISuccess>: .T. if the operation is successful

1.6.3.17.2.81 TRichEdit:SaveFile

Saves the control's content in a file.

Type	Only after Create()
Parameters	None
Return value	NIL

1.6.3.17.2.82 TRichEdit:SelectAll

Selects all the text.

Type	Only after Create()
Parameters	None
Return value	NIL

1.6.3.17.2.83 TRichEdit:SelectNone

De selects any text in the control.

Type	Only after Create()
Parameters	None
Return value	NIL

1.6.3.17.2.84 TRichEdit:SetBullet

Activates or deactivates the bullets on the actual or selected paragraph.

Type	Only after Create()
Parameters	<IBullet> Activate / Deactivate
Return value	<IOk> True if success

1.6.3.17.2.85 TRichEdit:SetEventMask

Sets the control event mask.

Type	Only after Create()
Parameters	<nMask> Activate / Deactivate See API message EM_SETEVENTMASK for further information
Return value	NIL

1.6.3.17.2.86 TRichEdit:SetHideSelection

Shows or hides actual control selection.

Type	Only after Create()
Parameters	<IOnOff> True or false
Return	NIL

value	
--------------	--

1.6.3.17.2.87 TRichEdit:SetHyperLinkTooltips

Enables tooltips on hyperlinks. Only supported with Riched20.dll version 5.0. See property IRE50.

Type	Only after Create()
Parameters	None
Return value	NIL

1.6.3.17.2.88 TRichEdit:SetIndent

Sets left, right and first line indentation on actual or selected paragraph.

Type	Only after Create()
Parameters	<nLeft> left indent <nRight> Right indent <nFirst> First line indent <nUnits> Measure units
Return value	<IOk> True if success

1.6.3.17.2.89 TRichEdit:SetModify

Sets or deletes the control updated indicator.

Type	Only after Create()
Parameters	<IOnOff> True or false
Return value	NIL

1.6.3.17.2.90 TRichEdit:SetNumbering

Activates or deactivates numbering on current or selected paragraph. It only works with RichEd20.dll version 3.0 or above.

Type	Only after Create()
Parameters	<nNumbering> Numbering type. See nNumbering <nNumStyle> Numbering style. See nNumStyle <lOnOff> True or false
Return value	<lOk> True if success

1.6.3.17.2.91 TRichEdit:SetOptions

Sets specific options to the control.

Type	Only after Create()
Parameters	<nFlags> New actions <nOptions> New options
Return value	<nOptions> New options See API message EM_SETOPTIONS for further information

1.6.3.17.2.92 TRichEdit:SetParagraphSpacing

Sets previous and next spacing on actual or selected paragraph.

Type	Only after Create()
Parameters	<nBefore> previous spacing <nAfter> next spacing <nUnits> Measure units
Return value	<lOk> True if success

1.6.3.17.2.93 TRichEdit:SetPos

Sets cursor position.

Type	Only after Create()
Parameters	<nPos> New position
Return value	NIL

1.6.3.17.2.94 TRichEdit:SetText

Asigns text control text.

Type	Only after Create()
Parameters	<cText> Textvalue <nFormat> rfRTF, rfTEXT <lInsert> True to insert text
Return value	NIL

1.6.3.17.2.95 TRichEdit:SetSel

Sets the selected text.

Type	Only after Create()
Parameters	<nStart> Start position <nEnd> End position
Return value	NIL

1.6.3.17.2.96 TRichEdit:SetTextMode

Sets the type of text for the control. It can be plain text (txt), or riched text (rtf). You can also indicate the level of undo operations.

Type	Only after Create()
Parameters	<nFlag>

	Type and level undo See the API message EM_SETTEXTMODE for further information
Return value	<IOk> True if success

1.6.3.17.2.97 TRichEdit:SetTypographyOptions

Sets the typographycs options. It only works with RichEd20.dll version 3.0 or above.

Type	Only after Create()
Parameters	<nFlag> Type and level undo See the API message EM_SETTYPOGRAPHYOPTIONS for further information
Return value	<IOk> True if success

1.6.3.17.2.98 TRichEdit:SetZoom

Sets the zoom factor. Values between 1/64 and 64. It only works with RichEd20.dll version 3.0 or above.

Type	Only after Create()
Parameters	<nNumerator> Value between 1 and 63 <nDenominator> Value between 1 and 63
Return value	<IOk> True if success
Sample	3/2 (150%), 5/4 (125%), 1/1 (100%), 3/4 (75%), 1/2 (50%), ...

1.6.3.17.2.99 TRichEdit:StopGroupTyping

Prevents the control to accumulate keyboard actions on its undo queue.

Type	Only after Create()
Parameters	NIL
Return value	<IOk> True if success

1.6.3.17.2.100 TRichEdit:Undo

Undoes the last action made.

Type	Only after Create()
Parameters	None
Return value	NIL

1.6.3.17.2.101 TRichEdit:WordToRtf

Converts the Word file to RTF format an inserts it in the control.

Type	Only after Create()
Parameters	<cFile> : File name in Word format <IInsert> : If true the text file is inserted.
Return value	<ISuccess> : .T. if the operation is successful

1.6.3.17.2.102 TRichEdit:WordXToRtf

Converts the Word DOCX file to RTF format an inserts it in the control.

Type	Only after Create()
Parameters	<cFile> : File name in Word format <IInsert> : If true the text file is inserted.
Return value	<ISuccess> : .T. if the operation is successful

1.6.3.17.2.103 TRichEdit:WriteToRtf

Converts the Write file to RTF format an inserts it in the control.

Type	Only after Create()
Parameters	<cFile> : file name in Write format <IInsert> :

	If true the text file is inserted.
Return value	<ISuccess>: .T. if the operation is successful

1.6.3.17.3 TRichEdit:Events

Name	
OnChange	
OnContextMenu	
OnHScroll	
OnInsertPicture	
OnPrinting	
OnRequestResize	

1.6.3.17.3.1 TRichEdit:OnChange

Event that is produced when there is an action that can change the control content, or when the selected text has changed.

Parameters	<oSender>: Reference to the object that triggers the event
Return value:	NIL

1.6.3.17.3.2 TRichEdit:OnContextMenu

Request a context menu. If this event is not trapped the default contextual menu of the control will be shown.

Parameters	<oSender>: Object that triggers the event <nPosX>: Mouse pointer X coordinate <nPosY>: Mouse pointer Y coordinate
Return value:	NIL

Description:

This event is produced when the user right-clicks the mouse over the control requesting the on context menu. Logically this event is used to show the Pop up menu and the coordinates reported are to situate the menu just in that position.

1.6.3.17.3.3 TRichEdit:OnHScroll

Event that is produced when the user clicks with the mouse pointer the control's horizontal scroll bar.

Parameters	<oSender>
:	Reference to the object that triggers the event
Return value:	NIL

1.6.3.17.3.4 TRichEdit:OnInsertPicture

Event that is produced when trying to insert a image to the control. If this method is not overloaded the classical File selection dialog is shown..

Parameters	<oSender>
:	Reference to the object that triggers the event
Return value:	<clmage> Filename to insert

1.6.3.17.3.5 TRichEdit:OnPrinting

Event that is produced in the printing process to indicate the printing percentage finished.

Parameters	<oSender>
:	Reference to the object that triggers the event <nPercentage> Printing percentage
Return value:	NIL

1.6.3.17.3.6 TRichEdit:OnRequestResize

Event that is produced when there are changes in the control dimensions.

Parameters	<oSender>
:	Reference to the object that triggers the event
Return value:	NIL

1.6.3.18 TStatusBar

This class represents a Windows Status Bar control. The panels are represented in the TStatusBarPanel class.



Hierarchy Inherits from TWinControl
File name \source\StatusBar.prg

1.6.3.18.1 TStatusBar:Properties

■ read Only ■ Assignable ■ Design assignable ■ Run-time assignable

Scope	Name	Type	Initial value
■	altems	Array	{}
■	cText	Character	""
■	IFullSize	Logic	.T.
■	ISimple	Logic	.T.
■	ISizeGrip	Logic	sgAUTO
■	nAlign	Numeric	alBOTTOM
■	nBorderStyle	Numeric	sbDEFAULT
■	nClrPane	Numeric	clBtnFace
■	nHeight	Numeric	22
■	nMinHeight	Numeric	0
■	olcon	Object	NIL
■	olmageList	Object	TImageList

1.6.3.18.1.1 TStatusBar:altems

Lists the control panels represented by TStatusBarPanel objects.

Scope	Design assignable
Type	Array
Initial value	{}

1.6.3.18.1.2 TStatusBar:cText

Panel text when it is in simple mode. See also the ISimple property.

Scope	Assignable
Type	Character
Initial value	""

1.6.3.18.1.3 TStatusBar:IFullSize

Indicates if the last panel must be enlarged to use all the remaining space.

Scope	Assignable
Type	Logic
Initial value	.T.

1.6.3.18.1.4 TStatusBar:ISimple

Shows the bar as it has only one panel and it shows the defined text in the cText property.

Scope	Assignable
Type	Logic
Initial value	.F.

1.6.3.18.1.5 TStatusBar:ISizeGrip

Indicates if the re dimension effect must be drawn in the control's right corner.

Scope	Design assignable
Type	Logic
Initial value	sgAUTO
Possible values	sgAUTO, sgYES, sgNO

1.6.3.18.1.6 TStatusBar:nAlign

Indicates the control alignment in its oParent container object.

Scope:	Assignable
Type:	Numeric
Initial value:	aIBOTTOM
Possible values:	aINONE, aILEFT, aITOP, aIRIGHT, aIBOTTOM, aIClient

1.6.3.18.1.7 TStatusBar:nBorderStyle

Indicates the border style to be painted.

Scope	Assignable
Type	Numeric
Initial value	sbDEFAULT
Possible values	sbDEFAULT, sbNOBORDER, sbRAISED

1.6.3.18.1.8 TStatusBar:nClrPane

Indicates the control's background color.

Scope:	Assignable
Type:	Numeric
Initial value:	clBtnFace

(See also the appendix to check colors available)

1.6.3.18.1.9 TStatusBar:nHeight

Indicates the control height.

Scope	Assignable
Type	Numeric
Initial value	22

1.6.3.18.1.10 TStatusBar:nMinHeight

Indicates the minimum control height.

Scope	Assignable
Type	Numeric
Initial value	0

1.6.3.18.1.11 TStatusBar:olcon

TIcon object to show in the bar.

Scope	Assignable
Type	Object
Initial value	NIL

1.6.3.18.1.12 TStatusBar:olmagedList

TImageList with all the images to be used by the control.

Scope	Assignable
Type	Object
Initial value	TImageList()

The TImageList object is instantiated since the control is create with its New() constructor. The first image included in the TImageList establishes the dimension for the next images to be added to the TImageList. If the first image includes more than one bitmap is important then to establishes the nHeight and nWidth TImageList properties before to add a bitmap.

1.6.3.18.2 TStatusBar:Methods

■ Constructor ■ Standard ■ Only after Create()

Typ e	Name
■	AddBitmap
■	AddIcon
■	AddItem
■	DeleteItem
■	DeleteItems

1.6.3.18.2.1 TStatusBar:AddBitmap

Adds a new image to the olmagedList.

Type	Standard
Parameters	<xImage>: Resource name, filename or bitmap handle
Return value	<nImage>: Position number for the new image in the olmagedList

1.6.3.18.2.2 TStatusBar:AddIcon

Adds a new icon to the olmageList.

Type	Standard
Parameters	<clcon> : Resource name or filename.
Return value	<nImage> : Position number for the new image in the olmageList

1.6.3.18.2.3 TStatusBar:AddItem

Adds a new TStatusPanel panel to the control.

Type	Standard
Parameters	[<cText>] : Text to be shown in the panel [<nWidth>] : Panel width [<cTooltip>] : Panel tooltip [<nImage>] : Image number to be used from its olmageList property [<nAlignment>] : Text alignment. By default alLEFT [<nBorderStyle>] : panel border [<nMinWidth>] : Minimum panel width [<IOwnerDraw>] : .T. if the panel will be painted by itself [<nType>] : Type of information to show on the panel. By default sbTEXT [<nResizeStyle>] If the panel dimension should be automatically adjusted All the parameters default values are established by the TStatusPanel class in its Create constructor method
Return value	TStatusPanel object

1.6.3.18.2.4 TStatusBar:DeleteItem

Deletes an specific panel from the control.

Type	Standard
Parameters	<nItem>: Panel number to delete
Return value	NIL

1.6.3.18.2.5 TStatusBar:DeleteItems

Deletes all the panels from the control.

Type	Standard
Parameters	None
Return value	NIL

1.6.3.18.3 TStatusBar:Events

Name	
OnClick	
OnDrawItem	
OnRClick	
OnSimpleMode	

1.6.3.18.3.1 TStatusBar:OnClick

Event that is produced when the user left-clicks the mouse.

Parameters	<oSender>: Reference to the object that triggers the event
:	<nPosX>: Cursor X coordinate
	<nPosY>: Cursor Y coordinate
Return value:	NIL

1.6.3.18.3.2 TStatusBar:OnDrawItem

Event that is produced when a IOwnerDraw panel type is going to be repainted.

Parameters	<oSender>: Reference to the object that triggers the event
:	<hDC>: Device context handle
	<nItem>: Panel number that needs to be re-painted
	<aRect>: Array with the rectangle when it must be printed
Return value:	NIL

1.6.3.18.3.3 TStatusBar:OnRClick

Event that is produced when the user right-clicks the mouse.

Parameters	<oSender>: Reference to the object that triggers the event
:	<nPosX>: Cursor X coordinate
	<nPosY>: Cursor Y coordinate
Return value:	NIL

1.6.3.18.3.4 TStatusBar:OnSimpleMode

Event that is produced when the control passes to simple mode, it means, its ISimple property goes to .T..

Parameters	<oSender>: Reference to the object that triggers the event
Return value:	NIL

1.6.3.19 TStatusPanel

This class represents the TStatusBar control panel.

Description:

Every panel created in the TStatusBar control is an object from the TStatusPanel class that is stored in its `altems` property.

Hierarchy Inherits from TComponent
File name \source\StatusPanel.prg

1.6.3.19.1 TStatusPanel:Properties

■ read Only ■ Assignable ■ Design assignable ■ Run-time assignable

Scope	Name	Type	Initial value
■	cText	Character	""
■	cToolTip	Character	""
■	IOwnerDraw	Logic	.F.
■	nAlignment	Numeric	taLEFT
■	nBorderStyle	Numeric	sbDEFAULT
■	nImage	Numeric	0
■	nMinWidth	Numeric	0
■	nResizeStyle	Numeric	sbNONE
■	nType	Numeric	sbTEXT
■	nWidth	Numeric	100
■	oControl	Object	NIL

1.6.3.19.1.1 TStatusPanel:cText

Text to be shown in the panel.

Scope	Assignable
Type	Character
Initial value	""

1.6.3.19.1.2 TStatusPanel:cToolTip

Tooltip to be shown when the text is not visible.

Scope	Assignable
Type	Character
Initial value	""

1.6.3.19.1.3 TStatusPanel:IOwnerDraw

Allows to configure the panel layout and paint it with a function defined by the user. To do that, the OnDrawItem event from the TStatusBar panel must be assigned.

Scope	Assignable
Type	Logic
Initial value	.F.

1.6.3.19.1.4 TStatusPanel:nAlignment

Panel text alignment.

Scope	Assignable
Type	Numeric
Initial value	taLEFT
Possible values	taLEFT, taCENTER and taRIGHT

1.6.3.19.1.5 TStatusPanel:nBorderStyle

Panel's border style.

Scope	Assignable
Type	Numeric
Initial value	sbDEFAULT
Possible values	sbDEFAULT, sbNOBORDER, sbRAISED

1.6.3.19.1.6 TStatusPanel:nImage

Image index number that is drawn in the panel. It can be displayed alone or with text.

Scope	Assignable
Type	Numeric
Initial value	0

1.6.3.19.1.7 TStatusPanel:nMinWidth

Minimum width specified in pixels [Not implemented].

Scope	Assignable
Type	Numeric
Initial value	0

1.6.3.19.1.8 TStatusPanel:nResizeStyle

This property permits the automatic adjustment of the panel when its form container dimensions change.

Scope	Assignable
Type	Numeric
Initial value	sbNONE
Possible values	sbNONE, sbAUTO

1.6.3.19.1.9 TStatusPanel:nType

Type of information to show.

Scope	Assignable
Type	Numeric
Initial value	sbTEXT
Possible values	sbTEXT, sbDATE, sbTIME, sbKEYCAPS, sbKEYINSERT, sbKEYNUMLOCK

1.6.3.19.1.10 TStatusPanel:nWidth

Panel width specified in pixels.

Scope	Assignable
Type	Numeric
Initial value	100

1.6.3.19.1.11 TStatusPanel:oControl

Any visual control to be shown in the panel.

This property allows to add to the panel any Xailer control, for example and image, a progress bar, etc.

Scope	Assignable
Type	Object
Initial value	NIL

1.6.3.19.2 TStatusPanel:Methods

■ Constructor ■ Standard

Typ e	Name
■	Create
■	End Destroy
■	New
■	Update

1.6.3.19.2.1 TStatusPanel:Create

Creates and configures a StatusPanel object.

Type	Constructor
Parameters	<oParent> TStatusBar object [<cText>] Tex to be shown in the panel [<nWidth>] Width in pixels. Default value: nWidth property value. [<cTooltip>] Tooltip that will be shown when the text is not visible. Default value: "" [<nImage>] Index image that will be drawn in the panel. It can be displayed alone or with text. Default value: the nImage property value [<nAlignment>] Text panel alignment. The possible values for this property are taLEFT, taCENTER y taRIGHT. Default value: taLEFT. [<nBorderStyle>] Panel's border style. Default value: nBordeStyle property value. [<nMinWidth>]

	<p>Minimum width specified in pixels. [Not implemented] Default value: nMinWidth property value.</p> <p>[<IOwnerDraw>] Allows to configure the panel layout. Default value: IOwnerDraw property value.</p> <p>[<nType>] Type of information to show. Default value: nType property value.</p> <p>[<nResizeStyle>] Automatic adjustment. By default sbNONE. nResizeStyle property value.</p> <p>All the parameters are optional but oParent.</p>
Return value	Self reference (Self)

1.6.3.19.2.2 TStatusPanel:End

Destroys the resources used by the panel and updates the TStatusBar object that contains the panel.

Type	Standard
Parameters	None
Return value	NIL

1.6.3.19.2.3 TStatusPanel:New

Builds a StatusPanel object.

Type	Constructor
Parameters	<oParent> TStatusBar object
Return value	Self reference (Self)

1.6.3.19.2.4 TStatusPanel:Update

Informs to the TStatusBar parent panel class, that there are changes in the panel.

Type	Standard
Parameters	None
Return	NIL

value	
--------------	--

1.6.3.19.3 TStatusPanel:Events

Name

OnClick

OnRClick

1.6.3.19.3.1 TStatusPanel:OnClick

Event that is triggered when the user clicks the mouse pointer on the panel. In case that there is not action assigned, it will be triggered the OnClick event from the StatusBar, if it is defined.

Parameters	<oSender> Reference to the object that triggers the event <x, y> Mouse pointer coordinates from the panel.
Return value	NIL

1.6.3.19.3.2 TStatusPanel:OnRClick

Event that is triggered when the user right-clicks the mouse pointer on the panel. In case that there is not action assigned, it will be triggered the OnClick event from the StatusBar, if it is defined.

Parameters	<oSender> Reference to the object that triggers the event <x, y> Mouse pointer coordinates from the panel.
Return value	NIL

1.6.3.20 TSysAnimate

This class represents a Windows animation control.



Description:

The TSysAnimate class represents a Windows animation control.

Hierarchy	Inherits from TControl
File name	\source\SysAnimate.prg

1.6.3.20.1 TSysAnimate:Properties

read Only
 Assignable
 Design assignable
 Run-time assignable

Scope	Name	Type	Initial value
<input type="checkbox"/>	cName	Character	""
<input type="checkbox"/>	IActive	Logic	.F.
<input type="checkbox"/>	IAutoPlay	Logic	.T.
<input type="checkbox"/>	ICentered	Logic	.F.
<input type="checkbox"/>	ITransparent	Logic	.T.
<input type="checkbox"/>	IUseTimer	Logic	.F.
<input type="checkbox"/>	nCommonAvi	Numeric	aviNONE
<input type="checkbox"/>	nFirstFrame	Numeric	1
<input type="checkbox"/>	nLastFrame	Numeric	-1
<input type="checkbox"/>	nRepeat	Numeric	-1

1.6.3.20.1.1 TSysAnimate:nCommonAvi

Indicates the standard Windows animation identifier.

Scope	Design assignable
Type	Character
Initial value	aviNONE
Possible values	aviNONE, aviFIND, aviFINDFILES, aviFINDCOMPUTER, aviCOPYFILES, aviCOPYFILE, aviRECYCLEFILE, aviDELETEFILE, aviXPINTERNETSEARCH, aviXPDOWNLOADFILE

NOTE: The animations with the aviXP prefix are Windows XP specific.

1.6.3.20.1.2 TSysAnimate:cName

Animation to be shown. It can be a filename in disk or a resource.

Scope	Design assignable
Type	Character
Initial value	""

1.6.3.20.1.3 TSysAnimate:IActive

Indicates if the animation is being shown. Changing this value, starts or ends the animation.

Scope	Assignable
Type	Logic
Initial value	.F.

1.6.3.20.1.4 TSysAnimate:IAutoPlay

Automatically starts to play an animation after it is loaded.

Scope	Design assignable
Type	Logic
Initial value	.T.

1.6.3.20.1.5 TSysAnimate:ICentered

Centers the animation in the parent form.

Scope	Design assignable
Type	Logic
Initial value	.F.

1.6.3.20.1.6 TSysAnimate:ITransparent

Shows the same parent form background color.

Scope	Design assignable
Type	Logic
Initial value	.T.

1.6.3.20.1.7 TSysAnimate:!UseTimer

Uses a timer to show the animation in the same thread than the application.

Scope	Design assignable
Type	Logic
Initial value	.F.

1.6.3.20.1.8 TSysAnimate:nFirstFrame

Frame where the animation starts.

Scope	Design assignable
Type	Numeric
Initial value	1

1.6.3.20.1.9 TSysAnimate:nLastFrame

Last animation frame. A -1 value indicates that it will show all the frames.

Scope	Design assignable
Type	Numeric
Initial value	-1

1.6.3.20.1.10 TSysAnimate:nRepeat

Indicates the number of times that the application will be repeated. A -1 value indicates that it must be shown unlimitedly.

Scope	Design assignable
Type	Numeric

Initial value -1

1.6.3.20.2 TSysAnimate:Methods

■ Constructor ■ Standard

Typ	Name
■	Close
■	Create
■	Destroy
■	Load
■	LoadCommon
■	LoadFromFile
■	LoadFromResource
■	Play
■	Seek
■	Stop

1.6.3.20.2.1 TSysAnimate:Close

Closes the current animation file.

Type	Standard
Parameters	None
Return value	<ISuccess> .T. if was closed successfully

1.6.3.20.2.2 TSysAnimate:Create

Class constructor.

Type	Constructor
Parameters	<cName> Animation to be shown. It can be a file name in disk or a resource
Return value	Self reference (Self)

1.6.3.20.2.3 TSysAnimate:Destroy

Destroys the object and releases the resources used.

Type	Standard
Parameters	None
Return value	<ISuccess> .T. if the object was destroyed

1.6.3.20.2.4 TSysAnimate:Load

Loads an animation in the object from disk or resources.

Type	Standard
Parameters	<cName> Animation to be shown. It is equivalent to assign the cName property
Return value	<ISuccess> .T. if was open successfully. Otherwise, .F.

1.6.3.20.2.5 TSysAnimate:LoadCommon

Loads an standard animation indicated in the object.

Type	Standard
Parameters	<nAvi> System animation identifier. It is equivalent to assign the nCommonAvi property
Return value	<ISuccess> .T. if was open successfully. Otherwise, .F.

1.6.3.20.2.6 TSysAnimate:LoadFromFile

Loads an animation in the object from a file in disk.

Type	Standard
Parameters	<cName> Animation to be shown. It is equivalent to assign the cName property
Return value	<ISuccess> .T. if was open successfully. Otherwise, .F.

1.6.3.20.2.7 TSysAnimate:LoadFromResource

Loads and animation in the object from a resource file.

Type	Standard
Parameters	<cName> Animation to be shown. It is equivalent to assign the cName property
Return value	<ISuccess> .T. if was open successfully. Otherwise, .F.

1.6.3.20.2.8 TSysAnimate:Play

Shows the animation in the optional range specified by the nFirstFrame and nLastFrame properties and it repeats the number of times indicated by the nRepeat property.

Type	Standard
Parameters	None
Return value	<ISuccess> .T. if was open successfully. Otherwise, .F.

1.6.3.20.2.9 TSysAnimate:Seek

Moves the animation to the indicated frame or in the frame indicated by the nFirstFrame property.

Type	Standard
Parameters	<nFrame> Animation frame
Return value	<ISuccess> .T. if was open successfully. Otherwise, .F.

1.6.3.20.2.10 TSysAnimate:Stop

Stops the current animation.

Type	Standard
Parameters	None
Return value	<ISuccess> .T. if stops successfully. Otherwise, .F.

1.6.3.20.3 TSysAnimate:Events

■ Constructor ■ Standard

Name
OnClose
OnOpen
OnStart
OnStop

1.6.3.20.3.1 TSysAnimate:OnClose

Event that is triggered every time that an animation is closed.

Parameters	<oSender> Object that triggers the event
Return value	NIL

1.6.3.20.3.2 TSysAnimate:OnOpen

Event that is triggered every time that an animation is loaded.

Parameters	<oSender> Object that triggers the event
Return value	NIL

1.6.3.20.3.3 TSysAnimate:OnStart

Event that is triggered every time that starts an animation.

Parameters	<oSender> Object that triggers the event
Return value	NIL

1.6.3.20.3.4 TSysAnimate:OnStop

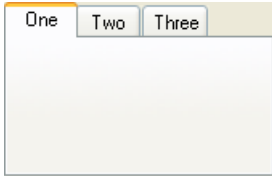
Event that is triggered every time that the animation is stopped.

Parameters	<oSender> Object that triggers the event
-------------------	--

Return value	NIL
---------------------	-----

1.6.3.21 TTabCtrl

This class represents a Windows Tab control type (tabs). Every control tab represents an object from the TTabItem class.



Any control that is inserted inside the TTabCtrl it will be shown regardless the active tab. If you want that every single tab has its own area and every tab has their own control, you should use a TFolder control type.

Hierarchy	Inherits from TWinControl
See also	TFolder
File name	\source\TabCtrl.prg

1.6.3.21.1 TTabCtrl:Properties

■ read Only ■ Assignable ■ Design assignable ■ Run-time assignable

Scope	Name	Type	Initial value
■	altems	Array	{}
■	IButtonStyle	Logic	.F.
■	IFixedWidth	Logic	.F.
■	IFlatButtons	Logic	.F.
■	IForceIconLeft	Logic	.F.
■	IForceLabelLeft	Logic	.F.
■	IHotTrack	Logic	.F.
■	IMultiLine	Logic	.F.
■	IRightJustify	Logic	.F.
■	IScrollOpposite	Logic	.F.
■	ITabCloseButtons	Logic	.F.
■	ITabStop	Logic	.F.
■	ITransparent	Logic	.F.
■	ITransparentBody	Logic	.F.
■	nHeight	Numeric	160
■	nIndex	Numeric	0
■	nMinTabWidth	Numeric	0

■	nOrientation	Numeric	orTOP
■	nWidth	Numeric	200
■	oImageList	Object	TImageList

1.6.3.21.1.1 TTabCtrl:altItems

List of tabs represented by TTabItem objects contained in the control.

Scope	Design assignable
Type	Array
Initial value	{}

1.6.3.21.1.2 TTabCtrl:IButtonStyle

Shows the tab as buttons and the control border is not painted.

Scope	Design assignable
Type	Logic
Initial value	.F.

1.6.3.21.1.3 TTabCtrl:IFixedWidth

If it is .T., the tabs will have a fixed width. This style is not compatible with the IRightJustify property.

Scope	Design assignable
Type	Logic
Initial value	.F.

1.6.3.21.1.4 TTabCtrl:IFlatButtons

The selected tab will be shown as pressed and the others will be shown as flat. It can be only used together with the IButtonStyle property.

Scope	Design assignable
Type	Logic
Initial value	.F.

1.6.3.21.1.5 TTabCtrl:IForcelconLeft

The icons are aligned to the left in every tab with fixed size. This style only can be used with the IFixedWidth property.

Scope	Design assignable
Type	Logic
Initial value	.F.

1.6.3.21.1.6 TTabCtrl:IForceLabelLeft

The labels are aligned to the left in every tab with fixed size. The label is shown immediately after the icon, instead to be centered. This style only can be used with the property IFixedWidth and implies to use the IForcelconLeft style.

Scope	Design assignable
Type	Logic
Initial value	.F.

1.6.3.21.1.7 TTabCtrl:IHotTrack

The labels under the mouse pointer are highlighted automatically.

Scope	Design assignable
Type	Logic
Initial value	.F.

1.6.3.21.1.8 TTabCtrl:IMultiLine

Allows to create multiple row labels.

Scope	Design assignable
Type	Logic
Initial value	.F.

1.6.3.21.1.9 TTabCtrl:IRightJustify

The tab width can be increased if it is needed and allows to show every tab row using the total control width. This style must be used together with the IMultiLine property.

Scope	Design assignable
Type	Logic
Initial value	.F.

1.6.3.21.1.10 TTabCtrl:IScrollOpposite

The tab that are not needed are moved to the opposite control side when one tab is selected.

Scope	Design assignable
Type	Logic
Initial value	.F.

1.6.3.21.1.11 TTabCtrl:ITabCloseButtons

If true, the tabs will show a close button on its upper right corner.

Scope	Design assignable
Type	Logic
Initial value	.F.

See also the OnCloseTab event.

1.6.3.21.1.12 TTabCtrl:ITabStop

The control receives the focus when the user presses the TAB key.

Scope	Design assignable
Type	Logic
Initial value	.F.

1.6.3.21.1.13 TTabCtrl:ITransparent

The control background is transparent.

Scope	Design assignable
Type	Logic
Initial value	.F.

1.6.3.21.1.14 TTabCtrl:ITransparentBody

The tab zone is transparent. It must be used together with the ITransparent property.

Scope	Design assignable
Type	Logic
Initial value	.F.

1.6.3.21.1.15 TTabCtrl:nHeight

Indicates the control height.

Scope	Assignable
Type	Numeric
Initial value	160

1.6.3.21.1.16 TTabCtrl:nIndex

Indicates the current tab selected.

Scope	Assignable
Type	Numeric
Initial value	0

1.6.3.21.1.17 TTabCtrl:nMinTabWidth

Indicates the minimum control tab width.

Scope	Assignable
Type	Numeric
Initial value	160

1.6.3.21.1.18 TTabCtrl:nOrientation

Indicates the control orientation.

Scope:	Design assignable
Type:	Numeric
Initial value:	orTOP

Possible values:	orTOP, orBOTTOM, orLEFT, orRIGHT
-------------------------	----------------------------------

1.6.3.21.1.19 TTabCtrl:nWidth

Indicates the control width.

Scope	Assignable
Type	Numeric
Initial value	200

1.6.3.21.1.20 TTabCtrl:olmImageList

TImageList object with all the images to be used by the control.

Scope	Assignable
Type	Object
Initial value	TImageList()

The TImageList object is instantiated since the control is create with its New() constructor. The first image included in the TImageList establishes the dimension for the next images to be added to the TImageList. If the first image includes more than one bitmap is important then to establishes the nHeight and nWidth TImageList properties before to add a bitmap.

1.6.3.21.2 TTabCtrl:Methods

■ Constructor ■ Standard ■ Only after Create()

Typ e	Name
■	AddImage
■	AddItem
■	Deleteltem
■	Deleteltems
■	InsertItem
■	SetIndex

1.6.3.21.2.1 TTabCtrl:AddImage

Adds a new image to the olmageList.

Type	Standard
-------------	----------

Parameters	<xImage> : Resource name, file name or bitmap handle
Return value	<nImage> : New image position number in the olmageList

1.6.3.21.2.2 TTabCtrl:AddItem

Adds a new TTabItem tab to the control.

Type	Standard
Parameters	[<cText>] : Text to be show in the tab [<nImage>] : Image number to be used from its olmageList property [<cTooltip>] : Tab tooltip [<nClrText>] : Text color [<IEnabled>] : .T. if the tab must be enabled The default values for all the parameters are established by the TTabItem class in its Create constructor method
Return value	TTabItem object

1.6.3.21.2.3 TTabCtrl:DeleteItem

Deletes an specific tab control.

Type	Standard
Parameters	<nItem> : Tab number to delete
Return value	NIL

1.6.3.21.2.4 TTabCtrl:DeleteItems

Deletes all the control tabs.

Type	Standard
-------------	----------

Parameters	None
Return value	NIL

1.6.3.21.2.5 TTabCtrl:InsertItem

Inserts a new TTabItem to the control.

Type	Standard
Parameters	<p><nPos>: New tab position</p> <p>[<cText>]: Text to be displayed in the tab</p> <p>[<nImage>]: Image number to be used from its olmageList property</p> <p>[<cTooltip>]: Tab tooltip</p> <p>[<nClrText>]: Text color</p> <p>[<IEnabled>]: .T. if the tab must be enabled</p> <p>The default values for all the parameters are established by the TTabItem class in its Create constructor method</p>
Return value	TTabItem object

1.6.3.21.2.6 TTabCtrl:SetIndex

Establishes the active tab. It is equivalent to the nIndex property, but it includes a parameter to avoid to trigger the OnChange event.

Type	Standard
Parameters	<p><nPos>: Tab number to activate</p> <p>[<ITriggerOnChange>]: if it is .T. the OnChange event is triggered. Default: .F.</p>
Return value	<nPos>

1.6.3.21.3 TTabCtrl:Events

Name
OnChange
OnChanging
OnCloseTab
OnRClick

1.6.3.21.3.1 TTabCtrl:OnChange

Event that is produced when the current tab changes. See also the OnChanging event.

Parameters	<oSender> : Reference to the object that triggers the event <nNewTab> : New active tab <nOldTab> : Old active tab
Return value:	NIL

1.6.3.21.3.2 TTabCtrl:OnChanging

Event that is produced when there is an attempt to change the current tab.

Parameters	<oSender> : Reference to the object that triggers the event <nNewTab> : New active tab <nOldTab> : Old active tab
Return value:	<IChange> : If it returns a NIL value or a .T. logic value, it allowed to change the tab

1.6.3.21.3.3 TTabCtrl:OnCloseTab

Event that is produced when a tab is closed pushing its close button.

Parameters	<oSender> : Reference to the object that triggers the event <nTab> : Tab to close
Return	NIL

value:

Sample:

```
METHOD FolderCloseTab( oSender, nIndex ) CLASS TFrmMain
  oSender:aItems[ nIndex ]:Delete()
RETURN Nil
```

1.6.3.21.3.4 TTabCtrl:OnRClick

Event that is produced when the user right clicks the mouse.

Parameters	<oSender>: Reference to the object that triggers the event <nPosX>: X cursor coordinate <nPosY>: Y cursor coordinate
Return value:	NIL

1.6.3.22 TTabItem

Class to manipulate tabs.

Description:

This class represents every tab from the TTabControl or TFolder control type.

Hierarchy	Inherits from TComponent
File name	\source\TabItem.prg

1.6.3.22.1 TTabItem:Properties

read Only
 Assignable
 Design assignable
 Run-time assignable

Scope	Name	Type	Initial value
<input type="checkbox"/>	cText	Character	""
<input type="checkbox"/>	cToolTip	Character	""
<input type="checkbox"/>	IEnabled	Logic	.T.
<input type="checkbox"/>	nCrText	Numeric	NIL
<input type="checkbox"/>	nClrTextHot	Numeric	NIL
<input type="checkbox"/>	nImage	Numeric	0
<input type="checkbox"/>	nItem	Numeric	0

1.6.3.22.1.1 TTabItem:cText

Text to be shown in the tab. Default value "".

Scope	Assignable
Type	Character
Initial value	""

1.6.3.22.1.2 TTabItem:cToolTip

Tab tooltip to be shown.

Scope	Assignable
Type	Character
Initial value	""

1.6.3.22.1.3 TTabItem:IEnabled

.T. if the tab can be selected.

Scope	Assignable
Type	Logic
Initial value	.T.

1.6.3.22.1.4 TTabItem:nClrText

Tab text color.

Scope	Assignable
Type	Numeric
Initial value	<NIL> By default its parent nClrText property

Consult the appendix for the list of available colors

1.6.3.22.1.5 TTabItem:nClrTextHot

Tab text color when the mouse is over it.

Scope	Assignable
--------------	------------

Type	Numeric
Initial value	<NIL> By default its parent nClrTextHot property

Consult the appendix for the list of available colors

1.6.3.22.1.6 TTabItem:nImage

Image to be shown in the tab. It corresponds to image number in the oImageList object from its TTabControl container object.

Scope	Assignable
Type	Numeric
Initial value	0

1.6.3.22.1.7 TTabItem:nItem

Tab creation order in its TTabControl container object.

Scope	Assignable
Type	Numeric
Initial value	0

1.6.3.22.2 TTabItem:Methods

■ Constructor ■ Standard

Typ	Name
■	Create
■	Delete
■	Select
■	SetItem

1.6.3.22.2.1 TTabItem>Create

This methods allows to create new tabs in a TTabControl object.

Type	Constructor
Parameters	<oParent> Container object. [<nItem>] Creation tab order. Default: after the last tab. [<cText>]

	<p>Text to be shown in the tab</p> <p>[<xImage>] Image to be used in the tab. If the data type is character indicate an resource image and in that case this image will be added to the olmageList object from its TTabControl container.</p> <p>[<cTooltip>] Tooltip to show</p> <p>[<nClrText>] Text color</p> <p>[<IEnabled>] If .T. the tab is enabled.</p> <p>The default values for all the parameters are indicated by the properties with the same name</p>
Return value	Self reference (Self)

1.6.3.22.2.2 TTabItem:Delete

Deletes the current tab from its TTabControl objet container.

Type	Standard
Parameters	None
Return value	NIL

1.6.3.22.2.3 TTabItem>Select

Selects the current tab in the TTabControl container object.

Type	Standard
Parameters	None
Return value	NIL

1.6.3.22.2.4 TTabItem:SetItem

This methods allows to modify in one step several tab features.

Type	Standard
Parameters	<p><[cText]> Text to be shown in the tab</p> <p><[xImage]> Image to be used in the tab. If the data type is</p>

character indicate an resource image and in that case this image will be added to the olmageList object from its TTabControl container.

<[cTooltip]>

Tooltip to be shown

<[nClrText]>

Text color.

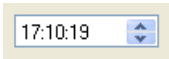
The default values for all the parameters are indicated by the properties with the same name

Return value
NIL

Consult the appendix for the list of available colors

1.6.3.23 TTimePicker

This class represents an standard windows time edit control.



Hierarchy Inherits from TStdControl
See also TDateEdit, TDatePicker
File name \source\TimePicker.prg

1.6.3.23.1 TTimePicker:Properties

■ read Only ■ Assignable ■ Design assignable ■ Run-time assignable

Scope	Name	Type	Initial value
■	cCustomFormat	Character	"".
■	cMaxTime	Character	""
■	cMinTime	Character	""
■	cTime	Character	""
■	nHeight	Numeric	20
■	nWidth	Numeric	90

1.6.3.23.1.1 TTimePicker:cCustomFormat

Time format defined by the user.

Scope:	Design assignable
Type:	Character
Initial value:	""

The time format is set based in the following templates:

"h"	Hour with one or two digits in 12 hours format.
"hh"	Hour with two digits in 12 hours format. Hours with only one digit are preceding by a zero.
"H"	Hour with one or two digits in 24 hours format.
"HH"	Hour with two digits in 24 hours format. Hours with only one digit are preceding by a zero.
"m"	Minutes with one or two digits.
"mm"	Minutes with two digits. Minutes with only one digit are preceding by a zero.
"t"	AM/PM abbreviation with one character. (AM is shown like "A").
"tt"	AM/PM abbreviation.

1.6.3.23.1.2 TTimePicker:cMaxTime

Indicates the maximum time allowed by the control.

Scope:	Assignable
Type:	Character
Initial value:	""

1.6.3.23.1.3 TTimePicker:cMinTime

Indicates the minimum time allowed by the control.

Scope:	Assignable
Type:	Character
Initial value:	""

1.6.3.23.1.4 TTimePicker:cTime

Indicates the control value.

Scope:	Assignable
Type:	Character
Initial value:	Blank time

1.6.3.23.1.5 TTimePicker:nHeight

Indicates the control height.

Scope:	Assignable
Type:	Numeric
Initial value:	20

1.6.3.23.1.6 TTimePicker:nWidth

Indicates the control width.

Scope:	Assignable
Type:	Numeric
Initial value:	90

1.6.3.23.2 TTimePicker:Events

Name	
OnChange	

1.6.3.23.2.1 TTimePicker:OnChange

Event that is produced when the control changes its value.

Parameters	<oSender>:
:	Reference to the object that triggers the event
	<cTime>:
	Newtime
Return value:	NIL

1.6.3.24 TToolBar

This class represents the 'ToolBar' Windows standard control. Normally any application contains a main window with a button bar like this.



Hierarchy Inherits from TControl

See also TToolButton
File name \source\ToolBar.prg

1.6.3.24.1 TToolBar:Properties

■ read Only ■ Assignable ■ Design assignable ■ Run-time assignable

Scope	Name	Type	Initial value
■	altems	Array	{}
■	lAlwaysTip	Logic	.F.
■	lAutoSize	Logic	.T.
■	lBallons	Logic	.F.
■	lFlat	Logic	.T.
■	lList	Logic	.F.
■	lTransparent	Logic	.T.
■	nAlign	Numeric	aTOP
■	nBorderStyle	Numeric	bvNONE
■	nBtnHeight	Numeric	0
■	nBtnWidth	Numeric	0
■	nHeight	Numeric	28
■	nIndent	Numeric	0
■	nWidth	Numeric	120
■	oImageList	Object	NIL
■	oImageListDis	Object	NIL
■	oImageListHot	Object	NIL

1.6.3.24.1.1 TToolBar:altems

Array with all the TToolButton type buttons included in the ToolBar.

Scope	read Only
Type	Array
Initial value	{}

1.6.3.24.1.2 TToolBar:lAlwaysTip

If it is .T., the button tooltips will be shown even when the container form from the TToolBar control does not have the focus.

Scope	Design assignable
Type	Logic
Initial value	.F.

1.6.3.24.1.3 TToolBar:IAutoSize

If it is .T., adjusts its dimensions every time that the bitmap or the button is resized, or the first time when the literals are defined.

Scope	Design assignable
Type	Logic
Initial value	.F.

1.6.3.24.1.4 TToolBar:IBalloons

If it is .T. the tooltips to be shown in the button will be like balloons.

Scope	Design assignable
Type	Logic
Initial value	.F.

1.6.3.24.1.5 TToolBar:IFlat

If it is .T. the ToolBar is create and its button will be all flat, transparent and with 'Hot tracking' support.

Scope	Design assignable
Type	Logic
Initial value	.T.

1.6.3.24.1.6 TToolBar: IList

If it is .T. the Toolbar is create and its buttons will be all flat, transparent and with 'Hot tracking' support.

Scope	Design assignable
Type	Logic
Initial value	.F.

1.6.3.24.1.7 TToolBar:ITransparent

If it is .T. the ToolBar will be created and will be transparent. The buttons will not be transparent.

Scope	Design assignable
Type	Logic
Initial value	.T.

1.6.3.24.1.8 TToolBar:nAlign

Control alignment in its oParent container object.

Scope:	Design assignable
Type:	Numeric
Initial value:	aITOP
Possible values	aINONE, aILEFT, aITOP, aIRIGHT, aIBOTTOM, aIClient

Description:

This property allows to adjust the control dimensions and positions in its oParent container object. The alignment can be:

- **None:** Default value
- **Left:** The control is aligned to the left of its oParent control and takes its height from its container client.
- **Up:** The control is aligned to the upper part of its oParent control and takes its width from its container client.
- **Right:** The control is aligned to the right of its oParent control and takes its height from its container client.
- **Bottom:** The control is aligned to the lower part of its oParent control and takes its width from its container client.
- **Client:** The control is aligned to all the area of its oParent control client adjusting its size to it.

1.6.3.24.1.9 TToolBar:nBorderStyle

Indicates the border style.

Scope	Design assignable
Type	Numeric
Initial value	bvNONE
Possible values	bvNONE, bvRAISED, bvSUNKEN, bvBUMP, bvETCHED, bvFLAT

1.6.3.24.1.10 TToolBar:nBtnHeight

Indicates the button height in the ToolBar. If this value is 0, the height to be used for the buttons will be defined by the first existing image in its `olmageList` property.

Scope	Design assignable
Type	Numeric
Initial value	0

1.6.3.24.1.11 TToolBar:nBtnWidth

Indicates the button width in the ToolBar. If this value is 0, the width to be used for the buttons will be defined by the first existing image in its `olmageList` property.

Scope	Design assignable
Type	Numeric
Initial value	0

1.6.3.24.1.12 TToolBar:nHeight

Indicates the control height.

Scope:	Design assignable
Type:	Numeric
Initial value:	28

1.6.3.24.1.13 TToolBar:nWidth

Indicates the control width.

Scope:	Design assignable
Type:	Numeric
Initial value:	120

1.6.3.24.1.14 TToolBar:nIndent

Indents the first button in the ToolBar. The value is specified in pixels.

Scope:	Design assignable
---------------	-------------------

Type:	Numeric
Initial value:	0

1.6.3.24.1.15 TToolBar:olmageList

TImageList object with all the images to be used by the control.

Scope	Design assignable
Type	Object
Initial value	TImageList()

The TImageList object is instantiated from the same control creation with its New() constructor. The first image included in the TImageList establishes the dimensions of the following images added. If the first image includes more than one bitmap, it is important to establish the nHeight and nWidth TImageList properties before to add any bitmap.

1.6.3.24.1.16 TToolBar:olmageListDis

TImageList object with all the images to be used in the control when the buttons are disabled. If those images are not defined, the control will automatically make the effect to disable the buttons.

Scope	Design assignable
Type	Object
Initial value	TImageList()

The TImageList object is instantiated from the same control creation with its New() constructor. The first image included in the TImageList establishes the dimensions of the following images added. If the first image includes more than one bitmap, it is important to establish the nHeight and nWidth TImageList properties before to add any bitmap.

1.6.3.24.1.17 TToolBar:olmageListHot

TImageList object with all the images to be used in the control when the buttons have the mouse pointer on them. If those images are not defined, the control will show always the same image defined in its olmageList property.

Scope	Design assignable
Type	Object
Initial value	TImageList()

The TImageList object is instantiated from the same control creation with its New() constructor. The first image included in the TImageList establishes the dimensions of the following images added. If the first image includes more than one bitmap, it is important to establish the nHeight and nWidth TImageList properties before to add any bitmap.

1.6.3.24.2 TToolBar:Methods

■ Constructor ■ Standard

Type	Name
■	AddButton
■	AddItem
■	AddSeparator
■	DeleteItem
■	InsertItem
■	InsertSeparator

1.6.3.24.2.1 TToolBar:AddButton

Adds a net button to the ToolBar. This is an obsolete method. We strongly recommend the AddItem method.

Type	Standard
Parameters	<p>[<cText>] Text to be shown in the button</p> <p>[<clmage>] Resource or file image to be used by the button</p> <p>[<clmage2>] Resource or file image to be used in the button when it is disabled</p> <p>[<clmage3>] Resource or file image to be used when the button has the mouse pointer on it.</p> <p>[<cTooltip>] Tooltip to be shown by the button</p> <p>[<OnClick>] Codeblock or number for the OnClick event</p> <p>[<oMenu>] TMenu object to include in the button like popup menu</p>
Return value	Object TToolBarButton

1.6.3.24.2.2 TToolBar:AddItem

Adds a new button to the ToolBar.

Type	Standard
Parameters	<p>[<cText>] Text to be show in the button</p>

	<p>[<nImage>] Image number to be used in the button in its olmageList, olmageListDis and shouldperties</p> <p>[<cTooltip>] Tooltip to be shown in the button.</p> <p>[<IAutoSize>] If it is .T. the button will indicate to the ToolBar that it should not assign to the button the standard width and height values, but it should calculate it based in the image and text dimensions. Default: .F.</p> <p>[<IGroup>] If it is .T. it will create a separator with the previous button. Default: .F.</p> <p>[<IChecked>] If it is .T. it will show the button checked. Default: .F.</p> <p>[<IEnabled>] If it is .T. it will show the button enables. Default: .T.</p> <p>[<IVisible>] If it is .T. it will show the button. Default: .T.</p> <p>[<oMenu>] TMenu object to include in the button as popup menu</p>
Return value	Object TToolButton

1.6.3.24.2.3 TToolBar:AddSeparator

Adds a separator in the ToolBar.

Type	Standard
Parameters	<p>[<nWidth>] Separator width. Default: 0</p> <p>[<IVisible>] True if visible. Default: True</p>
Return value	TToolButton object

1.6.3.24.2.4 TToolBar:DeleteItem

Deletes a button from the ToolBar.

Type	Standard
Parameters	<p>[<nItem>] Button number to delete</p>
Return	NIL

value	
--------------	--

1.6.3.24.2.5 TToolBar:InsertItem

Inserts a new button in the ToolBar.

Type	Standard
Parameters	<p><nPos> New button position</p> <p>[<cText>] Text to be show in the button</p> <p>[<nImage>] Image number to be used in the button in its olmageList, olmageListDis and shouldperties</p> <p>[<cTooltip>] Tooltip to be shown in the button.</p> <p>[<IAutoSize>] If it is .T. the button will indicate to the ToolBar that it should not assign to the button the standard width and height values, but it shoud calculate it based in the image and text dimensions. Default: .F.</p> <p>[<IGroup>] If it is .T. it will create a separator with the previous button. Default: .F.</p> <p>[<IChecked>] If it is .T. it will show the button checked. Default: .F.</p> <p>[<IEnabled>] If it is .T. it will show the button enables. Default: .T.</p> <p>[<IVisible>] If it is .T. it will show the button. Default: .T.</p> <p>[<oMenu>] TMenu object to include in the button as popup menu</p>
Return value	TToolBarButton object

1.6.3.24.2.6 TToolBar:InsertSeparator

Inserts a separator in the ToolBar.

Type	Standard
Parameters	<p><nPos> Position for the new separator</p> <p>[<nWidth>]</p>

	Separator width. Default: 0 [<IVisible>] True if visible. Default: True
Return value	TToolBar object

1.6.3.24.3 TToolBar:Events

Name	OnDropDown
-------------	------------

1.6.3.24.3.1 TToolBar:OnDropDown

Event that is produced when the context menu button is pushed.

Parameters	<oSender>: Reference to the object that triggers the event
:	<nItem>: Ordinal item that provoke the event
Return value:	<IResult>: Only a false return value aborts the menu display

1.6.3.25 TToolBarButton

Class to create and manage buttons from the TToolBar controls. The TToolBarButton class is used internally for the TToolBar class to create and manage buttons inside its bar.

Hierarchy	Inherits from TComponent
See also	TToolBar
File name	\source\ToolBarButton.prg

1.6.3.25.1 TToolBarButton:Properties

■ read Only ■ Assignable ■ Design assignable ■ Run-time assignable

Scope	Name	Type	Initial value
■	cText	Character	""
■	cTooltip	Character	""
■	IAutoSize	Logic	.F.
■	IChecked	Logic	.F.
■	IEnabled	Logic	.T.
■	IGroup	Logic	.F.
■	ISeparator	Logic	.F.
■	IVisible	Logic	.T.

■	nImage	Numeric	0
■	nIndex	Numeric	0
■	oMenu	Object	NIL

1.6.3.25.1.1 TToolButton:cText

Text to be shown in the button.

Scope	Assignable
Type	Character
Initial value	""

1.6.3.25.1.2 TToolButton:cTooltip

Tooltip to be shown in the button.

Scope	Assignable
Type	Character
Initial value	""

1.6.3.25.1.3 TToolButton:IAutoSize

If it is .T., the button indicates to the TToolBar that it should not assign to the button the standard width and height values, but it should calculate them based in the its image and text dimensions.

Scope	Design assignable
Type	Logic
Initial value	.F.

1.6.3.25.1.4 TToolButton:IChecked

If it is .T. it will show the button checked.

Scope	Assignable
Type	Logic
Initial value	.F.

1.6.3.25.1.5 TToolButton:LEnabled

If it is .F. it will show the button disabled.

Scope	Assignable
Type	Logic
Initial value	.T.

1.6.3.25.1.6 TToolButton:IGroup

If it is .T. the button will be separated from the previous button by some space.

Scope	Design assignable
Type	Logic
Initial value	.F.

1.6.3.25.1.7 TToolButton:ISeparator

If it is .T. the button will be only a separator line.

Scope	Design assignable
Type	Logic
Initial value	.F.

1.6.3.25.1.8 TToolButton:IVisible

If it is .F., the button will not be shown.

Scope	Assignable
Type	Logic
Initial value	.T.

1.6.3.25.1.9 TToolButton:nImage

Image to be used in the button. It corresponds to the olmageList object number from its TToolBar container object.

Scope	Assignable
Type	Numeric

Initial value 0

1.6.3.25.1.10 TToolButton:nIndex

Control creation number in its TToolBar container object.

Scope	read Only
Type	Numeric
Initial value	0

1.6.3.25.1.11 TToolButton:oMenu

TMenu object to include in the button like popup menu.

Scope	Design assignable
Type	Object
Initial value	NIL

1.6.3.25.2 TToolButton:Methods

■ Constructor ■ Standard ■ Only after Create() ■ Internal use

Typ	Name
■	Create
■	Delete
■	Click

1.6.3.25.2.1 TToolButton:Create

Class constructor.

Type	Constructor
Parameters	<oParent> Button container object <nPos> : New button position <ISeparator> If it is .T. the button will be the same type as the separator <cText> Button text <nImage>

	<p>Image to be used in the button. It corresponds to the <code>oImageList</code> object number in its shouldtainer</p> <p><cTooltip>: Tooltip to be shown in the button</p> <p>[<IAutoSize>] if it is <code>.T.</code> the button will indicate to the <code>TToolBar</code> that it should not assign to the button the standard height and width, but it should calculate them based on the image and text dimensions</p> <p>[<IGroup>] if it is <code>.T.</code> it will create a separation from the previous button</p> <p>[<IChecked>] If it is <code>.T.</code> it will show the button pressed</p> <p>[<IEnabled>] If it is <code>.T.</code> it will show the button enabled</p> <p>[<IVisible>] If it is <code>.T.</code> it will show the button</p> <p>[<oMenu>] <code>TMenu</code> object to be included in the button like popup menu</p> <p>Default values correspond to its equivalent properties</p>
Return value	<oToolButton> reference to the new created button

1.6.3.25.2.2 TToolButton:Delete

Deletes and destroys the button from its `TToolBar` container object.

Type	Only after <code>Create()</code>
Parameters	None
Return value	NIL

1.6.3.25.2.3 TToolButton:Click

Clicks the button.

Type	Only after <code>Create()</code>
Parameters	None
Return value	Value returned by the <code>OnClick</code> event

1.6.3.25.3 TToolButton:Events

Name

OnClick

1.6.3.25.3.1 TToolButton:OnClick

Event that is produced when the user clicks the button.

Parameters	<oSender>:
:	Reference to the object that triggers the event
Return value:	Any

1.6.3.26 TTrackBar

This class represents a Windows TrackBar control.



Hierarchy	Inherits from TStdControl
See also	TScrollBar
File name	\source\TrackBar.prg

1.6.3.26.1 TTrackBar:Properties

■ read Only ■ Assignable ■ Design assignable ■ Run-time assignable

Scope	Name	Type	Initial value
■	IAutoTicks	Logic	.T.
■	IDownIsLeft	Logic	.F.
■	IFixedLength	Logic	.F.
■	INoThumb	Logic	.F.
■	INoTicks	Logic	.F.
■	IReversed	Logic	.F.
■	ISelRange	Logic	.F.
■	IToolTips	Logic	.F.
■	ITransparent	Logic	.F.
■	nHeight	Numeric	30
■	nLineSize	Numeric	1
■	nMax	Numeric	100
■	nMin	Numeric	0
■	nOrientation	Numeric	orHORIZONTAL
■	nPageSize	Numeric	1

■	nSelEnd	Numeric	0
■	nSelStart	Numeric	0
■	nTickFreq	Numeric	10
■	nTickStyle	Numeric	tbtTOP
■	nTipSide	Numeric	tbsBOTTOM
■	nValue	Numeric	0
■	nWidth	Numeric	120

1.6.3.26.1.1 TTrackBar:IAutoTicks

Draws a marker in every control position.

Scope:	Design assignable
Type:	Logic
Initial value:	.F.

1.6.3.26.1.2 TTrackBar:IDownIsLeft

By default, the control manages the bottom/right and top/left margins in the same way. This property changes this behavior.

Scope:	Design assignable
Type:	Logic
Initial value:	.F.

1.6.3.26.1.3 TTrackBar:IFixedLength

Allows to change the track-bar size.

Scope:	Design assignable
Type:	Logic
Initial value:	.F.

1.6.3.26.1.4 TTrackBar:INoThumb

Disables the track-bat.

Scope:	Design assignable
Type:	Logic
Initial value:	.F.

1.6.3.26.1.5 TTrackBar:INoTicks

Disables the markers.

Scope:	Design assignable
Type:	Logic
Initial value:	.F.

1.6.3.26.1.6 TTrackBar:IReversed

Inverts the control causing and the lower limit is the biggest value and the upper limit is the lowest value.

Scope:	Design assignable
Type:	Logic
Initial value:	.F.

1.6.3.26.1.7 TTrackBar:ISelRange

Activates the possibility to select a range.

Scope:	Design assignable
Type:	Logic
Initial value:	.F.

1.6.3.26.1.8 TTrackBar:ITooltips

If it is .T., it will show a tooltip with the current indicated value.

Scope:	Design assignable
Type:	Logic
Initial value:	.F.

1.6.3.26.1.9 TTrackBar:ITransparent

If it is `.T.` the control's background is transparent. In that case, the `nClrPane` property makes no sense.

Scope:	Design assignable
Type:	Logic
Initial value:	<code>.F.</code>

1.6.3.26.1.10 TTrackBar:nHeight

Indicates the control height.

Scope:	Assignable
Type:	Numeric
Initial value:	30

1.6.3.26.1.11 TTrackBar:nLineSize

Indicates the scroll value when the track-bar is moved with the cursor keys.

Scope:	Assignable
Type:	Numeric
Initial value:	1

1.6.3.26.1.12 TTrackBar:nMax

Indicates the maximum scroll value.

Scope:	Assignable
Type:	Numeric
Initial value:	10

1.6.3.26.1.13 TTrackBar:nMin

Indicates the minimum scroll value.

Scope:	Assignable
Type:	Numeric

Initial value: 0

1.6.3.26.1.14 TTrackBar:nOrientation

Indicates the track-bar orientation.

Scope:	Design assignable
Type:	Numeric
Initial value:	orHORIZONTAL
Possible values:	orHORIZONTAL, orVERTICAL

1.6.3.26.1.15 TTrackBar:nPageSize

Indicates the scroll value when the track-bar is moved with the PgUp and PgDn keys.

Scope:	Assignable
Type:	Numeric
Initial value:	1

1.6.3.26.1.16 TTrackBar:nSelEnd

Indicates the upper limit for the selected range. The control must have the ISelRange property active.

Scope:	Assignable
Type:	Numeric
Initial value:	0

1.6.3.26.1.17 TTrackBar:nSelStart

Indicates the lower limit for the selected range. The control must have the ISelRange property active.

Scope:	Assignable
Type:	Numeric
Initial value:	0

1.6.3.26.1.18 TTrackBar:nTickFreq

Establishes the interval to draw the markers.

Scope:	Assignable
Type:	Numeric
Initial value:	10

1.6.3.26.1.19 TTrackBar:nTickStyle

Establishes the marker styles.

Scope:	Design assignable
Type:	Numeric
Initial value:	tbTOP
Possible values:	tbTOP, tbBOTTOM, tbLEFT, tbRIGHT, tbBOTH

1.6.3.26.1.20 TTrackBar:nTipSide

Establishes the tooltip location. The IToolTips property must be active.

Scope:	Design assignable
Type:	Numeric
Initial value:	tbsBOTTOM
Possible values:	tbsTOP, tbsLEFT, tbsBOTTOM, tbsRIGHT

1.6.3.26.1.21 TTrackBar:nValue

Indicates the current value from the indicator.

Scope:	Assignable
Type:	Numeric
Initial value:	0

1.6.3.26.1.22 TTrackBar:nWidth

Indicates the control width.

Scope:	Assignable
Type:	Numeric
Initial value:	100

1.6.3.26.2 TTrackBar:Methods

■ Constructor ■ Standard ■ Only after Create()

Type	Name
■	ClearTicks
■	GetNumTicks
■	GetTick
■	GetTickPos
■	GetTicks
■	SetTick

1.6.3.26.2.1 TTrackBar:ClearTicks

Deletes all the existing markers. It must be used together with the IAutoTicks property set to .F..

Type	Only after Create()
Parameters	<IRedraw> If it is .T., it will refresh the control
Return value	NIL

1.6.3.26.2.2 TTrackBar:GetNumTicks

Returns the number of existing markers. It must be used together with the IAutoTicks property set to .F..

Type	Only after Create()
Parameters	None
Return value	<nTotal> Total number of markers

1.6.3.26.2.3 TTrackBar:GetTick

Returns the position **nTick** marker position. It must be used together with the IAutoTicks property set to .F..

Type	Only after Create()
Parameters	<nTick>: Marker number
Return value	<nPos>: Marker position

1.6.3.26.2.4 TTrackBar:GetTickPos

Returns the coordinates from the marker specified in **nTick**. It must be used together with the IAutoTicks property set to .F..

Type	Only after Create()
Parameters	<nTick>: Marker number
Return value	<nPos>: Marker coordinates

1.6.3.26.2.5 TTrackBar:GetTicks

Returns an array with the position of every marker. It must be used together with the IAutoTicks property set to .F..

Type	Only after Create()
Parameters	None
Return value	<aPos>: Array with the markers position

1.6.3.26.2.6 TTrackBar:SetTick

Draws a marker in the position indicated by **nTick**. It must be used together with the IAutoTicks property set to .F..

Type	Only after Create()
Parameters	<nTick>: Position
Return value	NIL

1.6.3.26.3 TTrackBar:Events

Name

OnChange

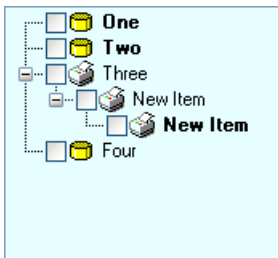
1.6.3.26.3.1 TTrackBar:OnChange

Event that is produced when the indicator changes its position.

Parameters	<oSender>: Reference to the object that triggers the event
:	<nPos>: New indicator position
Return value:	NIL

1.6.3.27 TTreeView

This class represents an TreeView windows type. This class shows tree structure list allowing to the user to select any element from the list.



Hierarchy	Inherits from TStdControl
File name	\source\TreeView.prg

1.6.3.27.1 TTreeView:Properties

■ read Only ■ Assignable ■ Design assignable ■ Run-time assignable

Scope	Name	Type	Initial value
■	altems	Array	{}
■	IAutoHScroll	Logic	.F.
■	IBorder	Logic	.T.
■	ICheckBoxes	Logic	.F.
■	IDoubleBuffer	Logic	.F.
■	IDragDropltem	Logic	.F.

■	IEditLabels	Logic	.F.
■	IFadeInOutExp	Logic	.F.
■	IFullRowSelect	Logic	.F.
■	IHasButtons	Logic	.T.
■	IHasLines	Logic	.T.
■	IHotTrack	Logic	.F.
■	ILinesAtRoot	Logic	.T.
■	IRichTooltip	Logic	.F.
■	IShowSelAlways	Logic	.T.
■	ISingleExpand	Logic	.F.
■	ITooltips	Logic	.T.
■	nClrLine	Numeric	clWindowText
■	nClrPane	Numeric	clWindoww
■	nClrText	Numeric	clBtnText
■	nClrSelFocus	Numeric	clHihgite
■	nClrSelItem	Numeric	clButtonFace
■	nHeight	Numeric	90
■	nIndent	Numeric	16
■	nWidth	Numeric	120
■	oImageList	Object	TImageList()

1.6.3.27.1.1 TTreeView:alters

TTreeViewItem object list that is displayed by the control in the first level. At the same time, every TTreeViewItem object has the same alters property with the possible existing lower branches.

Scope:	Assignable
Type:	Array
Initial value:	{}

It is not common that you need to manipulate this property directly. It is more common to use the AddItem, InsertItem and DeleteItem methods instead.

1.6.3.27.1.2 TTreeView:IAutoHScroll

If it is .T, removes the horizontal scroll bar and auto-scroll depending on mouse position.

Scope	Design assignable
Type	Logic
Initial value	.F.

1.6.3.27.1.3 TTreeView:IBorder

If it is .T. the control will have a border.

Scope	Design assignable
Type	Logic
Initial value	.T.

1.6.3.27.1.4 TTreeView:ICheckBoxes

If it is .T, every element from the control will show a checkbox that will allow to the user select/deselect it.

Scope	Design assignable
Type	Logic
Initial value	.F.

1.6.3.27.1.5 TTreeView:IDoubleBuffer

If it is .T uses double-buffer technique to avoid flickering.

Scope	Design assignable
Type	Logic
Initial value	.F.

1.6.3.27.1.6 TTreeView:IDragDropItem

If it is .T., it will allows Drag & Drop operations among the control elements.

Scope	Design assignable
Type	Logic
Initial value	.F.

1.6.3.27.1.7 TTreeView:IEditLabels

If it is .T., allows to edit the control elements through the TTreeViewItem:Edit method.

Scope	Design assignable
Type	Logic

Initial value	.F.
----------------------	-----

1.6.3.27.1.8 TTreeView:IFadeInOutExp

If it is .T., fade expand buttons in or out when the mouse moves away or into a state of hovering over the control.

Scope	Design assignable
Type	Logic
Initial value	.F.

1.6.3.27.1.9 TTreeView:IFullRowSelect

If it is .T., shows the complete line from the selected element. Otherwise, it will show the element text.

Scope	Design assignable
Type	Logic
Initial value	.F.

Note: This style is incompatible with the property IHasLines to true.

1.6.3.27.1.10 TTreeView:IHasButtons

If it is .T., every branch from the tree that is expandable will show buttons to expand and compress the branch.

Scope	Design assignable
Type	Logic
Initial value	.T.

1.6.3.27.1.11 TTreeView:IHasLines

If it is .T., it will show the lines that join the different tree branches.

Scope	Design assignable
Type	Logic
Initial value	.T.

1.6.3.27.1.12 TTreeView:IFHotTrack

If it is .T., it will show the elements underlined when the user moves the mouse pointer on them.

Scope	Design assignable
Type	Logic
Initial value	.F.

1.6.3.27.1.13 TTreeView:ILinesAtRoot

If it is .T., the first tree level will show lines.

Scope	Design assignable
Type	Logic
Initial value	.T.

1.6.3.27.1.14 TTreeView:IRichTooltip

If it is .T., allows rich tooltips in the tree view (custom drawn with icon and text).

Scope	Design assignable
Type	Logic
Initial value	.F.

1.6.3.27.1.15 TTreeView:IShowSelAlways

The selected element is always shown, even when the control does not have the focus.

Scope	Design assignable
Type	Logic
Initial value	.T.

1.6.3.27.1.16 TTreeView:ISingleExpand

Cause to expand the selected elements and compress the deselected one. If the user keeps the CTRL key pressed when selects and element, the deselected element will not be compressed.

Scope	Design assignable
Type	Logic

Initial value	.F.
----------------------	-----

1.6.3.27.1.17 TTreeView:ITooltips

If it is .T., it will show a Tooltip when all the element's text does not fit in the visible control zone.

Scope	Design assignable
Type	Logic
Initial value	.T.

1.6.3.27.1.18 TTreeView:nClrLine

Indicates the line color.

Scope	Assignable
Type	Numeric
Initial value	clWindowText

(See the appendix to check the colors available)

1.6.3.27.1.19 TTreeView:nClrPane

Indicates the control's background color.

Scope	Assignable
Type	Numeric
Initial value	clWindow

(See the appendix to check the colors available)

1.6.3.27.1.20 TTreeView:nClrText

Indicates the text color.

Scope	Assignable
Type	Numeric
Initial value	clBtnText

(See the appendix to check the colors available)

1.6.3.27.1.21 TTreeView:nClrSelItem

Indicates the text color of the item selected.

Scope	Assignable
Type	Numeric
Initial value	clBtnText

(See the appendix to check the colors available)

1.6.3.27.1.22 TTreeView:nClrSelFocus

Indicates the text color of the item selected with focus.

Scope	Assignable
Type	Numeric
Initial value	clHilighte

(See the appendix to check the colors available)

1.6.3.27.1.23 TTreeView:nHeight

Indicates the control's height.

Scope	Assignable
Type	Numeric
Initial value	90

1.6.3.27.1.24 TTreeView:nIndent

Establishes the TreeView Indent.

Scope	Assignable
Type	Numeric
Initial value	16

1.6.3.27.1.25 TTreeView:nWidth

Indicates the control's width.

Scope	Assignable
Type	Numeric
Initial value	120

1.6.3.27.1.26 TTreeView:oImageList

TImageList object with all the images that will be used by the control.

Scope	Assignable
Type	Object
Initial value	TImageList()

The TImageList object is instantiated since the control is created with its New() constructor. The first image included in TImageList establishes the dimensions for the next images to be used. If the first image is an image that includes more than one bitmap it is important to establish the TImageList nHeight and nWidth properties before to add a bitmap.

1.6.3.27.2 TTreeView:Methods

■ Constructor ■ Standard ■ Only after Create()

Typ	Name
■	AddImage
■	AddItem
■	CollapseAll
■	DeleteItems
■	EndEditLabelNow
■	ExpandAll
■	GetAllItems
■	GetCount
■	GetItemByName
■	GetItemByHandle
■	GetSelectedItem
■	GetVisibleCount
■	HitTest
■	InsertItem
■	IsEmpty
■	RestoreExpandState
■	SaveExpandState
■	SelectItem

- SelectRoot

1.6.3.27.2.1 TTreeView:AddImage

Adds a new image to the olmageList object.

Type	Standard
Parameters	<xImage> : Resource name, file name or bitmap handle <IMasked> : If it is .T., the image will be converted to mask format (all the color are modified to gray scale colors)
Return value	<nImage> : Position number from the new image in olmageList

1.6.3.27.2.2 TTreeView:AddItem

Add a new element to the control list of values.

Type	Standard
Parameters	<cItem> : New element description [<xImage>] : Image number in the olmageList or resource name or file that will be shown with the item. Default: zero, that means no image [<xSellImage>] : Image number in the olmageList or resource name or file that will be shown with the item when it is selected. Default: xImage [<IBold>] : If it is .T., it will show the item in bold. Default: .F. [<IChecked>] : If it is .T., it will show the item checked. Default: .F. [<IParam>] : Container for a numeric data to be used in the sort process with the methods SortChildren() and AltSortChildren() from the TTreeViewItem object
Return value	<oItem> : TTreeViewItem object created

1.6.3.27.2.3 TTreeView:CollapseAll

Collapse all the tree branches.

Type	Only after Create()
Parameters	None
Return value	NIL

1.6.3.27.2.4 TTreeView:DeleteItems

Deletes all the control elements.

Type	Standard
Parameters	None
Return value	NIL

1.6.3.27.2.5 TTreeView:EndEditLabelNow

Finalizes the edit operation in the item.

Type	Only after Create()
Parameters	[<ICancel>]: If it is .T., the edit operation is canceled and its values is not changed. Default: .T.
Return value	NIL

1.6.3.27.2.6 TTreeView:ExpandAll

Expand all the tree branches.

Type	Only after Create()
Parameters	None
Return value	NIL

1.6.3.27.2.7 TTreeView: GetAllItems

Returns in an array, all the TTreeViewItem elements from the control, regardless the branch level .

Type	Standard
Parameters	[<oFirst>]: First element to be searched
Return value	<aTreeViewItems>

1.6.3.27.2.8 TTreeView: GetCount

Returns the number of existing items in the control.

Type	Only after Create()
Parameters	None
Return value	<nTotal>

1.6.3.27.2.9 TTreeView: GetItemByName

Searches a branch in the control with the text specified.

Type	Standard
Parameters	<cText>: Text to search [<INoCase>]: If it is .T., the search will be case insensitive. Default: .F. [<oFirst>]: First element to be searched
Return value	<oItem> First TTreeViewItem found o NIL

1.6.3.27.2.10 TTreeView: GetItemByHandle

Searches a branch in the control with the handle specified.

Type	Only after Create()
Parameters	<hItem>: Item handle (TTreeViewItem) [<oFirst>]: First element to be searched

Return value	<oltem> First TTreeViewItem found o NIL
---------------------	--

1.6.3.27.2.11 TTreeView:GetSelectedItem

Returns the current selected TTreeViewItem.

Type	Only after Create()
Parameters	None
Return value	<oltem> TTreeViewItem selected or NIL

1.6.3.27.2.12 TTreeView:GetVisibleCount

Returns the number of visible items in the control.

Type	Only after Create()
Parameters	None
Return value	<nTotal>

1.6.3.27.2.13 TTreeView:HitTest

Returns the existing item in an specific coordinate.

Type	Only after Create()
Parameters	<nX>: X Coordinate <nY>: Y Coordinate
Return value	<oltem> Existing TTreeViewItem item, or NIL

1.6.3.27.2.14 TTreeView:InsertItem

Inserts a new item in the control list of values.

Type	Standard
Parameters	<citem>: New item description

	<p>[<xImage>]: Image number in the olmageList or resource name or file that will be shown with the item. Default: zero, that means no image</p> <p>[<xSellImage>]: Image number in the olmageList or resource name or file that will be shown with the item when it is selected. Default: xImage</p> <p>[<IBold>]: If it is .T. it will show the item in bold. Default: .F.</p> <p>[<IChecked>]: If it is .T. it will show the item checked. Default: .F.</p> <p>[<IParam>]: Container for a numeric data to be used in the sort process with the methods SortChildren() and AltSortChildren() from the TTreeViewItem object</p> <p>[<nPos>]: Indicates the relative position for the new element in its container item. Default: last position</p>
Return value	<oltem>: TTreeViewItem item created

1.6.3.27.2.15 TTreeView:IsEmpty

Returns .T. if the control is empty.

Type	Standard
Parameters	None
Return value	<IEmpty> .T. if the control is empty

1.6.3.27.2.16 TTreeView:RestoreExpandState

Restores the expand/collapse status of every tree branch based in a literal gotten through the SaveExpandState method.

Type	Only after Create()
Parameters	<cState> Literal with the control state
Return value	NIL

Nota: Is important to keep the configuration for the control elements between SaveExpandState and RestoreExpandState operations. Otherwise the effects will not be the desired ones.

1.6.3.27.2.17 TTreeView:SaveExpandState

Saves the expand/compress state of every tree branch in a literal. It is used together with RestoreExpandState.

Type	Only after Create()
Parameters	<cState> Literal with the control state
Return value	NIL

Nota: Is important to keep the configuration for the control elements between SaveExpandState and RestoreExpandState operations. Otherwise the effects will not be the desired ones.

1.6.3.27.2.18 TTreeView:SelectItem

Selects an specific TTreeViewItem object from the control.

Type	Only after Create()
Parameters	<oltem> TTreeViewItem object to select [<IFirstVisible>]: If it is .T., the item will be shown, if possible, as the first. Default: .F.
Return value	<ISuccess>: .T. if the operation is successful

1.6.3.27.2.19 TTreeView:SelectRoot

Selects the first item in from the control.

Type	Only after Create()
Parameters	None
Return value	NIL

1.6.3.27.3 TTreeView:Events

Name
OnBeginDragItem
OnBeginLabelEdit
OnCheckStateChanged

OnClick
OnDragOverItem
OnEndDragItem
OnEndLabelEdit
OnItemExpanded
OnItemExpanding
OnKeyDown
OnRClick
OnSelChanged
OnSelChanging
OnSingleExpand

1.6.3.27.3.1 TTreeView:OnBeginDragItem

Event that is produced when starts a Drag & drop operation among items.

Parameters	<oSender>: Reference to the object that triggers the event
:	<oltem>: TTreeViewItem object that starts the operation
Return value:	NIL

1.6.3.27.3.2 TTreeView:OnBeginLabelEdit

Event that is produced when starts the edition of any item.

Parameters	<oSender>: Reference to the object that triggers the event
:	<oltem>: TTreeViewItem in edit mode
Return value:	NIL

1.6.3.27.3.3 TTreeView:OnCheckStateChanged

Event that is produced when the check state of any item changes.

Parameters	<oSender>: Reference to the object that triggers the event
:	<oltem>: TTreeViewItem that has changed

Return value:	NIL
----------------------	-----

1.6.3.27.3.4 TTreeView:OnClick

Event that is produced when the user clicks on the control.

Parameters :	<oSender>: Reference to the object that triggers the event <nPoxX>: X mouse position <nPosY>: Y mouse position
Return value:	NIL

Nota: Use the HitTest method to get the item in the X,Y position.

1.6.3.27.3.5 TTreeView:OnDragOverItem

Event that is produced when in a Drag & Drop operation among items, the cursor is located on an specific item.

Parameters :	<oSender>: Reference to the object that triggers the event <oItemTo>: TTreeViewItem where the cursor is located
Return value:	<ISuccess>: If returns .F., the 'prohibited' cursor will be shown indicating that is not possible to make the Drop operation in that item

1.6.3.27.3.6 TTreeView:OnEndDragItem

Event that is produced when finalizes a Drag & Drop operation among item.

Parameters :	<oSender>: Reference to the object that triggers the event <oItemFrom>: TTreeViewItem object that started the operation <oItemTo>: TTreeViewItem target destination
------------------------	---

Return value:	Only if returns NIL, the oItemFrom will be moved just after the oItemTo item
----------------------	--

1.6.3.27.3.7 TTreeView:OnEndLabelEdit

Event that is produced when ends an edit operation.

Parameters	<oSender>: Reference to the object that triggers the event <oItem>: TTreeViewItem that ends the edition <cTex>: Text introduced by the user. If it pressed the ESC key its value will be NIL
Return value:	A logical FALSE return value prevents item text change

1.6.3.27.3.8 TTreeView:OnItemExpanded

Event that is produced when the user has expanded or compresses a branch from the tree.

Parameters	<oSender>: Reference to the object that triggers the event <hItem>: Item's handle <nState>: Item state
Return value:	NIL

nState is a numeric "flag" value that indicates the item state. For more information, check the API documentation: [Microsoft MSDN Item states \(Internet\)](#)

1.6.3.27.3.9 TTreeView:OnItemExpanding

Event that is produced when the user is expanding or compressing a branch from the tree.

Parameters	<oSender>: Reference to the object that triggers the event <hItem>: Item's Handle <nState>: Item state
-------------------	--

Return value:	<ICancel> If returns .T. the process is canceled
----------------------	--

nState is a numeric "flag" value that indicates the item state. For more information, check the API documentation: [Microsoft MSDN Item states \(Internet\)](#)

1.6.3.27.3.10 TTreeView:OnKeyDown

Event that is produced when the user presses a key.

Parameters	<oSender> : Reference to the object that triggers the event
:	<nKey> : Key pressed
Return value:	NIL

1.6.3.27.3.11 TTreeView:OnRClick

Event that is produced when the user right-clicks in the control.

Parameters	<oSender> : Reference to the object that triggers the event
:	<nPoxX> : X mouse position
	<nPosY> : Y mouse position
Return value:	NIL

Nota: Use the `HitTest` method to get the item in the X,Y position.

1.6.3.27.3.12 TTreeView:OnSelChanged

Event that is produced when the users changes to the selected item.

Parameters	<oSender> : Reference to the object that triggers the event
:	<nAction> : Change method: TVC_BYKEYBOARD, TVC_BYMOUSE, TVC_UNKNOWN
	<hOldItem> : Old Item's handle

	<hNewItem> : New item's handle
Return value:	NIL

1.6.3.27.3.13 TTreeView:OnSelChanging

Event that is produced when the users is changing the selected item.

Parameters	<oSender> : Reference to the object that triggers the event
:	<nAction> : Change method: TVC_BYKEYBOARD, TVC_BYMOUSE, TVC_UNKNOWN
	<hOldItem> : Old Item's handle
	<hNewItem> : New item's handle
Return value:	<ICancel> If returns .T. the process is canceled

1.6.3.27.3.14 TTreeView:OnSingleExpand

Event that is produced when the user expands a branch clicking on it and closes the previously selected branch. It is used together with the ISingleExpand property.

Parameters	<oSender> : Reference to the object that triggers the event
:	<hOldItem> : Old item's handle (compressed branch)
	<hNewItem> : New item's handle (branch to expand)
Return value:	<nOper> <ul style="list-style-type: none"> • NIL: The operation continues • TVNRET_SKIPOLD: Don't compress the old item • TVNRET_SKIPOLD: Don't expand new item It is possible to use both options with the OR() operator

1.6.3.28 TTreeViewItem

This class represents a TTreeView control items.

Description:

This class manages the TTreeView control items.

Hierarchy Inherits from TComponent
File name \source\TreeViewItem.prg

1.6.3.28.1 TTreeViewItem:Properties

■ read Only ■ Assignable ■ Design assignable ■ Run-time assignable

Scope	Name	Type	Initial value
■	altems	Array	{}
■	cText	Character	""
■	Handle	Numeric	0
■	IBold	Logic	.F.
■	IChecked	Logic	.F.
■	IParam	Numeric	0
■	nImage	Numeric	0
■	nIndex	Numeric	0
■	nSellImage	Numeric	0
■	oParent	Object	NIL
■	oTreeView	Object	NIL

1.6.3.28.1.1 TTreeViewItem:altems

Array with all the items that depend from the current one.

Scope	readOnly
Type	Array
Initial value	{}

1.6.3.28.1.2 TTreeViewItem:cText

Text to show in the item.

Scope	Assignable
Type	Character
Initial value	""

1.6.3.28.1.3 TTreeViewItem:Handle

Windows handle to the item.

Scope	read Only
Type	Numeric
Initial value	0

1.6.3.28.1.4 TTreeViewItem:IBold

If .T., it will show the item in bold.

Scope	Assignable
Type	Logic
Initial value	.F.

1.6.3.28.1.5 TTreeViewItem:IChecked

If .T. it will show the item with the checkbox active (the ICheckBoxes property from its TTreeView container must be active).

Scope	Assignable
Type	Logic
Initial value	.F.

1.6.3.28.1.6 TTreeViewItem:IParam

Container for a numeric data to be used in the sort process by the SortChildren() and AltSortChildren() methods

Scope	Assignable
Type	Numeric
Initial value	0

1.6.3.28.1.7 TTreeViewItem:nImage

Image to show in the item. It corresponds with the image number in the oImageList object from its TTreeView container.

Scope	Assignable
Type	Numeric
Initial value	0

1.6.3.28.1.8 TTreeViewItem:nIndex

Created item order in its TTreeView container.

Scope	read Only
Type	Numeric
Initial value	0

1.6.3.28.1.9 TTreeViewItem:nSellImage

Image to show in the item when is selected. It corresponds with the image number in the oImageList object from its TTreeView container.

Scope	Assignable
Type	Numeric
Initial value	0

1.6.3.28.1.10 TTreeViewItem:oParent

Container object from the current item. It can be another item from a upper or same level than the TTreeView object if is in the main tree root.

Scope	read Only
Type	Object
Initial value	NIL

1.6.3.28.1.11 TTreeViewItem:oTreeView

TTreeView property object from all the items.

Scope	read Only
Type	Object
Initial value	NIL

1.6.3.28.2 TTreeViewItem:Methods

■ Constructor ■ Standard

Type	Name
■	AddItem
■	AltSortChildren
■	Collapse
■	Create
■	Delete
■	DeleteChilds DeleteItems
■	Edit
■	EnusreVisible
■	Expand
■	ExpandPartial
■	GetNextSibling
■	GetPrevSibling
■	InsertItem
■	IsExpanded
■	MoveAfter
■	MoveDown
■	MoveInto
■	MoveUp
■	Select
■	SetExpandable
■	SortChildren

1.6.3.28.2.1 TTreeViewItem:AddItem

Adds a new item in the sub-tree from the current item.

Type	Standard
Parameters	<cltem> Item text <xImage> Image to use in the item. If it is a character data

type indicates a resource image file and in that case the image will be added to the olmageList object from its TTreeView container object.

<xSellImage>

Image to use in the item when is selected. If it is a character data type indicates a resource image file and in that case the image will be added to the olmageList object from its TTreeView container object.

<IBold>

If it is .T. the text will be shown in bold

<Checked>

If it is .T. the item will show a checkbox active (the ICheckboxes property from its TTreeView container must be active).

<IParam>

Container for a numeric data to be used in the sort processes with the SortChildren and AltSortChildren methods. The default values are their property equivalents

Return value	<oltem> Reference to the new item created.
---------------------	--

1.6.3.28.2.2 TTreeViewItem:AltSortChildren

Sort the son levels from the current item based in the IParam property value..

Type	Standard
Parameters	None
Return value	<ISuccess> .T. if the operation was successful

1.6.3.28.2.3 TTreeViewItem:Collapse

Compress the branch from the current item (if any).

Type	Standard
Parameters	None
Return value	NIL

1.6.3.28.2.4 TTreeViewItem:Create

Class constructor. It receives as parameters its container item or its own TTreeView object.

Type	Constructor
Parameters	oParent
Return value	Self reference (Self)

1.6.3.28.2.5 TTreeViewItem:Delete

Deletes the current item. If the item is on all the branches, all of them will be deleted as well.

Type	Standard
Parameters	None
Return value	<ISuccess> .T. if the operation is successful

1.6.3.28.2.6 TTreeViewItem:DeleteChilds

Deletes all the daughter branches from the current item. The current item is not deleted.

Type	Standard
Parameters	None
Return value	<ISuccess> .T. if the operation is successful

1.6.3.28.2.7 TTreeViewItem>Edit

Edits de label of the Item. The property IEditLabels of his Treeview parent should be set to TRUE.

Type	Standard
Parameters	None
Return value	<ISuccess> .T. if the operation is successful

1.6.3.28.2.8 TTreeViewItem:EnsureVisible

Forces de Item to be visible.

Type	Standard
Parameters	None
Return value	<ISuccess> .T. if the operation is successful

1.6.3.28.2.9 TTreeViewItem:Expand

Expand the branch from the current item (if any).

Type	Standard
Parameters	None
Return value	NIL

1.6.3.28.2.10 TTreeViewItem:ExpandPartial

Expands partially the branch from the current item (if any).

Type	Standard
Parameters	None
Return value	NIL

1.6.3.28.2.11 TTreeViewItem:GetNextSibling

Returns the next TTreeViewItem object in the branch (lower neighbor).

Type	Standard
Parameters	None
Return value	<oltem> Reference to the lower neighbor or NIL if was not found.

1.6.3.28.2.12 TTreeViewItem:GetPrevSibling

Returns the previous TTreeViewItem object from the branch (upper neighbor).

Type	Standard
Parameters	None
Return value	<oltem> Reference to the upper neighbor or NIL if was not found

1.6.3.28.2.13 TTreeViewItem:InsertItem

Inserts a new item in the sub-branch from the current item.

Type	Standard
Parameters	<cltem> Item text <xImage> Image to use in the item. If it is a character data type indicates a resource image file and in that case the image will be added to the olmagineList object from its TTreeView container object. <xSellImage> Image to use in the item when is selected. If it is a character data type indicates a resource image file and in that case the image will be added to the olmagineList object from its TTreeView container object. <IBold> If it is .T. the text will be shown in bold <Checked> If it is .T. the item will show a checkbox active (the lCheckboxes property from its TTreeView container must be active). <IParam> Container for a numeric data to be used in the sort processes with the SortChildren and AltSortChildren methods. <nPos> Item position in its container branch. Default value: after the last existing item in the branch. The default values are their property equivalents.
Return value	<oltem> Reference to the new created item

1.6.3.28.2.14 TTreeViewItem:IsExpanded

Returns true if the item is expanded.

Type	Standard
Parameters	None
Return value	<IState> Expandstate

1.6.3.28.2.15 TTreeViewItem:MoveAfter

Moves the current item one position after the item passed as parameter.

Type	Standard
Parameters	<oltem> Item where the current item will be moved
Return value	<ISuccess> .T. if the operation is successful

1.6.3.28.2.16 TTreeViewItem:MoveDown

Moves the current item one position down in its container branch.

Type	Standard
Parameters	None
Return value	<ISuccess> .T. if the operation is successful

1.6.3.28.2.17 TTreeViewItem:MoveInto

Moves the current item into the branch in the item passed as parameter.

Type	Standard
Parameters	<oltem> Item where the current item will be moved
Return value	<ISuccess> .T. if the operation is successful

1.6.3.28.2.18 TTreeViewItem:MoveUp

Moves up the current item in its container branch.

Type	Standard
Parameters	None
Return value	<ISuccess> .T. if the operation is successful

1.6.3.28.2.19 TTreeViewItem:Select

Selects the current item.

Type	Standard
Parameters	[<IFirstVisible>] If .T. the Item selected will be the first visible. By default .F.
Return value	<ISuccess> .T. if the operation is successful

1.6.3.28.2.20 TTreeViewItem:SetExpandable

Shows the plus button even when it has not any child items. You can add any child item on the OnItemExpanding event.

Type	Standard
Parameters	None
Return value	<ISuccess> .T. if the operation is successful

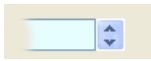
1.6.3.28.2.21 TTreeViewItem:SortChildren

Sorts the sons from the current item in descending order.

Type	Standard
Parameters	<IRecursive> If the sort will affect all the lower branches or only the branches from the first level
Return value	<ISuccess> .T. if the operation is successful

1.6.3.29 TUpDown

This class represents a Windows UpDown control (spinner or buttons to increase / decrease values).



Description:

This class represents a Windows UpDown control (spinner or buttons to increase / decrease values).

Normally is used together with an Edit type control.

Hierarchy Inherits from Tcontrol
File name \source\UpDown.prg

1.6.3.29.1 TUpDown:Properties

■ read Only ■ Assignable ■ Design assignable ■ Run-time assignable

Scope	Name	Type	Initial value
■	aAccels	Array	{}
■	IArrowKeys	Logic	.T.
■	AutoBuddy	Logic	.F.
■	IHotTrack	Logic	.T.
■	INoThousands	Logic	.F.
■	ISyncBuddy	Logic	.T.
■	IWrap	Logic	.F.
■	nAccelTime	Numeric	5
■	nBase	Numeric	10
■	nBuddyAlign	Numeric	alRIGHT
■	nIncrement	Numeric	1
■	nMax	Numeric	0
■	nMin	Numeric	0
■	nOrientation	Numeric	orVERTICAL
■	nValue	Numeric	0
■	oBuddy	Object	

1.6.3.29.1.1 TUpDown:aAccels

Two dimension array that indicates the increment that will be applied when the arrow keys are pressed. Every element contains the acceleration indicated in seconds, and the increment value to apply.

Scope	Assignable
■	■

Type	Array
Initial value	{}

1.6.3.29.1.2 TUpDown:IArrowKeys

Allows to manage the control with the arrow keys.

Scope	Design assignable
Type	Logic
Initial value	.T.

1.6.3.29.1.3 TUpDown:IAutoBuddy

If there is not a TEdit control partner established, this property will join the closest TEdit control.

Scope	Design assignable
Type	Logic
Initial value	.F.

1.6.3.29.1.4 TUpDown:IHotTrack

Highlights the arrow indicators when the mouse pointer is over them.

Scope	Design assignable
Type	Logic
Initial value	.T.

1.6.3.29.1.5 TUpDown:INoThousands

Formats the TEdit control partner content with a thousand mask.

Scope	Design assignable
Type	Logic
Initial value	.F.

1.6.3.29.1.6 TUpDown:ISyncBuddy

Updates automatically the TEdit object partner every time that nValue changes its value.

Scope	Assignable
Type	Logic
Initial value	.T.

1.6.3.29.1.7 TUpDown:IWrap

Allows to rotate between the lower and upper range limits. for example, if nValue is equal to nMax and the nValue continues increasing, the nValue property value becomes the nMin value.

Scope	Design assignable
Type	Logic
Initial value	.F.

1.6.3.29.1.8 TUpDown:nAccelTime

Minimum acceleration time.

Scope	Assignable
Type	Numeric
Initial value	5

1.6.3.29.1.9 TUpDown:nBase

Numeric base for the control. It can be Decimal (10) or Hexadecimal (16).

Scope	Assignable
Type	Numeric
Initial value	10

1.6.3.29.1.10 TUpDown:nBuddyAlign

Control alignment based in its partner control (Normally a TEdit)

Scope	Design assignable
Type	Numeric

Initial value	alRIGHT
Possible values	alNONE, alLEFT, alRIGHT

1.6.3.29.1.11 TUpDown:nIncrement

Position change increment to use after the time specified by nAccelTime elapses.

Scope	Assignable
Type	Numeric
Initial value	1

1.6.3.29.1.12 TUpDown:nMax

Maximum value in the selection range.

Scope	Assignable
Type	Numeric
Initial value	0

1.6.3.29.1.13 TUpDown:nMin

Minimum value in the selection range.

Scope	Assignable
Type	Numeric
Initial value	0

1.6.3.29.1.14 TUpDown:nOrientation

Indicates the position which the control will be shown. It can be horizontal or vertical.

Scope	Design assignable
Type	Numeric
Initial value	orVERTICAL
Possible values	orHORIZONTAL, orVERTICAL

1.6.3.29.1.15 TUpDown:nValue

Current control's value. Normally it is equivalent to the TEdit control partner value, if present.

Scope	Assignable
Type	Numeric
Initial value	0

1.6.3.29.1.16 TUpDown:oBuddy

TEdit object that is managed together with the control.

Scope	Design assignable
Type	Object
Initial value	NIL

1.6.3.29.2 TUpDown:Events

Name	
OnChange	

1.6.3.29.2.1 TUpDown:OnChange

Event that is activated before to change the control value and allows to decide if it will accept the change or not.

Parameters	<oSender> Reference to the object that triggers the event <nPos> Current value <nDelta> Increment
Return value	<IChange> if it is .T., it will not allow the change

1.6.4 Communications

1.6.4.1 TWebView

This control is a complete Internet browser based on **Microsoft Edge** (Chromium)

For this to work, the WebView2 runtime must be installed and the WebView2Loader.Dll library must be located in your application directory. You can locate this file in the same folder where

the Xailer executable itself is located. That is, \Xailer\Bin.

Important note:

TWebView does not work with BCC, it is only available with MinGW.

Further information:

<https://docs.microsoft.com/en-en/microsoft-edge/webview2/>

Hierarchy TStdControl descendant
File \source\WebView.prg

1.6.4.1.1 TWebView:Properties

■ Read only ■ Assignable ■ Design assignable ■ Run-time assignable

Scope	Name	Type	Initial value
■	cAdditionalArgs	Character	""
■	cRequestedFilter	Character	""
■	cRuntimeDir	Character	""
■	cUserDataDir	Character	""
■	IAllowDevTools	Logical	.T.
■	IAllowScripts	Logical	.T.
■	ICleanUserData	Logical	.T.
■	IContextMenu	Logical	.T.
■	IStatusBar	Logical	.T.
■	nBorderStyle	Numeric	bvNONE
■	nHeight	Numeric	200
■	nWidth	Numeric	300
■	nZoomFactor	Numeric	1.0

1.6.4.1.1.1 TWebView:cAdditionalArgs

It is used to pass additional parameters to the TWebview. More information in:

<https://peter.sh/experiments/chromium-command-line-switches>

Scope	Design assignable
Type	Character
Initial value	""

1.6.4.1.1.2 TWebView:cRequestedFilter

It is a filter for OnWebResourceRequested event. It is used with wildcards

Scope	Design assignable
Type	Character
Initial value	""

1.6.4.1.1.3 TWebView:cRuntimeDir

It is used to indicate a path where the "fixed" run time of the WebView is installed. Only the fixed one; for the "evergreen" it is not necessary to indicate anything. By default its value is GetFolderCommonAppData() + "\WebView", that in Win10 is in "C:\ProgramData\WebView".

Scope	Design assignable
Type	Character
Initial value	GetFolderCommonAppData()\WebView

1.6.4.1.1.4 TWebView:cUserDataDir

Browser temporary file storage directory.

Scope	Design assignable
Type	Character
Initial value	""

1.6.4.1.1.5 TWebView:lAllowDevTools

If true it is allowed to invoke the developer tools.

Scope	Design assignable
Type	Logical
Initial value	.T.

1.6.4.1.1.6 TWebView:lAllowScripts

If true, JavaScript code execution is allowed.

Scope	Design assignable
--------------	-------------------

Type	Logical
Initial value	.T.

1.6.4.1.1.7 TWebView:ICleanUserData

If true, all user browsing information will be erased when the control is destroyed

Scope	Design assignable
Type	Logical
Initial value	.T.

1.6.4.1.1.8 TWebView:IContextMenu

If true, the context menu will be displayed when the right mouse button is clicked in the navigation window.

Scope	Design assignable
Type	Logical
Initial value	.T.

1.6.4.1.1.9 TWebView:IStatusBar

If true, the browser status bar will be displayed..

Scope	Design assignable
Type	Logical
Initial value	.T.

1.6.4.1.1.10 TWebView:nBorderStyle

Style with which the border is drawn.

Scope	Design assignable
Type	Numeric
Initial value	bvETCHED
Possible values	bvNONE, bvRAISED, bvSUNKEN, bvBUMP, bvETCHED, bvFLAT

1.6.4.1.1.11 TWebView:nHeight

Control height.

Scope	Asignable
Type	Numeric
Initial value	200

1.6.4.1.1.12 TWebView:nWidth

Control width.

Scope	Asignable
Type	Numeric
Initial value	300

1.6.4.1.1.13 TWebView:nZoomFactor

Control zoom.

Scope	Asignable
Type	Numeric
Initial value	1.0
Possible values	1.0 equals 100% zoom 1.5 it would be a zoom of 150%.

1.6.4.1.2 TWebView:Methods

■ Constructor ■ Standard ■ Only after Create()

Type	Name
■	CanGoBack
■	CanGoForward
■	CleanUserData
■	DeleteAllCookies
■	ExecuteScript
■	GetTitle
■	GoBack
■	GoForward
■	IsInstalled
■	Navigate
■	NavigateEx

<input type="checkbox"/>	Reload
<input type="checkbox"/>	SetContent
<input type="checkbox"/>	SetVirtualHostNameToFolder
<input type="checkbox"/>	Stop

1.6.4.1.2.1 TWebView:CanGoBack

True if the browser can go backwards in its browsing history.

Type	Only after Create()
Parameters	None
Return value	IResult

1.6.4.1.2.2 TWebView:CanGoForward

True if the browser can go forward in its browsing history.

Type	Only after Create()
Parameters	None
Return value	IResult

1.6.4.1.2.3 TWebView:CleanUserData

Deletes all the user's browsing data.

Type	Only after Create()
Parameters	None
Return value	NIL

1.6.4.1.2.4 TWebView>DeleteAllCookies

Delete all existing cookies.

Type	Only after Create()
Parameters	None
Return value	NIL

1.6.4.1.2.5 TWebView:ExecuteScript

Execute a JavaScript script. You have to write the javascript code as is for it to execute. E.g:

```
oSender:OnExecuteScriptCompleted := { | oSender, cResult, nErrorCode | LogDebug( cResult ) }  
oSender:ExecuteScript( "document.body.innerText;" )
```

This would display in the debugger window all the HTML content of the page.

Type	Only after Create()
Parameters	None
Return value	NIL

1.6.4.1.2.6 TWebView:GetTitle

Return the page title.

Type	Only after Create()
Parameters	None
Return value	cTitle

1.6.4.1.2.7 TWebView:GoBack

Return to the last navigated URL.

Type	Only after Create()
Parameters	None
Return value	NIL

1.6.4.1.2.8 TWebView:GoForward

Return to the next navigated URL.

Type	Only after Create()
Parameters	None
Return value	NIL

1.6.4.1.2.9 TWebView:IsInstalled

Returns true if WebView2 run-time is installed on the computer.

Type	Only after Create()
Parameters	None
Return value	NIL

1.6.4.1.2.10 TWebView:Navigate

Navigate to a specified URL.

Type	Only after Create()
Parameters	<cUrl> Internet address
Return value	-1 si error

1.6.4.1.2.11 TWebView:NavigateEx

Navigates to a specific URL and can set values in the header and a postscript.

Type	Only after Create()
Parameters	<cUrl> Internet address [<cHeaders>] Optional headers [<cPostData>] Postdata
Return value	-1 if error

1.6.4.1.2.12 TWebView:Reload

Reload the current page.

Type	Only after Create()
Parameters	None
Return value	NIL

1.6.4.1.2.13 TWebView:SetContent

Directly sets the content to be displayed in the browser.

Type	Only after Create()
Parameters	<cContent> Content to display [<Utf8>] Indicates if the string is already in UTF8 format.
Return value	-1 if error

1.6.4.1.2.14 TWebView:SetVirtualHostNameToFolder

It is used to **map** a URL (host) to a local folder. For example:

```
::oWebView:SetVirtualHostNameToFolder( "test.com", "C:\Test" )
```

It causes all resources that are attempted to be downloaded from **test.com** to be searched directly in the local folder **C:\Test**, even if there are sub directories.

Type	Only after Create()
Parameters	<cUrl> URL host <cDir> local directory
Return value	True if success

1.6.4.1.2.15 TWebView:Stop

Stops page loading.

Type	Only after Create()
Parameters	None
Return value	NIL

1.6.4.1.3 TWebView:Events

Name
OnAcceleratorKeyPressed
OnContentLoading
OnDocumentTitleChanged
OnExecuteScriptCompleted
OnHistoryChanged

OnNavigationCompleted
OnNavigationStarting
OnNewWindowRequested
OnSourceChanged
OnWebMessageReceived
OnWebResourceRequested

1.6.4.1.3.1 TWebView:OnAcceleratorKeyPressed

Event that occurs every time a keyboard shortcut is pressed.

Parameters	<oSender>: Reference to the object that fires the event <nEventType>: CORE_WEBVIEW2_KEY_EVENT_KIND_KEY_DOWN CORE_WEBVIEW2_KEY_EVENT_KIND_KEY_UP CORE_WEBVIEW2_KEY_EVENT_KIND_KEY_DOWN CORE_WEBVIEW2_KEY_EVENT_KIND_SYSTEM_KEY_DOWN <nVirtualKey>: Key code of VK_????? <lParam>: Is the LPARAM value of the same WM_KEYDOWN message received by the WebView
Return value	0 Nil = continue, <>0 = Do not process

1.6.4.1.3.2 TWebView:OnContentLoading

Event that occurs when the page starts loading.

Parameters	<oSender>: Reference to the object that fires the event <IsErrorPage>: If true, it is an error page
Return value	NIL

1.6.4.1.3.3 TWebView:OnDocumentTitleChanged

Event that occurs when the page title is changed.

Parameters	<oSender>: Reference to the object that fires the event
Return value	NIL

1.6.4.1.3.4 TWebView:OnExecuteScriptCompleted

Event that occurs when the execution of a script is finished.

Parameters	<oSender>: Reference to the object that fires the event <cResult>: Result returned by the script <nError>: Error code. 0 if success
Return value	NIL

1.6.4.1.3.5 TWebView:OnHistoryChanged

Event that occurs every time there is a change in the history of visited pages.

Parameters	<oSender>: Reference to the object that fires the event
Return value	NIL

1.6.4.1.3.6 TWebView:OnNavigationCompleted

Event that occurs when navigation is terminated.

Parameters	<oSender>: Reference to the object that fires the event <nError>: Error code. 0 if successful
Return value	NIL

1.6.4.1.3.7 TWebView:OnNavigationStarting

Event that occurs when the page navigation starts.

Parameters	<oSender>: Reference to the object that fires the event <cUrl>: Internet address
Return value	0 Nil = Allow, <>0 = Do Not Allow

1.6.4.1.3.8 TWebView:OnNewWindowRequested

Event that occurs when the page navigation starts..

Parameters	<oSender> : Reference to the object that fires the event <cUrl> : Internet address
Return value	0 Nil = Allow, 1 = Open in the same window 2 = Do Not Allow

1.6.4.1.3.9 TWebView:OnSourceChanged

Event that occurs when a change in the page content is initiated.

Parameters:	<oSender> : Reference to the object that fires the event <llsNewDocument> : True, if this is a new page
Return value:	NIL

1.6.4.1.3.10 TWebView:OnWebMessageReceived

Event that occurs when a JSON type message is received.

Parameters:	<oSender> : Reference to the object that fires the event <cJSON> : son with the received information
Return value:	NIL

1.6.4.1.3.11 TWebView:OnWebResourceRequested

Event that occurs when requesting any Web resource.

Parameters:	<oSender> : Reference to the object that fires the event <cUri> :
--------------------	---

Internet address

<**@cData**>:

Data stream replacement if applicable

<**@cHeaders**>:

Header

<**cPostData**>:

The data that is being sent by POST to the request

<**cMethod**>:

Method of the request (GET, POST, PUT, ...). It is informational, i.e. the value is received in the event, but the user cannot change it.

Return value: NIL

1.6.4.2 TAsyncDownload

This class allows you to download in asynchronous mode from the Internet using the HTTP protocol any information. The **TAsynFileDownload** class is almost identical to this one with the only difference that it specifies the name of the files in remote and local and the class does the whole process of reception.

It is important to take into account the error values that can occur because they are exactly the same as when accessing any Web page with a browser.

Hierarchy File

TWinObject descendant
source\enterprise\AsyncDownload.prg

1.6.4.2.1 TAsyncDownload:Properties

■ Read only ■ Assignable ■ Design assignable ■ Run-time assignable

Scope	Name	Type	Initial value
■	cBuffer	Character	""
■	cError	Character	""
■	cUrl	Character	""
■	nStatus	Numeric	0
■	lConnected	Logical	.F.
■	lSuccess	Logical	.F.

1.6.4.2.1.1 TAsyncDownload:cBuffer

Buffer size for reception.

Scope	Read only
Type	Character
Initial value	""

1.6.4.2.1.2 TAsyncDownload:cError

Last error produced.

Scope	Read only
Type	Character
Initial value	""

1.6.4.2.1.3 TAsyncDownload:cUrl

URL to perform the download.

Scope	Assignable
Type	Character
Initial value	""

1.6.4.2.1.4 TAsyncDownload:nStatus

Last error produced. The error values are exactly the same you would receive when accessing any web page with a browse navigator.

Scope	Read only
Type	Numeric
Initial value	0

1.6.4.2.1.5 TAsyncDownload:lConnected

Returns true if the connection is performed.

Scope	Read only
Type	Logical
Initial value	.F.

1.6.4.2.1.6 TAsyncDownload:lSuccess

Returns true if download has end with success.

Scope	Read only
Type	Logical

Initial value .F.

1.6.4.2.2 TAsyncDownload:Methods

■ Constructor ■ Standard

Type	Name
■	Cancel / Close
■	Run

1.6.4.2.2.1 TAsyncDownload:Cancel

Cancels download.

Type	Standard
Parameters	Ninguno
Return value	NIL

1.6.4.2.2.2 TAsyncDownload:Run

Starts download.

Type	Standard
Parameters	Ninguno
Return value	NIL

1.6.4.2.3 TAsyncDownload:Events

Name
OnConnect
OnDisconnect
OnError
OnTotalDnLoadInfo
OnTotalLengthInfo

1.6.4.2.3.1 TAsyncDownload:OnConnect

Event triggered when the connection is established.

Parameters	<oSender> Reference to the object that triggers the event
-------------------	---

Return value	NIL
---------------------	-----

1.6.4.2.3.2 TAsyncDownload:OnDisconnect

Event triggered when the connection is closed.

Parameters	<oSender> Reference to the object that triggers the event
Return value	NIL

1.6.4.2.3.3 TAsyncDownload:OnError

Event triggered when any error is produced.

Parameters	<oSender> Reference to the object that triggers the event <cError> Error description
Return value	NIL

1.6.4.2.3.4 TAsyncDownload:OnTotalDnLoadInfo

Event triggered periodically to indicate the bytes received.

Parameters	<oSender> Reference to the object that triggers the event <nTotalDnLoad> Total bytes received
Return value	NIL

1.6.4.2.3.5 TAsyncDownload:OnTotalLengthInfo

Event triggered at the beginning of the download to indicate the total length of bytes to receive.

Parameters	<oSender> Reference to the object that triggers the event <nTotalLength> Total bytes to receive
Return value	NIL

1.6.4.3 TAsynFileDownload

Asynchronous file download class through the Web via HTTP protocol.

Description:

With this class you can download any file through the Web on a second thread, leaving your application completely operative.

This class is descendant from TMTObject class which is not documented since is a internal class. The next class in his hierarchy is TWinObject.

Hierarchy Descendent of **TMTObject**
File name \source\Enterprise\AsynFileDownload.prg

1.6.4.3.1 TAsynFileDownload:Properties

■ Read only ■ Assignable ■ Design assignable ■ Run-time assignable

Scope	Name	Type	Initial value
■	cError	Character	""
■	cLocalFile	Character	""
■	cRemoteFile	Character	""
■	IConnected	Logical	.F.
■	ISuccess	Logical	.F.
■	nStatus	Numeric	0

1.6.4.3.1.1 TAsynFileDownload:cError

Possible error string.

Scope	Read only
Type	Character
Initial value	""

1.6.4.3.1.2 TAsynFileDownload:cLocalFile

Local file name.

Scope	Assignable before Run()
Type	Character
Initial value	""

1.6.4.3.1.3 TAsynFileDownload:cRemotelFile

Remote file name. Must be a HTTP link to the exact file.

Scope	Assignable before Run()
Type	Character
Initial value	""

1.6.4.3.1.4 TAsynFileDownload:IConnected

True if connected.

Scope	Read only
Type	Logical
Initial value	.F.

1.6.4.3.1.5 TAsynFileDownload:ISuccess

True if success.

Scope	Read only
Type	Logical
Initial value	.F.

1.6.4.3.1.6 TAsynFileDownload:nStatus

Final download state. A value of 200 means correct download.

Scope	Read only
Type	Numeric
Initial value	0

1.6.4.3.2 TAsynFileDownload:Methods

■ Constructor ■ Standard

Typ	Name
■	Cancel Close
■	Run

1.6.4.3.2.1 TAsynFileDownload:Cancel | Close

Cancel or closes the download process.

Type	Standard
Parameters	None
Return value	NIL

1.6.4.3.2.2 TAsynFileDownload:Run

Starts the download process.

Be aware that the process is completely asynchronous, so the control is immediately returned to the application. You will have to use any of the class events to perform any additional tasks.

Type	Standard
Parameters	None
Return value	NIL

1.6.4.3.3 TAsynFileDownload:Events

Name
OnConnect
OnDisconnect
OnError
OnTotalDnLoadInfo
OnTotalLengthInfo

1.6.4.3.3.1 TAsynFileDownload:OnConnect

Event that is triggered in the moment that starts a connection with a server.

Parameters	<oSender> Reference to the object that triggers the event
Return value	NIL

1.6.4.3.3.2 TAsynFileDownload:OnDisconnect

Event that is triggered in the moment that ends a connection with a server.

Parameters	<oSender> Reference to the object that triggers the event
Return value	NIL

1.6.4.3.3.3 TAsynFileDownload:OnError

Event that is triggered when a error is generated.

Parameters	<oSender> Reference to the object that triggers the event
Return value	NIL

1.6.4.3.3.4 TAsynFileDownload:OnTotalDnLoadInfo

Event that is triggered periodically indicating the total length received.

Parameters	<oSender> Reference to the object that triggers the event <nTotalLength> Total length received
Return value	NIL

1.6.4.3.3.5 TAsynFileDownload:OnTotalLengthInfo

Event that is triggered to indicate the the total length of the file.

Parameters	<oSender> Reference to the object that triggers the event <nTotalLength> File length
Return value	NIL

1.6.4.4 TBlatMail

This class permits to send Email messages. The messages are sent directly to the Email SMTP server without need of any other local Email client software. If you need a proof of every mail sent is better use the TMapi class as long your Email client supports that MAPI interface.

This class leans on the dynamic library [BLAT](#) which is a completely free component with absolutely no restrictions on its distribution. Is necessary that the library **Blad.dll** is included with your application. That library may be installed on the same application directory, the windows system directory (\Windows\System32) or any other directory included in the 'Path'.

Hierarchy Descendent of TComponent
File \source\BlatMail.prg

1.6.4.4.1 TBlatMail:Properties

■ Read only ■ Assignable ■ Design assignable ■ Run-time assignable

Scope	Name	Type	Initial Value
■	aAttachments	Array	{}
■	aBCC	Array	{}
■	aCC	Array	{}
■	aReceipts	Array	{}
■	cAddress	Character	""
■	cBody	Character	""
■	cCharset	Character	"ISO-8859-1"
■	cExtra	Character	""
■	cPassword	Character	""
■	cServer	Character	""
■	cSubject	Character	""
■	cUser	Character	""
■	IAuth *	Logic	.F.
■	ILog	Logic	.F.
■	IUndisclosedReceipts	Logic	.F.
■	IReceipt	Logic	.F.
■	nAttachAs	Numeric	baBINARY
■	nAuth	Numeric	bmAUTH

■	nEncoding	Numeri c	beNONE
■	nLastError	Numeri c	0
■	nPort	Numeri c	25
■	nPriority	Numeri c	bpNORMAL
■	nTimeout	Numeri c	0
■	nTries	Numeri c	0
■	nType	Numeri c	btTEXT

* Obsolete, use nAuth instead.

1.6.4.4.1.1 TBlatMail:aAttachments

Array with a list of files to send with the message.

Scope	Assignable
Type	Array
Initial Value	{}

1.6.4.4.1.2 TBlatMail:aBCC

Array with a list of Email receivers of type BCC (Blind Courtesy Copy).

Scope	Assignable
Type	Array
Initial Value	{}

Note: RFC addresses supported: Name <name@mail.com>

1.6.4.4.1.3 TBlatMail:aCC

Array with a list of Email receivers of type CC (Courtesy Copy)..

Scope	Assignable
Type	Array
Initial Value	{}

Note: RFC addresses supported: Name <name@mail.com>

1.6.4.4.1.4 TBlatMail:aReceipts

Array with a list of Email receivers.

Scope	Assignable
Type	Array
Initial Value	{}

Note: RFC addresses supported: Name <name@mail.com>

1.6.4.4.1.5 TBlatMail:cAddress

Sender Email address.

Scope	Assignable
Type	Character
Initial Value	""

Note: RFC addresses supported: Name <name@mail.com>

1.6.4.4.1.6 TBlatMail:cBody

Messagebody.

Scope	Assignable
Type	Character
Initial Value	""

1.6.4.4.1.7 TBlatMail:cCharset

Charset to use.

Scope	Assignable
Type	Character
Initial Value	"ISO-8859-1"

1.6.4.4.1.8 TBlatMail:cExtra

Any extra flag value to add following Blatmail documentation.

Scope	Assignable
Type	Character
Initial Value	""
Example	"- overwritelog" (to delete previous log file)

For further information consult the BLAT documentation. <http://www.blat.net/syntax/syntax.html>

1.6.4.4.1.9 TBlatMail:cPassword

Password to use in case SMTP server requires. You must also set to true the property nAuth.

Scope	Assignable
Type	Character
Initial Value	""

1.6.4.4.1.10 TBlatMail:cServer

SMTP server name or IP address.

Scope	Assignable
Type	Character
Initial Value	""

1.6.4.4.1.11 TBlatMail:cSubject

Message subject.

Scope	Assignable
Type	Character
Initial Value	""

1.6.4.4.1.12 TBlatMail:cUser

Name to use in case the SMTP server requires it. You must also set to true the property nAuth.

Scope	Assignable
Type	Character
Initial Value	""

Note: RFC addresses supported: Name <name@mail.com>

1.6.4.4.1.13 TBlatMail:IAuth

Set to true if the SMTP server requires authentication. You must also se the properties cUser and cPassword. A true value sets nAuth to bmAUTH, a false value sets nAuth to bmNONE

Obsolete, use nAuth instead.

Scope	Assignable
Type	Logic
Initial Value	.F.
See also	nAuth

1.6.4.4.1.14 TBlatMail:ILog

If true it creates a blat.log file on the application directory.

Scope	Assignable
Type	Logic
Initial Value	.F.

1.6.4.4.1.15 TBlatMail:IUndisclosedRecipients

If true sets the TO value to 'undisclosed recipients'.

Scope	Assignable
Type	Logic
Initial Value	.F.

1.6.4.4.1.16 TBlatMail:lReceipt

If true requires a notification receipt to receiver.

Scope	Assignable
Type	Logic
Initial Value	.F.

1.6.4.4.1.17 TBlatMail:nAttachAs

Set the attached files type.

Scope	Assignable
Type	Numeric
Initial Value	baBINARY
Possible values	baBINARY, baTEXT y baINLINETEXT

For further information consult the BLAT documentation. <http://www.blat.net/syntax/syntax.html>

1.6.4.4.1.18 TBlatMail:nAuth

Sets the authentication method.

Scope	Assignable
Type	Numeric
Initial Value	bmNONE
Possible values	bmNONE, bmAUTH, bmPOP3, bmIMAP

For further information consult the BLAT documentation. <http://www.blat.net/syntax/syntax.html>

1.6.4.4.1.19 TBlatMail:nEncoding

Sets the codification to use for attached files.

Scope	Assignable
Type	Numeric
Initial Value	beNONE
Possible values	beNONE, beBASE64, beUUENCODE

For further information consult the BLAT documentation. <http://www.blat.net/syntax/syntax.html>

1.6.4.4.1.20 TBlatMail:nLastError

Last generated error.

Scope	Read only
Type	Numeric
Initial Value	0
Possible values	<ul style="list-style-type: none"> -100: Blat.dll not found -101: SendMail function not found -2: The server actively denied our connection The mail server doesn't like the sender name -1: Unable to open SMTP socket SMTP get line did not return 220 Command unable to write to socket Server does not like To: address Mail server error accepting message data 0: Ok 1: File name (message text) not given Bad argument given 2: File (message text) does not exist 3: Error reading the file (message text) or attached file 4: File (message text) not of type FILE_TYPE_DISK 5: Error Reading File (message text) 12: -server or -f options not specified and not found in registry 13: Error opening temporary file in temp directory

For further information consult the BLAT documentation. <http://www.blat.net/syntax/syntax.html>

1.6.4.4.1.21 TBlatMail:nPort

Server TCP/IP port.

Scope	Assignable
Type	Numeric
Initial Value	25

1.6.4.4.1.22 TBlatMail:nPriority

Sets message priority.

Scope	Assignable
Type	Numeric
Initial Value	bpNORMAL
Possible values	bpNORMAL, bpLOW, bpHIGH

For further information consult the BLAT documentation. <http://www.blat.net/syntax/syntax.html>

1.6.4.4.1.23 TBlatMail:nTimeout

Sets the server response timeout.

Scope	Assignable
Type	Numeric
Initial Value	0 (not set)

1.6.4.4.1.24 TBlatMail:nTries

Sets the tries value to send the message.

Scope	Assignable
Type	Numeric
Initial Value	0 (not set)

1.6.4.4.1.25 TBlatMail:nType

Sets the type of text to be used in the message body.

Scope	Assignable
Type	Numeric
Initial Value	btTEXT
Possible values	btTEXT, btHTML

For further information consult the BLAT documentation. <http://www.blat.net/syntax/syntax.html>

1.6.4.4.2 TBlatMail:Methods

■ Constructor ■ Standard

Typ	Name
■	AddBCC
■	AddCC
■	AddReceipt
■	Create
■	Destroy
■	Send

1.6.4.4.2.1 TBlatMail:AddBCC

Adds a new receiver of type BCC (Blind Courtesy Copy).

Type	Standard
Parameters	<cEmail> Email account
Return value	NIL

1.6.4.4.2.2 TBlatMail:AddCC

Adds a new receiver of type CC (Courtesy Copy)..

Type	Standard
Parameters	<cEmail> Email account
Return value	NIL

1.6.4.4.2.3 TBlatMail:AddReceipt

Adds a new receiver to the message.

Type	Standard
Parameters	<cEmail> Email account
Return value	NIL

1.6.4.4.2.4 TBlatMail:Create

Constructs the TBlatmail object.

Type	Constructor
Parameters	<oParent> Reference to the container object
Return value	<Self> Self Reference

1.6.4.4.2.5 TBlatMail:Destroy

Destructs the TBlatmail object.

Type	Standard
Parameters	None
Return value	NIL

1.6.4.4.2.6 TBlatMail:Send

Sends the message.

Type	Standard
Parameters	None
Return value	<ISuccess> True if success

Samples:

- To delete information profile:

```
::Send( '-q -profile -delete "<default>" )
```

```
::Send( '-q -profile -delete " + cProfileName )
```
- To delete all profiles:

```
::Send( '-q -profile -delete "<all>" )
```

1.6.4.4.3 TBlatMail:Events

Name	
OnError	

1.6.4.4.3.1 TBlatMail:OnError

Event triggered when an error is produced. The return value is the same that the property `nLastError`. Consult that property for further information.

Parameters	<oSender> : Reference to the object that triggers the event
:	<nError> : Error value
Return value:	Nil

1.6.4.5 TCefBrowser

This control is itself a full Internet browser based on the product CEF (Chromium Embedded Framework) which allows any application to incorporate the Google Chrome browser.

This control requires a large part of the CEF binaries, which carries some weight (12.4 Mb compressed), so it was decided not to include them in the official Xailer install program and leave it for download in our download area:

CEF binaries download

Downloaded binaries should be copied into the application directory or into the '**Application\cef**' directory. The files should be copied with the same directory structure of the RAR file.

The absence of such binaries causes no runtime error, showing in this case a blank control. For the same reason when using the control from the IDE you can not navigate with it.

Hierarchy Inherits from TStdControl
File \source\CefBrowser.prg

1.6.4.5.1 TCefBrowser:Properties

■ read Only ■ Assignable ■ Design assignable ■ Run-time assignable

Scope	Name	Type	Initial value
■	cCachePath	Character	GetTempDir() + "\cef_cache"
■	cLocale	Character	en-US
■	nBorderStyle	Numeric	bvNONE
■	nHeight	Numeric	300
■	nWidth	Numeric	400

1.6.4.5.1.1 TCefBrowser:cCachePath

Storage directory for web browser temporary files.

Scope:	Design assignable
Type:	Character
Initial value:	GetTempDir() + "\cef_cache"

1.6.4.5.1.2 TCefBrowser:cLocale

Language in which certain messages are displayed by the browser, which are defined in the 'local' directory of the 'libcef.rar' file that incorporates all CEF binaries.

Scope:	Design assignable
Type:	Character
Initial value:	en-US

1.6.4.5.1.3 TCefBrowser:nBorderStyle

Border style.

Scope	Design assignable
Type	Numérico
Initial value	bvETCHED
Possible values	bvNONE, bvRAISED, bvSUNKEN, bvBUMP, bvETCHED, bvFLAT

1.6.4.5.1.4 TCefBrowser:nHeight

Control height.

Scope:	Assignable
Type:	Numeric
Initial value:	300

1.6.4.5.1.5 TCefBrowser:nWidth

Control width.

Scope:	Assignable
Type:	Numeric
Initial value:	400

1.6.4.5.2 TCefBrowser:Methods

■ Constructor ■ Standard ■ Only after Create()

Type	Name
■	CanGoBack
■	CanGoForward
■	ClearHistory
■	CloseDevTools
■	GetZoomLevel
■	GoBack
■	GoForward
■	HasDocument
■	HidePopup
■	IsPopupVisible
■	Navigate
■	SetText
■	SetZoomLevel
■	ShowDevTools
■	StopLoad

1.6.4.5.2.1 TCefBrowser:CanGoBack

True if the browser can go back in its browsing history.

Type	Only after Create()
Parameters	none
Return value	IResult

1.6.4.5.2.2 TCefBrowser:CanGoForward

True if the browser can go forward in its browsing history

Type	Only after Create()
Parameters	none

Return value	IResult
---------------------	---------

1.6.4.5.2.3 TCefBrowser:ClearHistory

Clears navigation history.

Type	Only after Create()
Parameters	None
Return value	NIL

1.6.4.5.2.4 TCefBrowser:CloseDevTools

Closes browse debugger window.

Type	Only after Create()
Parameters	None
Return value	NIL
See also	ShowDevTools

1.6.4.5.2.5 TCefBrowser:GetZoomLevel

Retrieves actual zoom .

Type	Only after Create()
Parameters	None
Return value	<nZoom>
See also	SetZoomLevel

1.6.4.5.2.6 TCefBrowser:GoBack

Goes back on navigation history.

Type	Only after Create()
Parameters	None
Return value	NIL

1.6.4.5.2.7 TCefBrowser:GoForward

Goes forward on navigation history.

Type	Only after Create()
Parameters	None
Return value	NIL

1.6.4.5.2.8 TCefBrowser:HasDocument

True if the browser has some content.

Type	Only after Create()
Parameters	None
Return value	<IResult>

1.6.4.5.2.9 TCefBrowser:HidePopup

Hide popup window.

Type	Only after Create()
Parameters	None
Return value	NIL

1.6.4.5.2.10 TCefBrowser:IsPopupVisible

True if popup window is visible

Type	Only after Create()
Parameters	None
Return value	<IResult>

1.6.4.5.2.11 TCefBrowser:Navigate

Navigate to a specific URL.

Type	Only after Create()
Parameters	<cURL> URL address

Return value	NIL
---------------------	-----

1.6.4.5.2.12 TCefBrowser:Reload

Reloads current URL.

Type	Only after Create()
Parameters	[<IgnoreCache> If true cache will be ignored. By default false
Return value	NIL

1.6.4.5.2.13 TCefBrowser:SetText

Enter HTML content directly into the browser discarding the existing text. The text must be a full html document, that is, with all its <header> <html> <body> and their closures.

Type	Only after Create()
Parameters	<cText> Text to include
Return value	NIL

1.6.4.5.2.14 TCefBrowser:SetZoomLevel

Sets zoom value. By default 1.00

Type	Only after Create()
Parameters	<nZoomLevel> New zoom value
Return value	NIL
See also	GetZoomLevel

1.6.4.5.2.15 TCefBrowser:ShowDevTools

Shows debugger window.

Type	Only after Create()
Parameters	None
Return value	NIL
See also	CloseDevTools

1.6.4.5.2.16 TCefBrowser:StopLoad

Stops URL load.

Type	Only after Create()
Parameters	None
Return value	NIL

1.6.4.5.3 TCefBrowser:Events

Name
OnAdressChange
OnAfterCreated
OnAuthCredentials
OnBeforePopup
OnBeforeResourceLoad
OnClose
OnDownloadComplete
OnDownloadData
OnDownloadFile
OnDownloadStart
OnLoadEnd
OnLoadError
OnLoadStart
OnNavStateChange
OnProtocolExecution
OnStatusMessage
OnTitleChange

1.6.4.5.3.1 TCefBrowser:OnAddressChange

Event that is triggered every time the URL is changed.

Parameters:	<oSender>: Reference to the object that triggers the event
	<cURL>: NewURL
Return value:	NIL

1.6.4.5.3.2 TCefBrowser:OnAfterCreated

Event that is triggered after the creation of the control and before any URL is loaded. The best place to set the first URL.

Parameters:	<oSender>: Reference to the object that triggers the event
Return value:	NIL

1.6.4.5.3.3 TCefBrowser:OnAuthCredentials

Event that occurs when an attempt for authorization credentials. Must set the cUserName and cPassword parameters and should return T. or F. as the user clicks 'OK' or 'Cancel'.

Parameters:	<oSender>: Reference to the object that triggers the event
	<cHost>: Hostname
	<nPort>: Port number
	<IProxy>: True if proxy
	<@cUsername>: Username
	<@cPassword>: Passowrd
Return value:	<IContinue> True if success

1.6.4.5.3.4 TCefBrowser:OnBeforePopup

Event that occurs before any URL is displayed in a popup window. If it returns false the operation is canceled.

Parameters:	<oSender>: Reference to the object that triggers the event
Return value:	<IAllow>

1.6.4.5.3.5 TCefBrowser:OnBeforeResourceLoad

Event that is triggered when charging any external source. Return false to cancel its charge, even though it is possible to modify the resource to load.

Parameters:	<oSender>: Reference to the object that triggers the event <@cURL>: Resource URL address <@cData>: New data to supplant the URL data with any internal data <@cMimeType>: Mime type
Return value:	<IAllow>

1.6.4.5.3.6 TCefBrowser:OnClose

Event that occurs when you close the control. Is the equivalent of the End () method on any other control. If it returns false does the operation is canceled.

Parameters:	<oSender>: Reference to the object that triggers the event
Return value:	<IAllow>

1.6.4.5.3.7 TCefBrowser:OnDownloadComplete

Event fired when the download finishes. Is your responsibility to close the file handle on this event.

Parameters:	<oSender>: Reference to the object that triggers the event <hFile>: File handle
Return value:	NIL
See also:	OnDownloadData OnDownloadFile OnDownloadStart

1.6.4.5.3.8 TCefBrowser:OnDownloadData

Event triggered every time a chunk of data is received. Should return true to continue the download.

Parameters:	<oSender>: Reference to the object that triggers the event <hFile>: File handle <cData>:
--------------------	---

	Data <nSize>: Data size
Return value:	<IAllow> True to continue with download
See also:	OnDownloadFile OnDownloadStart OnDownloadComplete

1.6.4.5.3.9 TCefBrowser:OnDownloadFile

Event that is fired when asking for a file download. Should provide a valid file handle or zero to cancel the process. If it returns NIL or the event is not assigned, Xailer will directly call the TFileSaveDlg() class to have a valid file handle.

Parameters:	<oSender>: Reference to the object that triggers the event <cFilename>: File name <cMimeType>: Mime type <nFileSize>: File size
Return value:	<hFile>: File handle
See also:	OnDownloadData OnDownloadStart OnDownloadComplete

1.6.4.5.3.10 TCefBrowser:OnDownloadStart

Event that is fired when asking for a file download. Should provide a valid file handle or zero to cancel the process. If it returns NIL or the event is not assigned, Xailer will directly call the TFileSaveDlg() class to have a valid file handle..

You should use this event to control the start download process with a progress bar.

Parameters:	<oSender>: Reference to the object that triggers the event <cFilename>: File name <cMimeType>: Mime type <nFileSize>: File size
Return value:	<hFile>:

	File handle
See also:	OnDownloadData OnDownloadFile OnDownloadComplete

1.6.4.5.3.11 TCefBrowser:OnLoadEnd

Event triggered when the URL load process finishes.

Parameters:	<oSender>: Reference to the object that triggers the event <nStatutsCode>: HTTP status code
Return value:	NIL

1.6.4.5.3.12 TCefBrowser:OnLoadError

Event triggered when a load URL error arises.

Parameters:	<oSender>: Reference to the object that triggers the event <nErrorCode>: HTTP status code <cURL>: URL that fired the error <@cErrorText>: Error message or personal URL
Return value:	NIL

1.6.4.5.3.13 TCefBrowser:OnLoadStart

Event that is triggered when a URL load starts.

Parameters:	<oSender>: Reference to the object that triggers the event
Return value:	NIL

1.6.4.5.3.14 TCefBrowser:OnNavStateChange

Event that is triggered every time the browser changes its state.

Parameters:	<oSender>: Reference to the object that triggers the event
	<ICanGoBack>: True if it can go back
	<ICanGoForward>: True if it can go forward
Return value:	NIL

1.6.4.5.3.15 TCefBrowser:OnProtocolExecution

Event triggered every time a new URL is loaded with a different protocol. For example, from http:// or https://, to mailto.

Parameters:	<oSender>: Reference to the object that triggers the event
	<cURL>: URL to load
	<@IAllowExecution>: True to allow operation
Return value:	<IDefault> Default logic value

1.6.4.5.3.16 TCefBrowser:OnStatusMessage

Event triggered every time the status message area changes.

Parameters:	<oSender>: Reference to the object that triggers the event
	<cMessage>: New message
	<nMessageType>: Internal message type
Return value:	NIL

1.6.4.5.3.17 TCefBrowser:OnTitleChange

Event triggered every time the window or tag title is changed.

Parameters:	<oSender>: Reference to the object that triggers the event <cTitle>: Newtitle
Return value:	NIL

1.6.4.6 TCdoMail

This class allows sending emails through the component CDO.Message, which is a COM component found in most Windows installations either because it is included in the operating system itself or because it was included when installing Microsoft Outlook.

You can also send the mail directly, directly accessing the mail server with the class TBlatMail.

Hierarchy	TComponent descendant
File	source\CdoMail.prg

1.6.4.6.1 TCdoMail:Properties

■ Read only ■ Assignable ■ Design assignable ■ Run-time assignable

Scope	Name	Type	Initial value
■	aAttachments	Array	{}
■	cBCC	Character	""
■	cCC	Character	""
■	cFrom	Character	""
■	cMessage	Character	""
■	cPassword	Character	""
■	cServer	Character	""
■	cSubject	Character	""
■	cTO	Character	""
■	cUser	Character	""
■	IAuthenticate	Logical	.F.
■	IGmailOptions	Logical	.F.
■	IHtml	Logical	.F.
■	IInstalled	Logical	.F.
■	ISSL	Logical	.F.
■	nPort	Numeric	0

1.6.4.6.1.1 TCdoMail:aAttachments

File names array to include with the Email

Scope	Assignable
Type	Array
Initial value	{}

1.6.4.6.1.2 TCdoMail:cBCC

Blind carbon copy address.

Scope	Assignable
Type	Character
Initial value	""

1.6.4.6.1.3 TCdoMail:cCC

Carbon copy address.

Scope	Assignable
Type	Character
Initial value	""

1.6.4.6.1.4 TCdoMail:cFrom

Message sender

Scope	Assignable
Type	Character
Initial value	""

1.6.4.6.1.5 TCdoMail:cMessage

Message text

Scope	Assignable
Type	Character
Initial value	""

1.6.4.6.1.6 TCdoMail:cPassword

Userpassword

Scope	Assignable
Type	Character
Initial value	""

1.6.4.6.1.7 TCdoMail:cServer

Server name that handles the message.

Scope	Assignable
Type	Character
Initial value	""

1.6.4.6.1.8 TCdoMail:cSubject

Message subject.

Scope	Assignable
Type	Character
Initial value	""

1.6.4.6.1.9 TCdoMail:cTO

Message receiver.

Scope	Assignable
Type	Character
Initial value	""

1.6.4.6.1.10 TCdoMail:cUser

User name sending the message.

Scope	Assignable
Type	Character
Initial value	""

1.6.4.6.1.11 TCdoMail:IAuthenticate

True if server needs authentication.

Scope	Assignable
Type	Logical
Initial value	.F.

1.6.4.6.1.12 TCdoMail:IGmailOptions

True to use default GMail options.

Scope	Assignable
Type	Logical
Initial value	.F.

1.6.4.6.1.13 TCdoMail:IHtml

If true the text message is in HTML format.

Scope	Assignable
Type	Logical
Initial value	.F.

1.6.4.6.1.14 TCdoMail:IInstaled

If true the CDO component is installed on the computer.

Scope	Sólo lectura
Type	Logical
Initial value	.F.

1.6.4.6.1.15 TCdoMail:ISSL

If true uses a SSL connection with the mail server.

Scope	Assignable
Type	Logical
Initial value	.F.

1.6.4.6.1.16 TCdoMail:nPort

TCP/IP port to use.

Scope	Assignable
Type	Numeric
Initial value	0

1.6.4.6.2 TCdoMail:Methods

■ Constructor ■ Standard

Type	Name
■	Send

1.6.4.6.2.1 TCdoMail:Send

Sends the message.

Type	Standard
Parameters	None
Return value	<ISuccess> Return if success

1.6.4.7 TGMailSend

This class allows the sending of emails through the **GMail** mail servers using the protocol **oAuth2** which is an authorization protocol that allows third parties (clients) to access content owned by a user (hosted in trusted applications, server Resources) without them having to handle or know the user's credentials. That is, third-party applications can access user-owned content, but these applications do not know the authentication credentials. With this scenario it becomes necessary an authorization protocol that is based on the following aspects:

- Owner of the email account that will be used to send the mail on your behalf
- The application that requests to send a mail on behalf of the owner
- Service provider (resources and authorization), in our case **GMail**

The first step in sending is to create a 'Google API Console project' by going to the following Internet address: <https://console.developers.google.com/projectselector/apis/library>. From there you should create a *new project* and once created, select the menu option (in the sidebar) of '**Credentials**'.

Then select the option to create credential for **oAuth2 client ID** and when asked the Type of application, enter '**Other**'. For the first credential you will need to include some information that will be displayed on the authorization screen **oAuth2**.

For each client **ID** for **oAuth2** Google will supply you with two data, which are the '**Client ID**' and '**Secret ID**'. These two values should be entered into the cClientID and cSecretID properties of the **TGMailSend** control.

However, you need an additional data that is the '**Token**'. The first time you try to send an email message through **TGMailSend**, the program will show the user the classic **GMail** login page asking the user for the login credentials and indicating that a certain service (your project) wants to send Email on your behalf. If the user correctly enters his credentials and also authorizes the 'project' to send mail on his behalf, the mail should be sent without problems, but also a '**token**' will be generated and stored in the cToken property of the control. If you are careful to assign the cToken property with that token before sending the mail, you will not be prompted for your credentials again. Note that the '**token**' may expire and therefore a shipping error may occur. Circumstance that must be controlled to obtain a new token,

The last step is to enable the use of **GMail API** to the newly created project, for this you must select the '**Control Panel**' menu option and press the '**Enable API**' button and select the '**GMail API**' link and finally press the '**Enable**' button.

You can also send the mail directly and easily, directly accessing the mail server with the class TBlatMail.

Hierarchy TComponent descendant
File Notavailable

1.6.4.7.1 TGMailSend:Properties

■ Read only ■ Assignable ■ Design assignable ■ Run-time assignable

Scope	Name	Type	Initial value
■	cBCC	Carácter	""
■	cCC	Carácter	""
■	cClientId	Carácter	""
■	cError	Carácter	""
■	cLanguage	Carácter	""
■	cMessageld	Carácter	""
■	cSecretId	Carácter	""
■	cSubject	Carácter	""
■	cText	Carácter	""

■	cTo	Carácter	""
■	cToken	Carácter	""
■	lHtml	Lógico	.F.
■	nError	Numérico	0

1.6.4.7.1.1 TGMailSend:cBCC

Black carbon copy receiver for the message.

Scope	Assignable
Type	Carácter
Initial value	""

1.6.4.7.1.2 TGMailSend:cCC

Carbon copy receiver for the message.

Scope	Assignable
Type	Carácter
Initial value	""

1.6.4.7.1.3 TGMailSend:cClientId

Client identifier used by **GMail** service. See introduction.

Scope	Assignable
Type	Carácter
Initial value	""

1.6.4.7.1.4 TGMailSend:cError

Description of last error produced.

Scope	Read only
Type	Carácter
Initial value	""

1.6.4.7.1.5 TGMailSend:cLanguage

Language identifier.

Scope	Assignable
Type	Carácter
Initial value	""

1.6.4.7.1.6 TGMailSend:cMessageId

Last sent message identifier.

Scope	Read only
Type	Carácter
Initial value	""

1.6.4.7.1.7 TGMailSend:cSecretId

Secret identifier used by **GMail** service. See introduction.

Scope	Assignable
Type	Carácter
Initial value	""

1.6.4.7.1.8 TGMailSend:cSubject

Text subject for the message.

Scope	Assignable
Type	Carácter
Initial value	""

1.6.4.7.1.9 TGMailSend:cText

Text message.

Scope	Assignable
Type	Carácter
Initial value	""

1.6.4.7.1.10 TGMailSend:cTo

Message receiver.

Scope	Assignable
Type	Carácter
Initial value	""

1.6.4.7.1.11 TGMailSend:IHtml

True if text message is in HTML.

Scope	Assignable
Type	Lógico
Initial value	.F.

1.6.4.7.1.12 TGMailSend:cToken

Token used by **GMail** service. See introduction.

Scope	Assignable
Type	Carácter
Initial value	""

1.6.4.7.1.13 TGMailSend:nError

Last error code produced.

Scope	Read only
Type	Numérico
Initial value	0

1.6.4.7.2 TGMailSend:Methods

■ Constructor ■ Standard

Type	Name
■	Authorize

■ Send

1.6.4.7.2.1 TGMailSend:Authorize

Authorizes the mensaje. Retrieves the '**token**' for further sends in case it does not exist. See introduction.

Type	Standard
Parameters	None
Return value	<ISuccess> True if success

1.6.4.7.2.2 TGMailSend:Send

Sends the message.

Type	Standard
Parameters	Ninguno
Return value	<ISuccess> True if success

1.6.4.8 TPop3Client

Class for receiving e-mail via POP3 protocol.

This class is mainly intended for the treatment of emails M2M, that is to say, machine-to-machine, as for example could be the reception of a new order in a Prestashop server or equivalent. Usually M2M mail messages always have the same structure and it is easy to obtain from them the relevant information for our business that can be perfectly treated by our management application.

In order to use the class it is necessary first to indicate all the parameters of the connection that are: cServer, cUsername and cPassword.

The received mails are deposited directly in the matrix of literals aMails and are stored as they are received, that is to say, without absolutely no treatment. It is necessary to use the methods included in the class to extract information from them, such as: recipient, sender, subject, message body in text or HTML format and attached files.

Note: This POP3 client does not accept SSL connections, like for example, *Gmail*. In case you need it we recommend to use the stunnel utility in conjunction with this class. Its use is really simple.

Hierarchy **TComponent** descendant
File \source\Pop3Client.prg

1.6.4.8.1 TPop3Client:Properties

■ Read only ■ Assignable ■ Design assignable ■ Run-time assignable

Scope	Name	Type	Initial value
■	aMails	Array	{}
■	cLastMail	Character	""
■	cServer	Character	""
■	cPassword	Character	""
■	cUsername	Character	""
■	nPort	Numeric	110
■	nRcvBuSize	Numeric	1024
■	nTimeOut	Numeric	1000
■	lConnected	Logical	.F.
■	lDelOnRead	Logical	.F.
■	lDigest	Logical	.F.

1.6.4.8.1.1 TPop3Client:aMails

Array of literals with all received emails. To obtain specific information of the same you must use the methods of support that includes the class for it.

Scope	Read only
Type	Array
Initial value	{}
See also	MailFrom, MailTo, MailSubject, MailText, MailHtml, MailFiles

1.6.4.8.1.2 TPop3Client:cLastMail

Last received email. To obtain specific information of the same you must use the methods of support that includes the class for it.

Scope	Read only
Type	Character
Initial value	""
See also	MailFrom, MailTo, MailSubject, MailText, MailHtml, MailFiles

1.6.4.8.1.3 TPop3Client:cServer

POP3 server name.

Scope	Assignable (before connection)
Type	Character
Initial value	""
See also	cUsername, cPassword

1.6.4.8.1.4 TPop3Client:cPassword

POP3 server user password.

Scope	Assignable (before connection)
Type	Character
Initial value	""
See also	cUsername, cServer

1.6.4.8.1.5 TPop3Client:cUsername

POP3 server user name.

Scope	Assignable (before connection)
Type	Character
Initial value	""
See also	cPassword, cServer

1.6.4.8.1.6 TPop3Client:nPort

POP3 server port.

Scope	Assignable (before connection)
Type	Numeric
Initial value	110

1.6.4.8.1.7 TPop3Client:nRcvBufSize

Reception buffer size.

Scope	Assignable (before connection)
Type	Numeric
Initial value	1024

1.6.4.8.1.8 TPop3Client:nTimeOut

Timeout to connect.

Scope	Assignable (before connection)
Type	Numeric
Initial value	10000

1.6.4.8.1.9 TPop3Client:lConnected

Connection state.

Scope	Assignable
Type	Logical
Initial value	.F.
See also	Open, Close

1.6.4.8.1.10 TPop3Client:lDelOnRead

If true the mailbox will be emptied before retrieve new messages.

Scope	Assignable
Type	Logical
Initial value	.F.

1.6.4.8.1.11 TPop3Client:lDigest

If true the connection will be made using the APOP command that prevents the user's Name and password from being sent as plain text. Not all POP3 servers support it.

Scope	Assignable
--------------	------------

Type	Logical
Initial value	.F.

1.6.4.8.2 TPop3Client:Methods

Constructor
 Standard
 Only after Create()

Type	Name
<input type="checkbox"/>	Close
<input type="checkbox"/>	Count
<input type="checkbox"/>	Delete
<input type="checkbox"/>	List
<input type="checkbox"/>	MailFiles
<input type="checkbox"/>	MailFrom
<input type="checkbox"/>	MailHtml
<input type="checkbox"/>	MailText
<input type="checkbox"/>	MailTo
<input type="checkbox"/>	MailSubject
<input checked="" type="checkbox"/>	New
<input type="checkbox"/>	Open
<input type="checkbox"/>	Retrieve
<input type="checkbox"/>	RetrieveAll

1.6.4.8.2.1 TPop3Client:Close

Close connection with the server.

Type	Standard
Parameters	None
Return value	NIL

1.6.4.8.2.2 TPop3Client:Count

Returns the total number of Email received on last connection.

Type	Standard
Parameters	None
Return value	<nCount> Email total

1.6.4.8.2.3 TPop3Client:Delete

Deletes a message from the mailbox.

Type	Standard (with connection established)
Parameters	<nId> Email identifier
Return value	<ISuccess> True if success
See also	List

1.6.4.8.2.4 TPop3Client:List

Returns an Array with the identifiers of all the emails in the mailbox.

Type	Standard (with connection established)
Parameters	None
Return value	<aList> Email list

1.6.4.8.2.5 TPop3Client:MailFiles

Retrieves files included in a Email message.

Type	Standard
Parameters	[<cMail>] String with all the mail information. By default cLastMail value [<cDirectory>] Directory to download all files. By default, temporary system folder
Return value	<aFiles> Names of files received

1.6.4.8.2.6 TPop3Client:MailFrom

Email sender.

Type	Standard
Parameters	[<cMail>] String with all the mail information. By default cLastMail value
Return value	<cFrom> Sender name

1.6.4.8.2.7 TPop3Client:MailHtml

Text message in HTML format.

Type	Standard
Parameters	[<cMail>] String with all the mail information. By default cLastMail value
Return value	<cHtml> Text in HTML format

1.6.4.8.2.8 TPop3Client:MailText

Body message as plain text.

Type	Standard
Parameters	[<cMail>] String with all the mail information. By default cLastMail value
Return value	<cTxt> body message

1.6.4.8.2.9 TPop3Client:MailTo

Email Receiver

Type	Standard
Parameters	[<cMail>] String with all the mail information. By default cLastMail value
Return value	<cTo> Receiver name

1.6.4.8.2.10 TPop3Client:MailSubject

Email subject.

Type	Standard
Parameters	[<cMail>] String with all the mail information. By default cLastMail value
Return value	<cSubject> Subject

1.6.4.8.2.11 TPop3Client:New

Class constructor.

Type	Constructor
Parameters	[<oParent> Parent object
Return value	<Self> A reference to the new created object

1.6.4.8.2.12 TPop3Client:Open

Opens connection with the server.

Type	Standard
Parameters	[<IDigest> If true it will use APOP protocol. By default, IDigest property value
Return value	<ISuccess> True if success

1.6.4.8.2.13 TPop3Client:Retrieve

Retrieves one email.

Type	Standard (with connection established)
Parameters	<nId> Email identifier [<IDelete> If true the message will be deleted from the mailbox. By default, DelOnRead property value
Return value	<cMail> String with all the mail information

1.6.4.8.2.14 TPop3Client:RetrieveAll

Retrieves all the Emails from the mailbox.

Type	Standard (with connection established)
Parameters	[<IDelete>

	If true the message will be deleted from the mailbox. By default, DelOnRead property value
Return value	<aMails> Array of strings with all the mail information

1.6.4.8.3 TPop3Client:Events

Name	
	OnNewMail
	OnTimeOut

1.6.4.8.3.1 TPop3Client:OnNewMail

Event triggered when an Email is received.

Parameters	<oSender> Object that triggers the event <cMail> String with all the mail information <nMail> Number of the email received <nTotal> Total of email received
Return value	NIL

1.6.4.8.3.2 TPop3Client:OnTimeOut

Event fired when a timeout is produced.

Parameters	<oSender> Object that triggers the event
Return value	NIL

1.6.4.8.4 TPop3Client:Functions

Name	
	ReadPop3Asyn

1.6.4.8.4.1 ReadPop3Asyn

Retrieves all the Email asynchronously. Requires the mult-thread library **hbvmmt** instead of **hbvm**.

Parameters	<oPop3Client> TPop3Client object <bOnEnd> Code block to execute when all the Emails have been received.
Return value	<oThread> Thread

1.6.4.9 TInternet

Class to control connections and access to internet.

Description:

The TInternet class allows to manipulate the connections and access to internet. It is important to include the "WinINet.api" header file when the properties of this class are used, because includes most of their definitions.

Hierarchy	Inherits from TWinObject
File name	\source\Internet.prg

1.6.4.9.1 TInternet:Properties

■ read Only
 ■ Assignable
 ■ Design assignable
 ■ Run-time assignable

Scope	Name	Type	Initial value
■	cAgent	Character	"XailerBrowser"
■	cPassword	Character	""
■	cUser	Character	""
■	hSession	Numeric	0
■	nLastError	Numeric	0
■	nPort	Numeric	INTERNET_DEFAULT_HTTP_PORT
■	nService	Numeric	INTERNET_SERVICE_HTTP
■	nTimeOut	Numeric	3000

1.6.4.9.1.1 TInternet:cAgent

Application identifier in HTTP connections.

Scope	Assignable
Type	Character
Initial value	""XailerBrowser"

1.6.4.9.1.2 TInternet:cPassword

Password to access the resource.

Scope	Assignable
Type	Character
Initial value	""

1.6.4.9.1.3 TInternet:cUser

Username.

Scope	Assignable
Type	Character
Initial value	""

1.6.4.9.1.4 TInternet:hSession

Handle for the session open with Connect()

Scope	readOnly
Type	Numeric
Initial value	0

1.6.4.9.1.5 TInternet:nLastError

Error code if there were an error after calling any method.

Scope	readOnly
Type	Numeric
Initial value	0

1.6.4.9.1.6 TInternet:nPort

Port used for the connection. By default: HTTP.

Scope	Assignable
Type	Numeric
Initial value	INTERNET_DEFAULT_HTTP_PORT

Possible values	INTERNET_DEFAULT_HTTP_PORT, INTERNET_DEFAULT_HTTPS_PORT, INTERNET_DEFAULT_FTP_PORT, INTERNET_DEFAULT_GOPHER_PORT, INTERNET_DEFAULT SOCKS_PORT
------------------------	---

Note: You need to include the header file "WinINet.api".

1.6.4.9.1.7 TInternet:nService

Service type to be accessed. Default: HTTP.

Scope	Assignable
Type	Numeric
Initial value	INTERNET_SERVICE_HTTP
Possible values	INTERNET_SERVICE_HTTP, INTERNET_SERVICE_FTP, INTERNET_SERVICE_GOPHER

Note: You need to include the header file "WinINet.api".

1.6.4.9.1.8 TInternet:nTimeOut

Time in milliseconds after all the petitions will be canceled if there is not answer.

Scope	Assignable
Type	Numeric
Initial value	3000

1.6.4.9.2 TInternet:Methods

■ Constructor ■ Standard

Typ Name	Standard
AddRequestHeader	■
AttemptConnect	■
CanonicalizeUrl	■
CheckConnection	■
Close	■
CloseFile	■
CloseRequest	■
CloseURL	■
ConfirmZoneCrossing	■

■ Connect
■ Destroy
■ ErrorDialog
■ GetConnectedState
■ GetErrorDescription
■ GetFileSize
■ GetRequest
■ GetVersion
■ Open
■ OpenRequest
■ OpenURL
■ PostRequest
■ QueryInfo
■ ReadFile
■ SendRequest
■ SetProxyData
■ SetTimeOut
■ WriteFile

1.6.4.9.2.1 TInternet:AddRequestHeader

Adds HTTP headers to GetRequest/PostRequest calls.

Type	Standar
Parameters	<hRequest> Handle returned byr GetRequest or PostRequest. <cData> List of headers separated by CRLF. For example: "Content-Type: text/html" + CRLF <nFlags> Flags. See HTTP_ADDREQ_FLAG_ on Wininet.ch
Return value	<ISuccess> .T. if success

1.6.4.9.2.2 TInternet:AttemptConnect

Starts a connection to Internet.

Type	Standard
Parameters	None
Return value	NIL

1.6.4.9.2.3 TInternet:CanonicalizeUrl

Converts illegal characters and spaces to escape sequences on URL strings.

Type	Standard
Parameters	<cURL> URL string
Return value	<cURL> Converted URL string

1.6.4.9.2.4 TInternet:CheckConnection

Checks that there is a connection trying to ping the indicated URL.

Type	Standard
Parameters	<cURL> Resource address <Protocol>://<server>
Return value	<ISuccess> .T. if there is connection

1.6.4.9.2.5 TInternet:Close

Closes the Internet session and triggers the OnClose event.

Type	Standard
Parameters	None
Return value	<ISuccess> .T. if the session ended successfully

1.6.4.9.2.6 TInternet:CloseFile

Closes a remote file.

Type	Standard
Parameters	<hFile> File handle
Return value	<ISuccess> .T. if the file was closed successfully

1.6.4.9.2.7 TInternet:CloseRequest

Closes a petition open with GetRequest/PostRequest.

Type	Standard
Parameters	<hRequest> Petition handle
Return value	NIL

1.6.4.9.2.8 TInternet:CloseURL

Closes a remote resource previously open with OpenURL.

Type	Standard
Parameters	<hURL> Handle for the remote resource
Return value	<ISuccess> .T. if the resource was closed successfully

1.6.4.9.2.9 TInternet:ConfirmZoneCrossing

Confirms that it will change from a secure URL to other secure URL.

Type	Standard
Parameters	<cPrevURL> Accessed URL <cNewURL> URL to be accessed
Return value	<ISuccess> .T. if the user authorizes the action

1.6.4.9.2.10 TInternet:Connect

Connects to a remote server.

Type	Standard
Parameters	<cServer> Server name
Return value	<ISuccess> .T. if the session started successfully

1.6.4.9.2.11 TInternet:Destroy

Closes the connection and releases system resources.

Type	Standard
Parameters	None
Return value	NIL

1.6.4.9.2.12 TInternet:ErrorDialog

Shows an standard dialog with the error description, if the system finds a Standard dialog to show the error.

Type	Standard
Parameters	<nError> Optional error code. Default: nLastError.
Return value	NIL

1.6.4.9.2.13 TInternet:GetConnectedState

Checks if there is an Internet connection (it can be not active).

Type	Standard
Parameters	[<nType>] Type of connection to check. It can be the OR conjunction of any of the following : <ul style="list-style-type: none">• INTERNET_CONNECTION_CONFIGURED (0x40) Local system has a valid connection to the Internet, but it might or might not be currently connected• INTERNET_CONNECTION_LAN (0x02) Local system uses a local area network to connect to the Internet• INTERNET_CONNECTION_MODEM (0x01) Local system uses a modem to connect to the Internet• INTERNET_CONNECTION_OFFLINE (0x20) Local system is in offline mode.• INTERNET_CONNECTION_PROXY (0x04) Local system uses a proxy server to connect to the Internet

	<ul style="list-style-type: none"> INTERNET_RAS_INSTALLED (0x10) Local system has RAS installed
Return value	<ISuccess> .T. if there is a connection of the type requested

1.6.4.9.2.14 TInternet:GetErrorDescription

Returns the standard error description, if there as any after a method call.

Type	Standard
Parameters	None
Return value	<cError> Error message

1.6.4.9.2.15 TInternet:GetFileSize

Returns the size of a remote file.

Type	Standard
Parameters	<hFile> File handle
Return value	<nSize> File size

1.6.4.9.2.16 TInternet:GetRequest

Sends a data petition to a remote resource. It is equivalent to a GET type call from any browser.

Type	Standard
Parameters	<cURL> Resource address
Return value	<hRequest> Petition handle

1.6.4.9.2.17 TInternet:GetVersion

HTTP protocol version used.

Type	Standard
Parameters	None

Return value	<aVersion> HTTP protocol version used. It is equivalent to the aVersion property
---------------------	--

1.6.4.9.2.18 TInternet:Open

Starts an Internet session and triggers the OnOpen event.

Type	Standard
Parameters	<nAccessType> Access Type. It corresponds with the INTERNET_OPEN_TYPE_constants. Optional <cProxy> Proxy server name. Optional. <cProxyByPass> Server name that will not be routed through the proxy server <nFlags> Access options. If corresponds to the INTERNET_FLAG_constants. Optional
Return value	<ISuccess> .T. if the session started successfully

Note: You need to include the header file "WinINet.api".

1.6.4.9.2.19 TInternet:OpenRequest

Creates a HTTP request.

Type	Standard
Parameters	[<cVerb>] GET or POST. By default GET [<cURL>] Resource name [<nFlags>] Constants defined in WinINet.api. By default INTERNET_FLAG_KEEP_CONNECTION [<cVersion>] HTTP Version [<cReferer>] URL of the document from which the URL in the request (cUrl) was obtained
Return value	<hRequest> Request handle

Note: You need to include the header file "WinINet.api".

1.6.4.9.2.20 TInternet:OpenURL

Opens a remote resource to be read or downloaded.

Type	Standard
Parameters	<cURL> Resource address to be open <nFlags> Options to open the resource
Return value	<hURL> Resource handle

1.6.4.9.2.21 TInternet:QueryInfo

Retrieves headers information about a request created by OpenRequest.

Type	Standard
Parameters	<hRequest> Request handle <nFlags> Constants HTTP_QUERY_ defined in WinINet.api. <@cInfo> Headers info. Passed by reference. Filled on output
Return value	<ISuccess> True if success

1.6.4.9.2.22 TInternet:PostRequest

Prepares a remote resource to receive data. It is equivalent to a POST type call from any browser.

Type	Standard
Parameters	<cURL> Resource address <cData> Data to be sent to the server
Return value	<hRequest> Petition handle

1.6.4.9.2.23 TInternet:ReadFile

Recovers a file or parts of it (if it is supported by the server).

Type	Standard
Parameters	<hFile> File handle <cBuffer> Variable with holds the received data <nSize> Length in bytes of received data
Return value	<ISuccess> .T. if data has been received successfully

1.6.4.9.2.24 TInternet:SendRequest

Sends a request created by OpenRequest.

Type	Standard
Parameters	<hRequest> Request handle [<cHeaders>] Additional header list to send with the request [<cOptionalData>] Additional data to be sent after the headers.
Return value	<ISuccess> True if success

1.6.4.9.2.25 TInternet:SetProxyData

Sets proxy connections login credentials.

Type	Standard
Parameters	<cUser> User name <cPassword> User password
Return value	<ISuccess> True if success

1.6.4.9.2.26 TInternet:SetTimeOut

Time in milliseconds after all the petitions will be canceled if there is not answer.

Type	Standard
Parameters	<nTimeOut> Time in milliseconds
Return value	NIL

1.6.4.9.2.27 TInternet:WriteFile

Writes a file or parts of it (if it is supported by the server).

Type	Standard
Parameters	<hFile> File handle <cBuffer> Variable with the data to be written in te file <nSize> Buffer size in bytes
Return value	<ISuccess> .T. if the file was written successfully

1.6.4.9.3 TInternet:Events

Name
OnClose
OnConnect
OnGetData
OnOpen

1.6.4.9.3.1 TInternet:OnClose

Event that is produced when a session is closed.

Parameters	<oSender> Reference to the object that triggers the event
Return value	NIL

1.6.4.9.3.2 TInternet:OnConnect

Event that is triggered in the moment that starts a connection with a server.

Parameters	<oSender> Reference to the object that triggers the event
Return value	NIL

1.6.4.9.3.3 TInternet:OnGetData

Event that is triggered when a session is closed.

Parameters	<oSender> Reference to the object that triggers the event <cData> Information that is received after the SendGet method is called
Return value	NIL

1.6.4.9.3.4 TInternet:OnOpen

Event that is produced when a session in open.

Parameters	<oSender> Reference to the object that triggers the event
Return value	NIL

1.6.4.10 TFTP

Class to control FTP sessions.

Description:

Controls FTP sessions. It is important to include the "**WinINet.api**" header file, because it includes the definitions needed to use the properties of this class.

Hierarchy	Inherits from TInternet
File name	\source\Ftp.prg

1.6.4.10.1 TFTP:Properties

read Only
 Assignable
 Design assignable
 Run-time assignable

Scope	Name	Type	Initial value
<input type="checkbox"/>	IPassive	Logic	.F.
<input type="checkbox"/>	nBuffer	Numeric	32768
<input type="checkbox"/>	nTransferType	Numeric	ftpBinary

1.6.4.10.1.1 TFTP:IPassive

Indicates if it is needed to use a passive connection.

Scope	Assignable
Type	Logic
Initial value	.F.

1.6.4.10.1.2 TFTP:nBuffer

Indicates buffer size for upload and download operations.

Scope	Assignable
Type	Numeric
Initial value	32768 (32 Kb)

1.6.4.10.1.3 TFTP:nTransferType

Transfer connection type.

Scope	Assignable
Type	Numeric
Initial value	ftpBinary
Possible values	ftpBinary, ftpASCII

1.6.4.10.2 TFTP:Methods

Constructor
 Standard

Type	Name
<input type="checkbox"/>	CloseFile
<input type="checkbox"/>	Command
<input type="checkbox"/>	Create
<input type="checkbox"/>	CreateDirectory

■	DeleteFile
■	Directory
■	DownloadFile
■	GetCurrentDirectory
■	GetFile
■	GetFileSize
■	OpenFile
■	OpenFileRead
■	OpenFileWrite
■	PutFile
■	RemoveDirectory
■	RenameFile
■	SetCurrentDirectory
■	UploadFile

1.6.4.10.2.1 TFTP:Create

Class constructor.

Type	Constructor
Parameters	oParent
Return value	Self reference (Self)

1.6.4.10.2.2 TFTP:CreateDirectory

Creates a directory in the FTP server.

Type	Standard
Parameters	<cDirectory> Remote directory name
Return value	<ISuccess> .T. if the process is successful

1.6.4.10.2.3 TFTP:CloseFile

Closes a file open with OpenFile in the server.

Type	Standard
Parameters	<hFile> Remote file handle
Return	<ISuccess>

value	.T. if the process is successful
--------------	----------------------------------

1.6.4.10.2.4 TFTP:Command

Sends a command to the FTP server.

Type	Standard
Parameters	<cCmd> Command to be executed in the server <lResponse> Waits for the server answer. Default: .T.
Return value	<lSuccess> .T. if the server answered the petition

If the command is executed properly, the OnCommand event is triggered.

1.6.4.10.2.5 TFTP>DeleteFile

Deletes a file from the FTP server.

Type	Standard
Parameters	<cFile> Remote file name
Return value	<lSuccess> .T. if the process is successful

1.6.4.10.2.6 TFTP:Directory

Returns a list with the directory content.

Type	Standard
Parameters	<cMask> Mask for the files to be included in the list
Return value	<aFiles> File directory list. Every element has the following format: [1] Filename [2] Size in bytes [3] Date [4] Time [5] 3 position string with the file attributes. The format is: [A] File [D] Directory

[R] Read Only

NOTE: Some FTP server will not provide the attributes or they dot get them with the right values, you can check if it is a directory if the file size is 0.

1.6.4.10.2.7 TFTP:DownloadFile

Downloads a file from the FTP server.

Type	Standard
Parameters	<cRemoteFile> Remote file name <cLocalFile> Local file name
Return value	<ISuccess> .T. if the process is successful

1.6.4.10.2.8 TFTP:GetCurrentDirectory

Recovers the working directory in the FTP server.

Type	Standard
Parameters	None
Return value	<cDirectory> Remote directory name

1.6.4.10.2.9 TFTP:GetFile

Recovers a file from the FTP server and saves it locally.

Type	Standard
Parameters	<cFtpFile> Remote file name <cLocalFile> Local filename <IFail> If it is .T. and the file exists, the file petition fails. <nAttributes> Local file attributes
Return value	<ISuccess> .T. if the process is successful

1.6.4.10.2.10 TFTP:GetFileSize

Recovers the file size from the FTP server.

Type	Standard
Parameters	<hFile> File handle got with OpenFile
Return value	<nBytes> File size specified in bytes

1.6.4.10.2.11 TFTP:OpenFile

Opens a file to read/write in the FTP server.

Type	Standard
Parameters	<cFile> Remote file name <nAccess> Access type. It can be GENERIC_READ o GENERIC_WRITE
Return value	<hFile> Remote file handle

1.6.4.10.2.12 TFTP:OpenFileRead

Opens a file to read in the FTP server.

Type	Standard
Parameters	<cFile> Remote file name
Return value	<hFile> Remote file handle

1.6.4.10.2.13 TFTP:OpenFileWrite

Opens a file to write in the FTP server.

Type	Standard
Parameters	<cFile> Remote file name
Return value	<hFile> Remote file handle

1.6.4.10.2.14 TFTP:PutFile

Sends a file to the FTP server.

Type	Standard
Parameters	<cFile> Local filename <cFtpFile> Remote file name
Return value	<ISuccess> .T. if the process is successful

1.6.4.10.2.15 TFTP:RemoveDirectory

Deletes a directory in the FTP server.

Type	Standard
Parameters	<cDirectory> Remote directory name
Return value	<ISuccess> .T. if the process is successful

1.6.4.10.2.16 TFTP:RenameFile

Renames a file in the FTP server.

Type	Standard
Parameters	<cFile> Remote file name <cNewFile> New remote file name
Return value	<ISuccess> .T. if the process is successful

1.6.4.10.2.17 TFTP:SetCurrentDirectory

Changes the directory in the FTP server.

Type	Standard
Parameters	<cDirectory> Remote directory name

Return value	<ISuccess> .T. if the process is successful
---------------------	---

1.6.4.10.2.18 TFTP:UploadFile

Uploads a file to the FTP server.

Type	Standard
Parameters	<cLocalFile> Local file name <cRemoteFile> Remote file name
Return value	<ISuccess> .T. if the process is successful

1.6.4.10.3 TFTP:Events

Name
OnCommand
OnComplete
OnDirectory
OnError
OnProgress
OnStart

1.6.4.10.3.1 TFTP:OnCommand

Event that is triggered after a instruction is sent to the FTP server.

Parameters	<oSender> Reference to the object that triggers the event <hFile> File handle that contains the answer from the server. To get the server answer, you need to call to ReadFile.
Return value	NIL

1.6.4.10.3.2 TFTP:OnComplete

Event that is triggered after a transfer file with methods UploadFile or DownloadFile.

Parameters	<oSender> Reference to the object that triggers the event
Return value	NIL

1.6.4.10.3.3 TFTP:OnDirectory

Event that is triggered every time a file is retrieved on Directory method.

Parameters	<oSender> Reference to the object that triggers the event <cFile> Retrieved file name
Return value	NIL

1.6.4.10.3.4 TFTP:OnError

Event that is triggered after a error is produced.

Parameters	<oSender> Reference to the object that triggers the event <nError> Error code <cError> Error description
Return value	NIL

1.6.4.10.3.5 TFTP:OnProgress

Event that is triggered to indicate the progress of file transfer with methods UploadFile or DownloadFile.

Parameters	<oSender> Reference to the object that triggers the event <nBytes> Total bytes transfered
Return value	NIL

1.6.4.10.3.6 TFTP:OnStart

Event that is triggered to indicate the start of file transfer with methods UploadFile or DownloadFile

Parameters	<oSender> Reference to the object that triggers the event <nTotalBytes> Total bytes to transfer
Return value	NIL

1.6.4.11 TMapi

This class encapsulates the Windows API for simple messaging.

Description:

Thought this class it is possible to send e-mails through the Windows API message. It is needed that the API is present in the operating system and not always is possible.

Hierarchy	Inherits from TComponent
File name	\source\Mapi.prg

1.6.4.11.1 TMapi:Properties

■ read Only ■ Assignable ■ Design assignable ■ Run-time assignable

Scope	Name	Type	Initial value
■	aAttachments	Array	{}
■	aReceipts	Array	{}
■	cAccount	Character	""
■	cFromAddress	Character	""
■	cFromName	Character	""
■	cMessage	Character	""
■	cName	Character	""
■	cPassword	Character	""
■	cSubject	Character	""
■	cTime	Character	Time()
■	dDate	Date	Date()
■	IAskPassword	Logic	.F.
■	IAskRecipients	Logic	.F.
■	ILoggedIn	Logic	.F.
■	INewSession	Logic	.F.
■	IReceipt	Logic	.F.
■	IUseUI	Logic	.F.
■	nLastError	Numeric	.F.

1.6.4.11.1.1 TMapi:aAttachments

Array with a file list to send with the message.

Scope	Assignable
Type	Array
Initial value	{}

Every element in the array is a two elements sub-array:

- Describing filename
- Full filename with the full path included

Example:

```
:aAttachments:= { { "File.txt", "c:\tests\file.txt" }, ;  
                 { "App.log", "c:\test\app.log" } }
```

1.6.4.11.1.2 TMapi:aReceipts

Array with the receipts message list.

Scope	Assignable
Type	Array
Initial value	{}

Every Array element is a sub-array of three elements:

- Describing name
- Email address
- Receipt class:
 1. Normal receipt: mapiTO
 2. Carbon copy: mapiCC
 3. Blind carbon copy: mapiBCC

Example:

```
:aReceipts := { { "Friend", "test@mail.zzz", mapiTO }, ;  
               { "Another", "another@web.com", mapiCC } }
```

1.6.4.11.1.3 TMapi:cAccount

Sender's email address.

Scope	Assignable
Type	Character
Initial value	""

1.6.4.11.1.4 TMapi:cFromAddress

Sender's email address.

Scope	Assignable
Type	Character
Initial value	""

1.6.4.11.1.5 TMapi:cFromName

Sender's name to be shown.

Scope	Assignable
Type	Character
Initial value	""

1.6.4.11.1.6 TMapi:cMessage

Message body.

Scope	Assignable
Type	Character
Initial value	""

1.6.4.11.1.7 TMapi:cName

Profile to be used.

Scope	Assignable
Type	Character
Initial value	""

1.6.4.11.1.8 TMapi:cPassword

Password to use in the case that the profile needs it.

Scope	Assignable
Type	Character
Initial value	""

1.6.4.11.1.9 TMapi:cSubject

Text string containing the message subject.

Scope	Assignable
Type	Character
Initial value	""

1.6.4.11.1.10 TMapi:cTime

Message time.

Scope	Assignable
Type	Character
Initial value	Time()

1.6.4.11.1.11 TMapi:dDate

Message date.

Scope	Assignable
Type	Date
Initial value	Date()

1.6.4.11.1.12 TMapi:AskPassword

If it is .T., it will shows the password dialog.

Scope	Assignable
Type	Logic
Initial value	.F.

1.6.4.11.1.13 TMapi:AskRecipients

If it is .T., it will show the sender dialog and other options for the message. If this property is .T. the aReceipts array must be blank.

Scope	Assignable
Type	Logic
Initial value	.F.

1.6.4.11.1.14 TMapi:LoggedOn

It is .T. when there is a connection.

Scope	read Only
Type	Logic
Initial value	.F.

1.6.4.11.1.15 TMapi:NewSession

Creates a new session instead to use a system sessions to share.

Scope	Assignable
Type	Logic
Initial value	.F.

1.6.4.11.1.16 TMapi:Receipt

Asks for a receipt notification.

Scope	Assignable
Type	Logic

Initial value	.F.
----------------------	-----

1.6.4.11.1.17 TMapi:UseUI

Shows the connection dialog.

Scope	Assignable
Type	Logic
Initial value	.F.

1.6.4.11.1.18 TMapi:nLastError

Code for the last error produced.

Scope	read Only
Type	Numeric
Initial value	0

For more information, see also the Windows MAPI documentation.

1.6.4.11.2 TMapi:Methods

■ Constructor ■ Standard

Typ	Name
■	Create
■	Destroy
■	GetAccountNames
■	Installed
■	Logoff
■	Logon
■	Send

1.6.4.11.2.1 TMapi:Create

Builds the MAPI object.

Type	Constructor
Parameters	<oParent> Container form
Return	<Self>

value	Self reference (Self)
--------------	-----------------------

1.6.4.11.2.2 TMapi:Destroy

Destroys the MAPI object.

Type	Standard
Parameters	None
Return value	NIL

1.6.4.11.2.3 TMapi:GetAccountNames

Retrieves the list of available mailing addresses. Optionally the list can be sorted.

Type	Standard
Parameters	[<ISorted>] If TRUE the array will be sorted. By default FALSE.
Return value	<aData> Array with all the mailing addresses

1.6.4.11.2.4 TMapi:Installed

Indicates if the operating systems supports or has the MAPI installed.

Type	Standard
Parameters	None
Return value	<ISuccess> .T. if the operation is successful

1.6.4.11.2.5 TMapi:Logoff

Ends the MAPI session.

Type	Standard
Parameters	None
Return value	<ISuccess> .T. if the operation is successful

1.6.4.11.2.6 TMapi:Lagon

Starts the MAPI session.

Type	Standard
Parameters	None
Return value	<ISuccess> .T. if the operation is successful

1.6.4.11.2.7 TMapi:Send

Sends the MAPI message.

Type	Standard
Parameters	None
Return value	<ISuccess> .T. if the operation is successful

1.6.4.12 TMailSlotClient

Class to transfer information between two applications. Must be used in conjunction with the class TMailSlotServer.

Description:

This class is used in conjunction with the TMailSlotServer class to pass information between two applications. For this, one of the applications must use the **TMailSlotServer** class, which is the one that stays listening mode and the other applications must use the TMailSlotClient class which starts the conversation.

Hierarchy	Descendent from TComponent
File name	\source\MailSlot.prg
See also	TMailSlotClient

1.6.4.12.1 TMailSlotClient:Properties

■ read Only ■ Assignable ■ Design assignable ■ Run-time assignable

Scope	Name	Type	Initial value
■	cName	Character	"xmailslot"
■	cServer	Character	". "

1.6.4.12.1.1 TMailSlotClient:cName

Mailbox name to receive messages.

Scope	Assignable
Type	Character
Initial value	"xmailslot"

1.6.4.12.1.2 TMailSlotClient:cServer

Server name for communications. The dot "." is used on local communications. For LAN must be changed with the machine name.

Scope	Assignable
Type	Character
Initial value	"."

1.6.4.12.2 TMailSlotClient:Methods

■ Constructor ■ Standard ■ Only after Create()

Type	Name
■	Close
■	IsOpen
■	Open
■	Write

1.6.4.12.2.1 TMailSlotClient:Close

Closes the mailbox.

Type	Only after create
Parameters	None
Return value	<IValue> True if success

1.6.4.12.2.2 TMailSlotClient:IsOpen

Checks if mailbox is open.

Type	Standard
-------------	----------

Parameters	None
Return value	<IValue> True if mailbox open

1.6.4.12.2.3 TMailSlotClient:Open

Opens the mailbox for reading.

Type	Only after create
Parameters	None
Return value	<IValue> True if success

1.6.4.12.2.4 TMailSlotClient:Write

Writes a message on the mailbox.

Type	Only after create
Parameters	<cValue> Message text
Return value	<IValue> True if success

1.6.4.12.3 TMailSlotClient:Events

Name
OnClose
OnOpen
OnRead

1.6.4.12.3.1 TMailSlotClient:OnClose

Event triggered when closing the mailbox.

Parameters:	<oSender>: Object that triggers the event.
Return value:	NIL

1.6.4.12.3.2 TMailSlotClient:OnOpen

Event triggered when the mailbox is opened.

Parameters:	<oSender>: Object that triggers the event
Return value:	NIL

1.6.4.12.3.3 TMailSlotClient:OnWrite

Event that is triggered when the mailbox is written.

Parameters:	<oSender>: Object that triggers the event. <cMessage>: Textmessage
Return value:	NIL

1.6.4.13 TMailSlotServer

Class to transfer information between two applications. Must be used in conjunction with the class TMailSlotClient.

Description:

This class is used in conjunction with the TMailSlotClient class to pass information between two applications. For this, one of the applications must use the **TMailSlotServer** class, which is the one that stays listening mode and the other applications must use the TMailSlotClient class which starts the conversation.

Hierarchy	Descendent from TComponent
Filename	\source\MailSlot.prg
See also	TMailSlotClient

1.6.4.13.1 TMailSlotServer:Properties

■ read Only ■ Assignable ■ Design assignable ■ Run-time assignable

Scope	Name	Type	Initial value
■	cName	Character	"xmailslot"
■	cServer	Character	". "

1.6.4.13.1.1 TMailSlotServer:cName

Mailbox name to receive messages.

Scope	Assignable
Type	Character
Initial value	"xmailslot"

1.6.4.13.1.2 TMailSlotServer:cServer

Server name for communications. The dot "." is used on local communications. For LAN must be changed with the machine name.

Scope	Assignable
Type	Character
Initial value	"."

1.6.4.13.2 TMailSlotServer:Methods

■ Constructor ■ Standard ■ Only after Create()

Type	Name
■	Close
■	GetMessageCount
■	HasMessages
■	IsOpen
■	Open
■	Read
■	SetReadTimeOut

1.6.4.13.2.1 TMailSlotServer:Close

Closes the mailbox.

Type	Only after create
Parameters	None
Return value	<IValue> True if success

1.6.4.13.2.2 TMailSlotServer:GetMessageCount

Returns the number of messages pending to read.

Type	Only after create
Parameters	None
Return value	<nValue> Number of messages

1.6.4.13.2.3 TMailSlotServer:HasMessages

Checks if there are messages to read.

Type	Only after create
Parameters	None
Return value	<IValue> True if there are messages pending to read

1.6.4.13.2.4 TMailSlotServer:IsOpen

Checks if mailbox is open.

Type	Standard
Parameters	None
Return value	<IValue> True if mailbox open

1.6.4.13.2.5 TMailSlotServer:Open

Opens the mailbox for reading.

Type	Only after create
Parameters	None
Return value	<IValue> True if success

1.6.4.13.2.6 TMailSlotServer:Read

Reads next message of the mailbox.

Type	Only after create
Parameters	None
Return value	<cValue> Message text

1.6.4.13.2.7 TMailSlotServer:SetReadTimeout

Sets the reading timeout.

Type	Standard
Parameters	<nValue> Time in milliseconds. With a value of 0, the Read method does not wait if there are not messages. A value of -1 waits until a message arrives
Return value	<IValue> True if the value could be set

1.6.4.13.3 TMailSlotServer:Events

Name
OnClose
OnOpen
OnRead

1.6.4.13.3.1 TMailSlotServer:OnClose

Event triggered when closing the mailbox.

Parameters:	<oSender>: Object that triggers the event.
Return value:	NIL

1.6.4.13.3.2 TMailSlotServer:OnOpen

Event triggered when the mailbox is opened.

Parameters:	<oSender>:
--------------------	------------

Object that triggers the event
Return value: NIL

1.6.4.13.3.3 TMailSlotServer:OnRead

Event that is triggered when the mailbox is readed.

Parameters: <oSender>:
Object that triggers the event.
<cMessage>:
Text message

Return value: NIL

1.6.4.14 TWebBrowser

This control is itself a complete Internet browser that is based on the Microsoft COM component that controls Internet Explorer.

As you can see by reviewing the class, all your code is nothing more than members of the type wrapper to the real properties, methods and events of the COM component. In the present documentation only the small comment that shows the COM component itself has been included. For more information than the one provided here, consult the Microsoft documentation:

[https://msdn.microsoft.com/en-us/library/aa752084\(v=vs.85\).aspx](https://msdn.microsoft.com/en-us/library/aa752084(v=vs.85).aspx)

Hierarchy Descendiente de TOcx
File \source\WebBrowser.prg

1.6.4.14.1 TWebBrowser:Properties

■ Read only ■ Assignable ■ Design assignable ■ Run-time assignable

Scope	Name	Type	Comment
■	AddressBar	Logical	Controls whether address bar is shown
■	Application	IDispatch	Returns the application automation object if accessible, this automation object otherwise..
■	Busy	Logico	Query to see if something is still in progress
■	Container	IDispatch	Returns the container/parent automation object, if any
■	Document	IDispatch	Returns the active Document automation object, if any.
■	FullName	IDispatch	Returns file specification of the application, including path.
■	FullScreen	Logical	Maximizes window and turns off

			statusbar, toolbar, menubar, and titlebar.
■	Height	Numeric	The vertical dimension (pixels) of the frame window/object.
■	Left	Numeric	The horizontal position (pixels) of the frame window relative to the screen/container.
■	LocationName	Carácter	Gets the short (UI-friendly) name of the URL/file currently viewed.
■	LocationURL	Carácter	Gets the full URL/path currently viewed
■	MenuBar	Logical	Controls whether menubar is shown.
■	Offline	Logical	Controls if the frame is offline (read from cache)
■	Parent	IDispatch	Returns the automation object of the container/parent if one exists or this automation object.
■	Path	IDispatch	Returns the path to the application.
■	RegisterAsBrowse	Logico	Registers OC as a top-level browser (for target name resolution)
■	RegisterAsDropTarget	Logical	Registers OC as a drop target for navigation
■	Resizable	Logical	Controls whether the window is resizable
■	Silent	Logical	Controls if any dialog boxes can be shown
■	StatusBar	Logical	Turn on or off the statusbar.
■	StatusText	Carácter	Text of Status window.
■	TheaterMode	Logical	Controls if the browser is in theater mode
■	ToolBar	Logical	Controls which toolbar is shown
■	Top	Numeric	The vertical position (pixels) of the frame window relative to the screen/container.
■	TopLevelContainer	Logical	Returns True if this is the top level object.
■	Type	Carácter	Returns the type of the contained document object.
■	Visible	Logical	Determines whether the application is visible or hidden.
■	Width	Numeric	The horizontal dimension (pixels) of the frame window/object

1.6.4.14.2 TWebBrowser:Methods

■ Constructor ■ Standard ■ Only after Create()

Tipo	Nombre	Comentario
■	ClientToWindow (@npcx, @npcy)	Converts client sizes into window sizes.
■	ExecWB (ucmdID,	IOleCommandTarget::Exec

	ucmdexecopt, vpvaln, vpvaOut)	
■	GetProperty (cProperty)	Retrieve the Associated value for the property vtValue in the context of the object.
■	GoBack ()	Navigates to the previous item in the history list
■	GoForward ()	Navigates to the next item in the history list
■	GoHome ()	Go home/start page
■	GoSearch ()	Go Search Page
■	Navigate (cURL, vFlags, vTargetFrameName, vPostData, vHeaders)	Navigates to a URL or file
■	Navigate2 (vURL, vFlags, vTargetFrameName, vPostData, vHeaders)	Navigates to a URL or file or pidl
■	PutProperty (cProperty, vvtValue)	Associates vtValue with the name szProperty in the context of the object.
■	QueryStatusWB (ucmlID)	IOleCommandTarget::QueryStatus
■	Quit ()	Exits application and closes the open document.
■	Refresh2 (vLevel)	Refresh the currently viewed page.
■	ShowBrowserBar (vpvaClsid, vpvarShow, vpvarSize)	Set BrowserBar to Clsid
■	Stop ()	Stops opening a file.

1.6.4.14.3 TWebBrowser:Events

Name	Comment
BeforeNavigate2 (disppDisp, vURL, vFlags, vTargetFrameName, vPostData, vHeaders, @ICancel)	Fired before navigate occurs in the given WebBrowser (window or frameset element). The processing of this navigation may be modified.
BeforeScriptExecute (disppDispWindow)	Fired prior to the first script execution.
ClientToHostWindow (@nCXX, @nCY)	Fired to request client sizes be converted to host window sizes
CommandStateChange (nCommand, IEnable)	The enabled state of a command changed.
DocumentComplete (disppDisp, vURL)	Fired when the document being navigated to reaches ReadyState_Complete.
DownloadBegin ()	Download of a page started.
DownloadComplete ()	Download of page complete.
FileDownload (IActiveDocument, @ICancel)	Fired to indicate the File Download dialog is opening
NavigateComplete2 (disppDisp, vURL)	Fired when the document being navigated to becomes visible and enters the navigation stack.

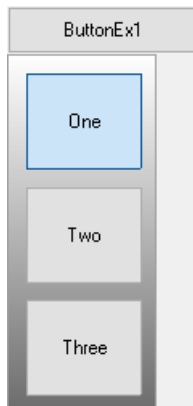
NavigateError (disppDisp, vURL, vFrame, vStatusCode, @ICancel)	Fired when a binding error occurs (window or frameset element).
NewProcess (nCauseFlag, disppWB2, @ICancel)	A new, hidden, non-navigated process is created to handle the navigation.
NewWindow2 (@dispppDisp, @ICancel)	A new, hidden, non-navigated WebBrowser window is needed.
NewWindow3 (@dispppDisp, @ICancel, ndwFlags, cbstrUrlContext, cbstrUrl)	A new, hidden, non-navigated WebBrowser window is needed.
OnFullScreen (IFullScreen)	Fired when fullscreen mode should be on/off
OnMenuBar (IMenuBar)	Fired when the menubar should be shown/hidden
OnQuit ()	Fired when application is quitting
OnStatusBar (IStatusBar)	Fired when the statusbar should be shown/hidden
OnTheaterMode (ITheaterMode)	Fired when theater mode should be on/off
OnToolBar (IToolBar)	Fired when the toolbar should be shown/hidden
OnVisible (IVisible)	Fired when the window should be shown/hidden
PrintTemplateInstantiation (disppDisp)	Fired when a print template is instantiated.
PrintTemplateTeardown (disppDisp)	Fired when a print template destroyed.
PrivacyImpactedStateChange (IblImpacted)	Fired when the global privacy impacted state changes
ProgressChange (nProgress, nProgressMax)	Fired when download progress is updated.
PropertyChange (cszProperty)	Fired when the PutProperty method has been called.
RedirectXDomainBlocked (disppDisp, vStartURL, vRedirectURL, vFrame, vStatusCode)	Fired when a x-domain redirect is blocked.
SetPhishingFilterStatus (nPhishingFilterStatus)	Fired to indicate the progress and status of the Phishing Filter analysis of the current web page
SetSecureLockIcon (nSecureLockIcon)	Fired to indicate the security level of the current web page contents
StatusTextChange (cText)	Statusbar text changed.
ThirdPartyUrlBlocked (vURL, ndwCount)	Fired when a third-party URL is blocked.
TitleChange (cText)	Document title changed.
UpdatePageStatus (disppDisp, vnPage, vfDone)	Fired when a page is spooled. When it is fired can be changed by a custom template.
WebWorkerFinsihed (ndwUniqueID)	Fired after a Web Worker has closed
WebWorkerStarted (ndwUniqueID, cbstrWorkerLabel)	Fired after a Web Worker has been started.
WindowClosing (IIsChildWindow, @ICancel)	Fired when the WebBrowser is about to be closed by script

WindowSetHeight (nHeight)	Fired when the host window should change its height
WindowSetLeft (nLeft)	Fired when the host window should change its Left coordinate
WindowSetResizable (IResizable)	Fired when the host window should allow/disallow resizing
WindowSetTop (nTop)	Fired when the host window should change its Top coordinate
WindowSetWidth (nWidth)	Fired when the host window should change its width
WindowStateChanged (ndwWindowStateFlags, ndwValidFlagsMask)	Fired to indicate that the browser window's visibility or enabled state has changed.

1.6.5 Modern

1.6.5.1 TButtonEx

This class represents a button-type control with the ability to display an image and a drop-down list. Its main feature is that it never receives focus, so it is especially suitable for applications of tactile type in which we do not want that by pressing a button, the focus changes towards him.



Hierarchy TControl descendant
See also TButton , TBtnBmp
File \source\ButtonEx.prg

1.6.5.1.1 TButtonEx:Properties

■ Read only ■ Assignable □ Design assignable ■ Run-time assignable

Scope	Name	Type	Initial value
■	altems	Array	{}
□	cShortcut	Character	""
■	IDropped	Logical	.F.
■	IMultiLine	Logical	.T.
■	IPushed	Logical	.F.
□	nAlignment	Numeric	taCENTER
□	nBmpHeight	Numeric	1

■	nBmpWidth	Numeric	1
■	nClrBorder	Numeric	0x675921
■	nClrPane	Numeric	clGrayText
■	nClrPanelItems	Numeric	clWhite
■	nEffects	Numeric	beSLIDE
■	nHeight	Numeric	30
■	nMargin	Numeric	-1
■	nOrientation	Numeric	orLEFT
■	nStyle	Numeric	bsDEFAULT
■	nVAlignment	Numeric	vaCENTER
■	nWidth	Numeric	120
■	oBitmaps	Object	NIL
■	oFontShortcut	Object	NIL

1.6.5.1.1.1 TButtonEx:alItems

Array with the literals from the drop-down list that will be displayed when the button is pressed.

Scope:	Assignable
Type:	Array
Initial value:	{}

1.6.5.1.1.2 TButtonEx:cShortCut

Text to display to indicate the keyboard shortcut for that button.

Scope:	Design assignable
Type:	Text
Initial value:	""

1.6.5.1.1.3 TButtonEx:IDropped

True if drop-down list is visible.

Scope:	Read only
Type:	Logical
Initial value:	.F.

1.6.5.1.1.4 TButtonEx:IMultiLine

True is multi-line.

Scope:	Assignable
Type:	Logical
Initial value:	.T.

1.6.5.1.1.5 TButtonEx:IPushed

True if button is pushed.

Scope:	Assignable
Type:	Logical
Initial value:	.F.

1.6.5.1.1.6 TButtonEx:nAlignment

Text horizontal alignment.

Scope:	Design assignable
Type:	Numeric
Initial value:	taCENTER
Possible values:	taLEFT, taCENTER, taRIGHT

1.6.5.1.1.7 TButtonEx:nBmpHeight

Bitmap height.

Scope:	Design assignable
Type:	Numeric
Initial value:	1

This property indicates the height of the Bitmap to be displayed on the button. The default value of 1 indicates that the actual height of the first bitmap indicated in the oBitmaps property will be used. If it indicates a lower height than the one actually existing in the Bitmap, the image will be truncated.

1.6.5.1.1.8 TButtonEx:nBmpWidth

Bitmap width.

Scope:	Design assignable
Type:	Numeric
Initial value:	1

This property indicates the width of the Bitmap to be displayed on the button. The default value of 1 indicates that the actual width of the first bitmap indicated in the oBitmaps property will be used. If it indicates a lower width than the one actually existing in the Bitmap, the image will be truncated.

1.6.5.1.1.9 TButtonEx:nClrBorder

Border color. This property is useless except when nStyle property is set to bsMODERN.

Scope:	Assignable
Type:	Numeric
Initial value:	0x675921

(Consult appendix for possible colors)

1.6.5.1.1.10 TButtonEx:nClrPane

Background color. This property is useless except when nStyle property is set to bsMODERN.

Scope:	Assignable
Type:	Numeric
Initial value:	clGrayText

(Consult appendix for possible colors)

1.6.5.1.1.11 TButtonEx:nClrPanelItems

Background color for drop down list.

Scope:	Assignable
Type:	Numeric
Initial value:	clWhite

(Consult appendix for possible colors)

1.6.5.1.1.12 TButtonEx:nEffects

Drop-down list display effect.

Scope:	Assignable
Type:	Numeric
Initial value:	beSLIDE
Possible values:	beSLIDE, beNONE

1.6.5.1.1.13 TButtonEx:nHeight

Control height.

Scope:	Assignable
Type:	Numeric
Initial value:	30

1.6.5.1.1.14 TButtonEx:nMargin

Space between bottom border and image.

Scope:	Assignable
Type:	Numeric
Initial value:	-1

A -1 value indicates that the margin will be calculated automatically.

1.6.5.1.1.15 TButtonEx:nOrientation

Bitmap orientation respect the text.

Scope:	Assignable
Type:	Numeric
Initial value:	orLEFT
Possible values:	orTOP, orBOTTOM, orLEFT, orRIGHT

1.6.5.1.1.16 TButtonEx:nStyle

Control style.

Scope:	Assignable
Type:	Numeric
Initial value:	bsDEFAULT
Possible values:	bsDEFAULT, bsWINDOWS7, bsMODERN

1.6.5.1.1.17 TButtonEx:nVAlignment

Text vertical alignment.

Scope:	Design assignable
Type:	Numeric
Initial value:	vaCENTER
Possible values:	vaTOP, vaCENTER, vaBOTTOM

1.6.5.1.1.18 TButtonEx:nWidth

Control width.

Scope:	Assignable
Type:	Numeric
Initial value:	120

1.6.5.1.1.19 TButtonEx:oBitmaps

TImageList object with all the images.

Scope:	Assignable
Type:	Object
Initial value:	NIL

The TButtonEx button can show 4 different images depending of its state. They can be:

- Normal, it corresponds to the first TImageList image.
- With the mouse pointer over it, that corresponds to the second TImageList image.
- Pressed, and corresponds to the third TImageList image.
- Disable, that corresponds to the fourth TImageList image.

The button will use the first bitmap for all the states where the image has not been defined.

There are several ways to assign this property:

- Through a previously created TImageList object.
- Through a literal with the resource name of Bitmap type file.
- Through an 1, 2, 3 or 4 elements array with the resource names or the bitmap type files for every possible button state.

For any of the last 2 cases the button will create a TImageList with the information provided.

1.6.5.1.1.20 TButtonEx:oFontShortcut

TFont object to use for shortcut text.

Scope:	Assignable
Type:	Object
Initial value:	Nil

1.6.5.1.1.21 TButtonEx:Events

Name
OnClick
OnDrop

Event triggered when the button is clicked.

Parameters:	<oSender>: Reference to the object that triggers the event <nItem>: Ordianl in altems fired. If altems is not defined this value is NIL
Return value:	<IResult>: If distinct from logical or .T. the drop-down list is closed

Event triggered when the menu is displayed.

Parameters:	<oSender>: Reference to the object that triggers the event <oMenu>: Menu object
Return value:	<NIL>

1.6.5.2 TButtonMod

This class represents a button with a Windows 10 appearance. The button text can use HTML formatting in the same way as the control TLabelEx.

Hierarchy	TStdControl descendant
See also	TButton
File	\source\ButtonMod.prg

1.6.5.2.1 TButtonMod:Properties

■ Read only ■ Assignable ■ Design assignable ■ Run-time assignable

Scope	Name	Type	Initial value
■	ICancel	Logical	.F.
■	IDefault	Logical	.F.
■	ILegacyBehaviour	Logical	.F.
■	IMultiLine	Logical	.F.
■	ITransparent	Logical	.F.
■	nBmpHeight	Numeric	1
■	nBmpMargin	Numeric	2
■	nBmpWidth	Numeric	1
■	nBorderRadius	Numeric	0
■	nLineSpacing	Numeric	100
■	nModalResult	Numeric	mrNONE
■	nOrientation	Numeric	orLEFT
■	nVAlignment	Numeric	vaTOP
■	oBitmaps	Object	NIL
■	oMenu	Object	NIL

1.6.5.2.1.1 TButtonMod:ICancel

Indicates that the OnClick event should be triggered when **ESC** key is pressed on its container form..

Scope:	Design assignable
Type:	Logical
Initial value:	.F.

1.6.5.2.1.2 TButtonMod:IDefault

Indicates to trigger the `OnClick` event when the **ENTER** key is pushed in its container form.

Scope:	Design assignable
Type:	Logical
Initial value:	.F.

1.6.5.2.1.3 TButtonMod:ILegacyBehaviour

By default, the button only activates its **OnClick** event when pressing **ENTER** when the `Application:UseReturn` property is true or it is a button with the `IDefault` property set to true. If you want **INTRO** to always fire **OnClick** while keeping the rest of Xailer's buttons working, set this property to true.

Scope:	Assignable
Type:	Logical
Initial value:	.F.

1.6.5.2.1.4 TButtonMod:IMultiLine

The control displays several lines of text.

Scope	Assignable
Type	Logical
Initial value	.F.

1.6.5.2.1.5 TButtonMod:ITransparent

The control is drawn transparent on the form to which it belongs.

Scope	Assignable
Type	Logical
Initial value	.T.

1.6.5.2.1.6 TButtonMod:nBmpHeight

Bitmap height.

Scope:	Design assignable
Type:	Numeric
Initial value:	1

This property indicates the height of the bitmap to be displayed on the button. The default value of 1 indicates that the real height of the first Bitmap indicated on its oBitmaps property will be used. If you specify a lower height value than the real one, the image will be truncated.

1.6.5.2.1.7 TButtonMod:nBmpMargin

Pixels between the button border and the image.

Scope:	Assignable
Type:	Numeric
Initial value:	2

1.6.5.2.1.8 TButtonMod:nBmpWidth

Bitmap width.

Scope:	Design assignable
Type:	Numeric
Initial value:	1

This property indicates the width of the Bitmap to be displayed on the button. The default value of 1 indicates that the actual width of the first bitmap indicated in the oBitmaps property will be used. If it indicates a lower width than the one actually existing in the Bitmap, the image will be truncated.

1.6.5.2.1.9 TButtonMod:nBorderRadius

Radius of curvature for the edges.

Scope:	Assignable
Type:	Numeric
Initial value:	0

1.6.5.2.1.10 TButtonMod:nLineSpacing

Leading in percentage. Maximum value: 1000.

Scope	Assignable
Type	Numeric
Initial value	100

1.6.5.2.1.11 TButtonMod:nModalResult

Value that indicates the close mode from the modal form where the button belongs..

Scope:	Assignable
Type:	Numeric
Initial value:	mrNONE
Possible values:	mrNONE, mrOK, mrCANCEL, mrABORT, mrRETRY, mrIGNORE, mrYES, mrNO, mrCLOSE, mrHELP, mrTRYAGAIN, mrCONTINUE, mrALL, mrNOTOALL, mrYESTOALL

This property is very useful when it is used together with modal forms. When its value is different to mrNONE, the simple push button operation implicates to close the form and the value of this property is assigned to the nModalResult property of the form where the button belongs, and this value is returned as returned value in the Showmodal() method calls..

The possible values are basically the description of the classic operations.

Example:

```
With Object TMyForm():New()
  If :ShowModal == mrOK
    . . . .
  Endif
End With
```

1.6.5.2.1.12 TButtonMod:nOrientation

Bitmap orientation respect the text.

Scope:	Assignable
Type:	Numeric
Initial value:	orLEFT
Possible values:	orTOP, orBOTTOM, orLEFT, orRIGHT

1.6.5.2.1.13 TButtonMod:nVAlignment

Text vertical alignment.

Scope	Assignable
Type	Numeric
Initial value	vaTOP
Possible values	vaTOP, vaBOTTOM, vaCENTER

1.6.5.2.1.14 TButtonMod:oBitmaps

TImageList object with all the images.

Scope:	Assignable
Type:	Object
Initial value:	NIL

The TButtonEx button can show 4 different images depending of its state. They can be:

- Normal, it corresponds to the first TImageList image.
- With the mouse pointer over it, that corresponds to the second TImageList image.
- Pressed, and corresponds to the third TImageList image.
- Disable, that corresponds to the fourth TImageList image.

The button will use the first bitmap for all the states where the image has not been defined.

There are several ways to assign this property:

- Through a previously created TImageList object.
- Through a literal with the resource name of Bitmap type file.
- Through an 1, 2, 3 or 4 elements array with the resource names or the bitmap type files for every possible button state.

For any of the last 2 cases the button will create a TImageList with the information provided.

1.6.5.2.1.15 TButtonMod:oMenu

TMenu object that will show the control.

Scope:	Assignable
Type:	Object
Initial value:	NIL

1.6.5.2.2 TButtonMod:Events

Name
OnCustomDraw
OnMenuClick

1.6.5.2.2.1 TButtonMod:OnCustomDraw

Event that is fired when the control is painted when the text to be displayed has not been specified.

Parameters	<oSender> A reference the object itself <hDC> Handle to device context <aRect> Rectangle with painted coordinates
Return value	NIL

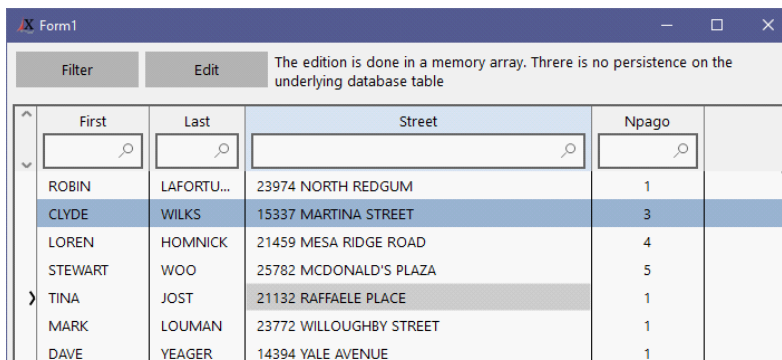
1.6.5.2.2.2 TButtonMod:OnMenuClick

Event that is triggered when trying to display the menu associated with the control.

Parameters	<oSender> A reference the object itself <oMenu> TMenu object
Return value	A .F. value avoids the display of the popup menu

1.6.5.3 TBrowseMod

This class represents a **TBrowse** adapted to Windows 10 style.



This control is actually a container for multiple controls:

- A TBrwModGrid that actually shows the data
- Three THeaderMod controls to manage the header, the footer and the super-header
- A TRecSelector which is a panel that occupies the record selector area.

And therefore it must take into account that many of the properties of this control are actually delegated to these controls.

Hierarchy Descendiente de TWinControl
File \source\BrowseMod.prg

1.6.5.3.1 TBrowseMod:Propiedades

■ Read only ■ Assignable ■ Design assignable ■ Run-time assignable

Scope	Name	Type	Initial value
■	aCols	Array	{}
■	aDisplayCols	Array	{}
■	aGridData	Array	{}
■	aSelected	Array	{}
■	cRecSelectorHeader	Character	""
■	IAllowAppend	Logical	.F.
■	IAllowColHiding	Logical	.T.
■	IAllowColSizing	Logical	.T.
■	IAllowColSwapping	Logical	.T.
■	IAllowDelete	Logical	.F.
■	IAllowEdit	Logical	.F.
■	IAllowInsert	Logical	.F.
■	IAllowSort	Logical	.F.
■	IAlternateColor	Logical	.F.
■	IAutoSave	Logical	.T.
■	IColumnSeparator	Logical	.F.
■	IFastAppend	Logical	.F.
■	IFastEdit	Logical	.F.
■	IFilterBar	Logical	.F.
■	IFooter	Logical	.F.
■	IHeader	Logical	.T.
■	IHideScrollBars	Logical	.T.
■	IHotTrack	Logical	.T.
■	IMultipleSel	Logical	.F.
■	IRowHeightButtons	Logical	.F.
■	IShowCellSelected	Logical	.F.
■	ISupraHeader	Logical	.F.
■	ITooltips	Logical	.T.
■	ITransparent	Logical	.F.
■	ITransparentGrid	Logical	.F.
■	IUseReturn	Logical	NIL
■	nClrDivider	Numeric	clGray

■	nClrHeaderDivider	Numeric	clGray
■	nClrHeaderEditPane	Numeric	clGradientInactiveCaption
■	nClrHeaderEditText	Numeric	clWindowText
■	nClrHeaderEditTextFocus	Numeric	clWindowText
■	nClrHeaderEditPaneFocus	Numeric	clWindow
■	nClrHeaderHotPane	Numeric	clActiveCaption
■	nClrHeaderHotText	Numeric	clWindowText
■	nClrHeaderPane	Numeric	clBtnFace
■	nClrHeaderSelPane	Numeric	clGradientInactiveCaption
■	nClrHeaderText	Numeric	clBtnText
■	nClrHotPane	Numeric	clActiveCaption
■	nClrHotText	Numeric	clWindowText
■	nClrPane	Numeric	clWindow
■	nClrSelPane	Numeric	clWindow
■	nClrSelText	Numeric	clWindowText
■	nClrText	Numeric	clWindowText
■	nEditMode	Numeric	beREAD
■	nFooterHeight	Numeric	0
■	nHeaderHeight	Numeric	0
■	nIndex	Numeric	0
■	nMargin	Numeric	5
■	nRecSelectorWidth	Numeric	0
■	nRowHeight	Numeric	0
■	nRowIncrement	Numeric	5
■	nSelectCol	Numeric	1
■	oDataGrid	Object	TBrwModGrid
■	oEditCol	Object	TBrwColMod
■	oFooter	Object	THeaderMod
■	oGroup	Object	TBrwGroup
■	oHeader	Object	THeaderMod
■	oImageList	Object	TImageList
■	oRecSelector	Object	TLabel
■	oSelectCol	Object	TBrwColMod
■	oSupra	Object	THeaderMod

1.6.5.3.1.1 TBrowseMod:aCols

TBrwColMod array object that hold the browse columns.

Scope	Assignable
Type	Array
Initial value	{}

1.6.5.3.1.2 TBrowserMod:aDisplayCols

TBrwColMod array objects with the actual visible columns.

Scope	Read only
Type	Array
Initial value	{}

1.6.5.3.1.3 TBrowserMod:aGridData

Multi-dimensional array with the data to show. Sults structure corresponds to an array of arrays. Each sub-array must have a dimension identical to the number of columns in the Browse (aCols) and the position of each element in the sub-array must match the position of each column in aCols.

Scope	Assignable
Type	Array
Initial value	{}

1.6.5.3.1.4 TBrowserMod:aSelected

Array with the record numbers of all currently selected rows. To use it, the MultipleSel property must be set to True.

Scope	Read only
Type	Array
Initial value	{}

1.6.5.3.1.5 TBrowserMod:cRecSelectorHeader

Text to be displayed in the header of the record selector if visible.

Scope	Assignable
Type	Character
Initial value	""

1.6.5.3.1.6 TBrowseMod:IAAllowAppend

If true, the user is allowed to add rows (records), either by pressing the cursor down key when in the last row of the Browse or by pressing the <Insert> key when the IAllowInsert is set to false.

Scope	Assignable
Type	Logical
Initial value	.F.

1.6.5.3.1.7 TBrowseMod:IAAllowColHiding

If True, columns can be hidden by the user using the context menu of the headers.

Scope	Assignable
Type	Logical
Initial value	.T.

1.6.5.3.1.8 TBrowseMod:IAAllowSizing

If True, the width of the columns can be modified by the user.

Scope	Assignable
Type	Logical
Initial value	.T.

1.6.5.3.1.9 TBrowseMod:IAAllowColSwapping

If True, the user is allowed to change the position of the columns.

Scope	Assignable
Type	Logical
Initial value	.T.

1.6.5.3.1.10 TBrowseMod:IAAllowDelete

If True, user deletion operation is allowed by pressing the <Delete> key..

Scope	Assignable
Type	Logical

Initial value	.F.
----------------------	-----

1.6.5.3.1.11 TBrowseMod:IAAllowEdit

If True, user editing operation is allowed by pressing the <Enter> key or **double-clicking** on the cell.

Scope	Assignable
Type	Logical
Initial value	.F.

1.6.5.3.1.12 TBrowseMod:IAAllowInsert

If true, user insertion operation is allowed by pressing the <Insert> key.

This option should only be used when the data on which the Browse is based is an array.

Scope	Assignable
Type	Logical
Initial value	.F.

1.6.5.3.1.13 TBrowseMod:IAAllowSort

If True, column sorting by the user is allowed..

Scope	Assignable
Type	Logical
Initial value	.F.

1.6.5.3.1.14 TBrowseMod:IAAlternateColor

If True, the pajama-like rows are shown alternating their color with a higher contrast and readability.

Scope	Assignable
Type	Logical
Initial value	.F.

1.6.5.3.1.15 TBrowseMod:IAutoSave

If True, the editing operations will be saved automatically. If this property is set to False, you must use the OnUpdate event to force the new data to be saved.

Scope	Assignable
Type	Logical
Initial value	.T.

1.6.5.3.1.16 TBrowseMod:IColumnSeparator

If True, it will show a line which separates each column.

Scope	Assignable
Type	Logical
Initial value	.F.

1.6.5.3.1.17 TBrowseMod:IFastAppend

If True when the user presses the cursor <Down> key in the last row of the Browse causes the entry into a append process as long as the IAllowAppend property is also set to True.

Scope	Assignable
Type	Logical
Initial value	.F.

1.6.5.3.1.18 TBrowseMod:IFastEdit

If you press a key other than a navigation key, it causes the cell to go into edit status and enters as the value of the cell, the value of the pressed key.

Scope	Assignable
Type	Logical
Initial value	.F.

1.6.5.3.1.19 TBrowseMod:IFilterBar

If True, the edit boxes for filtering are displayed for each of the visible columns.

Scope	Assignable
Type	Logical
Initial value	.F.

1.6.5.3.1.20 TBrowseMod:IFooter

If True, the column footers are displayed.

Scope	Assignable
Type	Logical
Initial value	.F.

1.6.5.3.1.21 TBrowseMod:IHeader

If True, the column headers are displayed.

Scope	Assignable
Type	Logical
Initial value	.T.

1.6.5.3.1.22 TBrowseMod:IHideScrollBars

If True, the scroll bars will be hidden and will only be displayed when the mouse hovers over the areas where it should be.

Scope	Assignable
Type	Logical
Initial value	.T.

1.6.5.3.1.23 TBrowseMod:IHotTrack

If True, the row where the mouse cursor is located is displayed in a different color.

Scope	Assignable
Type	Logical

Initial value	.T.
----------------------	-----

1.6.5.3.1.24 TBrowseMod:IMultipleSel

If True, multiple row selection is allowed. The aSelected property indicates the selected rows.

Scope	Assignable
Type	Logical
Initial value	.F.

1.6.5.3.1.25 TBrowseMod:IRowHeightButtons

If True is allowed, two buttons will be displayed allowing the user to modify the row height. It is necessary that the Browser has the IHeader property set to True and a value greater than zero of the nRecSelectorWidth property.

Scope	Assignable
Type	Logical
Initial value	.F.

1.6.5.3.1.26 TBrowseMod:IShowCellSelected

If True, displays the selected cell instead of the entire row.

Scope	Assignable
Type	Logical
Initial value	.F.

1.6.5.3.1.27 TBrowseMod:ISupraHeader

If True, the possible over-headers of the different columns are shown. For more information see the AddSupraCol method.

Scope	Assignable
Type	Logical
Initial value	.F.

1.6.5.3.1.28 TBrowseMod:ITooltips

If True, the control will display tool tips if the event OnDispTooltip is captured.

Scope	Assignable
Type	Logical
Initial value	.T.

1.6.5.3.1.29 TBrowseMod:ITransparent

The control is drawn transparent on the form to which it belongs.

Scope	Assignable
Type	Logical
Initial value	.F.

1.6.5.3.1.30 TBrowseMod:ITransparentGrid

The TDataGrid control is drawn transparent over its TBrowseMod container.

Scope	Assignable
Type	Logical
Initial value	.F.

1.6.5.3.1.31 TBrowseMod:IUseReturn

Allows to set the behavior of the **'Enter'** key in multiple column editing. A NIL value indicates that it will behave according to the Application:IUseReturn property. If True, the 'Enter' key will allow navigating between controls using that key.

Scope	Assignable in run-time
Type	Logical
Initial value	NIL

1.6.5.3.1.32 TBrowseMod:nClrDivider

Color of the column separator divider.

Scope:	Assignable
Type:	Numeric
Initial value:	clGray

(Consult appendix for possible colors)

1.6.5.3.1.33 TBrowseMod:nClrHeaderDivider

Color of the header column divider.

Scope:	Assignable
Type:	Numeric
Initial value:	clGray

(Consult appendix for possible colors)

1.6.5.3.1.34 TBrowseMod:nClrHeaderEditPane

Background color of the TEditMod responsible for filtering the header columns.

Scope:	Assignable
Type:	Numeric
Initial value:	clGradientInactiveCaption

(Consult appendix for possible colors)

1.6.5.3.1.35 TBrowseMod:nClrHeaderEditPaneFocus

Background color when focusing on the TEditMod control that is responsible for filtering the header columns.

Scope:	Assignable
Type:	Numeric
Initial value:	clWindow

(Consult appendix for possible colors)

1.6.5.3.1.36 TBrowseMod:nClrHeaderEditText

TEditMod text color responsible for filtering the header columns.

Scope:	Assignable
Type:	Numeric
Initial value:	clWindowText

(Consult appendix for possible colors)

1.6.5.3.1.37 TBrowseMod:nClrHeaderEditTextFocus

TEditMod text color **with focus** responsible for filtering the header columns.

Scope:	Assignable
Type:	Numeric
Initial value:	clWindowText

(Consult appendix for possible colors)

1.6.5.3.1.38 TBrowseMod:nClrHeaderHotPane

Background color of the columns when they are marked by the mouse.

Scope:	Assignable
Type:	Numeric
Initial value:	clActiveCaption

(Consult appendix for possible colors)

1.6.5.3.1.39 TBrowseMod:nClrHeaderHotText

Color of the text in the columns when they are marked by the mouse.

Scope:	Assignable
Type:	Numeric
Initial value:	clActiveCaption

(Consult appendix for possible colors)

1.6.5.3.1.40 TBrowseMod:nClrHeaderPane

Background color of the columns.

Scope:	Assignable
Type:	Numeric
Initial value:	clBtnFace

(Consult appendix for possible colors)

1.6.5.3.1.41 TBrowseMod:nClrHeaderSelPane

Background color of the selected rows.

Scope:	Assignable
Type:	Numeric
Initial value:	clGradientInactiveCaption

(Consult appendix for possible colors)

1.6.5.3.1.42 TBrowseMod:nClrHeaderText

Column headers text color.

Scope:	Assignable
Type:	Numeric
Initial value:	clBtnText

(Consult appendix for possible colors)

1.6.5.3.1.43 TBrowseMod:nClrHotPane

Columns background color when marked with the mouse.

Scope:	Assignable
Type:	Numeric
Initial value:	clActiveCaption

(Consult appendix for possible colors)

1.6.5.3.1.44 TBrowseMod:nClrHotText

Columns text color when marked with the mouse.

Scope:	Assignable
Type:	Numeric
Initial value:	clWindowText

(Consult appendix for possible colors)

1.6.5.3.1.45 TBrowseMod:nClrPane

Columns background color.

Scope:	Assignable
Type:	Numeric
Initial value:	clWindow

(Consult appendix for possible colors)

1.6.5.3.1.46 TBrowseMod:nClrSelPane

Background color of selected rows.

Scope:	Assignable
Type:	Numeric
Initial value:	clWindow

(Consult appendix for possible colors)

1.6.5.3.1.47 TBrowseMod:nClrSelText

Text color of selected rows.

Scope:	Assignable
Type:	Numeric
Initial value:	clWindowText

(Consult appendix for possible colors)

1.6.5.3.1.48 TBrowseMod:nClrText

Text color.

Scope:	Assignable
Type:	Numeric
Initial value:	clWindowText

(Consult appendix for possible colors)

1.6.5.3.1.49 TBrowseMod:nEditMode

Edit mode

Scope:	Read only
Type:	Numeric
Initial value:	beREAD
Possible values:	beREAD, beEDIT, beAPPEND, beINSERT

1.6.5.3.1.50 TBrowseMod:nFooterHeight

Footer height. If it is zero, it is calculated automatically.

Scope:	Assignable
Type:	Numeric
Initial value:	0

1.6.5.3.1.51 TBrowseMod:nHeaderHeight

Header height. If it is zero, it is calculated automatically.

Scope:	Assignable
Type:	Numeric
Initial value:	0

1.6.5.3.1.52 TBrowseMod:nIndex

Ordinal of active row according to current sorting and filtering

Scope:	Read only
---------------	-----------

Type:	Numeric
Initial value:	0

1.6.5.3.1.53 TBrowseMod:nMargin

argin to be used in the plotting of each column.

Scope:	Assignable
Type:	Numeric
Initial value:	5

1.6.5.3.1.54 TBrowseMod:nRecSelectorWidth

Width of the dummy column displayed to the left of the Browse indicating the active row. If the value is zero, the column is not displayed.

Scope:	Assignable
Type:	Numeric
Initial value:	0

1.6.5.3.1.55 TBrowseMod:nRowHeight

Row height. A value of zero forces it to be calculated automatically.

Scope:	Assignable
Type:	Numeric
Initial value:	0

1.6.5.3.1.56 TBrowseMod:nRowIncrement

Number of pixels by which the row height will be increased or decreased when the buttons shown when IRowHeightButtons is set to true.

Scope:	Assignable
Type:	Numeric
Initial value:	5

1.6.5.3.1.57 TBrowserMod:nSelectCol

Active column number according to the aDisplayCols. Use oSelectCol to get the TBrwColMod active column.

Scope:	Assignable
Type:	Numeric
Initial value:	0

1.6.5.3.1.58 TBrowserMod:oDataGrid

Object that manages Browse data.

Scope:	Read only
Type:	Object
Initial value:	TBrwModGrid

1.6.5.3.1.59 TBrowserMod:oEditCol

Column currently being edited in the Browse grid. This property has value only when the edition process is of a single cell. In all other cases, its value is NIL.

Scope:	Read only
Type:	Object
Initial value:	TBrwColMod

1.6.5.3.1.60 TBrowserMod:oFooter

Object that manages the browse footer.

Scope:	Read only
Type:	Object
Initial value:	THeaderMod

1.6.5.3.1.61 TBrowserMod:oGroup

Object that manages the Browse grouping when it has been defined with the GroupOn method

Scope:	Read only
Type:	Object
Initial value:	TBrwGroup

1.6.5.3.1.62 TBrowseMod:oHeader

Object managing the browse header.

Scope:	Read only
Type:	Object
Initial value:	THeaderMod

1.6.5.3.1.63 TBrowseMod:oImageList

TImageList object that holds all the images used by the control.

Scope	Assignable
Type	Object
Initial value	TImageList

The object TImageList is instantiated on the New() constructor of the control. The first image included on the TImageList sets the dimension of the following images introduced. If the first image is an image that includes more images inside is important that you set the nHeight and nWidth properties of the TImageList before adding any image.

1.6.5.3.1.64 TBrowseMod:oRecSelector

Object that manages the area marked by the nRecSelectorWidth property.

Scope:	Read only
Type:	Object
Initial value:	TLabel

1.6.5.3.1.65 TBrowseMod:oSelectCol

Selected column in the Browse grid. It corresponds to the nSelectCol property.

Scope:	Read only
Type:	Object
Initial value:	TBrwColMod

1.6.5.3.1.66 TBrowseMod:oSupra

Object that manages the browse supra-header. For more information, see the `AddSupraCol` method.

Scope:	Read only
Type:	Object
Initial value:	THeaderMod

1.6.5.3.2 TBrowseMod:Methods

■ Constructor ■ Standard ■ Only after Create()

Type	Name
■	AddCol
■	AdjustCols
■	AddSupraCol
■	AdjustColWidth
■	Append
■	CancelEdit
■	ClearFilter
■	ColAtPos
■	ColWithHeader
■	DelCol
■	Delete
■	Edit
■	GetDataRow
■	GoBottom
■	GoDown
■	GoLeft
■	GoLeftMost
■	GoRight
■	GoRightMost
■	GoTop
■	GroupOn
■	HitTest
■	HitTestCol
■	InsCol
■	Insert
■	IsFilter
■	IsOnEdit
■	IsSingleEdit
■	IsSorted
■	MinimumRowHeight
■	RestoreState
■	SaveState

■	Sort
■	StretchCols
■	Unsort
■	ToExcel

1.6.5.3.2.1 TBrowseMod:AddCol

Adds a new TBrwColMod object.

Type	Standard
Parameters	<oCol> TBrwColMod object
Return value	NIL

1.6.5.3.2.2 TBrowseMod:AdjustCols

Adjusts the width of all columns in the browse to display the value with the largest width.

Type	Standard
Parameters	[<IWithImage>] If true, the image width on TImagelist control will be taken into account (even if not assigned on that column). By default, false
Return value	NIL

1.6.5.3.2.3 TBrowseMod:AddSupraCol

Adds a super-header to certain Browse columns.

Each time you execute this method, a supra-header will be added to the columns you specify in the Browse, based on your aCols array. The supra-header will wrap as many columns as the value you specify in the <nCols> parameter starting from the first column. The next time you run this method, the initial column to wrap will be the one following the last one processed.

Es importante llamar a este método antes de que se muestre el Browse.

Type	Standard
Parameters	<cHeader> Name for supra header <nCols> Number of columns to be included since the last call to this method [<ISeparator>] Whether a column separator will be displayed. Default True

	[<IResize>] Whether it will be allowed to change the size of the super-column. By default True
	[<IDrag>] Whether it will be allowed to drag a super-column to a new position. By default True
Return value	NIL or THeaderItemMod

1.6.5.3.2.4 TBrowseMod:AdjustColWidth

Adjusts the width of a column to the longest existing text.

Type	Standard
Parameters	<oCol> TBrwColMod object
Return value	NIL

1.6.5.3.2.5 TBrowseMod:Append

Starts the process of adding all editable columns.

Type	Standard
Parameters	None
Return value	True if success

1.6.5.3.2.6 TBrowseMod:CancelEdit

Cancels the current edition.

Type	Standard
Parameters	[<ISave>] f True, saves the new value. By default False Returned Value: True if successful
Return value	True if success

1.6.5.3.2.7 TBrowseMod:ClearFilter

Removes all filters defined in the Browse columns.

Type	Standard
-------------	----------

Parameters	None
Return value	NIL

1.6.5.3.2.8 TBrowserMod:ColAtPos

Returns the existing TBrwColMod object at a given position according to the aDisplayCols array.

Type	Standard
Parameters	<nPos> Objecto position in aDisplayCols array
Return value	NIL or TBrwColMod object

1.6.5.3.2.9 TBrowserMod:ColWithHeader

Returns the existing TBrwColMod object with a given header name.

Type	Standard
Parameters	<cHeader> Header text
Return value	NIL or TBrwColMod object

1.6.5.3.2.10 TBrowserMod:DelCol

Deletes a column

Type	Standard
Parameters	<oCol> / nOrderCol TBrwColMod object / ordinal position
Return value	True if success

1.6.5.3.2.11 TBrowserMod>Delete

Starts the deletion process of the current row.

Type	Standard
Parameters	None
Return value	True if success

1.6.5.3.2.12 TBrowserMod:Edit

Starts the editing process of all editable columns.

Type	Standard
Parameters	[<oCol>] TBrwColMod object to be edited. If this values is NIL, all editable columns become editable
Return value	True if success

1.6.5.3.2.13 TBrowserMod:GetDataRow

Retorna the data array of a given row on its aGridData array.

Type	Standard
Parameters	<nIndex> Row to process
Return value	<aData>

1.6.5.3.2.14 TBrowserMod:GoBottom

Select the last row of the Browse.

Type	Standard
Parameters	None
Return value	NIL

1.6.5.3.2.15 TBrowserMod:GoDown

Select the next row of the Browse.

Type	Standard
Parameters	None
Return value	NIL

1.6.5.3.2.16 TBrowseMod:GoLeft

Selects the previous Browse column.

Type	Standard
Parameters	None
Return value	NIL

1.6.5.3.2.17 TBrowseMod:GoLeftMost

Select the first column of the Browse.

Type	Standard
Parameters	None
Return value	NIL

1.6.5.3.2.18 TBrowseMod:GoRight

Selects the next Browse column.

Type	Standard
Parameters	None
Return value	NIL

1.6.5.3.2.19 TBrowseMod:GoRightMost

Select the last column of the Browse.

Type	Standard
Parameters	None
Return value	NIL

1.6.5.3.2.20 TBrowseMod:GoTop

Select the first row of the Browse.

Type	Standard
Parameters	None
Return value	NIL

1.6.5.3.2.21 TBrowseMod:GroupOn

Performs grouping by values in certain columns.

Type	Standard
Parameters	<p><xCol> Column or array of columns to be included in the grouping.</p> <p>[<ICollapsed>] If the groupings are to be displayed summarized. By default False</p> <p>[<IFooter>] If the groupings will have a footer. By default False</p> <p>[<ILine>] Whether a separator line will be displayed in the header of each grouping. By default False</p>
Return value	TBrwGroup object

1.6.5.3.2.22 TBrowseMod:HitTest

Returns the row of given coordinates

Type	Standard
Parameters	<p><{nX, nY}> Search coordinates</p>
Return value	Coordinate row

1.6.5.3.2.23 TBrowseMod:HitTestCol

Returns the column object that is located at a given coordinate.

Type	Standard
Parameters	<p><nX> X coordinate</p>
Return value	TBrwColMod object or NIL

1.6.5.3.2.24 TBrowseMod:InsCol

Inserts a new TBrwColMod object.

Type	Standard
-------------	----------

Parameters	<oCol> TBrwColMod object <nPos> Position
Return value	NIL

1.6.5.3.2.25 TBrowseMod:Insert

Starts the insertion process of all editable columns..

Type	Standard
Parameters	None
Return value	True if success

1.6.5.3.2.26 TBrowseMod:IsFilter

True if any filter is present.

Type	Standard
Parameters	None
Return value	<IValue>

1.6.5.3.2.27 TBrowseMod:IsOnEdit

True if the browse is in edit mode.

Type	Standard
Parameters	None
Return value	<IValue>

1.6.5.3.2.28 TBrowseMod:IsSingleEdit

True if the browse is in single-column edition status.

Type	Standard
Parameters	None
Return value	<IValue>

1.6.5.3.2.29 TBrowseMod:IsSorted

True if sorted.

Type	Standard
Parameters	None
Return value	<IValue>

1.6.5.3.2.30 TBrowseMod:MinimumRowHeight

Minimum number of pixels high in a row for the image and text to be displayed as a whole.

Type	Standard
Parameters	None
Return value	<nValue>

1.6.5.3.2.31 TBrowseMod:RestoreState

This function retrieves the Browse state through a simple string obtained by calling the SaveState method.

Type	Only after Create()
Parameters	<cState> String with browse configuration
Return value	NIL

1.6.5.3.2.32 TBrowseMod:SaveState

This function returns a simple string with the complete Browse configuration: Row height and visibility and current width and position of each of the columns. It is used in conjunction with the RestoreState method.

Type	Only after Create()
Parameters	None
Return value	<cState> String with the current browse configuration

1.6.5.3.2.33 TBrowserMod:Sort

Sort by a given column.

Type	Standard
Parameters	<oCol> TBrwColMod object [<nSort>] Possible values: <ul style="list-style-type: none">• hsASCENDING• hsDESCENDING By default, it is the opposite of the value you had. If its current value is hsNONE, hsASCENDING will be set.
Return value	True if success

1.6.5.3.2.34 TBrowserMod:StretchCols

Adjusts the width of all Browse columns so that they are visible in their complete width.

Type	Standard
Parameters	<lRight> If true, columns that are not visible from the left will not be taken into account. By default false.
Return value	NIL

1.6.5.3.2.35 TBrowserMod:UnSort

It eliminates the current sorting.

Type	Standard
Parameters	None
Return value	NIL

1.6.5.3.2.36 TBrowserMod:ToExcel

Envía el contenido del control a Excel.

Type	Only after Create()
Parameters	<[cFile]> Name of the XLS file type to be created. If this parameter is left blank the report is sent to Excel and when finished the spreadsheet is displayed.

	<[IXIsNumeric]> If True, all columns of numeric type will be transferred to Excel, also in numeric form. By default is set to True
Return value	<ISuccess> True if success

Nota: Es necesario que **Excel** esté instalado en el equipo.

1.6.5.3.3 TBrowseMod:Events

Name
OnAppend
OnCancelEdit
OnChange
OnChangeSelected
OnClick
OnClickRecSelector
OnContextMenu
OnDelete
OnDbClick
OnDispTooltip
OnDrawActiveRow
OnDrawCell
OnDrawFooter
OnDrawGroupFooter
OnDrawGroupHeader
OnDrawHeader
OnDrawRecSelector
OnEdit
OnEditCol
OnEditColBtnClick
OnFilterEnd
OnFilterStart
OnFooterClick
OnFooterRClick
OnHeaderClick
OnHeaderRClick
OnInsert
OnPostSave
OnSave
OnSaveCol
OnUpdate

1.6.5.3.3.1 TBrowseMod:OnAppend

Event that occurs before the browse row append operation.

Parameters:	<oSender>: Object that fires the event.
Return value:	NIL <IValue> If False, the append process is canceled

1.6.5.3.3.2 TBrowseMod:OnCancelEdit

Event that occurs when the user cancel the edit process.

Parameters:	<oSender>: Object that fires the event <ISave> True if cancellation is desired for the saving of the new values <nKey> Key pressed to quit editing. If zero, it means that you quit editing with the mouse.
Return value:	NIL <IValue> If False, the cancellation process is canceled

1.6.5.3.3.3 TBrowseMod:OnChange

Event that occurs when the active cell is changed.

Parameters:	<oSender>: Object that fires the event <IRowChanged> True if the row has been changed
Return value:	NIL

1.6.5.3.3.4 TBrowseMod:OnChangeSelected

Event that occurs when the selected cells are changed.

Parameters:	<oSender>: Object that fires the event <aSelected> An array with the selected elements
Return value:	NIL

1.6.5.3.3.5 TBrowseMod:OnClick

Event that occurs when the browse data area is clicked by the mouse.

Parameters:	<oSender>: Object that fires the event <nKey> Keyboard and/or mouse status. It can be a combination of any of the following values: MK_CONTROL CTRL key pressed MK_LBUTTON Left mouse button pressed MK_MBUTTON Middle mouse button pressed MK_RBUTTON Right mouse button pressed MK_SHIFT SHIFT key pressed Use the IAnd() function to check status <oCol> TBrwColMod object that was clicked on <nIndex> Actual row number
Return value:	NIL <IValue> If False, the row selection process is canceled.

1.6.5.3.3.6 TBrowseMod:OnClickRecSelector

Event that occurs when the browse record selector area is clicked.

Parameters:	<oSender>: Object that fires the event <nIndex> Current row number <aPos> Cursor position
Return value:	NIL

1.6.5.3.3.7 TBrowseMod:OnContextMenu

Event that occurs when a contextual menu is intended to be displayed when the right mouse button is pressed.

Parameters:	<oSender>: Object that fires the event <nX> X coordinate
--------------------	---

	<nY> Y coordinate <oCol> TBrwColMod object that was clicked on <nIndex> Current row number
Return value:	NIL

1.6.5.3.3.8 TBrowseMod:OnDbClick

Event that occurs when the browse data area is double-clicked by the mouse.

Parameters:	<oSender> : Object that fires the event <nKey> Keyboard and/or mouse status. It can be a combination of any of the following values: MK_CONTROL CTRL key pressed MK_LBUTTON Left mouse button pressed MK_MBUTTON Middle mouse button pressed MK_RBUTTON Right mouse button pressed MK_SHIFT SHIFT key pressed Use the IAnd() function to check status <oCol> TBrwColMod object that was clicked on <nIndex> Actual row number
Return value:	NIL <IValue> If False, the row selection process is canceled.

1.6.5.3.3.9 TBrowseMod:OnDispTooltip

Event that occurs when a browse tool-tip is to be displayed.

Parameters:	<oSender> : Object that fires the event <@cText> Text to be displayed. Passed by reference. If left blank, the tool-tip is not displayed. [<@cTitle>] Title [<@nIcon>] Index of the icon <nIndex> Current record <oCol> Column object on which the cursor is located
--------------------	---

Return value: NIL | <IValue>
If False, the tool-tip display process is canceled.

oSender, @cText, @cTitle, @nIcon, nIndex, oCol)

1.6.5.3.3.10 TBrowseMod:OnDelete

Event that occurs when a deletion process starts.

Parameters: <oSender>
Object that fires the event.

Return value: NIL | <IValue>
If False, the deletion process is canceled.

1.6.5.3.3.11 TBrowseMod:OnDrawActiveRow

Event that occurs when painting the active row. When this event is overloaded, the default painting process stops and it is the user's responsibility to paint.

Parameters: <oSender>
Object that fires the event
<hDC>
ontext device handler
<aRect>
Rectangle with coordinates of the row to be painted
<aSel>
NIL or rectangle with the coordinates of the selected cell
<IFocus>
True if Browse has focus

Return value: NIL

1.6.5.3.3.12 TBrowseMod:OnDrawCell

Event that occurs when each Browse cell is painted.

Parameters: <oSender>
Object that fires the event
<oCol>
TBrwColMod object
<nIndex>
Row number to paint according to present display
<xValue>
Cell original value
<@cValue>

Value transformed to string that will be show ,which can be modified since is passed by reference

<@nImage>
Image ordinal to show of its olmageList property. Its value must be set in any case if you want to display any image. cualquier imagen

<@nClrText>
Text color, subject to change due to being passed by reference

<@nClrPane>
Background color, subject to change due to being passed by reference

<nState>
Cell state:

- STATE_NORMAL (0)
- STATE_SELECTED (1)
- STATE_HOT (2)
- STATE_ACTIVE (3)

<hDC>
Context device handler

<aRect>
Rectangle with coordinates to be painted

Return value: NIL

1.6.5.3.3.13 TBrowseMod:OnDrawFooter

Event that occurs when the Browse footer is painted.

Parameters:

<oSender>:
Object that fires the event

<oltem>
THeaderItemMod object to be painted

<@cText>
Text to be displayed,which can be modified since is passed by reference

<@nImage>
Image ordinal to show of its olmageList property. Its value must be set in any case if you want to display any image. cualquier imagen

<@nClrText>
Text color, subject to change due to being passed by reference

<@nClrPane>
Background color, subject to change due to being passed by reference

<hDC>
Context device handler

<aRect>
Rectangle with coordinates to be painted

Return value: NIL

1.6.5.3.3.14 TBrwMod:OnDrawGroupFooter

Event that occurs when painting the footer of any group.

Parameters:	<p><oSender>: Object that fires the event</p> <p><oSt> TBrwGroupSt object that includes the accounting of this group</p> <p><oCol> TBrwColMod object</p> <p><@cValue> Text to be show, passed by reference so it can be modified. By default an empty string. Normally you will use the method from the oSt object to set its value. For example: <pre>cValue := Transform(oSt:Sum(oCol), "999999")</pre></p> <p><@nClrText> Text color, passed by reference so it can be modified</p> <p><@nClrPane> Background color, passed by reference so it can be modified</p> <p><@ILine> If true a separator line will be shown before the text</p> <p><hDC> Context device handler</p> <p><aRect> Rectangle with the coordinates of the row to be painted</p>
Return value:	<p>NIL <IValue> If false, the painting process is canceled</p>

1.6.5.3.3.15 TBrwMod:OnDrawGroupHeader

Event that occurs when painting the header of any group.

Parameters:	<p><oSender>: Object that fires the event</p> <p><oSt> TBrwGroupSt object that includes the accounting of this group</p> <p><@cGroup> Text to show passed by reference, so it can be modified. By default oSt:Header()</p> <p><nRecord> Ordinal row position based on current browse display</p> <p><hDC> Context device handler</p> <p><aRect> Rectangle with coordinates of the row to be painted</p>
Return value:	<p>NIL <IValue> If False, the painting process is canceled.</p>

1.6.5.3.3.16 TBrowseMod:OnDrawHeader

Event that occurs when the Browse header is painted

Parameters:	<oSender>: Object that fires the event <oltem> THeaderItemMod object to be painted <@cText> Text to be displayed, which can be modified since it is passed by reference <@nImage> Image ordinal to show of its olmageList property. Its value must be set in any case if you want to display any image. cualquier imagen <@nClrText> Text color, subject to change due to being passed by reference <@nClrPane> Background color, subject to change due to being passed by reference <hDC> Context device handler <aRect> Rectangle with coordinates to be painted
Return value:	NIL

1.6.5.3.3.17 TBrowseMod:OnDrawRecSelector

Event that occurs when the Browse record selector is painted.

Parameters:	<oSender>: Object that fires the event <oltem> THeaderItemMod object to be painted <@cText> Text to be displayed, which can be modified since it is passed by reference <@nImage> Image ordinal to show of its olmageList property. Its value must be set in any case if you want to display any image. cualquier imagen <@nClrText> Text color, subject to change due to being passed by reference <@nClrPane> Background color, subject to change due to being passed by reference <hDC> Context device handler <aRect> Rectangle with coordinates to be painted
--------------------	---

Return value: NIL

1.6.5.3.3.18 TBrowserMod:OnEdit

Event that occurs when the editing process starts.

Parameters:	<oSender>: Object that fires the event.
Return value:	NIL <IValue> If true the process is canceled

1.6.5.3.3.19 TBrowserMod:OnEditCol

Event that occurs when a single column editing process starts.

Parameters:	<oSender>: Object that fires the event. <oCol> TBrwColMod object <@Value> Cell value passed by reference, so it can be modified <nKey> Key pushed in case IFastEdit is set to true
Return value:	NIL

1.6.5.3.3.20 TBrowserMod:OnEditColBtnClick

Event that occurs when you click on the button that is part of the editing control.

Parameters:	<oSender>: Object that fires the event. <oCol> TBrwColMod object <@Value> Cell value passed by reference, so it can be modified
Return value:	<IValue> If true its value is updated

1.6.5.3.3.21 TBrowseMod:OnFilterEnd

Event that occurs when the filter ends.

Parameters:	<oSender>: Object that fires the event. <oCol> TBrwColMod that triggers the event
Return value:	NIL

1.6.5.3.3.22 TBrowseMod:OnFilterStart

Event that occurs when the filtering process starts.

Parameters:	<oSender>: Object that fires the event. <oCol> TBrwColMod that triggers the event
Return value:	NIL

1.6.5.3.3.23 TBrowseMod:OnFooterClick

Event that occurs when you click on the browse footer.

Parameters:	<oSender>: Object that fires the event. <oItem> THeaderItemMod pressed <aPos> X,Y coordinates
Return value:	NIL

1.6.5.3.3.24 TBrowseMod:OnFooterRClick

Event that occurs when you right click on the browse footer.

Parameters:	<oSender>: Object that fires the event. <oItem> THeaderItemMod pressed <aPos> X,Y coordinates
Return value:	NIL

1.6.5.3.3.25 TBrowseMod:OnHeaderClick

Event that occurs when you click on the browse header.

Parameters:	<oSender>: Object that fires the event. <oItem> THeaderItemMod pressed <aPos> X,Y coordinates
Return value:	NIL

1.6.5.3.3.26 TBrowseMod:OnHeaderRClick

Event that occurs when you right click on the browse header.

Parameters:	<oSender>: Object that fires the event. <oItem> THeaderItemMod pressed <aPos> X,Y coordinates
Return value:	NIL

1.6.5.3.3.27 TBrowseMod:OnInsert

Event that occurs when an insertion process begins.

Parameters:	<oSender>: Object that fires the event.
Return value:	NIL <IValue> If False, the insertion process is canceled.

1.6.5.3.3.28 TBrowseMod:OnPostSave

Event that occurs when the edit process has been successfully completed.

Parameters:	<oSender>: Object that fires the event.
Return value:	NIL

1.6.5.3.3.29 TBrowserMod:OnSave

Event that occurs when the save process starts after editing. This event is triggered before the OnSaveCol event.

Parameters:	<oSender>: Object that fires the event.
Return value:	NIL <IValue> If false, the save process is canceled

1.6.5.3.3.30 TBrowserMod:OnSaveCol

Event that occurs when the save process of each column of the browser is started.

Parameters:	<oSender>: Object that fires the event. <oCol> TBrwColMod object <@Value> Edition value passed by reference so it can be modified
Return value:	NIL

1.6.5.3.3.31 TBrowserMod:OnUpdate

Event responsible for saving the edit controls in the underlying database when the IAutoSave property is set to false. The **oEditFilter** property of the TBrwColMod object saves the edited values.

Parameters:	<oSender>: Object that fires the event.
Return value:	<IValue> True on correct save

1.6.5.4 TBrwColMod

This class represents each of the columns of the control TBrowserMod.

Hierarchy	TComponent descendant
File	\source\BrwColMod.prg

1.6.5.4.1 TBrwColMod:Properties

■ Read only ■ Assignable ■ Design assignable ■ Run-time assignable

Scope	Name	Type	Initial value
■	aEditListBound	Array	{}
■	aEditListText	Array	{}
■	cFilter	Character	""
■	cFooter	Character	""
■	cHeader	Character	""
■	cPicture	Character	""
■	cTooltip	Character	""
■	FilterEval	Cualquiera	NIL
■	ICanDrag	Logical	.T.
■	ICanEdit	Logical	.T.
■	ICanFilter	Logical	.T.
■	ICanHide	Logical	.T.
■	ICanHot	Logical	.T.
■	ICanResize	Logical	.T.
■	ICanSort	Logical	.T.
■	IDataMultiline	Logical	.F.
■	LeftPos	Numeric	0
■	IMultLine	Logical	.F.
■	IVisible	Logical	.T.
■	ITransparent	Logical	.T.
■	nAlignment	Numeric	taLEFT
■	nBmpAlignment	Numeric	taLEFT
■	nClrPane	Numeric	NIL
■	nDataAlignment	Numeric	taLEFT
■	nDataBmpAlignment	Numeric	taLEFT
■	nDataCol	Numeric	0
■	nDataVAlignment	Numeric	vaCENTER
■	nDroppedHeight	Numeric	300
■	nDroppedWidth	Numeric	0
■	nEditLength	Numeric	-1
■	nEditType	Numeric	beGET
■	nFilterEditStyle	Numeric	esSEARCH
■	nImage	Numeric	0
■	nIndex	Numeric	0
■	nVAlignment	Numeric	vaCENTER
■	oEditFilter	Object	TEditMod
■	oFooterCol	Object	THeaderItemMod
■	oGridEdit	Object	TEditForGrid
■	oHeaderCol	Object	THeaderItemMod
■	oSupraCol	Object	THeaderItemMod
■	RightPos	Numeric	0

■	uDefaultValue	Any	NIL
---	---------------	-----	-----

1.6.5.4.1.1 TBrwColMod:aEditListBound

Array with a list of values in conjunction with the aEditListText array to be used for assignment to the record.

Scope:	Assignable
Type:	Array
Initial value:	{}

When the column is editable with an nEditType style, beLISTBOX type or beGET_LISTBOX, the EditListText array takes the literal list with all the elements from the listbox. You can establish the aEditListBound array with any type of values that will be related to every aEditListText value.

Example:

```
WITH OBJECT oCol
  :nEditType      := beLISTBOX
  :aEditListText := { "Contado", "Talón", "Letra" }
  :aEditListBound := { 1, 2, 3 }
END WITH
```

1.6.5.4.1.2 TBrwColMod:aEditListText

Array with list of values to show in the list-box when the column is editable and has a listbox..

Scope:	Assignable
Type:	Array
Initial value:	{}

When the column is editable with a nEditType style, beLISTBOX type or beGET_LISTBOX, the EditListText array takes the literal list with all the elements from the listbox. You can establish the aEditListBound array with any type of values that will be related to every aEditListText value.

Example:

```
WITH OBJECT oCol
  :nEditType      := beLISTBOX
  :aEditListText := { "Contado", "Talón", "Letra" }
  :aEditListBound := { 1, 2, 3 }
END WITH
```

1.6.5.4.1.3 TBrwColMod:cFilter

Filter text. Normally this property is assigned in run-time by the user when the IFilterBar property of the Browse is True.

Scope:	Assignable
Type:	Character
Initial value:	""

When the Browse to which the column belongs has the property IFilterBar set to True, the header of the Browse will show, in addition to the text, an edit box in which the user can set (type) a filter expression for that column.

The cFilter property will pick up this value. This value in conjunction with the FilterEval property will be used to set the filter of the records to be displayed

The use of the operators '<' and '>' in numeric and date fields is also supported.

1.6.5.4.1.4 TBrwColMod:cFooter

Footer text. This text is only visible in the case that the Browse has its IFooter property set to True.

Scope:	Assignable
Type:	Character
Initial value:	""

1.6.5.4.1.5 TBrwColMod:cHeader

Header text. This text is only visible in the case that the Browse has its IHeader property set to True.

Scope:	Assignable
Type:	Character
Initial value:	""

1.6.5.4.1.6 TBrwColMod:cPicture

Mask based on the xBase standards for the oGridEdit control used in editing.

It will also be used for the painting of the column.

Scope:	Assignable
Type:	Character

Initial value: ""

When the column is editable with an nEditType style of beGET or beGET_LISTBOX or beGET_BUTTON, a mask (or picture) can be set for the oGridEdit control used for editing. The mask follows the same convention as the CA-Clipper mask.

Refer to the [x]Harbour or Clipper documentation for more information on mask (pictures) types.

1.6.5.4.1.7 TBrwColMod:cTooltip

Tool-tip to display..

Scope:	Assignable
Type:	Character
Initial value:	""

Popup window that displays the cTooltip text when the user hovers the mouse momentarily inside the column's 'Header'.



The Tool-tip will only be displayed when the column has its header active, i.e. the IHeader property of its TBrowse container object is set to True.

1.6.5.4.1.8 TBrwColMod:FilterEval

Filter expression to be used.

Scope:	Assignable
Type:	Character o Bloque
Initial value:	NIL

When the Browse to which the column belongs has its property IFilterBar set to True, the header of the Browse will show, in addition to the text, an edit box in which the user can type a filter expression for that column:

The cFilter property will retrieve this value, which in conjunction with this **FilterEval** property will be used to set up the filter of the records to be displayed.

This property works almost automatically with the TDBBrowseMod classes. The filter expressions must be made according to the type of data being handled.

It is preferable to use always an expression of character type because then most of the database engines will be able to solve the filtering optimally by themselves without the program having to do the filtering manually. When the expression is of character type, the following conditions must be taken into account::

- The text entered by the user and collected by the cFilter property must be replaced by the expression %
- Any literal incorporating the expression must necessarily be enclosed in single quotation marks or square brackets []

In the case of a block expression, this block always receives as first parameter a reference to the column object, and as second parameter the text cFilter entered by the user. It must return True for the records to be included in the filter, and False for the records to be excluded.

However when you are filtering arrays and therefore you are using the TBrowseMod class and not the TDBBrowseMod class, the expression becomes more complicated as it is not easy to reference the column being filtered and you need to use a function for filtering which can either be given as a string if you want to assign it in IDE or as a code block if you want to initialize its value by code. The function receives the column to filter and the actual filter value. The function should return the value that we really want to be used to filter the original arra.

Examples:

Possibles filters with TDBBrowseMod:

```
oCol:FilterEval := "Upper('%') $ Upper(Name)"
oCol:FilterEval := "Upper( '%' ) $ ( Familia + ' ' + Upper(" + ::oDbfFam:Alias() +
->Name ) )"
oCol:FilterEval := "Upper( '%' ) $ ( Provee + ' ' + Upper(" + ::oDbfPrv:Alias() + "->Name
) )"
oCol:FilterEval := "Upper('%') = iif( Descatalog, 'S', 'N' )"
```

Possible filters with TBrowseMod:

```
oCol:FilterEval := "FilterNPago(oCol, cFilter)"
-o-
oCol:FilterEval := {|oCol, cFilter| FilterNPago(oCol, cFilter)}

FUNCTION FilterNPago( oCol, cFilter )

    LOCAL nPos

    WITH OBJECT oCol
        nPos := AScan( :aEditListText, {|v| Upper(v) = Upper( cFilter ) } )
        if nPos > 0
            cFilter := ToString( :aEditListBound[ nPos ] )
        ENDIF
    END WITH

return cFilter
```

1.6.5.4.1.9 TBrwColMod:ICanDrag

True if the column can be dragged and moved to another position with the mouse.

Scope:	Assignable
Type:	Logical
Initial value:	.T.

1.6.5.4.1.10 TBrwColMod:ICanEdit

True if the column can be edited.

Scope:	Assignable
Type:	Logical
Initial value:	.T.

1.6.5.4.1.11 TBrwColMod:ICanFilter

True if the column can be filtered.

Scope:	Assignable
Type:	Logical
Initial value:	.T.

1.6.5.4.1.12 TBrwColMod:ICanHide

True if the column can be hidden.

Scope:	Assignable
Type:	Logical
Initial value:	.T.

1.6.5.4.1.13 TBrwColMod:ICanHot

True if the column can be 'hot' when the mouse is hovered over it.

Scope:	Assignable
Type:	Logical
Initial value:	.T.

1.6.5.4.1.14 TBrwColMod:ICanResize

True if the column can be change its size by the user.

Scope:	Assignable
---------------	------------

Type:	Logical
Initial value:	.T.

1.6.5.4.1.15 TBrwColMod:ICanSort

True if the column can be sorted.

Scope:	Assignable
Type:	Logical
Initial value:	.T.

1.6.5.4.1.16 TBrwColMod:IDataMultiLine

True if the text displayed by the browse for this column is multi-line.

Scope:	Assignable
Type:	Logical
Initial value:	.F.

1.6.5.4.1.17 TBrwColMod:LeftPos

Left column coordinate.

Scope:	Read only
Type:	Numeric
Initial value:	0.

1.6.5.4.1.18 TBrwColMod:IMultiLine

True if the header text is multi-line.

Scope:	Assignable
Type:	Logical
Initial value:	.F.

1.6.5.4.1.19 TBrwColMod:IVisible

True if the column is visible.

Scope:	Assignable
Type:	Logical
Initial value:	.T.

1.6.5.4.1.20 TBrwColMod:ITransparent

If True, the column background will be transparent and it respects the background color of its TBrowseMod object.

Scope:	Assignable
Type:	Logical
Initial value:	.T.

1.6.5.4.1.21 TBrwColMod:nAlignment

Column header alignment.

Scope:	Assignable
Type:	Numeric
Initial value:	taLEFT
Possible values:	taLEFT, taRIGHT, taCENTER

1.6.5.4.1.22 TBrwColMod:nBmpAlignment

Alignment of the column header image.

Scope:	Assignable
Type:	Numeric
Initial value:	taLEFT
Possible values:	taLEFT, taRIGHT, taCENTER

1.6.5.4.1.23 TBrwColMod:nClrPane

Background color of the column. This property will only be used if the column is not transparent. In case that this value is NIL, the nClrPane property of its oParent will be used as background color.

Scope:	Assignable
Type:	Numeric
Initial value:	NIL

(Consult appendix for possible colors)

1.6.5.4.1.24 TBrwColMod:nDataAlignment

Data alignment in the column.

Scope:	Assignable
Type:	Numeric
Initial value:	taLEFT
Possible values:	taLEFT, taRIGHT, taCENTER

1.6.5.4.1.25 TBrwColMod:nDataBmpAlignment

Image alignment for data column.

Scope:	Assignable
Type:	Numeric
Initial value:	taLEFT
Possible values:	taLEFT, taRIGHT, taCENTER

1.6.5.4.1.26 TBrwColMod:nDataCol

Index in the second dimension of the aGridData array to which the data to be displayed must refer.

Scope:	Assignable
Type:	Numeric
Initial value:	-1

1.6.5.4.1.27 TBrwColMod:nDataVAlignment

Vertical alignment of data column.

Scope:	Assignable
Type:	Numeric
Initial value:	vaCENTER
Possible values:	vaTOP, vaBOTTOM, vaCENTER

1.6.5.4.1.28 TBrwColMod:nDroppedHeight

Height of the list or drop-down control used for editing cell data.

Scope:	Assignable
Type:	Numeric
Initial value:	300.

1.6.5.4.1.29 TBrwColMod:nDroppedWidth

Width of the list or drop-down control used for editing cell data. A value of zero causes it to be automatically calculated based on the width of the column.

Scope:	Assignable
Type:	Numeric
Initial value:	0.

1.6.5.4.1.30 TBrwColMod:nEditLength

Maximum column editing length.

Scope:	Assignable
Type:	Numeric
Initial value:	-1

This property allows you to specify the maximum column editing length through its oGridEdit control. The value -1 means unlimited length. A value of 0 indicates that the maximum edit length will be automatically adjusted based on the initial value of the edit length.

To be able to edit any column you must use the nEditType property to indicate the

way of editing. If you also want to use a list-box to display a list of possible values, you must also populate the array `aEditListTxt`.

1.6.5.4.1.31 TBrwColMod:nEditType

Type of column edit mode.

Scope:	Assignable
Type:	Numeric
Initial value:	beNONE
Possible values:	beNONE, beGET, beBUTTON, beLISTBOX, beGET_LISTBOX, beGET_BUTTON, beGET_CALENDAR, beCALENDAR, beGET_MEMO, beMEMO

Description:

This property allows you to specify how the column is edited:

beGET	TEditMod control
beBUTTON	TEditMod (Read only) + Button
beLISTBOX	TListboxMod control
beGET_LISTBOX	TEditMod and TListboxMod controls
beGET_BUTTON	TEditMod + Button
beGET_CALENDAR	TEditMod and TCalendarMod controls
beCALENDAR	TCalendarMod control
beGET_MEMO	TEditMod and TMemoMod controls
beMEMO	TMemoMod control

When a column has an edit type other than **beNONE** defined, when pressing **Enter**, **double-clicking** on that column or call the Edit method from code, will enter directly into the edit mode.

The object that is responsible for editing is an object of type `TEditForGrid` which you can access through the `oGridEdit` property. The `oGridEdit` object is available from the creation of the Browse and remains hidden until it is going into edit mode with the `Edit` method, therefore it can also work with the events and properties of this control.

Editing is not modal, i.e. the program does not stop to wait for the user to finish editing the cell, it is the responsibility of the programmer to trap the `OnUpdate` event or or set the `IAutoSave` property to `True`.

To be able to edit any column you must use the **nEditType** property to indicate the edit mode you want. If you also want to use a list-box to display a list of possible values, you must populate the array `aEditListTxt` with all possible values.

1.6.5.4.1.32 TBrwColMod:nFilterEditStyle

Style of the edit control to be used for filtering. For more information see the nEditStyle property of the TEditMod class.

Scope:	Assignable
Type:	Numeric
Initial value:	esNONE
Possible values:	esNONE, esSEARCH, esCLEAR, esPASSWORD, esVERTDOTS, esHORZDOTS, esCOMBO, esARROW, esUSER

1.6.5.4.1.33 TBrwColMod:nImage

Index of image to display from the olmageList property of its container object.

Scope:	Assignable
Type:	Numeric
Initial value:	0

1.6.5.4.1.34 TBrwColMod:nIndex

Number that indicates the order of creation of the control in its container.

Scope:	Read only
Type:	Numeric
Initial value:	0.

1.6.5.4.1.35 TBrwColMod:nVAlignment

Vertical alignment of the column header.

Scope:	Assignable
Type:	Numeric
Initial value:	vaCENTER
Possible values:	vaTOP, vaBOTTOM, vaCENTER

1.6.5.4.1.36 TBrwColMod:oEditFilter

TEditMod object responsible of the filter edition.

Scope:	Read only
Type:	TEditMod object
Initial value:	TEditMod

1.6.5.4.1.37 TBrwColMod:oFooterCol

Reference to the THeaderItemMod object shown on column footer.

Scope:	Read only
Type:	THeaderItemMod object
Initial value:	THeaderItemMod

1.6.5.4.1.38 TBrwColMod:oGridEdit

Reference to the TEditForGrid object responsible of data editing.

Scope:	Read only
Type:	TEditForGrid object
Initial value:	TEditForGrid

1.6.5.4.1.39 TBrwColMod:oHeaderCol

Reference to the THeaderItemMod object that shows the column header.

Scope:	Read only
Type:	THeaderItemMod object
Initial value:	THeaderItemMod

1.6.5.4.1.40 TBrwColMod:oSupraCol

Reference to the THeaderItemMod object that show the supra-header on this column.

Scope:	Read only
Type:	THeaderItemMod object
Initial value:	NIL

1.6.5.4.1.41 TBrwColMod:RightPos

Right column coordinate.

Scope:	Read only
Type:	Numeric
Initial value:	0.

1.6.5.4.1.42 TBrwColMod:uDefaultValue

Default value of the column for adding and inserting processes.

Scope:	Assignable
Type:	Cualquiera
Initial value:	Nil

This property allows you to set the default value that the oGridEdit object will have in the add and insert processes. If this property has not been assigned its value will be automatically calculated based on its type.

1.6.5.4.2 TBrwColMod:Methods

■ Constructor ■ Standard ■ Only after Create()

Type	Name
■	Value

1.6.5.4.2.1 TBrwColMod:Value

Returns the column value for a specific row.

Type	Standard
Parameters	[<nRecord>] Number of the row according to current display. By default the active row
Return value	<xValue>

1.6.5.5 TBrwModGrid

This class is responsible for displaying all the information of the data grid. It is basically a class that inherits from TListBoxMod and adds the following read-only properties that allow accessing members of its container in a easier way, which are:

- **oBrowse** oints to your TBrowseMod
- **aCols** which points to TBrowseMod:aCols
- **aDisplayCols** which points to TBrowseMod:aDisplayCols
- **oImageList** which points to TBrowseMod:oImageList
- **aGridData** which points to TBrowseMod:aGridData

1.6.5.6 TEditForGrid

This class is responsible for editing the grid data. It is basically a class that inherits from TEditMod and adds some additional properties to be able to display a list, calendar or multi-line editor:

- **oBrowse** wich points to your TBrowseMod
- **oCol** wich points to the TBrwColMod column on which relies
- **IDropped** which indicates whether the control is expanded. This is a Read-Only property
- **IDropAnimation** if true, an animation is displayed when unfolding
- **nClrDropPane, nClrDropText, nClrDropHotPane, nClrDropHotText, nClrDropSelPane, nClrDropSelText** hat allow you to modify the colors to be displayed in the drop-down list
- **nDropMagin** to set the margin with which the list is displayed. Default to 10
- **nDropItemHeight** o indicate the height of each row in the drop-down list. Default to 25

1.6.5.7 TBrwGroup

This class is responsible for performing all calculations and painting of the groups when the **GroupOn** method is used in the TBrowseMod control.

Hierarchy TComponent descendant
File \source\BrowseMod.prg

1.6.5.7.1 TBrwGroup:Properties

■ Read only ■ Assignable ■ Design assignable ■ Run-time assignable

Scope	Name	Type	Initial value
■	aGroupCols	Array	{}
■	lCollapsed	Logical	.F.
■	lFooter	Logical	.T.
■	lLine	Logical	.T.

1.6.5.7.1.1 TBrwGroup:aGroupCols

Array with the TBrwColMod involved in the grouping. The columns have to be set through the GroupOn method.

Scope:	Read only
Type:	Array
Initial value:	{}

1.6.5.7.1.2 TBrwGroup:!Collapsed

If True, by default, the groups will be displayed in summary (collapsed) form.

Scope:	Assignable
Type:	Logical
Initial value:	.F.

1.6.5.7.1.3 TBrwGroup:!Footer

If True, a footnote will be displayed for each group.

Scope:	Assignable
Type:	Logical
Initial value:	.F.

1.6.5.7.1.4 TBrwGroup:!Line

If True, a line will be displayed in the header of each group.

Scope:	Assignable
Type:	Logical
Initial value:	.F.

1.6.5.7.2 TBrwGroup:Methods

■ Constructor ■ Standard ■ Only after Create()

Type	Name
■	SumTotal

1.6.5.7.2.1 TBrwGroup:SumTotal

Returns the total sum of all grouping counts for a specific column.

Type	Standard
Parameters	<xCol> TBrwColMod object on which the information or number in the second dimension of the data array aGridData is required
Return value	<nValue>

1.6.5.8 TBrwGroupSt

This class represents each of the parts that the Browse grouping has for all the columns of the browse.

Hierarchy	TComponent descendant
File	\source\BrowseMod.prg

1.6.5.8.1 TBrwGroupSt:Methods

■ Constructor ■ Standard ■ Only after Create()

Type	Name
■	Average
■	Header
■	Max
■	Min
■	Sum

1.6.5.8.1.1 TBrwGroupSt:Average

Returns the arithmetic average of the group count for a specific column.

Type	Standard
Parameters	<xCol> TBrwColMod object on which the information or the number in the second dimension of the data array aGridData is required
Return value	<nValue>

1.6.5.8.1.2 TBrwGroupSt:Header

Returns the default text to be displayed in the header of the group list. This text consists of a simple concatenation of all the values of the columns involved in the grouping.

Type	Standard
Parameters	None
Return value	<cValue>

1.6.5.8.1.3 TBrwGroupSt:Max

Returns the maximum value of the group count for a specific column.

Type	Standard
Parameters	<xCol> TBrwColMod object on which the information or the number in the second dimension of the data array aGridData is required
Return value	<nValue>

1.6.5.8.1.4 TBrwGroupSt:Min

Returns the minimum value of the group count for a specific column.

Type	Standard
Parameters	<xCol> TBrwColMod object on which the information or the number in the second dimension of the data array aGridData is required
Return value	<nValue>

1.6.5.8.1.5 TBrwGroupSt:Sum

Returns the sum of the group count for a specific column.

Type	Standard
Parameters	<xCol> TBrwColMod object on which the information or the number in the second dimension of the data array aGridData is required
Return value	<nValue>

1.6.5.9 TBtnPanelMod

This class represents a Panel control, container of controls, that behaves as a button with a 'hot' effect and an OnClick event that works in the whole area of the control even if it has other static controls as labels..

Hierarchy TWinControl descendant
See also TScrollBar
File \source\Panel.prg

1.6.5.9.1 TBtnPanelMod:Properties

■ Read only ■ Assignable ■ Design assignable ■ Run-time assignable

Scope	Name	Type	Initial value
■	IHotBorder	Logical	.T.
■	nClrPane	Numeric	clBlack
■	nHeight	Numeric	100
■	nHotOpacity	Numeric	25
■	nPushedOpacity	Numeric	50
■	nWidth	Numeric	100

1.6.5.9.1.1 TBtnPanelMod:IHotBorder

Displays a border on the panel when the mouse is over it.

Scope	Assignable
Type	Logical
Initial value	.T.

1.6.5.9.1.2 TBtnPanelMod:nClrPane

Background color of the control.

Scope:	Assignable
Type:	Numeric
Initial value:	clBlack

(Consult the appendix for possible colors)

1.6.5.9.1.3 TBtnPanelMod:nHeight

Height of the control.

Scope	Assignable
Type	Numeric
Initial value	100

1.6.5.9.1.4 TBtnPanelMod:nHotOpacity

Opacity level when the panel has the 'hot' effect due to the mouse being over it.

Scope	Assignable
Type	Numeric
Initial value	25

1.6.5.9.1.5 TBtnPanelMod:nPushedOpacity

Opacity level when the panel has been clicked.

Scope	Assignable
Type	Numeric
Initial value	50

1.6.5.9.1.6 TBtnPanelMod:nWidth

Control width.

Scope	Assignable
Type	Numeric
Initial value	120

1.6.5.10 TCalendarMod

This class represents a Calendar control adapted to Windows 10 style.

Hierarchy	TWinControl descendant
File	source\CalendarMod.prg

1.6.5.10.1 TCalendarMod:Properties

■ Read only ■ Assignable ■ Design assignable ■ Run-time assignable

Scope	Name	Type	Initial value
■	aDays	Array	Windows local information
■	aDaysSelected	Array	{}
■	aMonths	Array	Windows local information
■	aSDays	Array	Windows local information
■	aSMonths	Array	Windows local information
■	dValue	Date	Date()
■	lHighLiteToday	Logical	.T.
■	lShowDOW	Logical	.T.
■	lShowFirstOfGroup	Logical	.F.
■	lShowLines	Logical	.T.
■	lTransparent	Logical	.F.
■	nClrBorder	Numeric	clActiveBorder
■	nClrDaysDisabled	Numeric	cl3DLight
■	nClrHot	Numeric	clGray
■	nClrSelection	Numeric	clSystem
■	nClrTextHeader	Numeric	clWindowText
■	nCrToday	Numeric	clSystem
■	nDisplayMode	Numeric	dmMonth
■	nNumberOfWeeks	Numeric	6
■	nSelectionMode	Numeric	smSingle
■	oFontDOW	TFont	NIL
■	oFontHeader	TFont	NIL
■	oHeader	TButtonMod	NIL
■	oNext	TButtonMod	NIL
■	oPrev	TButtonMod	NIL

1.6.5.10.1.1 TCalendarMod:aDays

Array with the literal of long names of each day of the week. By default it collects the information provided by the operating system.

Scope	Run-time assignable
Type	Array
Initial value	Windows local information

1.6.5.10.1.2 TCalendarMod:aDaysSelected

Date array with all selected dates..

Scope	Run-time assignable
Type	Array
Initial value	{}

1.6.5.10.1.3 TCalendarMod:aMonths

Array with the literal of long names of each month. By default it collects the information provided by the operating system..

Scope	Run-time assignable
Type	Array
Initial value	Windows local information

1.6.5.10.1.4 TCalendarMod:aSDays

Array with the literal of short names of each day of the week. By default it collects the information provided by the operating system..

Scope	Run-time assignable
Type	Array
Initial value	Windows local information

1.6.5.10.1.5 TCalendarMod:aSMonths

Array with the literal of short names of each month. By default it collects the information provided by the operating system.

Scope	Run-time assignable
Type	Array
Initial value	Windows local information

1.6.5.10.1.6 TCalendarMod:dValue

Current date selected in the control.

Scope	Assignable
Type	Date
Initial value	Date()

1.6.5.10.1.7 TCalendarMod:IHighLiteToday

If true, the current day will be highlighted.

Scope	Assignable
Type	Logical
Initial value	.T.

1.6.5.10.1.8 TCalendarMod:IShowDOW

If true, the days of the week will be displayed.

Scope	Assignable
Type	Logical
Initial value	.T.

1.6.5.10.1.9 TCalendarMod:IShowFirstOfGroup

If true, a group heading will be displayed depending on the type of display.

Scope	Assignable
Type	Logical
Initial value	.F.

1.6.5.10.1.10 TCalendarMod:IShowLines

If true, separator lines will be displayed for each sub-element of the control.

Scope	Assignable
Type	Logical
Initial value	.T.

1.6.5.10.1.11 TCalendarMod:ITransparent

The control is drawn transparent on the form to which it belongs.

Scope	Assignable
Type	Logical
Initial value	.T.

1.6.5.10.1.12 TCalendarMod:nClrBorder

Color for border and inner lines

Scope	Assignable
Type	Numeric
Initial value	clActiveBorder

(See the Appendix for possible colors)

1.6.5.10.1.13 TCalendarMod:nClrDaysDisabled

Color for disabled days.

Scope	Assignable
Type	Numeric
Initial value	cl3DLight

(See the Appendix for possible colors)

1.6.5.10.1.14 TCalendarMod:nClrHot

Color for the item that has the mouse over it.

Scope	Assignable
Type	Numeric
Initial value	clGray

(See the Appendix for possible colors)

1.6.5.10.1.15 TCalendarMod:nClrSelection

Color for selected days.

Scope	Assignable
Type	Numeric
Initial value	clSystem

(See the Appendix for possible colors)

1.6.5.10.1.16 TCalendarMod:nClrTextHeader

Text color for header texts.

Scope	Assignable
Type	Numeric
Initial value	clWindowText

(See the Appendix for possible colors)

1.6.5.10.1.17 TCalendarMod:nClrToday

Background color of the current day.

Scope	Assignable
Type	Numeric
Initial value	clSystem

(See the Appendix for possible colors)

1.6.5.10.1.18 TCalendarMod:nDisplayMode

Display mode.

Scope	Assignable
Type	Numeric
Initial value	dmMonth
Valores posibles	dmMonth: por meses dmYear: por años dmDecade: por décadas

1.6.5.10.1.19 TCalendarMod:nNumberOfWeeks

Number of weeks to display when display mode is by months.

Scope	Assignable
Type	Numeric
Initial value	6

1.6.5.10.1.20 TCalendarMod:nSelectionMode

Dates selection method.

Scope	Assignable
Type	Numeric
Initial value	smSingle
Valores posibles	smSingle: Sólo una Date smMultiple: Múltiples Dates smNone: Ninguna Date es seleccionable

1.6.5.10.1.21 TCalendarMod:oFontDOW

Font to be used for weekday texts.

Scope	Assignable
Type	TFont
Initial value	NIL

1.6.5.10.1.22 TCalendarMod:oFontHeader

Font to be used for header texts.

Scope	Assignable
Type	TFont
Initial value	NIL

1.6.5.10.1.23 TCalendarMod:oHeader

Object of type TButtonMod that displays the selected month, year or decade, according to the value of nDisplayMode.

Scope	Read only
Type	TButtonMod
Initial value	NIL

1.6.5.10.1.24 TCalendarMod:oNext

Object of type TButtonMod that displays the button of the next selected month, year or decade, according to the value of nDisplayMode.

Scope	Read only
Type	TButtonMod
Initial value	NIL

1.6.5.10.1.25 TCalendarMod:oPrev

Object of type TButtonMod that displays the button of the previous selected month, year or decade, according to the value of nDisplayMode.

Scope	Read only
Type	TButtonMod
Initial value	NIL

1.6.5.10.2 TCalendarMod:Events

Name
OnChange
OnChangeView
OnCheckDate
OnDrawItem
OnSelect

1.6.5.10.2.1 TCalendarMod:OnChange

Event that occurs when there is a change in the calendar.

Parameters:	<oSender>
--------------------	-----------

	Object the fires the event
Return value:	<NIL>

1.6.5.10.2.2 TCalendarMod:OnChangeView

Event that occurs when there is a change in the display mode due to a change in the nDisplayMode property.

Parameters:	<oSender> : Object the fires the event <nOldMode> Old mode <nNewMode> New mode
Return value:	<NIL>

1.6.5.10.2.3 TCalendarMod:OnCheckDate

Event that occurs when there is a date validation.

Parameters:	<oSender> : Object the fires the event <dDate> Date to validate
Return value:	<NIL> or <.T.> validates the date. A false value marks it as not selectable and the date is strike-out

1.6.5.10.2.4 TCalendarMod:OnDrawItem

Event that occurs when any day of the calendar is painted.

Parameters:	<oSender> : Object the fires the event <dDay> Date <@nClrText> Text color. Passed by reference. Editable <@nClrPane> Text background color. Passed by reference. Editable <hDC> Handle to context device <aRect> Rectangle to paint
--------------------	--

Return value: <NIL> or <.T.> process the painting. A return value forces the complete painting on the event using **hDC** and **aRect**.

1.6.5.10.2.5 TCalendarMod:OnSelect

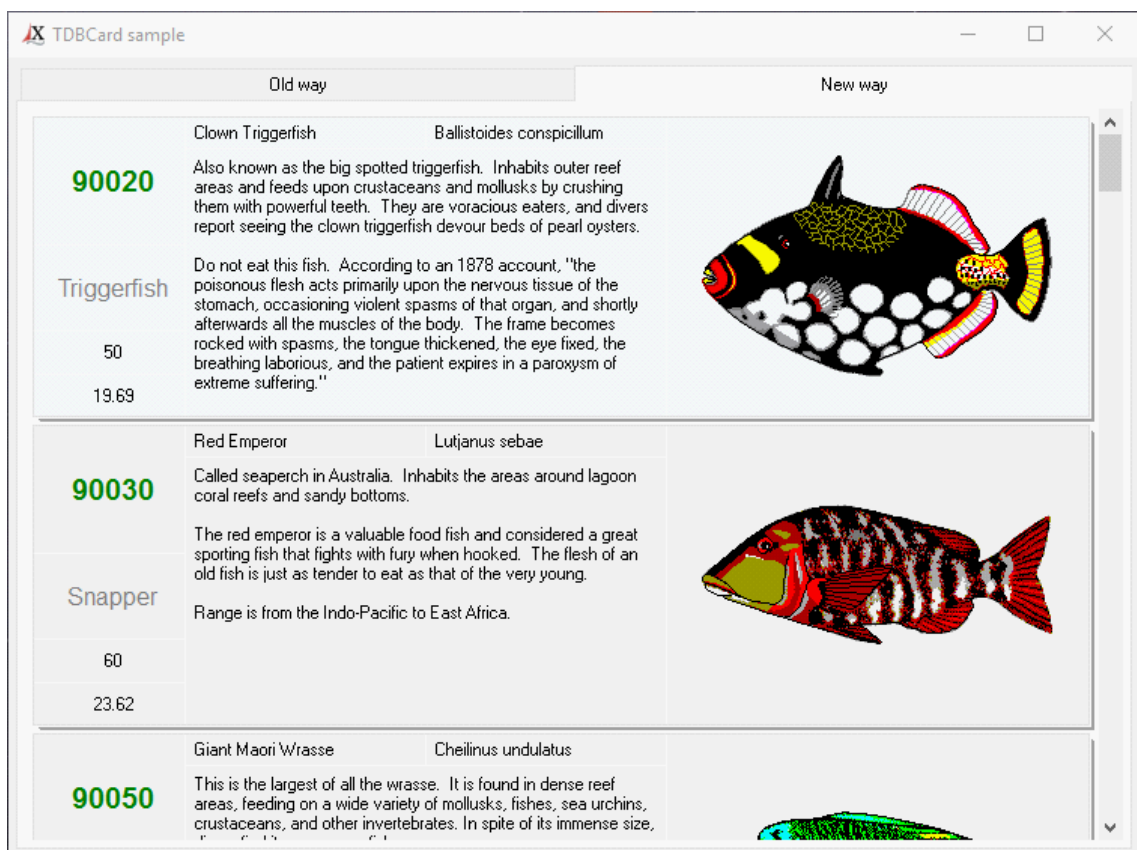
Event that occurs when a date is selected.

Parameters: <oSender>:
Object the fires the event
<dDate>
Date

Return value: <NIL>

1.6.5.11 TCardBox

This class represents a card box control which is specially indicated to replace a lot of browses, but specially to be a fundamental control for tablets.

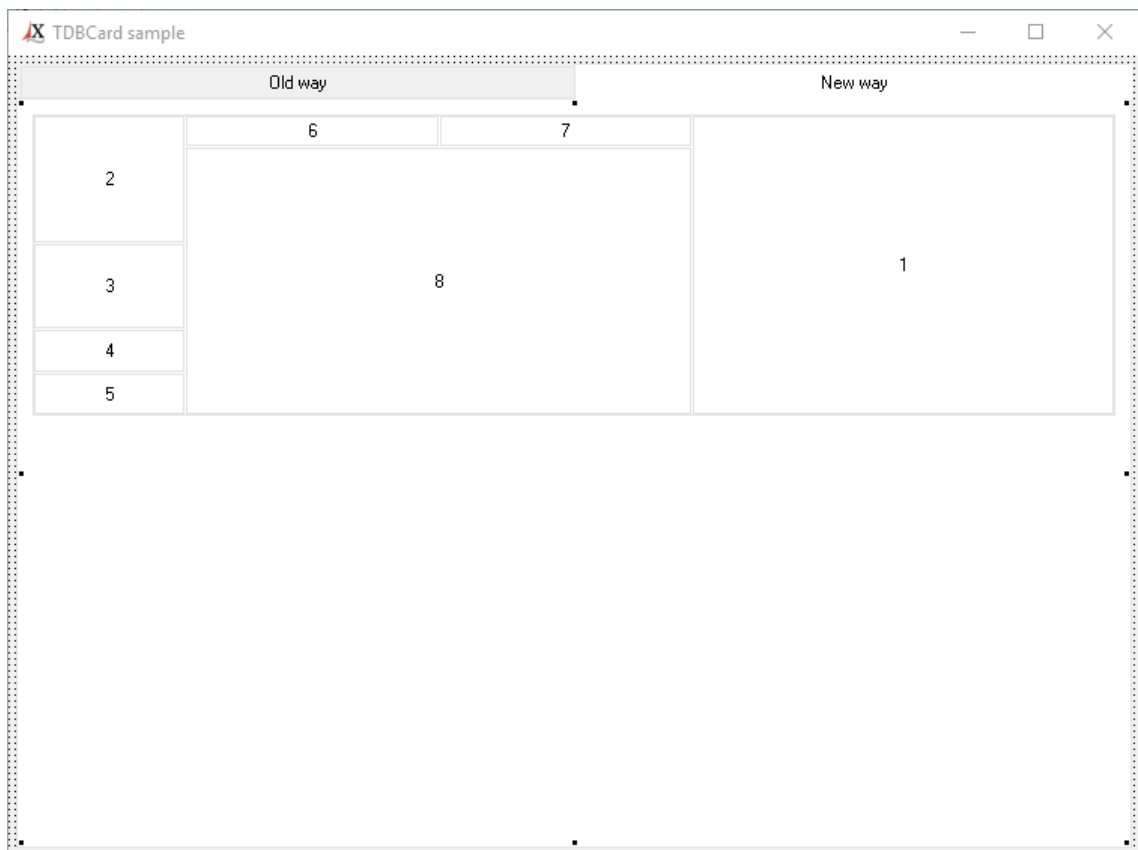


Hierarchy	TScrollingWinControl descendant
See also	TCardItem
File	\source\CardBox.prg
Samples	\samples\Cards\

1.6.5.11.1 TCardBox:Introduction

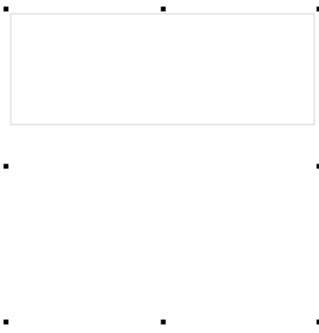
It might seem that the **TCardBox** control are just many panels aligned to **alTOP**, in which each one contains other controls with different alignments. Not at all, **TCardBox** is a unique control, with a single window handle that is prepared to process millions of records as fast as a dozen of them would treat. In terms of speed and resource consumption it is equivalent to a **TBrowse** control. Therefore, do not hesitate to use it as if it were a **TBrowse** object.

Keeping in mind the great speed and low consumption of resources required, an absolutely new control has been developed that does not exist in other development environments and whose design is totally visual and very simple. However, it requires a small introductory explanation that clearly shows how it works. Here is the card design shown at the beginning of this control main page:



As you can see, in design time only one card is displayed and it is divided into several sections, which will show each of the elements that we want to visualize. Each of these sections inside the card is a **TCardItem** control, which is a simple object inherited from the **TComponent** base class which has a series of properties to set its parameters, like Alignment, color, and text or image to display. The number that shows each of the **TCardItem** matches your creation order, which is just as important of controls alignment. When you create a **TCardBox** object this is what is

shown in the **IDE**:



Only a rectangle of what the card occupies is shown. That is all. In order to display something in the control it is necessary to add a **TCardItem** control, which would be the equivalent of adding a column in a **TBrowse**. When you add a control through its context menu you will see the following:



Notice how there is a new rectangle with the number '1' inside the card. This rectangle is a **TCardItem** control and would be the equivalent of a **TBrowse** column object. If you click on the rectangle with the mouse you can start editing all its properties just as you would with a column object.

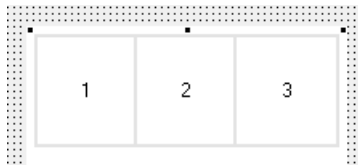
There are four properties that indicate the occupation of the item in its container, which are:

1. Its property **nSize** which indicates its size.
2. Its property **nAlign** which sets its alignment inside the card.
3. Its property **nAlignWeight** which indicates its weight with other items with same alignment and consecutive creation.
4. Its property **nSiblingWeight** (weight between siblings) which sets the the weight of an item respect a base item (its eldest sibling).

The **nSize** property indicates the size of the item. Note that if the orientation is vertical, its value indicates the width of the control, while if it is horizontal it will indicate its height.

The **nAlign** property indicates the alignment in the same way as a **TPanel** control on a form. In fact, it can have the same values: **alTOP**, **alBOTTOM**, **alLEFT**, **alRIGHT** and **alCLIENT** (except **alNONE**). The alignment **alLEFT** is the one that has been set in the image above.

The **nAlignWeight** property indicates the weight of the item relative to other items with the same alignment that are consecutive in creation. When this property is nonzero, the **nSize** property becomes useless since its size will be calculated based on the weight indicated in this property. Suppose three items we want to split the card into three vertical sections of equal size. Something like this:

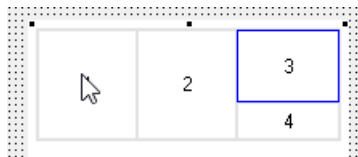


(items with same **nAlignWeight** value)

To do this, we would only have to indicate that all items have `alLEFT` orientation and that all of their **nAlignWeight** property has a value **1**, ie all items have the same weight. Note that it is the same to put all items in the value **1** in this property that any other value, because in any case all three will have the same weight and therefore, its size will be evenly distributed. As you may have guessed, to make one item double the other two, you only have to change the value of the **nAlignWeight** property to **2** for that particular item. As soon as you change the alignment of any item and make it different from the previous one the weight system is initialized for the new alignment.

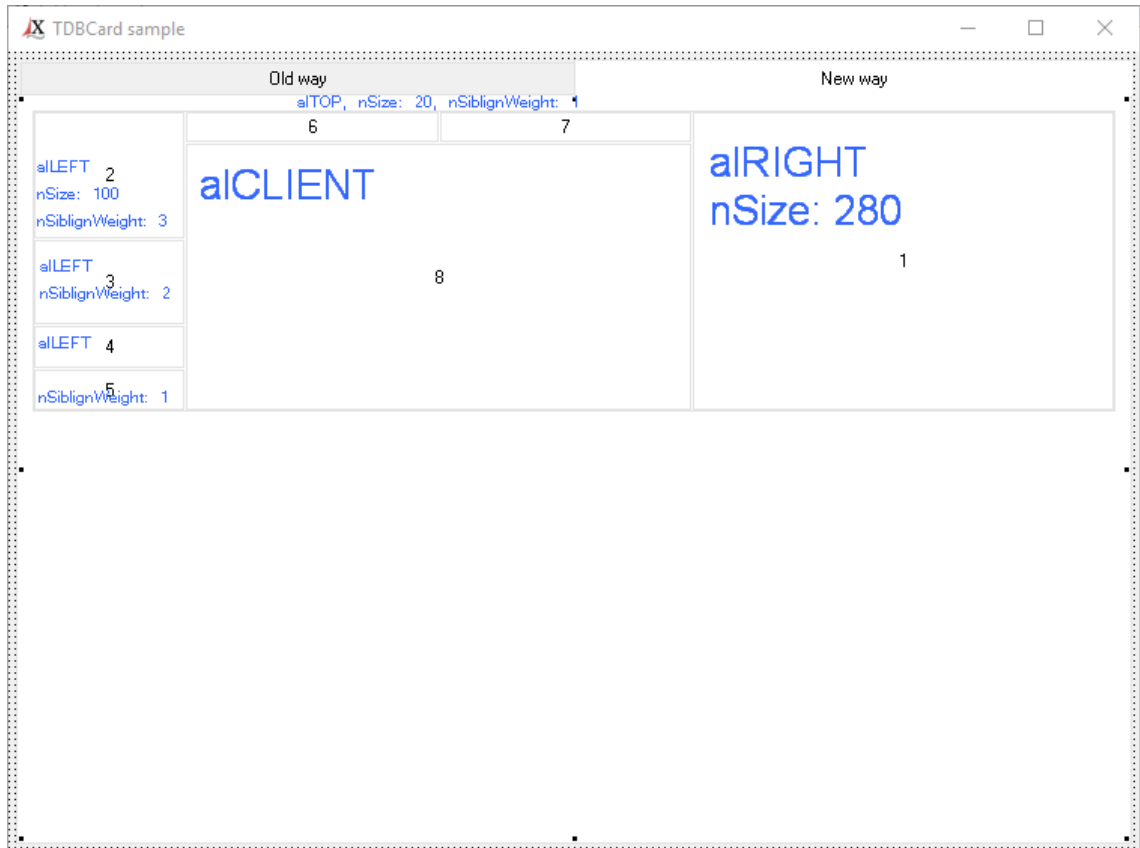
This alignment system can still be more powerful thanks to the **nSiblingWeight** property that allows you to set multiple items as siblings that share the size or weight of the older sibling. The older sibling would be the first to have a value on this nonzero property. Thereafter, all items that have the same alignment (`taLEFT`), their **nAlignWeight** property is set to zero and have a nonzero value in their **nSiblingWeight** property behave like siblings and they will all share the space that would occupy his older sibling as if it were alone.

It is easy to understand with an example: Suppose we want to split item 3, in two parts making the first piece triple that the second. Something like this:



We have simply created an additional item (4), we have given it the same alignment (`taLEFT`) as the rest, we have set its **nSiblingWeight** property to **1** and to Item 3 (oldest sibling) we have set its **nSiblingWeight** property to **3**. That is all.

As a final exercise we will see in the example card the values that have these properties:



As with the columns of a **TBrowse** control, in a **TCardBody** control, each item is responsible for displaying its information. In a **TBrowse** object, the number of rows in the **DB** is marked by the size of an array, the number of records in a **DBF** file, or the number of rows in a dataset. In a **TCardBody** object it is the same, but instead of rows, we will have cards. That is all.

The **TCardBody** control uses a multi-column array that is set to its **aData** property as the data source. Each **TCardBodyItem** of the control should indicate the column number to use and use its **nColumn** property. This name has been used precisely to see the similarities between a **TCardBody** object and a **TBrowse** object.

When you use a **TDbCardBody** control, the data source is no longer an array, but a dataset, and each **TCardBodyItem** control will have a property of **oDataField** That allows you to set the information to display.

Like a **TBrowse** there is always an active card that corresponds to the **nIndex** property, which can also be modified to change the active card by code.

TCardBody also supports a multi-selection card mode with **IMultipleSel** and **aSelected** properties, which works identically to **TBrowse**.

A **TCardBodyItem** object can display the information in different ways through **nType**, which can be:

- **ctLABEL**: Simple text.
- **ctLABLEX**: Text that supports the same properties as a **TLabelEX** control, such as HTML commands for example. In addition, when using this type, if the text does not fit on the card, clicking the object will show a scroll bar that will allow you to scroll through the entire text.
- **ctPICTURE**: An image, can be a **TPicture** object or a image stream.

- **ctIMAGEINDEX**: Image ordinal on its **TCardBody olmageList** property.

On the control main page indicates that the control inherits from **TScrollingWinControl**, but indeed inherits from **TGestureControl** class which simply adds gestures support to **TScrollingWinControl**. The class **TGestureControl** is not documented since it has not any public member for the programmer.

1.6.5.11.2 TCardBody:Properties

■ Read only ■ Assignable ■ Design assignable ■ Run-time assignable

Scope	Name	Type	Initial value
■	aData	Array	{}
■	altems	Array	{}
■	aSelected	Array	{}
■	lCardBody	Logical	.T.
■	lCardBodyShadow	Logical	.F.
■	lGapFirstCard	Logical	.T.
■	lHideScrollBars	Logical	.F.
■	lHotTrack	Logical	.F.
■	lMultipleSel	Logical	.F.
■	nCardBodyClrHot	Numeric	clHotLight
■	nCardBodyClrPane	Numeric	clWindow
■	nCardBodyClrAct	Numeric	NIL
■	nCardBodyClrLink	Numeric	clBlue
■	nCardBodyClrSel	Numeric	clActiveCaption
■	nCardBodyGap	Numeric	5
■	nCardBodyInnerGap	Numeric	0
■	nCardBodyMargin	Numeric	5
■	nCardBodySize	Numeric	70
■	nClickItem	Numeric	0
■	nClickPos	Numeric	0
■	nHeight	Numeric	200
■	nHotItem	Numeric	0
■	nHotPos	Numeric	0
■	nIndex	Numeric	1
■	nWidth	Numeric	200
■	olmageList	Object	TImageList
■	oLabelEx	Object	TLabelEx

1.6.5.11.2.1 TCardBody:aData

Multidimensional array with all the data to show.

Scope:	Assignable
Type:	Array
Initial value:	{}

Array format:

```
{ {Row1Col1, Row1Col2, ..., Row1ColN }, ..., {RowNCol1, RowNCol2, ..., RowNColN } }
```

1.6.5.11.2.2 TCardBox:altems

List of card box sections, represented by TCardItem objects.

Scope	Design assignable
Type	Array
Initial value	{}

1.6.5.11.2.3 TCardBox:aSelected

Array of numeric values with all the card box selected. This property is used in conjunction with IMultipleSel property.

Scope	Assignable
Type	Array
Initial value	{}

1.6.5.11.2.4 TCardBox:lCardBorder

If true the card box will have a border. The border is not modifiable, its always **cl3DLight**.

Scope	Assignable
Type	Logical
Initial value	.T.

1.6.5.11.2.5 TCardBox:lCardShadow

If true the card box will show a small shadow.

Scope	Assignable
Type	Logical
Initial value	.F.

1.6.5.11.2.6 TCardBox:IGapFirstCard

If true, the first card will be shown with a small gap, based on its property nCardGap.

Scope	Assignable
Type	Logical
Initial value	.F.

1.6.5.11.2.7 TCardBox:IHideScrollBars

If true, the scroll bars are only visible when the mouse cursor reaches the area were they should be.

Scope	Assignable en design
Type	Logical
Initial value	.F.

1.6.5.11.2.8 TCardBox:IHotTrack

If true the card box will support hot tracking.

Scope	Assignable
Type	Logical
Initial value	.F.

1.6.5.11.2.9 TCardBox:IMultipleSel

Permits the selection of more than one card. Use this property in conjunction with aSelected property to retrieve all the cards selected.

Scope:	Assignable
Type:	Logical
Initial value:	.F.

1.6.5.11.2.10 TCardBox:nCardClrHot

Color to use with the current hot tracked card.

Scope:	Assignable
---------------	------------

Type:	Numeric
Initial value:	clHotLight

Consult for appendix for possible colors.

1.6.5.11.2.11 TCardBox:nCardClrPane

Background card color.

Scope:	Assignable
Type:	Numeric
Initial value:	clWindow

Consult for appendix for possible colors.

1.6.5.11.2.12 TCardBox:nCardClrAct

Background active card color. If this value is NIL, the active card will be painted with semi-opaque effect.

Scope:	Assignable
Type:	Numeric
Initial value:	NIL

Consult for appendix for possible colors.

1.6.5.11.2.13 TCardBox:nCardClrLink

Default color for links on items with use TLabelEx type.

Scope:	Assignable
Type:	Numeric
Initial value:	clBlue

Consult for appendix for possible colors.

1.6.5.11.2.14 TCardBox:nCardClrSel

Background selected card color.

Scope:	Assignable
Type:	Numeric
Initial value:	clActiveCaption

Consult for appendix for possible colors.

1.6.5.11.2.15 TCardBox:nCardGap

Disntance in pixels between cards.

Scope:	Assignable
Type:	Numeric
Initial value:	5

1.6.5.11.2.16 TCardBox:nCardInnerGap

Distance in pixels between each section of card.

Scope:	Assignable
Type:	Numeric
Initial value:	0

1.6.5.11.2.17 TCardBox:nCardMargin

Left and right margin of cards.

Scope:	Assignable
Type:	Numeric
Initial value:	5

1.6.5.11.2.18 TCardBox:nCardSize

Card size (it corresponds with its height).

Scope:	Assignable
Type:	Numeric
Initial value:	70

1.6.5.11.2.19 TCardBox:nClickItem

Ordinal in altems of the last TCardItem object selected with the mouse. This property only has value while the mouse left button is pushed and therefore can only be trapped on its OnCardClick event.

Scope:	Read only
Type:	Numeric
Initial value:	0

1.6.5.11.2.20 TCardBox:nClickPos

Ordinal in aData of the last card selected. This property only has value while the mouse left button is pushed and therefore can only be trapped on its OnCardClick event.

Scope:	Read only
Type:	Numeric
Initial value:	0

1.6.5.11.2.21 TCardBox:nHeight

Control height.

Scope:	Assignable
Type:	Numeric
Initial value:	200

1.6.5.11.2.22 TCardBox:nHotItem

Ordinal in altems of the actual TCardItem with hot track.

Scope:	Read only
Type:	Numeric
Initial value:	0

1.6.5.11.2.23 TCardBox:nHotPos

Ordinal in aData of the actual card with hot track.

Scope:	Read only
Type:	Numeric
Initial value:	0

1.6.5.11.2.24 TCardBox:nIndex

Active card.

Scope:	Assignable
Type:	Numeric
Initial value:	1

1.6.5.11.2.25 TCardBox:nWidth

Control width.

Scope:	Assignable
Type:	Numeric
Initial value:	200

1.6.5.11.2.26 TCardBox:oImageList

TImageList object with all the images used by the control.


Scope	Assignable
Type	Object
Initial value	TImageList

The TImageList object is instantiate on the New() constructor. The first image included on the TImageList sets the dimension of future images introduced. If the first image is a image combination is important that you set the nHeight and nWidth TImageList properties before adding any image.

1.6.5.11.2.27 TCardBox:oLabelEx

TLabelEx object that is used solely to show TCardItem information with type **ctLABLELEX** and the text that nos fit on the card and is selected by the mouse. The rest of the time is hidden. Its presence can be noted for the existence of a vertical scroll bar on that section of the card.

Scope	Assignable
Type	Object
Initial value	TLabelEx

	Clown Triggerfish	Ballistoides conspicillum	
90020	Also known as the big spotted triggerfish. Inhabits outer reef areas and feeds upon crustaceans and mollusks by crushing them with powerful teeth. They are voracious eaters, and divers report seeing the clown triggerfish devour beds of pearl oysters.		
Triggerfish	Do not eat this fish. According to an 1878 account, "the poisonous flesh acts primarily upon the nervous tissue of the stomach, occasioning violent spasms of that organ, and shortly afterwards all the muscles of the body. The frame becomes rocked with spasms, the tongue thickened, the eye fixed, the breathing laborious, and the patient expires in a paroxsm of extreme suffering."		
50			
19.69			

1.6.5.11.3 TCardBox:Methods

■ Constructor ■ Standard ■ Only after Create()

Type	Name
■	AddItem
■	Deleteltem
■	Deleteltems
■	InsertItem
■	HitTest
■	RecCount

1.6.5.11.3.1 TCardBox:AddItem

Adds a new section to the card box, represented by a TCardItem control.

Type	Standard
Parameters	None
Return value	TCardItem

1.6.5.11.3.2 TCardBox:DeleteItem

Deletes a card section.

Type	Standard
Parameters	<nItem>: Section ordinal in altems
Return value	NIL

1.6.5.11.3.3 TCardBox:DeleteItems

Deletes all card sections.

Type	Standard
Parameters	None
Return value	NIL

1.6.5.11.3.4 TCardBox:InsertItem

Inserts a new card section, represented b a TCardItem control.

Type	Standard
Parameters	[<nItem>]: Section position
Return value	TCardItem

1.6.5.11.3.5 TCardBox:HitTest

Returns ordinal in aData of a card in a specific mouse position.

Type	Standard
Parameters	<nX>: Mouse X position <nY>: Mouse Y position
Return value	<nValue>

1.6.5.11.3.6 TCardBox:RecCount

Total of cards.

Type	Standard
Parameters	None
Return value	<nTotal>

1.6.5.11.4 TCardBox:Events

Name
OnCardChange
OnCardClick
OnCardDbClick
OnCardGapPaint
OnCardPaint
OnCardSelection

1.6.5.11.4.1 TCardBox:OnCardChange

Event triggered when active card is changed.

Parameters:	<oSender>: Reference to the object that triggers the event <nNewPos>: New active card <nOldPos>: Old active card
Return value:	NIL

1.6.5.11.4.2 TCardBox:OnCardClick

Event triggered when mouse left button is clicked.

Parameters:	<oSender>: Reference to the object that triggers the event <oItem>: TCardItem object pushed <nPos>: Card ordinal in aData <nX>: Mouse X position <nY>: Mouse Y position
--------------------	---

Return value: NIL

1.6.5.11.4.3 TCardBox:OnCardDbClick

Event triggered when mouse double-click is done.

Parameters: <oSender>:
Reference to the object that triggers the event
<oltem>:
TCardItem object pushed
<nPos>:
Card ordinal in aData
<nX>:
Mouse X position
<nY>:
Mouse Y position

Return value: NIL

1.6.5.11.4.4 TCardBox:OnCardGapPaint

Event triggered when space between cards is painted.

Parameters: <oSender>:
Reference to the object that triggers the event
<hDC>:
Device context handle
<aRect>:
Painting array rectangle

Return value: NIL

1.6.5.11.4.5 TCardBox:OnCardPaint

Event triggered when each section of the cards are printed.

Parameters: <oSender>:
Reference to the object that triggers the event
<oltem>:
TCardItem object to print
<@Value>:
Actual value
<@nClrText>:
Text color
<@nClrPane>:
Background color
<nPos>_

	Ordinal in aData of the card
	<IActive> : True if the card is active
	<hDC> : Device context handle
	<aRect> : Rectangle array
Return value:	NIL

1.6.5.11.4.6 TCardBox:OnCardSelection

Event triggered when cards selection has changed.

Parameters:	<oSender> : Reference to the object that triggers the event
	<aSelected> : Array with ordinals in aData of cards selected
Return value:	NIL

1.6.5.12 TCardItem

Class that represents each of the sections of a TCardBox control.

Hierarchy TComponent descendant
File \source\CardItem.prg

1.6.5.12.1 TCardItem:Properties

■ Read only ■ Assignable ■ Design assignable ■ Run-time assignable

Scope	Name	Type	Initial value
■	IButton	Logical	.F.
■	IHotTrack	Logical	.F.
■	IMultiLine	Logical	.F.
■	IParentFont	Logical	.T.
■	IStretch	Logical	.F.
■	IVisible	Logical	.T.
■	nAlign	Numeric	alLEFT
■	nAlignWeight	Numeric	0
■	nAlignment	Numeric	taCENTER
■	nClrLink	Numeric	NIL
■	nClrText	Numeric	NIL
■	nColumn	Numeric	0
■	nItem	Numeric	0
■	nSiblingWeight	Numeric	0

■	nSize	Numeric	20
■	nTextPad	Numeric	5
■	nType	Numeric	ctLABEL
■	nVAlignment	Numeric	vaCENTER
■	oCursor	Object	TCursor
■	oFont	Object	TFont
■	oParent	Object	TCardBox
■	Value	Any	NIL

1.6.5.12.1.1 TCardItem:IButton

If true the control will behave like a button.

Scope	Assignable
Type	Logical
Initial value	.F.

1.6.5.12.1.2 TCardItem:IHotTrack

If true the control will have mouse hot- track.

Scope	Assignable
Type	Logical
Initial value	.F.

1.6.5.12.1.3 TCardItem:IMultiLine

If true the text shown will be multi-line.

Scope	Assignable
Type	Logical
Initial value	.F.

1.6.5.12.1.4 TCardItem:IParentFont

If true it will its parent TCardBox font.

Scope	Assignable
Type	Logical

Initial value	.T.
----------------------	-----

1.6.5.12.1.5 TCardItem:IStretch

If true and its Type is ctPICTURE or ctIMAGEINDEX, its image will be stretched.

Scope	Assignable
Type	Logical
Initial value	.F.

1.6.5.12.1.6 TCardItem:IVisible

If true the control will be visible.

Scope	Assignable
Type	Logical
Initial value	.T.

1.6.5.12.1.7 TCardItem:nAlign

Control alignment on its oParent container.

Scope:	Assignable
Type:	Numeric
Initial value:	aLEFT
Possible values:	aLEFT, aTOP, aRIGHT, aBOTTOM, aCLIENT

This property permits to adjust the position and sizes of each section of the card box:

- **Left:** The control is aligned to the left of its oParent control and it takes the height of its client container.
- **Upper:** The control is aligned to the top of its oParent control and it takes the width of its client container.
- **Right:** The control is aligned to the right of its oParent control and it takes the height of its client container.
- **Bottom:** The control is aligned to the bottom of its oParent control and it takes the width of its client container.
- **Client:** The control is aligned the client area of its oParent control adjusts itself its size.

For further information TCardBox control introduction.

1.6.5.12.1.8 TCardItem:nAlignWeight

Sets the relative weight of the control from other control with same alignment and created consecutively.

Scope:	Assignable
Type:	Numeric
Initial value:	0

For further information consult TCardBox control introduction.

1.6.5.12.1.9 TCardItem:nAlignment

Horizontal alignment.

Scope	Assignable
Type	Numeric
Initial value	taLEFT
Valores posibles	taLEFT, taRIGHT, taCENTER

1.6.5.12.1.10 TCardItem:nClrLink

Color for text links using the <a> HTML tags described on TLabelEx control. This property is only aplicable when Type property is set to ctLBELEX. If this value its NIL, the property nCardClrLink of its oParent is used.

Scope	Assignable
Type	Numeric
Initial value	NIL.

(Consult appendix for possible colors)

1.6.5.12.1.11 TCardItem:nClrText

Text color. This property is only aplicable when Type property is set to ctLABEL or ctLBELEX. It this value is NIL, the nClrText property of its oParent is used.

Scope	Assignable
Type	Numeric
Initial value	NIL.

(Consult appendix for possible colors)

1.6.5.12.1.12 TCardItem:nColumn

Using its similarity to a TBrowse control, sets the column number to use to link with data array of its TCardBox container.

Scope:	Assignable
Type:	Numeric
Initial value:	0

For further information consult TCardBox control introduction.

1.6.5.12.1.13 TCardItem:nItem

Control order creation on its TCardBox container.

Scope:	Read only
Type:	Numeric
Initial value:	0

1.6.5.12.1.14 TCardItem:nSiblingWeight

Sets the relative weight of various **TCardItem** controls that have the same alignment and are defined as siblings among them.

Scope:	Assignable
Type:	Numeric
Initial value:	0

For further information consult TCardBox control introduction.

1.6.5.12.1.15 TCardItem:nSize

Control size. When its alignment is horizontal this property indicates its width. When is vertical, is height. This property is useless when its property nAlignWeight is different from zero.

Scope:	Assignable
Type:	Numeric
Initial value:	20

For further information consult TCardBox control introduction.

1.6.5.12.1.16 TCardItem:nTextPad

Text left margin in pixels. This property is only applicable when Type property is set to ctLABEL or ctLBELEX.

Scope	Assignable
Type	Numeric
Initial value	clWindowText.

(Consult appendix for possible colors)

1.6.5.12.1.17 TCardItem:nType

Display type.

Scope:	Assignable
Type:	Numeric
Initial value:	ctLABEL
Possible values:	ctLABEL, ctLBELEX, ctPICTURE, ctIMAGEINDEX

- **ctLABEL:** Simple text.
- **ctLBELEX:** Rich text based on TLabelEx control.
- **ctPICTURE:** A image, can be a TPicture object or a stream.
- **ctIMAGEINDEX:** Image ordinal on its olmageList property.

For further information consult TCardBox control introduction.

1.6.5.12.1.18 TCardItem:nVAlignment

Text vertical alignment.

Scope	Assignable
Type	Numeric
Initial value	vaTOP
Possible values	vaTOP, vaBOTTOM, vaCENTER

1.6.5.12.1.19 TCardItem:oCursor

TCursor to be used in the control.

Scope:	Assignable
Type:	Object
Initial value:	NIL

Description:

This property allows to change the cursor that will be used in the control. This cursor will be visible when the mouse pointer is located on the control.

1.6.5.12.1.20 TCardItem:oFont

Object TFont to use for text.

Scope:	Assignable
Type:	Object
Initial value:	Nil

This property permits to use a different font the item is printed. By default it uses its container font and also if its property IParentFont is set to true.

For further information consult TFont class.

1.6.5.12.1.21 TCardItem:oParent

TCardBody object container.

Scope:	Read only
Type:	Object
Initial value:	Nil

1.6.5.12.1.22 TCardItem:Value

Control actual value.

Scope:	Read only
Type:	Any
Initial value:	Nil

1.6.5.12.2 TCardItem:Methods

■ Constructor ■ Standard

Type	Name
■	Delete
■	IsSelected
■	IsActive

1.6.5.12.2.1 TCardItem:Delete

Deletes on its TCardbox container this card section.

Type	Standard
Parameters	None
Return value	NIL

1.6.5.12.2.2 TCardItem:IsSelected

Returns true if the control is selected.

Type	Standard
Parameters	None
Return value	<IValue>

1.6.5.12.2.3 TCardItem:IsActive

Returns true if the control belongs the current active card.

Type	Standard
Parameters	None
Return value	<IValue>

1.6.5.13 TCheckBoxMod

This class represents a CheckBox object with Windows 10 appearance. The checkbox text can use HTML formatting in the same way as the TLabelEx control.

Hierarchy	TStdControl descendant
See also	TCheckBox
File	\source\CheckBoxMod.prg

1.6.5.13.1 TCheckBoxMod:Properties

■ Read only ■ Assignable ■ Design assignable ■ Run-time assignable

Scope	Name	Type	Initial value
■	IChecked	Logical	.F.
■	ITransparent	Logical	.F.
■	nClrColorization	Numeric	cSystem
■	nClrTextDisabled	Numeric	c3DDkShadow
■	nLineSpacing	Numeric	100
■	nVAlignment	Numeric	vaTOP

1.6.5.13.1.1 TCheckBoxMod:IChecked

If true, the control is shown as checked.

Scope	Assignable
Type	Logical
Initial value	.F.

1.6.5.13.1.2 TCheckBoxMod:ITransparent

The control is drawn transparent on the form to which it belongs.

Scope	Assignable
Type	Logical
Initial value	.T.

1.6.5.13.1.3 TCheckBoxMod:nClrColorization

Box and marker color.

Scope:	Assignable
Type:	Numeric
Initial value:	cSystem

(Consult the appendix for possible colors)

1.6.5.13.1.4 TCheckBoxMod:nClrTextDisabled

Text color when the control is disabled.

Scope:	Assignable
Type:	Numeric
Initial value:	cl3DDkShadow

(Consult the appendix for possible colors)

1.6.5.13.1.5 TCheckBoxMod:nLineSpacing

Percentage spacing. Maximum value 1000.

Scope	Assignable
Type	Numeric
Initial value	100

1.6.5.13.1.6 TCheckBoxMod:nVAlignment

Vertical text alignment. Displays the box aligned up, centered or down with respect to the text.

Scope	Assignable
Type	Numeric
Initial value	vaTOP
Possible values	vaTOP, vaBOTTOM, vaCENTER

1.6.5.13.2 TCheckBoxMod:Events

Name
OnChange

1.6.5.13.2.1 TCheckBoxMod:OnChange

Event that is occurs when the state of the control changes.

As of Xailer 8.1 **INewValue** already incorporates the change of value and not the old value, but it is passed by reference in case you want to change it. The return value becomes useless.

Parameters	<oSender>
-------------------	-----------

	Reference to the object that fires the event <@INewValue> Newvalue
Return value	<NIL>

1.6.5.14 TCircularImage

This class represents a TPicture object framed in a circle.

Hierarchy TStdControl descendant
File \source\CircularImage.prg

1.6.5.14.1 TCircularImage:Properties

■ Read only ■ Assignable ■ Design assignable ■ Run-time assignable

Scope	Name	Type	Initial value
■	ITransparent	Logical	.F.
■	nClrPane	Numeric	cBtnFace
■	nClrPen	Numeric	cActiveBorder
■	nPenSize	Numeric	4

1.6.5.14.1.1 TCircularImage:ITransparent

The control is drawn transparent on the form to which it belongs.

Scope	Assignable
Type	Logical
Initial value	.T.

1.6.5.14.1.2 TCircularImage:nClrPane

Background color of the control.

Scope:	Assignable
Type:	Numeric
Initial value:	cBTNFACE

When the ITransparent property of some controls is set to true this property will also have no effect.

(Consult appendix for possible colors)

1.6.5.14.1.3 TCircularImage:nClrPen

Color of the pen or pencil used to paint the image border.

Scope:	Assignable
Type:	Numeric
Initial value:	clActiveBorder

(Consult appendix for possible colors)

1.6.5.14.1.4 TCircularImage:nPenSize

Size in pixels of the pen or pencil used to paint the image edge.

Scope:	Assignable
Type:	Numeric
Initial value:	4

1.6.5.14.2 TCircularImage:Events

Name
OnClick

1.6.5.14.2.1 TCircularImage:OnClick

Event triggered when clicking on the control.

Parameters	<oSender> Reference to the object that fires the event
Return value	<NIL>

1.6.5.15 TComboBoxMod

This class represents a **ComboBox** control adapted to the Windows 10 style.

Hierarchy	TEditMod descendant
File	\source\ComboBoxMod.prg

1.6.5.15.1 TComboBoxMod:Properties

■ Read only ■ Assignable ■ Design assignable ■ Run-time assignable

Scope	Name	Type	Initial value
■	altems	Logical	{}
■	cText	Character	""
■	IDropAnimation	Logical	.T.
■	IDropped	Logical	.F.
■	IFreeEdit	Object	.F.
■	nClrDropHotPane	Numeric	clActiveCaption
■	nClrDropHotText	Numeric	clWindowText
■	nClrDropPane	Numeric	clWindow
■	nClrDropText	Numeric	clWindowText
■	nClrDropSelPane	Numeric	clHighlight
■	nClrDropSelText	Numeric	clWindowText
■	nDropItemHeight	Numeric	25
■	nDropMargin	Numeric	10
■	nDroppedHeight	Numeric	300
■	nDroppedWidth	Numeric	0
■	nIndex	Numeric	0
■	nStyle	Numeric	esCOMBO
■	olmageList	Object	NIL

1.6.5.15.1.1 TComboBoxMod:altems

Character array with combo elements.

Scope	Assignable
Type	Matriz
Initial value	{}

1.6.5.15.1.2 TComboBoxMod:cText

Current control value as text.

Scope:	Assignable
Type:	Character
Initial value:	""

1.6.5.15.1.3 TComboBoxMod:IDropAnimation

If true, the combo will be displayed with animation.

Scope	Assignable
Type	Logical
Initial value	.T.

1.6.5.15.1.4 TComboBoxMod:IDropped

If is True the combo is expanded.

Scope	Sólo lectura
Type	Logical
Initial value	.F.

1.6.5.15.1.5 TComboBoxMod:IFreeEdit

If True, it is possible to enter any text that will try to validate with some elements of the array altems.

Scope	Assignable
Type	Logical
Initial value	.F.

1.6.5.15.1.6 TComboBoxMod:nClrDropHotPane

Background color of the element selected by the mouse in the drop-down list.

Scope:	Assignable
Type:	Numeric
Initial value:	clActiveCaption

(Consult appendix for possible colors)

1.6.5.15.1.7 TComboBoxMod:nClrDropHotText

Text color of the element selected by the mouse in the drop-down list.

Scope:	Assignable
---------------	------------

Type:	Numeric
Initial value:	clWindowText

(Consult appendix for possible colors)

1.6.5.15.1.8 TComboBoxMod:nClrDropPane

Background color of the drop-down list.

Scope:	Assignable
Type:	Numeric
Initial value:	clWindow

(Consult appendix for possible colors)

1.6.5.15.1.9 TComboBoxMod:nClrDropText

Color of the text in the drop-down list.

Scope:	Assignable
Type:	Numeric
Initial value:	clWindowText

(Consult appendix for possible colors)

1.6.5.15.1.10 TComboBoxMod:nClrDropSelPane

Background color of the selected element from the drop-down list.

Scope:	Assignable
Type:	Numeric
Initial value:	clHighlight

(Consult appendix for possible colors)

1.6.5.15.1.11 TComboBoxMod:nClrDropSelText

Text color of the selected item from the drop-down list.

Scope:	Assignable
Type:	Numeric
Initial value:	clWindowText

(Consult appendix for possible colors)

1.6.5.15.1.12 TComboBoxMod:nDropltemHeight

Height of each item in the drop-down list.

Scope:	Assignable
Type:	Numeric
Initial value:	25

1.6.5.15.1.13 TComboBoxMod:nDropMargin

Side painting margin of each item in the drop-down list.

Scope:	Assignable
Type:	Numeric
Initial value:	10

1.6.5.15.1.14 TComboBoxMod:nDroppedHeight

Height of the drop-down list.

Scope:	Assignable
Type:	Numeric
Initial value:	300

1.6.5.15.1.15 TComboBoxMod:nDroppedWidth

Width of the drop-down list. A value of 0 means that it uses the same width as the control.

Scope:	Assignable
Type:	Numeric
Initial value:	0

1.6.5.15.1.16 TComboBoxMod:nIndex

Current item selected. If IFreeEdit is set to true and the current value in edit does not match any item in the items list, its value would be 0.

Scope:	Assignable
Type:	Numeric
Initial value:	0

1.6.5.15.1.17 TComboBoxMod:nStyle

Button style to be displayed.

Scope:	Assignable
Type:	Numeric
Initial value:	esCOMBO
Possible values:	esCOMBO, esARROW

1.6.5.15.1.18 TComboBoxMod:olmageList

TImageList object with all the images that will be used by the control.

Scope	Assignable
Type	Object
Initial value	TImageList()

The TImageList object is instantiated since the control creation through its New(0 constructor. The first image included in the TImageList establishes the image dimensions. If the first image is an image that includes more than one bitmap, is important that you establishes the TImageList nHeight and nWidth properties before to add a bitmap.

1.6.5.15.2 TComboBoxMod:Events

Name
OnCloseUp
OnDrawItem
OnDropCreate
OnDropDown

1.6.5.15.2.1 TComboBoxMod:OnCloseUp

Evento que se produce cuando se cierra la lista desplegable.

Parameters:	<oSender>: Reference to the object that fires the event
Return value:	NIL

1.6.5.15.2.2 TComboBoxMod:OnDrawItem

Evento que se produce cuando se pinta cada elemento de la lista desplegable.

Parameters:	<oSender>: Reference to the object that fires the event <nItem>: Ordinal of the element <@cText>: Value with information. This parameter is received by reference. <@nImage>: Order of the image to be displayed according to the <code>oImageList</code> object of the control. This parameter is received by reference. By default its value is zero which means no image. <@nClrText>: Text color. This parameter is received by reference. <@nClrPane>: Background color. This parameter is received by reference. <hDC>: Context device handler <aRect>: Rectangle marking the possible painting area
Return value:	NIL or Logical. If Logical False is returned the painting is canceled and it is the responsibility of the event to do so.

1.6.5.15.2.3 TComboBoxMod:OnDropCreate

Evento que se produce cuando se intenta crear el control que muestra la lista desplegable.

Parameters:	<oSender>: Reference to the object that fires the event
Return value:	NIL or Logical. If Logical False is returned, the creation is canceled and no list is displayed.

1.6.5.15.2.4 TComboBoxMod:OnDropDown

Event that occurs when the drop-down list is displayed.

Parameters:	<oSender>: Reference to the object that fires the event <oDropForm>: Invisible form containing the list control <oControl>: List control of type ListboxMod
Return value:	NI

1.6.5.16 TDateEditMod

This class represents a TEditMod control specialized in date editing. Unlike the TDatePicker control it allows blank dates without using an additional checkbox..



Hierarchy	TEditMod descendant
File	\source\EditMod.prg

1.6.5.16.1 TDateEditMod:Properties

■ Read only ■ Assignable ■ Design assignable ■ Run-time assignable

Scope	Name	Type	Initial value
■	IDropAnimation	Logical	.T.
■	IDropShowDOW	Logical	.T.
■	IDropShowFirstOfGroup	Logical	.F.
■	IDropShowLines	Logical	.F.
■	nClrDropBorder	Numeric	clActiveBorder
■	nClrDropDaysDisabled	Numeric	cl3DLight
■	nClrDropHot	Numeric	clGray
■	nClrDropTextHeader	Numeric	clWindowText
■	nClrDropToday	Numeric	clSystem
■	nDropNumberOfWeeks	Numeric	6
■	nDroppedHeight	Numeric	0
■	nDroppedWidth	Numeric	0

1.6.5.16.1.1 TDateEditMod:IDropAnimation

If true, the calendar will be displayed with animation.

Scope	Assignable
Type	Logical
Initial value	.T.

1.6.5.16.1.2 TDateEditMod:IDropShowDOW

If true, the days of the week will be displayed.

Scope	Assignable
Type	Logical
Initial value	.T.

1.6.5.16.1.3 TDateEditMod:IDropShowFirstOfGroup

If true, the group header will be displayed.

Scope	Assignable
Type	Logical
Initial value	.F.

1.6.5.16.1.4 TDateEditMod:IDropShowLines

If true, separator lines will be displayed for each sub-element of the calendar.

Scope	Assignable
Type	Logical
Initial value	.T.

1.6.5.16.1.5 TDateEditMod:nClrDropBorder

Color for border and inner lines.

Scope	Assignable
Type	Numeric
Initial value	clActiveBorder

(Consult appendix for possible colors)

1.6.5.16.1.6 TDateEditMod:nClrDropDaysDisabled

Color for disabled days.

Scope	Assignable
Type	Numeric
Initial value	cl3DLight

(Consult appendix for possible colors)

1.6.5.16.1.7 TDateEditMod:nClrDropHot

Color for the day that has the mouse hovering on it.

Scope	Assignable
Type	Numeric
Initial value	clGray

(Consult appendix for possible colors)

1.6.5.16.1.8 TDateEditMod:nClrDropTextHeader

Text color for header texts.

Scope	Assignable
Type	Numeric
Initial value	clWindowText

(Consult appendix for possible colors)

1.6.5.16.1.9 TDateEditMod:nClrDropToday

Background color for the current day.

Scope	Assignable
Type	Numeric
Initial value	clSystem

(Consult appendix for possible colors)

1.6.5.16.1.10 TDateEditMod:nDropNumberOfWeeks

Number of weeks to be shown.

Scope	Assignable
Type	Numeric
Initial value	6

1.6.5.16.1.11 TDateEditMod:nDroppedHeight

Height of the calendar when it is displayed. A value of zero means that the default value of 250 pixels will be used.

Scope	Assignable
Type	Numeric
Initial value	0

1.6.5.16.1.12 TDateEditMod:nDroppedWidth

Width of the calendar when it is displayed. A value of zero means that the default value of 250 pixels will be used.

Scope	Assignable
Type	Numeric
Initial value	0

1.6.5.16.2 TDateEditMod:Events

Name
OnCloseUp
OnDrawItem
OnDropCreate
OnDropDown

1.6.5.16.2.1 TDateEditMod:OnDrawItem

Event that occurs when any day of the calendar is painted.

Parameters	<oSender> : Object that fires the event <dDay>
-------------------	--

	Dayt < @nClrText > Text color. By reference. Editable < @nClrPane > Background color. By reference. Editable < hDC > Handle of Device Context < aRect > Painting rectanble
Return value	< NIL > or <.T.> processes the painting. A false value forces the painting using hDC and aRect .

1.6.5.16.2.2 TDateEditMod:OnCloseUp

Event that occurs when the drop-down list is closed.

Parameters	< oSender >: Reference to the object that fires the event
Return value	NIL

1.6.5.16.2.3 TDateEditMod:OnDropCreate

Event that occurs when an attempt is made to create the control that displays the drop-down list.

Parameters	< oSender >: Reference to the object that fires the event
Return value	NIL or Logic. If logical false, the creation is canceled and no list is displayed.

1.6.5.16.2.4 TDateEditMod:OnDropDown

Event that occurs when the drop-down list is displayed.

Parameters	< oSender >: Reference to the object that fires the event < oDropForm >: Invisible form containing the list control < oControl >: Calendar control of type TCalendarMod
Return value	NIL

1.6.5.17 TEditMod

This class represents a TMaskEdit with multiple enhancements, like:

- Windows 10 style
- Show a text by default
- Password style, showing an eye that when pressed reveals real content
- Predefined buttons: search, delete, password an others
- Correct text vertical alignment regardless of the font used
- Adapt to the color theme used in Windows Vista or later



Hierarchy TMaskEdit descendant
File \source\EditMod.prg

1.6.5.17.1 TEditMod:Properties

■ Read only ■ Assignable ■ Design assignable ■ Run-time assignable

Scope	Name	Type	Initial value
■	cHint	Character	""
■	nClrColorization	Numeric	0
■	nStyle	Numeric	esNONE
■	oBitmaps	Object	NIL

1.6.5.17.1.1 TEditMod:cHint

Text to show when control has no value.

Scope:	Assignable
Type:	Character
Initial value:	""

1.6.5.17.1.2 TEditMod:nClrColorization

Border color when control receives focus. If its value is zero it will use the predefined system value.

Scope:	Assignable
Type:	Numeric
Initial value:	0

Consulte el apéndice para ver colores posibles

1.6.5.17.1.3 TEditMod:nStyle

Style.

Scope:	Asignable
Type:	Numeric
Initial value:	esNONE
Valores posibles:	esNONE, esSEARCH, esCLEAR, esPASSWORD, esVERTDOTS, esHORZDOTS, esCOMBO, esARROW, esUSER

1.6.5.17.1.4 TEditMod:oBitmaps

TImageList object with the images to show when the style is set to esUSER

Scope:	Asignable
Type:	Object
Initial value:	NIL

The **TEditmod** can show 4 different images depending of its state. They can be:

- Normal, it corresponds to the first TImageList image.
- With the mouse pointer over it, that corresponds to the second TImageList image.
- Pressed, and corresponds to the third TImageList image.
- Disable, that corresponds to the fourth TImageList image.

The button will use the first bitmap for all the states where the image has not been defined.

There are several ways to assign this property:

- Through a previously created TImageList object.
- Through a literal with the resource name of Bitmap type file.
- Through an 1, 2, 3 or 4 elements array with the resource names or the bitmap type files for every possible button state.

For any of the last 2 cases the button will create a TImageList with the information provided.

1.6.5.17.2 TEditMod:Events

Name
OnBtnClick
OnClearClick

1.6.5.17.2.1 TEditBtn:OnBtnClick

Event triggered when the image is clicked.

Parameters:	<oSender>: Reference to the object that triggers the event <nX>: Mouse X position <nY>: Mouse Y position
Return value:	NIL

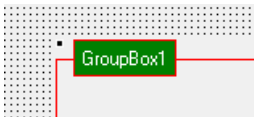
1.6.5.17.2.2 TEditMod:OnClearClick

Event triggered when the reset cross is clicked.

Parameters:	<oSender>: Reference to the object that triggers the event
Return value:	NIL

1.6.5.18 TGroupBoxMod

This class represents a TGroupBox control with enhancements.



Hierarchy TWinControl descendant
See also TPaneloupBox
File \source\GroupBoxMod.prg

1.6.5.18.1 TGroupBox:Properties

■ Read only ■ Assignable ■ Design assignable ■ Run-time assignable

Scope	Name	Type	Initial value
■	nAlignment	Numeric	taLEFT
■	nFrameClrPane	Numeric	clBtnFace
■	nFrameClrPen	Numeric	cl3DLight
■	nFrameEllipseX	Numeric	0
■	nFrameEllipseY	Numeric	0
■	nFrameRect	Numeric	trNONE
■	nFrameSize	Numeric	24

■	nLineGap	Numeric	10
■	nHeigth	Numeric	90
■	nTextGap	Numeric	10
■	nTextVAlignment	Numeric	vaCENTER
■	nWidth	Numeric	120

1.6.5.18.1.1 TGroupBoxMod:nAlignment

Text alignment shown on control title.

Scope	Assignable
Type	Numeric
Initial value	taLEFT
Possible values	taLEFT, taRIGHT, taCENTER

1.6.5.18.1.2 TGroupBoxMod:nFrameClrPane

Background color for the box text area.

Scope:	Assignable
Type:	Numeric
Initial value:	clBtnFace

(Check Appendix for possible colors)

1.6.5.18.1.3 TGroupBoxMod:nFrameClrPen

Pen color for box lines.

Scope:	Assignable
Type:	Numeric
Initial value:	cl3DLight

(Check Appendix for possible colors)

1.6.5.18.1.4 TGroupBoxMod:nFrameElipseX

Ellipse angle on its X coordinate for the box text area.

Scope:	Assignable
Type:	Numeric

Initial value:	0
-----------------------	---

1.6.5.18.1.5 TGroupBoxMod:nFrameEllipseY

Ellipse angle on its Y coordinate for the box text area.

Scope:	Assignable
Type:	Numeric
Initial value:	0

1.6.5.18.1.6 TGroupBoxMod:nFrameRect

Paint mode to use for the text box.

Scope:	Assignable
Type:	Numeric
Initial value:	trNONE
Possible values:	trNONE: no borders trFRAME: paint only the lines trFILL: Fill the box trALL: Both, fill the box and paint the lines

1.6.5.18.1.7 TGroupBoxMod:nFrameSize

Height of the are used by the text box.

Scope:	Assignable
Type:	Numeric
Initial value:	24

1.6.5.18.1.8 TGroupBoxMod:nLineGap

Space in pixels from the border to the text box.

Scope:	Assignable
Type:	Numeric
Initial value:	10

1.6.5.18.1.9 TGroupBoxMod:nHeight

Control height.

Scope	Assignable
Type	Numeric
Initial value	90

1.6.5.18.1.10 TGroupBoxMod:nTextGap

Extra space in pixels for the box text area.

Scope:	Assignable
Type:	Numeric
Initial value:	10

1.6.5.18.1.11 TGroupBoxMod:nTextVAlignment

Text vertical alignment on its box.

Scope:	Assignable en diseño
Type:	Numeric
Initial value:	vaCENTER
Possible values:	vaTOP, vaCENTER, vaBOTTOM

1.6.5.18.1.12 TGroupBoxMod:nWidth

Control width.

Scope	Assignable
Type	Numeric
Initial value	120

1.6.5.19 THeaderMod

This class represents a modern Windows 10 type of Header control. It is used to display headers of any columnar information. It is unlikely that you will use the control directly, however, it is used internally by all **TBrowseMod** type controls.



Hierarchy TWinControl descendant
File \source\HeaderMod.prg

1.6.5.19.1 THeaderMod:Properties

■ Read only ■ Assignable ■ Design assignable ■ Run-time assignable

Scope	Name	Type	Initial value
■	altems	Array	{}
■	ICanDrag	Logical	.T.
■	ICanHide	Logical	.T.
■	IFilterBar	Logical	.F.
■	IHotTrack	Logical	.T.
■	IResize	Logical	.T.
■	nBmpMargin	Numeric	5
■	nClrDivider	Numeric	clGray
■	nClrEditPane	Numeric	clGradientInactiveCaption
■	nClrHotPane	Numeric	clActiveCaption
■	nClrHotText	Numeric	clWindowText
■	nClrPane	Numeric	clBtnFace
■	nClrSelPane	Numeric	clGrandientInactiveCaption
■	nClrText	Numeric	clBtnText
■	nFilterTimeOut	Numeric	1000
■	nMarginHor	Numeric	5
■	nMarginVer	Numeric	2
■	nSelect	Numeric	0
■	olmgeList	Object	NIL

1.6.5.19.1.1 THeaderMod:altems

Array of THeaderItemMod objects containing the different columns of the control.

Scope	Assignable
Type	Array
Initial value	{}

1.6.5.19.1.2 THeaderMod:ICanDrag

If true, drag operations between columns are allowed.

Scope	Assignable
Type	Logical
Initial value	.T.

1.6.5.19.1.3 THeaderMod:ICanHide

If true, column hiding operations are allowed

Scope	Assignable
Type	Logical
Initial value	.T.

1.6.5.19.1.4 THeaderMod:IFilterBar

If true, edit controls are displayed above each of the columns.

Scope	Assignable
Type	Logical
Initial value	.F.

1.6.5.19.1.5 THeaderMod:IHotTrack

If true, the color of the column in which the mouse is located is changed.

Scope	Assignable
Type	Logical
Initial value	.T.

1.6.5.19.1.6 THeaderMod:IResize

If true, you are allowed to resize the columns with the mouse.

Scope	Assignable
Type	Logical
Initial value	.T.

1.6.5.19.1.7 THeaderMod:nBmpMargin

Side (lateral) margin in pixels of painting of the possible image of each column.

Scope	Assignable
Type	Numeric
Initial value	5

1.6.5.19.1.8 THeaderMod:nClrDivider

Color of the column divider.

Scope	Assignable
Type	Numeric
Initial value	clGray

(Consult the appendix for possible colors)

1.6.5.19.1.9 THeaderMod:nClrEditPane

Background color for edit controls when the IFilterBar property is set to true.

Scope	Assignable
Type	Numeric
Initial value	clGradientInactiveCaption

(Consult the appendix for possible colors)

1.6.5.19.1.10 THeaderMod:nClrHotPane

Color of the column when the mouse is over it and IHotTrack property is set to true.

Scope	Assignable
Type	Numeric
Initial value	clActiveCaption

(Consult the appendix for possible colors)

1.6.5.19.1.11 THeaderMod:nClrHotText

Color of the column text when the mouse is over it and IHotTrack property is set to true.

Scope	Assignable
Type	Numeric
Initial value	clActiveCaption

(Consult the appendix for possible colors)

1.6.5.19.1.12 THeaderMod:nClrPane

Background color of the columns.

Scope	Assignable
Type	Numeric
Initial value	clBtnFace

(Consult the appendix for possible colors)

1.6.5.19.1.13 THeaderMod:nClrSelPane

The background color of the selected column corresponds to the property nSelect.

Scope	Assignable
Type	Numeric
Initial value	clGradientInactiveCaption

(Consult the appendix for possible colors)

1.6.5.19.1.14 THeaderMod:nClrText

The color for the column text.

Scope	Assignable
Type	Numeric
Initial value	clBtnText

(Consult the appendix for possible colors)

1.6.5.19.1.15 THeaderMod:nFilterTimeOut

Time in milliseconds from when text is no longer entered in an edit control and the OnFilterTimeOut event is triggered.

Scope	Assignable
Type	Numeric
Initial value	1000

1.6.5.19.1.16 THeaderMod:nMarginHor

Horizontal margin for image and text painting for each item..

Scope	Assignable
Type	Numeric
Initial value	5

1.6.5.19.1.17 THeaderMod:nMarginVer

Vertical margin for image and text painting for each item.

Scope	Assignable
Type	Numeric
Initial value	2

1.6.5.19.1.18 THeaderMod:nSelect

Selected column.

Scope	Assignable
--------------	------------

Type	Numeric
Initial value	0

1.6.5.19.1.19 THeaderMod:oImageList

TImageList object with all the images to be used in the control.

Scope	Assignable
Type	Object
Initial value	TImageList()

The **TImageList** object is instantiated when the control is created with its **New()** constructor. The first image included in the **TImageList** defines the dimension of the successive images to be inserted. If the first image is an image that includes more than one Bitmap it is important to set the **nHeight** and **nWidth** properties of the **TImageList** before any Bitmap added.

1.6.5.19.2 THeaderMod:Methods

■ Constructor ■ Standard

Type	Name
■	AddItem
■	ClearAllFilters
■	ClearFilter
■	Deleteltem
■	Deleteltems
■	InsertItem
■	HitTest
■	IsEmpty
■	IsFilter
■	RestoreState
■	SaveState
■	SwapItems

1.6.5.19.2.1 THeaderMod:AddItem

Adds a new THeaderItemMod object to the THeaderMod object.

Type	Standard
Parameters	<cltem> Text to be displayed in the column [<nAlignment>]

	Alignment. Same as property nAlignment
	[<nWidth>] Control width. Default 100
	[<nImage>] Ordinal of the image to display in the olmageList object. By default 0.
	[<cTooltip>] Tooltip to display. Same as property cTooltip
	[<nVAlignment>] Vertical alignment. Same as property nVAlignment
Return value	THeaderItemMod

1.6.5.19.2.2 THeaderMod:ClearAllFilters

Clears all filters from the Header columns.

Type	Standard
Parameters	None
Return value	NIL

1.6.5.19.2.3 THeaderMod:ClearFilter

Clear the filter for a given column.

Type	Standard
Parameters	<nIndex> Column to delete the filter
Return value	NIL

1.6.5.19.2.4 THeaderMod:DeleteItem

Deletes a specific column..

Type	Standard
Parameters	<nIndex> The number of the column to delete
Return value	NIL

1.6.5.19.2.5 THeaderMod:DeleteItems

Deletes all columns from the Header.

Type	Standard
Parameters	None
Return value	NIL

1.6.5.19.2.6 THeaderMod:InsertItem

Inserts a new THeaderItemMod object to the THeaderMod object.

Type	Standard
Parameters	<nPos> Position of the new column <cItem> Text to be displayed in the column [<nAlignment>] Alignment. Same as property nAlignment [<nWidth>] Control width. Default 100 [<nImage>] Ordinal of the image to display in the olmageList object, by default 0. [<cTooltip>] Tooltip to display. Same as property cTooltip [<nVAlignment>] Vertical alignment. Same as property nVAlignment
Return value	THeaderItemMod

1.6.5.19.2.7 THeaderMod:HitTest

Checks the existing column for a given coordinate.

Type	Standard
Parameters	<aPos> Coordinate [x, y]
Return value	Ordinal of the column or zero

1.6.5.19.2.8 THeaderMod:IsEmpty

Returns true if Header has no columns created.

Type	Standard
Parameters	None
Return value	Logical

1.6.5.19.2.9 THeaderMod:IsFilter

Returns true if the Header has any filtered columns.

Type	Standard
Parameters	None
Return value	Logical

1.6.5.19.2.10 THeaderMod:RestoreState

Restores the Header status saved by the method SaveState.

Type	Standard
Parameters	<cState> CadenaString with Header status
Return value	NIL

1.6.5.19.2.11 THeaderMod:SaveState

Saves the Header status to a string returned by the method. You can restore its status with the RestoreState method..

Type	Standard
Parameters	None
Return value	<cState> String with Header status

1.6.5.19.2.12 THeaderMod:SwapItems

Swaps the position of several columns.

Type	Standard
Parameters	<oltem1> First column <oltem2> Second column <IRebuild> If true, the Header is reconstructed
Return value	NIL

1.6.5.19.3 THeaderMod:Events

Name
OnDividerDbClick
OnDrawItem
OnFilterBtnClick
OnFilterChange
OnFilterEdit
OnFilterTimeOut
OnItemBeginDrag
OnItemChanged
OnItemChanging
OnItemClick
OnItemEndDrag
OnItemRClick
OnRebuild

1.6.5.19.3.1 THeaderMod:OnDividerDbClick

Event that occurs when a double-click is performed just between two columns (on the divider).

Parameters	<oSender> Reference to the object that fires the event
Return value	NIL

1.6.5.19.3.2 THeaderMod:OnDrawItem

Event that occurs when each column is painted..

Parameters	<oSender>
-------------------	------------------------

	Object that fires the event (Self)..
	<oltem> THeaderItemMod object
	<@cText> Value with the text to be displayed. This parameter is received by reference.
	<@nImage> Ordinal of the image to be displayed according to the olmgeList object of the control. This parameter is received by reference. By default its value is zero which means no image.
	<@nClrText> Text color. This parameter is received by reference.
	<@nClrPane> Background color. This parameter is received by reference.
	<hDC> context device handler
	<@aRect> Rectangle marking the possible painting area
Return value	NIL or Logic. If logic false the painting is canceled and it is the responsibility of the event to do so.

1.6.5.19.3.3 THeaderMod:OnFilterBtnClick

Event that occurs when the edit control button is pressed when filtering is active.

Parameters	<oSender> Reference to the object that fires the event <oltem> THeaderItemMod that fires the event
Return value	None

1.6.5.19.3.4 THeaderMod:OnFilterChange

Event that occurs when there is a change in the edit control when filtering is active.

Parameters	<oSender> Reference to the object that fires the event <oltem> THeaderItemMod object that fires the event <oEdit> TEditMod object that fires the event
Return value	None

1.6.5.19.3.5 THeaderMod:OnFilterEdit

Event that occurs when any edit control receives the focus.

Parameters	<oSender> Reference to the object that fires the event <oltem> THeaderItemMod that fires the event <oEdit> TEditMod object that fires the event <oPrev> Last TEditMod with focus
Return value	None

1.6.5.19.3.6 THeaderMod:OnFilterTimeOut

Event that occurs when the time specified by the nFilterTimeOut property passes without any key being pressed.

Parameters	<oSender> Reference to the object that fires the event <oltem> THeaderItemMod that fires the event
Return value	None

1.6.5.19.3.7 THeaderMod:OnItemBeginDrag

Event that occurs when a column shift operation is started to move its position.

Parameters	<oSender> Reference to the object that fires the event <oltem> THeaderItemMod that fires the event
Return value	NIL or Logical. If Logical false the operation is canceled.

1.6.5.19.3.8 THeaderMod:OnItemChanged

Event that occurs when there is any change in the column.

Parameters	<oSender> Reference to the object that fires the event <oltem>
-------------------	--

	THeaderItemMod that fires the event
Return value	None

1.6.5.19.3.9 THeaderMod:OnItemChanging

Event that is raised when a change to the column width is about to occur.

Parameters	<oSender> Reference to the object that fires the event <oltem> THeaderItemMod that fires the event
Return value	NIL or Logical. If Logical false the operation is canceled.

1.6.5.19.3.10 THeaderMod:OnItemClick

Event that occurs when any column is clicked.

Parameters	<oSender> Reference to the object that fires the event <oltem> THeaderItemMod that fires the event <aPos> Click coordinates
Return value	None

1.6.5.19.3.11 THeaderMod:OnItemEndDrag

Event that occurs when a column shift operation is completed.

Parameters	<oSender> Reference to the object that fires the event <oltem> THeaderItemMod that fires the event
Return value	NIL or Logical. If Logical false the operation is canceled.

1.6.5.19.3.12 THeaderMod:OnItemRClick

Event that occurs when a right click is done on any column.

Parameters	<oSender> Reference to the object that fires the event <oltem> THeaderItemMod that fires the event <aPos> Click coordinates
Return value	NIL or Logical. If Logical false the operation is canceled.

1.6.5.19.3.13 THeaderMod:OnRebuild

Event that occurs when the control is rebuilt.

Parameters	<oSender> Reference to the object that fires the event
Return value	NIL

1.6.5.19.4 THeaderItemMod

This class represents each of the columns of the THeaderMod control.

Hierarchy	TComponent descendant
File	\source\HeaderModItem.prg

1.6.5.19.4.1 THeaderItemMod:Properties

■ Read only ■ Assignable ■ Design assignable ■ Run-time assignable

Scope	Name	Type	Initial value
■	cFilter	Character	""
■	cHeader	Character	""
■	cTooltip	Character	""
■	ICanDrag	Logical	.T.
■	ICanFilter	Logical	.T.
■	ICanHide	Logical	.T.
■	ICanHot	Logical	.T.
■	ICanResize	Logical	.T.
■	ICanSort	Logical	.T.
■	IMultiLine	Logical	.F.
■	IVisible	Logical	.T.

■	nAlignment	Numeric	taCENTER
■	nFilterEditStyle	Numeric	esVERDOTS
■	nImage	Numeric	0
■	nSort	Numeric	hsNONE
■	nVAlignment	Numeric	vaCENTER
■	nWidth	Numeric	100
■	oEditFilter	Object	NIL
■	oGridCol	Object	NIL

Current column filter

Scope	Assignable
Type	Character
Initial value	""

Text to be displayed in the column.

Scope	Assignable
Type	Character
Initial value	""

Tooltip to be displayed.

Scope	Assignable
Type	Character
Initial value	""

Popup window that displays the cTooltip text when the user pauses momentarily with the mouse inside the column header, or 'Header' of the column.

If true, Drag operations are allowed in this column.

Scope	Assignable
Type	Logical
Initial value	.T.

If true filtering operations are allowed in this column.

Scope	Assignable
Type	Logical
Initial value	.T.

If true, hiding operations are allowed in this column.

Scope	Assignable
Type	Logical
Initial value	.T.

If true, the column is displayed in a different color when the mouse is over it.

Scope	Assignable
Type	Logical
Initial value	.T.

If true, the width of the column can be changed with the mouse.

Scope	Assignable
Type	Logical
Initial value	.T.

If true, the column provides sorting functionality.

Scope	Assignable
Type	Logical
Initial value	.T.

If true, the column text can be multi-line.

Scope	Assignable
--------------	------------

Type	Logical
Initial value	.F.

If true the column is visible.

Scope	Assignable
Type	Logical
Initial value	.F.

Horizontal text alignment.

Scope	Assignable
Type	Numeric
Initial value	taCENTER
Possible values	taLEFT, taCENTER, taRIGHT

Style of the TEditMod control used for filter editing.

Scope	Assignable
Type	Numeric
Initial value	esVERTDOTS
Possible values	esNONE, esSEARCH, esCLEAR, esVERTDOTS, esHORZDOTS, esCOMBO, esARROW, esUPDOWN

Ordinal of image to display from the olmageList property of your container object.

Scope	Assignable
Type	Numeric
Initial value	0

Column sorting mode.

Scope	Assignable
Type	Numeric
Initial value	hsNONE
Possible values	hsNONE, hsASCENDING, hsDESCENDING

Vertical text alignment.

Scope	Assignable
Type	Numeric
Initial value	vaCENTER
Valores posibles	vaTOP, vaBOTTOM, vaCENTER

Column width.

Scope	Assignable
Type	Numeric
Initial value	100

TEditMod object that is responsible for editing the filter.

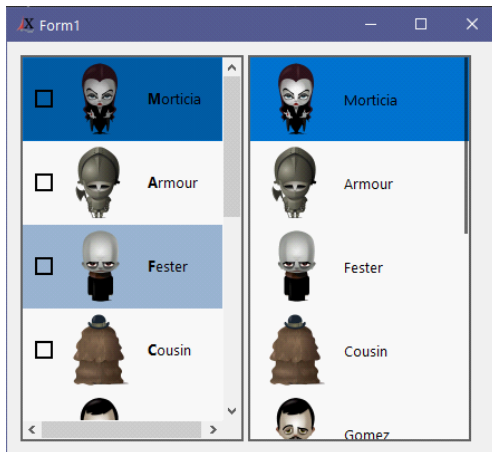
Scope	Assignable
Type	Object
Initial value	NIL

TBrwColMod object pointing to one of the columns of a TBrowseMod object.

Scope	Read only
Type	Object
Initial value	NIL

1.6.5.20 TListBoxMod

This class represents a TListBox control adapted to the Windows 10 style.



Hierarchy File

TScrollingWinControl descendant
 \source\ListBoxMod.prg

1.6.5.20.1 TListBoxMod:Properties

■ Read only ■ Assignable ■ Design assignable ■ Run-time assignable

Scope	Name	Type	Initial value
■	aChecked	Array	{}
■	altems	Array	{}
■	aSelected	Array	{}
■	cText	Character	""
■	ICheckBoxes	Logical	.F.
■	IHideScrollBars	Logical	.T.
■	IHotTrack	Logical	.T.
■	IHtmlText	Logical	.F.
■	IMultiLine	Logical	.F.
■	IMultipleSel	Logical	.F.
■	IShortCuts	Logical	.F.
■	IShowActive	Logical	.T.
■	nClrHotPane	Numeric	c\ActiveCaption
■	nClrHotText	Numeric	c\WindowText
■	nClrSelPane	Numeric	c\Highlight
■	nClrSelText	Numeric	c\WindowText
■	nIndex	Numeric	1
■	nItemHeight	Numeric	25
■	nMargin	Numeric	10
■	nOrientation	Numeric	orLEFT
■	nVAlignment	Numeric	vaCENTER
■	olmagedList	Object	NIL

1.6.5.20.1.1 TListBoxMod:aChecked

Numerical array with all the elements of the list marked.

Scope	Run-time assignable
Type	Array
Initial value	{}

1.6.5.20.1.2 TListBoxMod:altems

Character array with the elements of the list.

Scope	Assignable
Type	Array
Initial value	{}

1.6.5.20.1.3 TListBoxMod:aSelected

Numerical array with all selected items of the list.

Scope	Run-time assignable
Type	Array
Initial value	{}

1.6.5.20.1.4 TListBoxMod:cText

Text of the active list item.

Scope	Assignable
Type	Character
Initial value	""

1.6.5.20.1.5 TListBoxMod:lCheckBoxes

If true, each item in the list will display a checkbox that can be checked.

Scope	Assignable
--------------	------------

Type	Logical
Initial value	.F.

1.6.5.20.1.6 TListBoxMod:IHideScrollbars

If true, the scroll bars will only be displayed when you move the mouse close to the edges of the control. This property must be set in design-time.

Scope	Design assignable
Type	Logical
Initial value	.F.

1.6.5.20.1.7 TListBoxMod:IHotTrack

If true, the list item where the cursor is placed will be highlighted with the colors indicated as 'hot'.

Scope	Assignable
Type	Logical
Initial value	.T.

1.6.5.20.1.8 TListBoxMod:IHtmlText

If true, list items support basic HTML tags. See the TLabelEx class for more information about supported tags..

Scope	Assignable
Type	Logical
Initial value	.F.

1.6.5.20.1.9 TListBoxMod:IMultiLine

If true, the list items support multiple lines..

Scope	Assignable
Type	Logical
Initial value	.F.

1.6.5.20.1.10 TListBoxMod:IMultipleSel

If true, more than one item can be selected from the list. Use the `aSelected` property to determine which items are selected.

Scope	Asignable
Type	Logical
Initial value	.F.

1.6.5.20.1.11 TListBoxMod:IShortCuts

If true, keyboard shortcuts set with the '&' key will be displayed on any text in the list items. Pressing `Alt+'?'`, where '?' is the character following the '&' sign, will immediately select the item and execute its `OnSelect()` event which would be the equivalent as if you had pressed "Enter".

Scope	Asignable
Type	Logical
Initial value	.F.

1.6.5.20.1.12 TListBoxMod:IShowActive

If false, the selected element is not painted differently.

Scope	Asignable
Type	Logical
Initial value	.T.

1.6.5.20.1.13 TListBoxMod:nClrHotPane

Background color of the element pointed out by the mouse in the list.

Scope	Asignable
Type	Numeric
Initial value	<code>clActiveCaption</code>

(Consult appendix for possible colors)

1.6.5.20.1.14 TListBoxMod:nClrHotText

Text color of the element pointed out by the mouse in the list..

Scope	Asignable
Type	Numeric
Initial value	clWindowText

(Consult appendix for possible colors)

1.6.5.20.1.15 TListBoxMod:nClrSelPane

Background color of the selected items in the list when the IMultipleSel property is set to true.

Scope	Asignable
Type	Numeric
Initial value	clHighlight

(Consult appendix for possible colors)

1.6.5.20.1.16 TListBoxMod:nClrSelText

Text color of the selected items in the list when the IMultipleSel property is set to true.

Scope	Asignable
Type	Numeric
Initial value	clWindowText

(Consult appendix for possible colors)

1.6.5.20.1.17 TListBoxMod:nIndex

The active element.

Scope	Asignable
Type	Numeric
Initial value	0

1.6.5.20.1.18 TListBoxMod:nItemHeight

Height of each item in the list.

Scope	Assignable
Type	Numeric
Initial value	25

1.6.5.20.1.19 TListBoxMod:nMargin

Side margin of the text and/or image of each list element.

Scope	Assignable
Type	Numeric
Initial value	10

1.6.5.20.1.20 TListBoxMod:nOrientation

Orientation of the image that can display each item in the list.

Scope	Assignable
Type	Numeric
Initial value	orLEFT
Possible values:	orLEFT, orRIGHT

1.6.5.20.1.21 TListBoxMod:nVAlignment

Text vertical alignment.

Scope	Assignable
Type	Numeric
Initial value	vaCENTER
Possible values	vaTOP, vaBOTTOM, vaCENTER

1.6.5.20.1.22 TListBoxMod:olmImageList

TImageList object with all the images to be used in the control.

Scope	Asignable
Type	Object
Initial value	TImageList()

The TImageList object is instantiated when the control is created with its New() constructor. The first image that is included in the TImageList object determines the dimension of the successive images that are included. If the first image is an image that includes more than one Bitmap, it is important to set the nHeight and nWidth properties of the TImageList before adding any Bitmap.

1.6.5.20.2 TListBoxMod:Methods

■ Constructor ■ Standard ■ Only after Create()

Type	Name
■	AddImage
■	AddItem
■	CheckAll
■	Deleteltem
■	Deleteltems
■	FirstVisible
■	ForceltemVisible
■	InsertItem
■	IsItemVisible
■	ItemChecked
■	ItemHot
■	ItemPushed
■	ItemSelected
■	LastVisible
■	ModifyItem
■	RecCount
■	SetFirstVisible
■	UncheckAll

1.6.5.20.2.1 TListBoxMod:AddImage

Adds a new image to the control's olmImageList object..

Type	Standard
Parameters	<p><xImage>: Name of the resource, file name or Handle to Bitmap</p> <p><IMasked>: If true, the image will be converted to mask format, i.e. its colors</p>

	will be modified to shades of gray.
Return value	<nImage>: Ordinal position of the new image in olmageList

1.6.5.20.2.2 TListBoxMod:AddItem

Adds a new item to the list.

Type	Standard
Parameters	<cItem> Text of the new item
Return value	NIL

1.6.5.20.2.3 TListBoxMod:CheckAll

It marks the check of all the elements. It only makes sense to use it with the ICheckboxes property set to true.

Type	Standard
Parameters	<None>
Return value	NIL

1.6.5.20.2.4 TListBoxMod:DeleteItem

Deletes an item from the list.

Type	Standard
Parameters	<nItem> he number of item to deleted
Return value	NIL

1.6.5.20.2.5 TListBoxMod:DeleteItems

Deletes all items from the list.

Type	Standard
Parameters	None
Return value	NIL

1.6.5.20.2.6 TListBoxMod:FirstVisible

Returns the ordinal of the first visible item.

Type	Standard
Parameters	None
Return value	<nItem> Item number

1.6.5.20.2.7 TListBoxMod:ForceltemVisible

Forces an item in the list to be visible.

Type	Standard
Parameters	<nItem> umber of the item to be displayed
Return value	NIL

1.6.5.20.2.8 TListBoxMod:InsertItem

Inserts a new element in the list..

Type	Standard
Parameters	<nPos> Position of the item <cItem> Text for the new item
Return value	NIL

1.6.5.20.2.9 TListBoxMod:IsItemVisible

Returns whether a given item is visible.

Type	Standard
Parameters	<nItem> Number of the item
Return value	<IVisible> True if visible otherwise false

1.6.5.20.2.10 TListBoxMod:ItemChecked

Returns true if a given item is selected when the IMultipleSel property is set to true.

Type	Standard
Parameters	<nItem> Number of the given item
Return value	<IChecked> True if selected otherwise fail

1.6.5.20.2.11 TListBoxMod:ItemHot

Returns true if a given element is marked with the mouse.

Type	Standard
Parameters	<nItem> Number of the item
Return value	<IHot> True if marked, false otherwise

1.6.5.20.2.12 TListBoxMod:ItemPushed

Returns true if a given element is clicked by the mouse.

Type	Standard
Parameters	<nItem> Number of the item
Return value	<IPushed> True if pressed, false otherwise

1.6.5.20.2.13 TListBoxMod:ItemSelected

Returns true if a certain item is selected.

Type	Standard
Parameters	<nItem> Number of item
Return value	<ISelected> True if selected, false otherwise

1.6.5.20.2.14 TListBoxMod:LastVisible

Returns the ordinal of the last visible element..

Type	Standard
Parameters	None
Return value	<nItem> NúmeroItem number

1.6.5.20.2.15 TListBoxMod:ModifyItem

Modify an item in the list.

Type	Standard
Parameters	<nItem> Item number <cText> Newtext
Return value	NIL

1.6.5.20.2.16 TListBoxMod:RecCount

Returns the number of elements in the list.

Type	Standard
Parameters	None
Return value	<nTotal> Number of items

1.6.5.20.2.17 TListBoxMod:SetFirstVisible

Sets the first visible item in the list.

Type	Standard
Parameters	<nItem> Item number
Return value	<IOk> True if successful

1.6.5.20.2.18 TListBoxMod:UnCheckAll

Remove the check mark of all the items. It only makes sense to use it with the ICheckboxes property set to true.

Type	Standard
Parameters	<None>
Return value	NIL

1.6.5.20.3 TListBoxMod:Events

Name
OnCalcVirtualSize
OnChange
OnChanged
OnChangeSelected
OnCheckStateChanged
OnClick
OnDbClick
OnDrawItem
OnSelect

1.6.5.20.3.1 TListBoxMod:OnCalcVirtualSize

Event that occurs when calculating the actual width and height of the set of list items. It is rare that you will have to catch this event. However, if any list item makes use of HTML tags you will probably have to expand its width.

Parameters	<oSender> : Object that fires the event (Self) <@nVirtualWidth> : Total width of the list. This parameter is received by reference <@nVirtualHeight> : Total height of the list. This parameter is received by reference
Return value	NIL

1.6.5.20.3.2 TListBoxMod:OnChange

Event that occurs when the active item is changed.

Parameters	<oSender> : Object that fires the event (Self) <nOldItem> : Ordinal of the old item
-------------------	--

	<nNewItem>: Ordinal of the new item
Return value	NIL

1.6.5.20.3.3 TListBoxMod:OnChanged

Event that occurs in the change of the active element.

Parameters	<oSender>: Object that fires the event (Self)
Return value	NIL

1.6.5.20.3.4 TListBoxMod:OnChangeSelected

Event that occurs when the selected rows are changed.

Parameters	<oSender>: Object that fires the event (Self) <aSelected>: Array with the numbers of selected rows
Return value	NIL

1.6.5.20.3.5 TListBoxMod:OnCheckStateChanged

Event that occurs when the selected state of an element is changed.

Parameters	<oSender>: Object that fires the event (Self) <nItem>: Item ordinal <INewValue>: New selection status
Return value	NIL or Logic. If false is used, the change of state is canceled.

1.6.5.20.3.6 TListBoxMod:OnClick

Event that occurs when an item in the list is clicke

Parameters	<oSender>
-------------------	------------------------

	Reference to the object that fires the event <nKeyFlags> Keyboard and/or mouse status. It can be a combination of any of the following values: MK_CONTROL CTRL key pressed MK_LBUTTON Left mouse button pressed MK_MBUTTON Center mouse button pressed MK_RBUTTON Right mouse button pressed MK_SHIFT SHIFT key pressed Use the function IAnd() for your query <nRow> Pressed row
Return value	NIL

1.6.5.20.3.7 TListBoxMod:OnDbClick

Event that occurs when double clicking on a list item.

Parameters	<oSender> Reference to the object that fires the event <nKeyFlags> Keyboard and/or mouse status. It can be a combination of any of the following values: MK_CONTROL CTRL key pressed MK_LBUTTON Left mouse button pressed MK_MBUTTON Center mouse button pressed MK_RBUTTON Right mouse button pressed MK_SHIFT SHIFT key pressed Use the function IAnd() for your query <nRow> Pressed row
Return value	NIL

1.6.5.20.3.8 TListBoxMod:OnDrawItem

Event that occurs when each element of the list is painted.

Parameters	<oSender>: Object that fires the event (Self) <nItem>: Item ordinal <@cText>: Value with information. This parameter is received by reference. <@nImage>: Ordinal order of the image to be displayed according to the <code>oImageList</code> object of the control. This parameter is received by
-------------------	---

	reference. By default its value is zero which means no image.
	<@nClrText>: Text color. This parameter is received by reference.
	<@nClrPane>: Background color. This parameter is received by reference.
	<hDC>: context device handler
	<aRect>: Rectangle defining the possible painting area
Return value	NIL or Logic. If logical false the painting is canceled and it is the responsibility of the event to paint it.

1.6.5.20.3.9 TListBoxMod:OnSelect

Event that occurs when an item is selected and set as the active one in the list.

Parameters	<oSender>: Object that fires the event (Self) <nItem>: Item ordinal
Return value	NIL

1.6.5.21 TMemoMod

This class represents TMemo with the following enhancements:

- Windows 10 style
- Adapt to the color theme used in Windows Vista or later



Hierarchy File

TMemo descendant
 \source\MemoMod.prg

1.6.5.21.1 TMemotMod:Properties

■ Solo lectura ■ Asignable ■ Asignable en diseño ■ Asignable en run-time

Ámbito	Nombre	Tipo	Valor Inicial
■	IModBorder	Lógico	.T.
■	nClrColorization	Numérico	0

1.6.5.21.1.1 TMemotMod:IModBorder

Si verdadero el control mostrará un borde grueso al estilo de Windows 10.

Ámbito:	Asignable
Tipo:	Lógico
Valor inicial:	.T.

1.6.5.21.1.2 TMemotMod:nClrColorization

Border color when the control receives focus. If zero uses system color.

Scope	Assignable
Type:	Numeric
Initial value:	0

Consult appendix for possible colors

1.6.5.22 TProgressBarMod

This class represents a modern Windows 10 ProgressBar control.

Important note:

The functionality provided by the IMarquee property is obtained through the use of an internal timer, but this has two important limitations:

- You must constantly call the **ProcessMessages()** function in your work loop so that the progress bar animation does not stop.
- It is very likely that there will be stoppages in the animation due to the amount of work you are doing.

To avoid both problems, you should use a future (TFuture), but this forces you to use Harbour's Multi-thread library (mtvm.lib). For example:

```
METHOD MyHeavyTask() CLASS TForm1
    LOCAL oFuture, bTask, bComplete
```

```

bTask      := {|| MyHeavyTaskFunc() }
bComplete := {|| ::oProgress:IMarquee := .F. }

oFuture := TFuture():CreateFrom( bTask, bComplete )

::oProgress:IMarquee := .T.

RETURN nil

```

Hierarchy TStdControl descendant
File \source\ProgressBarMod.prg

1.6.5.22.1 TProgressBarMod:Properties

■ Read only ■ Assignable □ Design assignable ■ Run-time assignable

Scope	Name	Type	Initial value
□	IMarquee	Logical	.F.
■	nHeight	Numeric	20
■	nMarqueeSpeed	Numeric	100
■	nMax	Numeric	100
■	nMin	Numeric	0
□	nOrientation	Numeric	orHORIZONTAL
■	nStep	Numeric	10
■	nValue	Numeric	120
■	nWidth	Numeric	120

1.6.5.22.1.1 TProgressBarrMod:IMarquee

Progress indicator which does not increase in size, but by moving continuously along the length of the bar, it shows some activity, without specifying what percentage of progress has been completed.

Scope	Design assignable
Type	Logical
Initial value	.F.
See also	nMarqueeSpeed

1.6.5.22.1.2 TProgressBarrMod:nHeight

Vertical size of the control.

Scope	Assignable
Type	Numeric

Initial value	25
----------------------	----

1.6.5.22.1.3 TProgressBarrMod:nMax

Maximum value of the range indicated by the control. If not set, the operating system assigns 100.

Scope	Assignable
Type	Numeric
Initial value	100

1.6.5.22.1.4 TProgressBarrMod:nMarqueeSpeed

Speed of the animation in milliseconds.

Scope	Assignable
Type	Numeric
Initial value	100
See also	IMarquee

1.6.5.22.1.5 TProgressBarrMod:nMin

Minimum value of the range indicated by the control. If not set, the operating system assigns 0.

Scope	Assignable
Type	Numeric
Initial value	0

1.6.5.22.1.6 TProgressBarrMod:nOrientation

Indicates whether the control should be displayed horizontally or vertically.

Scope	Design assignable
Type	Numeric
Initial value	orHORIZONTAL
Possible values	orHORIZONTAL, orVERTICAL

1.6.5.22.1.7 TProgressBarrMod:nStep

Increment made by the progress bar every time it advances.

Scope	Assignable
Type	Numeric
Initial value	10

1.6.5.22.1.8 TProgressBarrMod:nValue

Current position on the progress bar.

Scope	Assignable
Type	Numeric
Initial value	0

1.6.5.22.1.9 TProgressBarrMod:nWidth

Horizontal control size.

Scope	Assignable
Type	Numeric
Initial value	120

1.6.5.22.2 TProgressBarMod:Events

Name
OnChange

1.6.5.22.2.1 TProgressBarMod:OnChange

Event that occurs when the control changes state.

Parameters	<oSender> : Reference to the object that fires the event <nPos> : New position
Return value	NIL

1.6.5.23 TProgressCircle

This class represents a **circular** ProgressBar control.

Important note:

The functionality provided by the `IMarquee` property is realized through the use of an internal timer, but this has two important limitations:

- You must be continuously calling the **ProcessMessages()** function in your work loop so that the progress bar animation does not stop.
- It is very likely that there will be stops in the animation due to the heavy work you are doing.

To avoid both problems, you should use a future (TFuture), but this forces you to use Harbour's Multi-thread library (mtvm.lib). For example:

```
METHOD MyHeavyTask() CLASS TForm1

    LOCAL oFuture, bTask, bComplete

    bTask      := {|| MyHeavyTaskFunc() }
    bComplete := {|| ::oProgressCircle:IMarquee := .F. }

    oFuture := TFuture():CreateFrom( bTask, bComplete )

    ::oProgressBar:IMarquee := .T.

RETURN nil
```

Hierarchy File

TStdControl descendant
\source\ProgressCircular.prg

1.6.5.23.1 TProgressCircle:Properties

■ Read only ■ Assignable □ Design assignable ■ Run-time assignable

Scope	Name	Type	Initial value
□	IMarquee	Logical	.F.
■	IShowPercent	Logical	.F.
■	nCircleGap	Numeric	20
■	nCircleSize	Numeric	20
■	nClrCircle	Numeric	clSystem
■	nHeight	Numeric	120
■	nMarqueeSpeed	Numeric	100
■	nMax	Numeric	100
■	nMin	Numeric	0
■	nValue	Numeric	120
■	nWidth	Numeric	120

1.6.5.23.1.1 TProgressCircular:IMarquee

The progress indicator does not increase in size, but moves around, all around the circular part of the control, indicating some activity, without specifying which part of the progress has been completed.

Scope	Design assignable
Type	Logical
Initial value	.F.
See also	nMarqueeSpeed

1.6.5.23.1.2 TProgressCircular:IShowPercent

If true, the percentage of progress is displayed.

Scope	Assignable
Type	Logical
Initial value	.F.

1.6.5.23.1.3 TProgressCircular:nCircleGap

Space in degrees is left empty on both sides of the progress circle to incorporate a control at the bottom.

Scope:	Assignable
Type:	Numeric
Initial value:	20

1.6.5.23.1.4 TProgressCircular:nCircleSize

Bar width in pixels..

Scope:	Assignable
Type:	Numeric
Initial value:	20

1.6.5.23.1.5 TProgressCircular:nClrCircle

Color of the progress circle.

Scope:	Assignable
Type:	Numeric
Initial value:	clSystem

(Consult appendix for possible colors)

1.6.5.23.1.6 TProgressCircular:nHeight

Vertical size of the control.

Scope	Assignable
Type	Numeric
Initial value	120

1.6.5.23.1.7 TProgressCircularMod:nMax

Maximum value of the range indicated by the control. If not set, the operating system assigns 100.

Scope	Assignable
Type	Numeric
Initial value	100

1.6.5.23.1.8 TProgressCircular:nMarqueeSpeed

Animation speed in milliseconds.

Scope	Assignable
Type	Numeric
Initial value	100
See also	IMarquee

1.6.5.23.1.9 TProgressCircular:nMin

Minimum value of the range indicated by the control. If not set, the operating system assigns 0.

Scope	Assignable
Type	Numeric
Initial value	0

1.6.5.23.1.10 TProgressCircular:nValue

Current position in the progress bar

Scope	Assignable
Type	Numeric
Initial value	0

1.6.5.23.1.11 TProgressCircular:nWidth

Horizontal control size.

Scope	Assignable
Type	Numeric
Initial value	120

1.6.5.23.2 TProgressCircular:Events

Name
OnChange

1.6.5.23.2.1 TProgressCircular:OnChange

Event that occurs when the control changes state.

Parameters	<oSender> : Reference to the object that fires the event <nPos> : New position
Return value	NIL

1.6.5.24 TRadioMod

This class represents a Radio object with Windows 10 appearance. The text of the control can use HTML formatting in the same way as the TLabelEx control.

Hierarchy	TStdControl descendant
See also	TRadio
File	\source\RadioMod.prg

1.6.5.24.1 TRadioMod:Properties

■ Read only ■ Assignable ■ Design assignable ■ Run-time assignable

Scope	Name	Type	Initial value
■	cGroup	Character	""
■	IChecked	Logical	.F.
■	ITransparent	Logical	.F.
■	nClrColorization	Numeric	c\System
■	nClrTextDisabled	Numeric	c\3DDkShadow
■	nLineSpacing	Numeric	100
■	nVAlignment	Numeric	vaTOP

1.6.5.24.1.1 TRadioMod:cGroup

Arbitrary name for the group of **TRadioMod** type controls that should work together. That is, when one is checked, the rest are unchecked.

Scope	Assignable
Type	Character
Initial value	""

1.6.5.24.1.2 TRadioMod:IChecked

If true, the control is checked.

Scope	Assignable
Type	Logical
Initial value	.F.

1.6.5.24.1.3 TRadioMod:ITransparent

The control is drawn transparent on the form to which it belongs.

Scope	Assignable
Type	Logical
Initial value	.T.

1.6.5.24.1.4 TRadioMod:nClrColorization

Color of the box and marker.

Scope:	Assignable
Type:	Numeric
Initial value:	clSystem

(Consult appendix for possible colors)

1.6.5.24.1.5 TRadioMod:nClrTextDisabled

Text color when the control is disabled.

Scope:	Assignable
Type:	Numeric
Initial value:	cl3DDkShadow

(Consult the appendix for possible colors)

1.6.5.24.1.6 TRadioMod:nLineSpacing

Percentage spacing. Maximum value 1000.

Scope	Assignable
Type	Numeric
Initial value	100

1.6.5.24.1.7 TRadioMod:nVAlignment

Vertical text alignment. Displays the box aligned up, centered or down with respect to the text.

Scope	Assignable
Type	Numeric
Initial value	vaTOP
Possible values	vaTOP, vaBOTTOM, vaCENTER

1.6.5.24.2 TRadioModMod:Events

Name
OnChange
OnClick

1.6.5.24.2.1 TRadioMod:OnChange

Event that is occurred when the state of the control changes.

As of Xailer 8.1 **INewValue** already incorporates the change of value and not the old value, but it is passed by reference in case you want to change it. The return value becomes useless.

Parameters	<oSender> Reference to the object that fires the event <@INewValue> Newvalue
Return value	<NIL>

1.6.5.24.2.2 TRadioMod:OnClick

Event that is triggered by clicking with the mouse on the control.

Parameters	<oSender> Reference to the object that fires the event
Return value	<NIL>

1.6.5.25 TScreenKeyboard

This class represents a screen keyboard control.



Hierarchy TWinControl descendant
File \source\ScreenKeyboard.prg

1.6.5.25.1 TScreenKeyboard:Properties

■ Read only ■ Assignable ■ Design assignable ■ Run-time assignable

Scope	Name	Type	Initial value
■	aClrKeys	Array	{}
■	aKeys	Array	{}
■	lHotTrack	Logical	.T.
■	lTransparent	Logical	.T.
■	nBorderStyle	Numeric	bvFLAT
■	nClrKeys	Numeric	cl3DDkShadow
■	nHeight	Numeric	230
■	nLayout	Numeric	klALPHA
■	nStyle	Numeric	bsDEFAULT
■	nWidth	Numeric	800
■	oFontShortcut	Object	NIL

1.6.5.25.1.1 TScreenKeyboard:aClrKeys

Array with all the keys colors.

Scope	Assignable
Type	Array
Initial value	{}

1.6.5.25.1.2 TScreenKeyboard:aKeys

Multi-dimensional array with all the keys that will be shown with the actual style.

Each element of the represents a key with the following information:

- The first four positions indicate its coordinates
- Fifth position indicates its type: normal (0), Shift (1) or Alt (2) key
- Sixth position indicates its ordinal on its object **TImageList**
- Eighth position indicates text in normal mode
- Ninth position indicates text in Alt mode
- Tenth position indicates text in Shift mode
- Following positions are used when the key is pushed for a short time

Scope	Assignable
Type	Array
Initial value	{}

1.6.5.25.1.3 TScreenKeyboard:IHotTrack

Keys change its color when mouse is over.

Scope	Assignable
Type	Logical
Initial value	.T.

1.6.5.25.1.4 TScreenKeyboard:ITransparent

Background is transparent.

Scope	Assignable
Type	Logical
Initial value	.T.

1.6.5.25.1.5 TScreenKeyboard:nBorderStyle

Borde style.

Scope	Assignable
--------------	------------

Type	Numeric
Initial value	bvFLAT
Valores posibles	bvNONE, bvRAISED, bvSUNKEN, bvBUMP, bvETCHED, bvFLAT

1.6.5.25.1.6 TScreenKeyboard:nClrKeys

Keys color. This property is useless except when its property nStyle is set to bsMODERN.

Scope:	Assignable
Type:	Numeric
Initial value:	cl3DDkShadow

(Consult appendix for possible colors)

1.6.5.25.1.7 TScreenKeyboard:nHeight

Control height.

Scope	Assignable
Type	Numeric
Initial value	230

1.6.5.25.1.8 TScreenKeyboard:nLayout

Keyboard layout.

Scope	Assignable
Type	Numeric
Initial value	kIALPHA
Valores posibles	kIALPHA, kALPHAINT, kICALCULATOR

1.6.5.25.1.9 TScreenKeyboard:nStyle

Control style.

Scope:	Assignable
Type:	Numeric
Initial value:	bsDEFAULT
Valores posibles:	bsDEFAULT, bsWINDOWS7, bsMODERN

1.6.5.25.1.10 TScreenKeyboard:nWidth

Control width.

Scope	Assignable
Type	Numeric
Initial value	230

1.6.5.25.1.11 TScreenKeyboard:oFontShortCut

TFont object to use for shortcuts.

Scope:	Assignable
Type:	Object
Initial value:	Nil

1.6.5.25.1.12 TScreenKeyboard:Events

Name	OnKeyClick
-------------	------------

Event fired when a screen key is pressed.

Parameters	<p><oSender>: Reference to the object that triggers the event</p> <p><nKey>: Ordinal key pressed on its aKeys array</p> <p><@cChar>: Key char pressed. Passed by reference</p> <p><@nKeyCode>: Key code pressed. Passed by reference</p>
Return value	<p><IResult>:</p> <ul style="list-style-type: none"> • If NIL or .T.: If nKeyCode not empty, nKeyCode pushing is simulated. If nKeyCode empty, but cChar not empty, cChar pushing is simulated. • if distinct from NIL and numeric or character, its return value pushing is simulated.

1.6.5.26 TShape

This class represents a Shape control, i.e. a shape, with different styles, which can contain other controls.



It supports a transparency mode, letting you see the background image of the control or form that contains it, both in XP and W98. It can also have a background image.

Hierarchy a descendant of TWinControl
File \source\Shape.prg

1.6.5.26.1 TShape:Properties

■ Read only ■ Assignable ■ Design assignable ■ Run-time assignable

Scope	Name	Type	Initial value
■	IFillShape	Logical	.F.
■	nClrLine	Numeric	clBlack
■	nLineWitdh	Numeric	1
■	nHeigth	Numeric	100
■	nOpacity	Numeric	0
■	nRadius	Numeric	0
■	nShapeType	Numeric	stHORZLINE
■	nVertices	Numeric	0
■	nWidth	Numeric	100

1.6.5.26.1.1 TShape:IFillShape

If true, the control will be filled with its background color.

Scope	Assignable
Type	Numeric
Initial value	100

1.6.5.26.1.2 TShape:nClrLine

Width of the brush to paint the lines.

Scope	Asignable
Type	Numeric
Initial value	1

1.6.5.26.1.3 TShape:nLineWidth

Color for paint lines

Scope	Asignable
Type	Numeric
Initial value	100

1.6.5.26.1.4 TShape:nHeight

Height of the control.

Scope	Asignable
Type	Numeric
Initial value	100

1.6.5.26.1.5 TShape:nOpacity

Opacity of control in percentage.

Scope:	Asignable
Type:	Numeric
Initial value:	0
Valores posibles:	0-100

1.6.5.26.1.6 TShape:nRadius

Radius of curvature for the junction of the control vertices.

Scope	Asignable
Type	Numeric
Initial value	0

1.6.5.26.1.7 TShape:nShapeType

Type of shape to be painted.

Scope:	Asignable
Type:	Numeric
Initial value:	stHORZLINE
Valores posibles:	stHORZLINE, stVERTLINE, stRECTANGLE, stPOLYGON, stCURVEDPOLYGON, stSTAR, stCURVEDSTAR, stELLIPSE

1.6.5.26.1.8 TShape:nVertices

Number of vertices of the shape when using the styles stPOLYGON, stCURVEDPOLYGON, stSTAR, stCURVEDSTAR in property nShapeType.

Scope	Asignable
Type	Numeric
Initial value	0

1.6.5.26.1.9 TShape:nWidth

Control width.

Scope	Asignable
Type	Numeric
Initial value	100

1.6.5.27 TSummerNote

This control is a complete HTML editor. It inherits from the TWebView control and therefore has the same requirements. See the TWebView documentation for more information.

Hierarchy	TWebView descendant
File	\source\SummerNote.prg

1.6.5.27.1 TSummerNote:Properties

Read only
 Assignable
 Design assignable
 Run-time assignable

Scope	Name	Type	Initial value
<input checked="" type="checkbox"/>	cHtml	Character	""
<input checked="" type="checkbox"/>	cLanguage	Character	""
<input checked="" type="checkbox"/>	cPlaceholder	Character	""
<input checked="" type="checkbox"/>	cToolbar	Character	"fontname fontsize style bold italic underline color paragraph ul ol table link picture undo redo"

1.6.5.27.1.1 TSummerNote:cHtml

HTML text of the control.

Scope	Assignable
Type	Character
Initial value	""

1.6.5.27.1.2 TSummerNote:cLanguage

Language to be used in editing.

Scope	Assignable
Type	Character
Initial value	""
Possible values	en-US es-ES ca-ES gl-ES eu-ES fr-FR de-DE it-IT pt-BR pt-PT

1.6.5.27.1.3 TSummerNote:cPlaceholder

Text to show when control is empty.

Scope	Assignable
Type	Character
Initial value	""

1.6.5.27.1.4 TSummerNote:cToolBar

Buttons to be displayed in the button bar Note how the "|" separator makes a separation in the button bar.

Scope	Assignable
Type	Character
Initial value	"fontname fontsize style bold italic underline color paragraph ul ol table link picture undo redo"
Possible values	Insert: picture link video table hr Font: fontname fontsize fontsizeunit color forecolor backcolor bold italic underline strikethrough superscript subscript clear Paragraph: style ol ul paragraph height Misc: fullscreen codeview undo redo help

1.6.5.27.2 TSummerNote:Methods

■ Constructor ■ Standard ■ Only after Create()

Type	Name
■	GetHtml
■	SetHtml

1.6.5.27.2.1 TSummerNote:GetHtml

Returns a string with the HTML text from the control

Type	Only after Create()
Parameters	<ILineBreak> If true, include carriage returns where paragraphs <p>or line

	returns are found.
Return value	cHtml

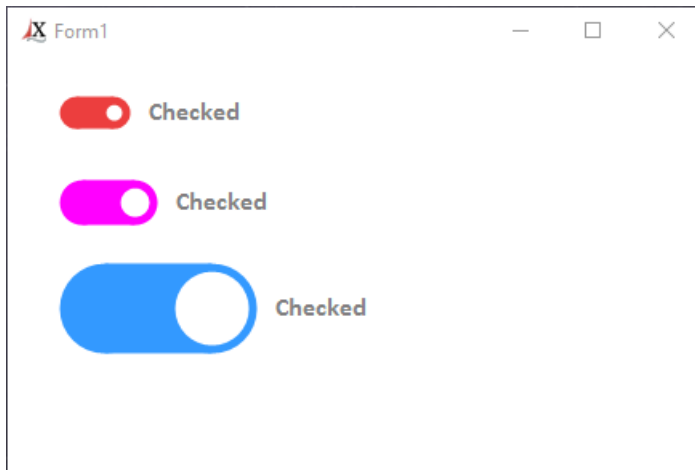
1.6.5.27.2.2 TSummerNote:SetHtml

Sets the HTML text of the control

Type	Only after Create()
Parameters	<cHtml> HTML text
Return value	NIL

1.6.5.28 TSwitch

This class represents a Windows 10 switch control.



Hierarchy TStdControl descendant
File \source\Switch.prg

1.6.5.28.1 TSwitch:Properties

■ Read only ■ Assignable ■ Design assignable ■ Run-time assignable

Scope	Name	Type	Initial value
■	IChecked	Logical	.F.
■	ITransparent	Logical	.T.
■	nClrColorization	Numeric	0
■	nHeight	Numeric	30
■	nWidth	Numeric	137

1.6.5.28.1.1 TSwitch:IChecked

If true the control will be checked.

Scope:	Assignable
Type:	Logical
Initial value:	.F.

1.6.5.28.1.2 TSwitch:ITransparent

if true the control background will be transparent

Scope:	Design assignable
Type:	Logical
Initial value:	.T.

1.6.5.28.1.3 TSwitch:nClrColorization

Color to use when control is checked. If zero system color will be used.

Scope:	Assignable
Type:	Numeric
Initial value:	0

Consult appendix for possible colors

1.6.5.28.1.4 TSwitch:nHeight

Control height.

Scope:	Assignable
Type:	Numeric
Initial value:	30

1.6.5.28.1.5 TSwitch:nWidth

Control width.

Scope:	Assignable
---------------	------------

Type:	Numeric
Initial value:	137

1.6.5.28.2 TSwitch:Methods

Constructor
 Standadr
 Only after create()

Type	Name
<input type="checkbox"/>	Click
<input type="checkbox"/>	Toggle

1.6.5.28.2.1 TSwitch:Click

Changes control state. Also triggers OnClick event.

Type	Only after Create()
Parameters	None
Return value	NIL

1.6.5.28.2.2 TSwitch:Toggle

Change control state.

Type	Only after Create()
Parameters	None
Returnn value	NIL

1.6.5.28.3 TSwitch:Events

Name
OnChange
OnClick

1.6.5.28.3.1 TSwitch:OnChange

Event triggered when the control changes its state.

Parameters:	<oSender>: Reference to the object that triggers the event
Return value:	NIL or Logical. A false value stops the change

1.6.5.28.3.2 TSwitch:OnClick

Event triggered when the control is clicked.

Parameters:	<oSender>: Reference to the object that triggers the event
Return value:	NIL

1.6.5.29 TTrackBarMod

This class represents a modern Windows 10 TrackBar control.

Hierarchy	TStdControl descendant
File	\source\TrackBarMod.prg

1.6.5.29.1 TTrackBarMod:Properties

Read only
 Assignable
 Design assignable
 Run-time assignable

Scope	Name	Type	Initial value
<input type="checkbox"/>	nClrBarBack	Numeric	cl3DDkShadow
<input type="checkbox"/>	nClrBarFore	Numeric	clSystem
<input type="checkbox"/>	nHeight	Numeric	32
<input type="checkbox"/>	nMax	Numeric	100
<input type="checkbox"/>	nMin	Numeric	0
<input type="checkbox"/>	nOrientation	Numeric	orHORIZONTAL
<input type="checkbox"/>	nValue	Numeric	120
<input type="checkbox"/>	nWidth	Numeric	120

1.6.5.29.1.1 TTrackBarMod:nHeight

Vertical size of the control.

Scope	Assignable
Type	Numeric
Initial value	32

1.6.5.29.1.2 TTrackBarMod:nClrBarBack

Bar color for the unselected part

Scope:	Assignable
Type:	Numeric
Initial value:	cl3DDkShadow

(Consult appendix for possible colors)

1.6.5.29.1.3 TTrackBarMod:nClrBarFore

Bar color for the selected part.

Scope:	Assignable
Type:	Numeric
Initial value:	clSystem

(Consult appendix for possible colors)

1.6.5.29.1.4 TTrackBarMod:nMax

Maximum value of the range indicated by the control. If not set, the operating system assigns 100.

Scope	Assignable
Type	Numeric
Initial value	100

1.6.5.29.1.5 TTrackBarMod:nMin

Minimum value of the range indicated by the control. If not set, the operating system assigns 0.

Scope	Assignable
Type	Numeric
Initial value	0

1.6.5.29.1.6 TTrackBarMod:nOrientation

Indicates whether it should be displayed horizontally or vertically.

Scope	Design assignable
Type	Numeric
Initial value	orHORIZONTAL
Possible values	orHORIZONTAL, orVERTICAL

1.6.5.29.1.7 TTrackBarMod:nValue

Current position on the progress bar..

Scope	Assignable
Type	Numeric
Initial value	0

1.6.5.29.1.8 TTrackBarMod:nWidth

Horizontal control size.

Scope	Assignable
Type	Numeric
Initial value	120

1.6.5.29.2 TTrackBarMod:Eventos

Name
OnChange

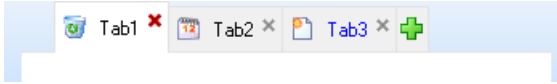
1.6.5.29.2.1 TTrackBarMod:OnChange

Event that occurs when the control changes state.

Parameters	<oSender>: Reference to the object that fires the event <nPos>: New position
Return value	NIL

1.6.5.30 TTabs

This class represents a Tab control that surpasses the limitations of Windows original control (TTabCtrl). Each tab is represented by TTabItem object control.



Any control you insert on a **TTabs** container will always be visible, independently of the active tab. If you need that each tab has its independent controls area you must use a TTabsFolder control.

Hierarchy	TScrollingWinControl descendant
See also	TTabsFolder
File	\source\Tabs.prg
Samples	\samples\Tabs\

1.6.5.30.1 TTabs:Properties

■ Read only ■ Assignable ■ Design assignable ■ Run-time assignable

Scope	Name	Type	Initial value
■	altItems	Array	{}
■	IFixedWidth	Logical	.F.
■	IPhantomTab	Logical	.F.
■	ISameFrameSize	Logical	.F.
■	IShowActive	Logical	.T.
■	IShowTabsMenu	Logical	.F.
■	ITabCloseButtons	Logical	.F.
■	ITabStop	Logical	.F.
■	ITabsCanMove	Logical	.F.
■	ITransparent	Logical	.F.
■	nFrame	Numeric	tfNONE
■	nHeight	Numeric	200
■	nIndex	Numeric	0
■	nItemPadding	Numeric	10
■	nOrientation	Numeric	orTOP
■	nPhantomImage	Numeric	0
■	nTabsClrPane	Numeric	clBtnFace
■	nTabsClrPaneHot	Numeric	clGradientInActiveCaption
■	nTabsClrPaneSel	Numeric	clWindow
■	nTabsClrPen	Numeric	cl3DLight
■	nTabsClrText	Numeric	clWindowText
■	nTabsClrTextHot	Numeric	clWindowText
■	nTabsClrTextSel	Numeric	clWindowText
■	nTabsMargin	Numeric	10

■	nTabSize	Numeric	27
■	nWidth	Numeric	200
■	oImageList	Object	TImageList

1.6.5.30.1.1 TTabs:altItems

Tabs list represented by TTabItem objects.

Scope	Assignable
Type	Array
Initial value	{}

1.6.5.30.1.2 TTabs:IFixedWidth

If true the tabs will have a fixed width

Scope	Assignable
Type	Logical
Initial value	.F.

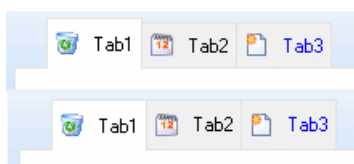
1.6.5.30.1.3 TTabs:IPhantomTab

If true a phantom tab will be shown with a plus icon which lets you create new tabs in run-time. See also OnNewTab event.

Scope	Assignable
Type	Logical
Initial value	.F.

1.6.5.30.1.4 TTabs:ISameFrameSize

If true all the tabs will have the same dimension. On the image you can see how the second tab control, all its tabs have the same height.



Scope	Assignable
--------------	------------

Type	Logical
Initial value	.F.

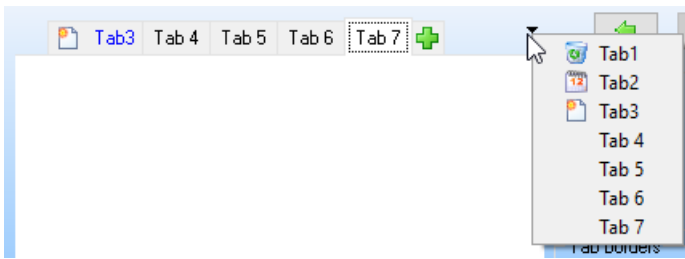
1.6.5.30.1.5 TTabs:IShowActive

If false the active item will not have a special painting.

Scope	Assignable
Type	Logical
Initial value	.T.

1.6.5.30.1.6 TTabs:IShowTabsMenu

If true a menu will be shown when all the tabs do not fit in the control.



Scope	Assignable
Type	Logical
Initial value	.F.

1.6.5.30.1.7 TTabs:ITabCloseButtons

If true, every tab will show a close icon to close the tab. Is possible to trap the event with the OnCloseTab event to modify its behaviour or to perform an additional action.

Scope	Assignable
Type	Logical
Initial value	.F.

1.6.5.30.1.8 TTabs:ITabStop

The control receives focus when pushing the TAB key.

Scope	Assignable
Type	Logical
Initial value	.F.

1.6.5.30.1.9 TTabs:ITabsCanMove

If true tabs control can be move by a drag & drop operation with the mouse.



Scope	Assignable
Type	Logical
Initial value	.F.

1.6.5.30.1.10 TTabs:ITransparent

If true the background control is transparent.

Scope	Assignable
Type	Logical
Initial value	.F.

1.6.5.30.1.11 TTabs:nFrame

Control border.

Scope	Assignable
Type	Numeric
Initial value	tfNONE
Possible values	tfNONE, tfTABS, tfALL

1.6.5.30.1.12 TTabs:nHeight

Control height.

Scope	Assignable
Type	Numeric
Initial value	200

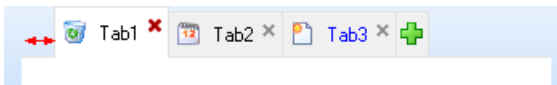
1.6.5.30.1.13 TTabs:nIndex

Actual tab selected.

Scope	Assignable
Type	Numeric
Initial value	0

1.6.5.30.1.14 TTabs:nItemPadding

Space in pixels added to left and right sizes of every tab .



Scope	Assignable
Type	Numeric
Initial value	10

1.6.5.30.1.15 TTabs:nOrientation

Control orientation.

Scope:	Assignable
Type:	Numeric
Initial value:	orTOP
Possible values:	orTOP, orBOTTOM, orLEFT, orRIGHT

1.6.5.30.1.16 TTabs:nPhantomImage

Ordinal image number on its `olmageList` property which it will be used to show in the phantom tab. If zero, the plus icon from Xailer resources will be used.

Scope	Assignable
Type	Numeric
Initial value	0

1.6.5.30.1.17 TTabs:nTabsClrPane

Background color for tabs. You can trap the `OnDrawTab` event to change its behaviour for a specific tab.

Scope	Assignable
Type	Numeric
Initial value	<code>clBtnFace</code>

1.6.5.30.1.18 TTabs:nTabsClrPaneHot

Background color for tabs when mouse is over them. You can trap the `OnDrawTab` event to change its behaviour for a specific tab.

Scope	Assignable
Type	Numeric
Initial value	<code>clGradientInActiveCaption</code>

1.6.5.30.1.19 TTabs:nTabsClrPaneSel

Background color for selected tabs. You can trap the `OnDrawTab` event to change its behaviour for a specific tab.

Scope	Assignable
Type	Numeric
Initial value	<code>clWindow</code>

1.6.5.30.1.20 TTabs:nTabsClrPen

Pen color for tabs border. See also propiedad nFrame.

Scope	Assignable
Type	Numeric
Initial value	cl3DLight

1.6.5.30.1.21 TTabs:nTabsClrText

Text color for tabs. You can trap the OnDrawTab event to change its behaviour for a specific tab.

Scope	Assignable
Type	Numeric
Initial value	clWindowText

1.6.5.30.1.22 TTabs:nTabsClrTextHot

Text color for tabs when mouse is over them. You can trap the OnDrawTab event to change its behaviour for a specific tab.

Scope	Assignable
Type	Numeric
Initial value	clWindowText

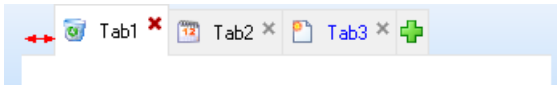
1.6.5.30.1.23 TTabs:nTabsClrTextSel

Text color for tabs selected. You can trap the OnDrawTab event to change its behaviour for a specific tab.

Scope	Assignable
Type	Numeric
Initial value	clWindow

1.6.5.30.1.24 TTabs:nTabsMargin

Space in pixels from the border of the control to its first tab.



Scope	Assignable
Type	Numeric
Initial value	0

1.6.5.30.1.25 TTabs:nTabsSize

Tab size. Width or height depending of its orientation.

Scope	Assignable
Type	Numeric
Initial value	24

1.6.5.30.1.26 TTabs:nWidth

Control width.

Scope	Assignable
Type	Numeric
Initial value	200

1.6.5.30.1.27 TTabs:olmageList

TImageList object with all the images used by the control.

Scope	Assignable
Type	Objeto
Initial value	TImageList

The TImageList object is instantiate on the New() constructor. The first image included on the TImageList sets the dimension of future images introduced. If the first image is a image combination is important that you set the nHeight and nWidth TImageList properties before adding any image.

1.6.5.30.2 TTabs:Methods

■ Constructor ■ Standard ■ Only after Create()

Type	Name
■	AddItem
■	Deleteltem
■	Deleteltems
■	InsertItem
■	MoveTab
□	RestoreState
□	SaveState

1.6.5.30.2.1 TTabs:AddItem

Adds a new TTabItem tab.

Type	Standard
Parameters	<p>[<cText>]: Text to show on tag</p> <p>[<nImage>]: Ordinal image on its olmageList property</p> <p>[<cTooltip>]: Tool tip for tab</p> <p>[<nSize>]: Tab size</p> <p>[<nAlignment>]: Text alignment: taLEFT, taRIGHT, taCENTER</p> <p>[<nBmpAlignment>]: Image alignment: taLEFT, taRIGHT</p> <p>[<IVisible>]: True if tab should be visible</p> <p>[<IEnabled>]: True if tab should be enabled</p> <p>Default values for all the parameters are the same that TTabsItem Create constructor method.</p>
Return value	TTabsItem object

1.6.5.30.2.2 TTabs:Deleteltem

Deletes tab.

Type	Standard
Parameters	<p><nItem>: Tab index to delete</p>
Return value	NIL

1.6.5.30.2.3 TTabs:DeleteItems

Deletes all tabs.

Type	Standard
Parameters	None
Return value	NIL

1.6.5.30.2.4 TTabs:InsertItem

Inserts a new TTabItem tab.

Type	Standard
Parameters	<p>[<nItem>]: Tab position</p> <p>[<cText>]: Text to show on tag</p> <p>[<nImage>]: Ordinal image on its olmageList property</p> <p>[<cTooltip>]: Tool tip for tab</p> <p>[<nSize>]: Tab size</p> <p>[<nAlignment>]: Text alignment: taLEFT, taRIGHT, taCENTER</p> <p>[<nBmpAlignment>]: Image alignment: taLEFT, taRIGHT</p> <p>[<IVisible>]: True if tab should be visible</p> <p>[<IEnabled>]: True if tab should be enabled</p> <p>Default values for all the parameters are the same that TTabItem Create constructor method..</p>
Return value	TTabItem object

1.6.5.30.2.5 TTabs:MoveTab

Moves tab position.

Type	Standard
Parameters	<p>[<nFrom>]: Initial position</p> <p>[<nTo>]:</p>

	Final position
Return value	True if success

1.6.5.30.2.6 TTabs:RestoreState

Restore tab positions saved by SaveState method.

Type	Only after create
Parameters	[<cState]: State string
Return value	True if success
See also	SaveState

1.6.5.30.2.7 TTabs:SaveState

Saves tabs actual position to a string.

Type	Only after create
Parameters	None
Return value	State string
See also	RestoreState

1.6.5.30.3 TTabs:Events

Name
OnCalcRect
OnChange
OnChanging
OnCloseTab
OnClick
OnDrawTab
OnDrawTabsBackground
OnNewTab
OnRClick

1.6.5.30.3.1 TTabs:OnCalcRect

Event triggered when the tabs sizes are calculated. The aRect sizes can be modified to adjust its size.

Parameters:	<oSender>: Reference to the object that triggers the event <nIndex>: Tab index <@aRect>: Calculated array size
Return value:	NIL

1.6.5.30.3.2 TTabs:OnChange

Event triggered when active tab is changed. See also OnChanging event.

Parameters:	<oSender>: Reference to the object that triggers the event <nNewTab>: New active <nOldTab>: Old active tab
Return value:	NIL

1.6.5.30.3.3 TTabs:OnChanging

Event triggered when active tab is changing.

Parameters:	<oSender>: Reference to the object that triggers the event <nNewTab>: New active tab <nOldTab>: Olds active tab
Return value:	<IChange>: If returns NIL or a .T. value the change operation is accepted

1.6.5.30.3.4 TTabs:OnCloseTab

Event triggered when a tab is closing.

Parameters:	<oSender>: Reference to the object that triggers the event <nIndex>: Closing tab
Return value:	NIL

Sample:

```
METHOD FolderCloseTab( oSender, nIndex ) CLASS TFrmMain
    oSender:aItems[ nIndex ]:Delete()
RETURN Nil
```

1.6.5.30.3.5 TTabs:OnClick

Event triggered when mouse left button is clicked.

Parameters:	<oSender>: Reference to the object that triggers the event <nIndex>: Tab index <nPosX>: Mouse X position <nPosY>: Mouse Y position
Return value:	NIL

1.6.5.30.3.6 TTabs:OnDrawTab

Event triggered when each of the tabs is painted.

Parameters:	<oSender>: Reference to the object that triggers the event <nIndex>: Tab index <@cText>: Text to show. Passed by reference <@nImage>: Ordinal image to show. Passed by reference <@nClrText>: Text color. Passed by reference <@nClrPane>: Background color. Passed by reference <nState>: Tab state: tsNORMAL (0) tsSELECTED (1) tsHOT (2) tsDISABLED (3) <hDC>: Device context handle <@aRect>: Rectangle array. Passed by referene
Return value:	NIL

1.6.5.30.3.7 TTabs:OnDrawTabsBackground

Event when tabs background is painted.

Parameters:	<oSender>: Reference to the object that triggers the event
	<hDC>: Device context handle
	<@aRect>: Rectangle array. Passed by reference
Return value:	NIL

1.6.5.30.3.8 TTabs:OnNewTab

Event triggered when a new tab is created.

Parameters:	<oSender>: Reference to the object that triggers the event
Return value:	NIL

1.6.5.30.3.9 TTabs:OnRClick

Event triggered when mouse right button is clicked.

Parameters:	<oSender>: Reference to the object that triggers the event
	<nIndex>: Tab index
	<nPosX>: Mouse X position
	<nPosY>: Mouse Y position
Return value:	NIL

1.6.5.31 TTabItem

Class to manipulate tabs from TTabs control.

Descripción:

This class represents each of the tabs on TTabs or TTabsFolder control.

Hierarchy	TComponent descendant
File	\source\TabItem.prg

1.6.5.31.1 TTabItem:Properties

■ read Only ■ Assignable ■ Design assignable ■ Run-time assignable

Scope	Name	Type	Initial value
■	cText	Character	""
■	cToolTip	Character	""
■	lEnabled	Logical	.T.
■	lVisible	Logical	.T.
■	nAlignment	Numeric	taCENTER
■	nBmpAlignment	Numeric	taLEFT
■	nCrLfText	Numeric	cWindowText
■	nImage	Numeric	0
■	nItem	Numeric	0
■	nSize	Numeric	0
■	oParent	Object	NIL

1.6.5.31.1.1 TTabItem:cText

Text to show on tab, by default "".

Scope	Assignable
Type	Character
Initial value	""

1.6.5.31.1.2 TTabItem:cToolTip

Tool tip to show on tab.

Scope	Assignable
Type	Character
Initial value	""

1.6.5.31.1.3 TTabItem:lEnabled

True if tab can be selected.

Scope	Assignable
Type	Logical
Initial value	.T.

1.6.5.31.1.4 TTabItem:IVisible

True if tab is visible.

Scope	Assignable
Type	Logical
Initial value	.T.

1.6.5.31.1.5 TTabItem:nAlignment

Text tab alignment.

Scope	Assignable
Type	Numeric
Initial value	taCENTER
Possible values	taLEFT, taRIGHT, taCENTER

1.6.5.31.1.6 TTabItem:nBmpAlignment

Image alignment on tab.

Scope	Assignable
Type	Numeric
Initial value	taLEFT
Possible values	taLEFT, taRIGHT

1.6.5.31.1.7 TTabItem:nCrLfText

Tab text color.

Scope	Assignable
Type	Logical
Initial value	clWindowText

Check appendix for possible color values

1.6.5.31.1.8 TTabItem:nImage

Image to show on tab, corresponds to image number on its parent **oImageList** TTab property.

Scope	Assignable
Type	Numeric
Initial value	0

1.6.5.31.1.9 TTabItem:nItem

Actual tab position on its container TTab.

Scope	Read only
Type	Numeric
Initial value	0

1.6.5.31.1.10 TTabItem:nSize

Tab size. If zero is calculated automatically based on its text and image.

Scope	Assignable
Type	Numeric
Initial value	0

1.6.5.31.1.11 TTabItem:oParent

TTabs container.

Scope	Read only
Type	Objeto
Initial value	NIL

1.6.5.31.2 TTabItem:Methods

■ Constructor ■ Standard

Type	Name
■	Create
■	Delete
■	Select
■	Selected

1.6.5.31.2.1 TTabItem:Create

This method creates a new tab on a TTabs control.

Type	Constructor
Parameters	<p>[<oParent>] Container TTabs</p> <p>[<nItem>]: Tab position</p> <p>[<cText>]: Text to show</p> <p>[<nImage>]: Ordinal image number on its olmageList container property</p> <p>[<cTooltip>]: Tool tip for the tab</p> <p>[<nSize>]: Tab size</p> <p>[<nAlignment>]: Text alignment: taLEFT, taRIGHT, taCENTER</p> <p>[<nBmpAlignment>]: Image alignment: taLEFT, taRIGHT</p> <p>[<IVisible>]: If true the tab will be visible</p> <p>[<IEnabled>]: If true the tab can be selected</p>
Return value	Self

1.6.5.31.2.2 TTabItem>Delete

Deletes the tab from its TTabs container.

Type	Standard
Parameters	None
Return value	NIL

1.6.5.31.2.3 TTabItem>Select

Selects this tab on its TTabs container.

Type	Standard
Parameters	None
Return value	NIL

1.6.5.31.2.4 TTabItem:Selected

True if tab is selected.

Type	Standard
Parameters	Ninguno
Return value	<IValue>

1.6.5.31.2.5 TTabItem:Events

Name
OnClick
OnClose
OnRClick

Event fired when left mouse button is clicked.

Parameters:	<oSender>: A reference to the object that triggers the event <nPosX>: Mouse X position <nPosY>: Mouse Y position
Return value:	NIL

Event fired when the tab is closed.

Parametros:	<oSender>: A reference to the object that triggers the event <nPosX>: Mouse X position <nPosY>: Mouse Y position
Return value:	<IValue> if different from Empty() the close process is aborted

Event fired when right mouse button is clicked..

Parameters:	<oSender>:
--------------------	-------------------------

A reference to the object that triggers the event

<nPosX>:

Mouse X position

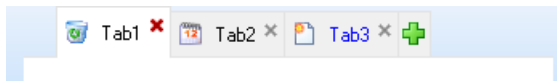
<nPosY>:

Mouse Y position

Return value: NIL

1.6.5.32 TTabsFolder

This class represents a advanced Folder control. Each tab of the control represents a TTabsFolderPage class object.



Any control you insert on a **TTabsFolder** object will only be visible when that tab is active. If you wan that all tabs shared the same area you must use a TTabs control instead.

Hierarchy	Descendiente de TTabs
See also	TTabs
File	\source\TabsFolder.prg

1.6.5.32.1 TTabsFolder:Properties

■ Read only ■ Assignable ■ Design assignable ■ Run-time assignable

Scope	Name	Type	Initial value
■	altems	Array	{}

1.6.5.32.1.1 TTabsFolder:altems

List of tab controls represented by TTabsFolderPage objects.

Scope	Design Assignable
Type	Arrayz
Initial value	{}

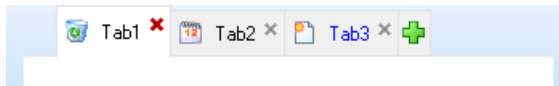
1.6.5.33 TTabsFolderPage

This class represents each of the container control created by each tab of a TTabsFolder.

You can can not create this control directly, they are created automatically by TTabsFolder control every time a tab is created.

Supports a background image and automatic scroll bars when its controls do not fit on its client

area.



Hierarchy TScrollingWinControl descendant
See also TTabsFolder
File \source\TabsFolderPage.prg

1.6.5.33.1 TTabsFolderPage:Properties

■ Read only ■ Assignable ■ Design assignable ■ Run-time assignable

Scope	Name	Type	Initial value
■	IEnabled	Logical	.T.
■	IVisible	Logical	.T.
■	ITransparent	Logical	.F.
■	cText	Character	""
■	cTooltip	Character	""
■	nAlignment	Numeric	taCENTER
■	nBmpAlignment	Numeric	taLEFT
■	nClrText	Numeric	clWindowText
■	nImage	Numeric	0
■	nItem	Numeric	0
■	nSize	Numeric	0
■	oParent	Object	NIL

1.6.5.33.1.1 TTabsFolderPage:IEnabled

True if tab can be selected.

Scope	Assignable
Type	Logical
Initial value	.T.

1.6.5.33.1.2 TTabsFolderPage:IVisible

True if tab is visible.

Scope	Assignable
Type	Logical
Initial value	.T.

1.6.5.33.1.3 TTabsFolderPage:ITransparent

The control is painted transparent over its container.

Scope	Assignable
Type	Logical
Initial value	.F.

1.6.5.33.1.4 TTabsFolderPage:cText

Text to show on tab.

Scope	Assignable
Type	Character
Initial value	""

1.6.5.33.1.5 TTabsFolderPage:cToolTip

Tool tip to show on tab.

Scope	Assignable
Type	Character
Initial value	""

1.6.5.33.1.6 TTabsFolderPage:nAlignment

Tab text alignment.

Scope	Assignable
Type	Numeric
Initial value	taCENTER
Possible values	taLEFT, taRIGHT, taCENTER

1.6.5.33.1.7 TTabFolderPage:nBmpAlignment

Tab image alignment.

Scope	Assignable
Type	Numeric
Initial value	taLEFT
Possible values	taLEFT, taRIGHT

1.6.5.33.1.8 TTabFolderPage:nClrText

Text color.

Scope:	Assignable
Type:	Numeric
Initial value:	clWindowText

(Consult appendix for possible colors)

1.6.5.33.1.9 TTabFolderPage:nImage

Image to show on tab. A ordinal value that corresponds with its position on the olmageList property of its container TTabFolder.

Scope	Assignable
Type	Numeric
Initial value	0

1.6.5.33.1.10 TTabFolderPage:nItem

Actual position of the tab on its TTabFolder container.

Scope	Read only
Type	Numeric
Initial value	0

1.6.5.33.1.11 TTabsFolderPage:nSize

Tab size. If is equal to zero, its size is calculated automatically based on its text and image.

Scope	Asignable
Type	Numeric
Initial value	0

1.6.5.33.1.12 TTabsFolderPage:oParent

TTabsFolder control container.

Scope	Read only
Type	Object
Initial value	NIL

1.6.5.33.2 TTabsFolderPage:Methods

■ Constructor ■ Standard ■ Only after Create()

Type	Name
■	Create
■	Delete
■	Select
■	Selected

1.6.5.33.2.1 TTabsFolderPage:Create

This method creates new tab on a TTabsFolder control.

Type	Constructor
Parameters	[<oParent>] Container TTabsFolder [<nItem>]: Tab position [<cText>]: Text t show [<nImage>]: Image number to use from its oImageList container property [<cTooltip>]: Tab tool tip [<nSize>]: Tab size [<nAlignment>]: Text alignment: taLEFT, taRIGHT, taCENTER [<nBmpAlignment>]:

	Image alignment: taLEFT, taRIGHT [<IVisible>]: True if tab should be visible. [<IEnabled>]: True if tab should be enabled.
Return value	Self

1.6.5.33.2.2 TTabsFolderPage:Delete

Deletes the TTabsFolderPage from its TTabsFolder container.

Type	Standard
Parameters	None
Return value	<IOk> True if success

1.6.5.33.2.3 TTabsFolderPage:Select

Select this TTabsFolderPage on its TTabsFolder container.

Type	Standard
Parameters	None
Return value	NIL

1.6.5.33.2.4 TTabsFolderPage:Selected

True if this TTabsFolderPage is selected.

Type	Standard
Parameters	None
Return value	<IResult>

1.6.5.33.3 TTabsFolderPage:Events

Name
OnClick
OnClose
OnRClick

1.6.5.33.3.1 TTabsFolderPage:OnClick

Event triggered when mouse left button is clicked.

Parameters:	<oSender>: Reference to the object that triggers the event <nPosX>: Mouse X position <nPosY>: Mouse Y position
Return value:	NIL

1.6.5.33.3.2 TTabsFolderPage:OnClose

Event that is triggered when closing the tab.

Parameters:	<oSender>: Reference to the object that triggers the event <nPosX>: Mouse X position <nPosY>: Mouse Y position
Return value:	<IValue> If distinct from Empty() closing is aborted

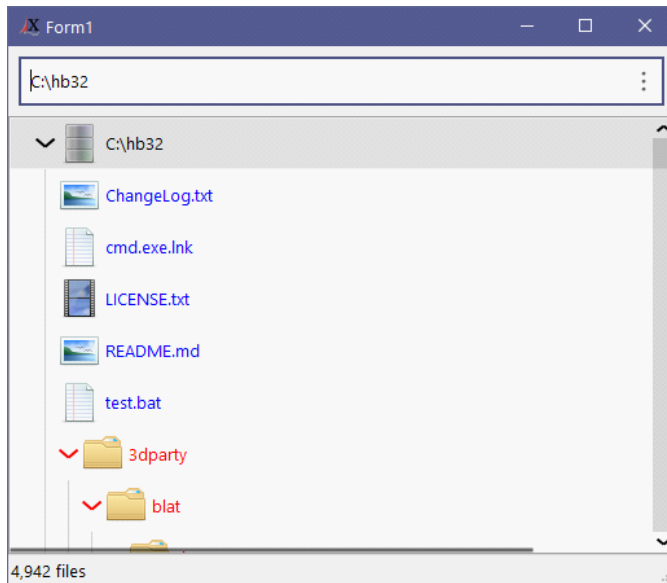
1.6.5.33.3.3 TTabsFolderPage:OnRClick

Event triggered when mouse right button is clicked.

Parameters:	<oSender>: Reference to the object that triggers the event <nPosX>: Mouse X position <nPosY>: Mouse Y position
Return value:	NIL

1.6.5.34 TTreeViewMod

This class represents a TreeView control of modern Windows 10 type.



Hierarchy File

TListBoxMod descendant
\source\TreeViewMod.prg

1.6.5.34.1 TTreeViewMod:Properties

■ Read only ■ Assignable ■ Design assignable ■ Run-time assignable

Scope	Name	Type	Initial value
■	altems	Array	{}
■	lEditLabels	Logical	.F.
■	lHasLines	Logical	.F.
■	lMultiLine	Logical	.F.
■	lSingleExpand	Logical	.T.
■	lTooltips	Logical	.T.
■	nGap	Numeric	4
■	nMargin	Numérico	20
■	oEdit	Object	TEditMod

1.6.5.34.1.1 TTreeViewMod:altems

Array of TTreeViewItemMod objects with the various items of the control.

Scope:	Run-time assignable
Type:	Array
Initial value:	{}

1.6.5.34.1.2 TTreeViewMod:IEditLabels

If True, TreeView elements can be edited by pressing F2 or by code..

Scope	Assignable
Type	Logical
Initial value	.F.

1.6.5.34.1.3 TTreeViewMod:IHasLines

If True, the branch lines will be displayed when the control has the mouse over them.

Scope	Assignable
Type	Logical
Initial value	.F.

1.6.5.34.1.4 TTreeViewMod:IMultiLine

If True, the text to be displayed for each item can be multi-line.

Scope	Assignable
Type	Logical
Initial value	.F.

1.6.5.34.1.5 TTreeViewMod:ISingleExpand

If True, the effect of expanding a branch will only open the current branch and not the lower ones.

Scope	Assignable
Type	Logical
Initial value	.T.

1.6.5.34.1.6 TTreeViewMod:ITooltips

If True, tool-tips of the tree elements will be displayed (if any).

Scope	Assignable
Type	Logical
Initial value	.T.

1.6.5.34.1.7 TTreeViewMod:nGap

The space between the text and the possible image of the item.

Scope	Assignable
Type	Numeric
Initial value	4

1.6.5.34.1.8 TTreeViewMod:nMargin

Initial left margin for painting tree elements.

Scope	Assignable
Type	Numeric
Initial value	20

1.6.5.34.1.9 TTreeViewMod:oEdit

TEditMod type object to perform editing when the IEditLabels property is set to True.

Scope	Read only
Type	Object
Initial value	TEditMod

1.6.5.34.2 TTreeViewMod:Methods

■ Constructor ■ Standard

Type	Name
■	AddItem
■	BeginPaint
■	CollapseAll

■	Deleteltems
■	EditLabel
■	EndEditLabelNow
■	EndPoint
■	ExpandAll
■	GetAllItems
■	GetCount
■	GetItemAtPos
■	GetItemByID
■	GetItemByName
■	GetItemPos
■	GetSelectedItem
■	InsertItem
■	IsEmpty
■	SelectItem
■	SelectRoot

1.6.5.34.2.1 TTreeViewMod:AddItem

Adds a new TTreeViewItemMod object to the TTreeViewMod object.

Type	Standard
Parameters	<p><cltem> Text to display the element</p> <p>[<nImage>] Number (index) of the image to display in the olmageList object, 0 by default</p> <p>[<nSellImage>] Number (index) for the image to display in the olmageList object when the item is selected, 0 by default</p> <p>[<IChecked>] True if the item is displayed with check box checked</p> <p>[<nClrText>] Text color</p>
Return value	TTreeViewItemMod

1.6.5.34.2.2 TTreeViewMod:BeginPaint

Indicates the start of a tree element painting process. You should call this method in conjunction with the EndPaint to avoid unnecessary painting.

Type	Standard
Parameters	None
Return value	NIL

1.6.5.34.2.3 TTreeViewMod:CollapseAll

Collapse all branches of the tree.

Type	Standard
Parameters	None
Return value	NIL

1.6.5.34.2.4 TTreeViewMod:DeleteItems

Deletes all elements of the control.

Type	Standard
Parameters	None
Return value	NIL

1.6.5.34.2.5 TTreeViewMod:EditLabel

Starts editing a control element.

Type	Standard
Parameters	<oltem> Object TTreeViewItem
Return value	NIL

1.6.5.34.2.6 TTreeViewMod:EndEditLabelNow

End of the possible edition that was in progress.

Type	Standard
Parameters	<ISave> If True, the value of the item text will be updated. By default .T.
Return value	NIL

1.6.5.34.2.7 TTreeViewMod:EndPaint

Indicates the end of a tree element painting process. You should call this method in conjunction with the BeginPaint method to avoid unnecessary painting.

Type	Standard
Parameters	None
Return value	NIL

1.6.5.34.2.8 TTreeViewMod:ExpandAll

Expand all branches of the tree.

Type	Standard
Parameters	None
Return value	NIL

1.6.5.34.2.9 TTreeViewMod:GetAllItems

Retrieves all items in the tree starting from a given item.

Type	Standard
Parameters	[<oFirst>] TTreeViewItemMod object. By default tree root
Return value	<altems>

1.6.5.34.2.10 TTreeViewMod:GetCount

Returns the total number of items in the tree.

Type	Standard
Parameters	None
Return value	<nCount>

1.6.5.34.2.11 TTreeViewMod:GetItemAtPos

Retrieves the position of a given Item in its containing Item. Corresponds to the number (index) in oltem:oParent:altems.

Type	Standard
Parameters	<nPos> Corresponds to the nIndex property of the element TTreeViewItemMod
Return value	TTreeViewItemMod object or <NIL>

1.6.5.34.2.12 TTreeViewMod:GetItemByID

Returns the item with a given ID. The ID corresponds to the **nID** property of the TTreeViewItemMod. All items have a unique ID.

Type	Standard
Parameters	<nID> Item ID. Corresponds to the nID property of the item TTreeViewItemMod
Return value	TTreeViewItem object or <NIL>

1.6.5.34.2.13 TTreeViewMod:GetItemByName

Returns the item with a given text. Corresponds to the **cText** property of the TTreeViewItemMod. All items have a unique ID.

Type	Standard
Parameters	<cItem> Text to search <INoCase> If True, performs the search case insensitive. By default is False. [<oFirst>] TTreeViewItem object. By default, tree root
Return value	TTreeViewItem object or <NIL>

1.6.5.34.2.14 TTreeViewMod:GetItemPos

Return the item position in its working array on the actual state view. So to say, collapsed branches are not included.

Type	Standard
Parameters	[<oItem>] TTreeViewItem object
Return value	<nPos>

1.6.5.34.2.15 TTreeViewMod:GetSelectedItem

Returns the current selected item.

Type	Standard
Parameters	None
Return value	TTreeViewItem object or <NIL>

1.6.5.34.2.16 TTreeViewMod:InsertItem

Inserts a new TTreeViewItemMod object to the TTreeViewMod object.

Type	Standard
Parameters	<cltem> Text to display the element [<nImage>] Number (index) of the image to display in the olmageList object, 0 by default [<nSellImage>] Number (index) for the image to display in the olmageList object when the item is selected, 0 by default [<IChecked>] True if the item is displayed with check box checked [<nClrText>] Text color [<nPos>] Position of the new item. By default at the end
Return value	TTreeViewItemMod object

1.6.5.34.2.17 TTreeViewMod:IsEmpty

Returns True if the TreeView has no items.

Type	Standard
Parameters	None
Return value	<IValue>

1.6.5.34.2.18 TTreeViewMod:SelectItem

Select a specific item.

Type	Standard
Parameters	<oltem>

	TTreviewItem object [<IFirstVisible>] If True makes that item the first visible item. By default True.
Return value	NIL

1.6.5.34.2.19 TTreeViewMod:SelectRoot

Select the root of the TreeView..

Type	Standard
Parameters	None
Return value	NIL

1.6.5.34.3 TTreeViewMod:Events

Name
OnBeginLabelEdit
OnClick
OnDbClick
OnDrawItem
OnEndLabelEdit
OnGetTooltipText
OnItemExpanded
OnItemExpanding

1.6.5.34.3.1 TTreeViewMo:OnBeginLabelEdit

Evento que se produce cuando se inicia la edición de un item.

Parameters:	<oSender> Reference to the object that fires the event <oltem> Item being edited
Return value:	NIL

1.6.5.34.3.2 TTreeViewMod:OnClick

Event that occurs when an item is clicked.

Parameters:	<oSender> Reference to the object that fires the event
--------------------	---

	<nKeyFlags> Keyboard and/or mouse status. This can be a combination of any of the following values: MK_CONTROL CTRL key pressed MK_LBUTTON Left mouse button pressed MK_MBUTTON Middle mouse button pressed MK_RBUTTON Right mouse button pressed MK_SHIFT SHIFT key pressed Use the IAnd() function to get the status <oltem> TTreeViewItemMod object that is been pressed <nRow> Row pressed
Return value:	NIL

1.6.5.34.3.3 TTreeViewMod:OnDbClick

Evento que se produce cuando se hace doble click en un item.

Parameters:	<oSender> Reference to the object that fires the event <nKeyFlags> Keyboard and/or mouse status. This can be a combination of any of the following values: MK_CONTROL CTRL key pressed MK_LBUTTON Left mouse button pressed MK_MBUTTON Middle mouse button pressed MK_RBUTTON Right mouse button pressed MK_SHIFT SHIFT key pressed Use the IAnd() function to get the status <oltem> TTreeViewItemMod object that is been pressed <nRow> Row pressed
Return value:	NIL

1.6.5.34.3.4 TTreeViewMod:OnDrawItem

Event that occurs when each treeview item is painted

Parameters:	<oSender>: Object que lanza el evento (Self). <nItem>: Element ordinal <@cText>: Value with the information. This parameter is received by reference
--------------------	--

<@nImage>:

The number (index) of the image to display according to the `olmageList` object of the control. This parameter is received by reference. By default its value is zero which means no image

<@nClrText>:

Color of the text. This parameter is received by reference

<@nClrPane>:

The background color. This parameter is received by reference

<hDC>:

context device handler

<aRect>:

Rectangle marking the possible painting area

Return value: NIL or Logical. If logical False, the painting is canceled and it is the responsibility of the event to do the painting.

1.6.5.34.3.5 TTreeViewMod:OnEndLabelEdit

Event that occurs when the editing of an item is finished.

Parameters:**<oSender>**

Reference to the object that fires the event

<oltem>

tem being edited

<@cText>

Edited text, which is passed by reference

<@ISave>

If True, the new value is saved in the item

Return value: NIL

1.6.5.34.3.6 TTreeViewMod:OnGetTooltipText

Event that occurs when an attempt is made to display the tool-tip of the control.

Parameters:**<oSender>**

Reference to the object that fires the event

<oltem>

Item being edited

<@cText>

Text to be displayed, which is passed by reference

Return value: NIL

1.6.5.34.3.7 TTreeViewMod:OnItemExpanded

An event that occurs when an item expands or collapses.

Parameters:	<oSender> Reference to the object that fires the event <oltem> Item that is expanding or collapsing <IState> True if a branch is opening
Return value:	NIL

1.6.5.34.3.8 TTreeViewMod:OnItemExpanding

An event that occurs when an item is to be expanded or collapsed.

Parameters:	<oSender> Reference to the object that fires the event <oltem> Item that is expanding or collapsing <IState> True if a branch is opening
Return value:	<NIL> o <IValue> If False the process is aborted

1.6.5.34.4 TTreeViewItemMod

This class represents each of the items of the TTreeViewMod control.

Hierarchy	TComponent descendant
File	\source\TreeViewItemMod.prg

1.6.5.34.4.1 TTreeViewItemMod:Properties

■ Read only ■ Assignable ■ Design assignable ■ Run-time assignable

Scope	Name	Type	Initial value
■	altems	Array	{}
■	cText	Character	""
■	cTooltip	Character	""
■	lChecked	Logical	.F.
■	lExpanded	Logical	.F.
■	lSelected	Logical	.F.
■	nClrText	Numeric	clWindowText

■	nID	Numeric	0
■	nImage	Numeric	0
■	nIndex	Numeric	0
■	nSellImage	Numeric	0
■	oParent	Object	TTreewViewltemMod owner
■	oTreeView	Object	TTreeviewMod owner

Array of TTreeViewltemMod objects with the various items dependent on this item. Each item in this array is a TTreeVeiwltemMod object that can also contain more items.

Scope:	Run-time assignable
Type:	Array
Initial value:	{}

Item text.

Scope:	Assignable
Type:	Character
Initial value:	""

Tool-tip that will display the item.

Scope:	Assignable
Type:	Character
Initial value:	""

Changes the checked state when the TreeView has the ICheckBoxes property checked.

Scope:	Assignable
Type:	Logical
Initial value:	.F.

Changes the expanded status.

Scope:	Assignable
Type:	Logical
Initial value:	.F.

Returns True if the item is selected.

Type	Standard
Parameters	None
Return value	<ISelected> True if selected

Text color.

Scope:	Assignable
Type:	Numeric
Initial value:	clWindowText

(Consult appendix for possible colors)

Unique identifier of the element in the TTreeViewMod tree.

Scope:	Read only
Type:	Numeric
Initial value:	0

Number (index) of image to display from the olmagedList property of its container object..

Scope:	Assignable
Type:	Numeric
Initial value:	0

Position in the property altems of its co-owner

Scope:	Read only
Type:	Numeric
Initial value:	0

Number of image to display from the olmageList property of its container object when the item is selected.

Scope:	Assignable
Type:	Numeric
Initial value:	0

Reference to its owner. Can be a TTreeViewItemMod object or the TTreeViewMod if its in the first branch of the first level.

Scope:	Read only
Type:	Numeric
Initial value:	NIL

TTreeViewMod owner.

Scope:	Read only
Type:	Numeric
Initial value:	NIL

1.6.5.34.4.2 TTreeViewItemMod:Methods

■ Constructor ■ Standard

Type	Name
■	AddItem
■	Collapse
■	Delete
■	DeleteChilds / Deleteltems
■	EnsureVisible
■	Expand

■	FirstChild
■	GetNextSibling
■	GetPrevSibling
■	HasChilds
■	InsertItem
■	LastChild
■	MoveAfter
■	MoveDown
■	MoveUp
■	Select
■	SortChildren

Adds a new TTreeViewItemMod object to the actual item.

Type	Standard
Parameters	<p><cItem> Text to display the element</p> <p>[<nImage>] Number (index) of the image to display in the olmageList object, 0 by default</p> <p>[<nSellImage>] Number (index) for the image to display in the olmageList object when the item is selected, 0 by default</p> <p>[<IChecked>] True if the item is displayed with check box checked</p> <p>[<nClrText>] Text color</p>
Return value	TTreeViewItemMod

Collapses the lower branches of the item.

Type	Standard
Parameters	<p><IPartial> If true only the firt branch level is collapsed. By default, the value of ISingleExpand property of its TTreeViewMod owner</p>
Return value	NIL

Deletes the item with all its branches.

Type	Standard
Parameters	None

Return value	NIL
---------------------	-----

Deletes all branches of the item.

Type	Standard
Parameters	None
Return value	NIL

Forces the item to be visible..

Type	Standard
Parameters	<IFirstVisible> If true, it will be the first item visible of the TTreeViewMod
Return value	NIL

Expands the lower branches of the item.

Type	Standard
Parameters	<IPartial> If true only the first branch level is expanded. By default, the value of ISingleExpand property of its TTreeViewMod owner
Return value	NIL

Returns the first child element of the Item.

Type	Standard
Parameters	None
Return value	NIL TTreeViewItemMod

Returns the next element in its container according to its order (nIndex).

Type	Standard
Parameters	None
Return value	NIL TTreeViewItemMod

Returns the previous element in its container according to its order (nIndex).

Type	Standard
Parameters	None
Return value	NIL TTreeViewItemMod

Returns True if the current item has branches with more items.

Type	Standard
Parameters	None
Return value	<IHas>

Insertes a new TTreeViewItemMod object to the present item.

Type	Standard
Parameters	<cltem> Text to display the element [<nImage>] Number (index) of the image to display in the olmageList object, 0 by default [<nSellImage>] Number (index) for the image to display in the olmageList object when the item is selected, 0 by default [<IChecked>] True if the item is displayed with check box checked [<nClrText>] Text color [<nPos>] Position of the new item. By default at the end
Return value	TTreeViewItemMod

Returns the last child element of the Item.

Type	Standard
Parameters	None
Return value	NIL TTreeViewItemMod

Moves the item after a specific item.

Type	Standard
Parameters	<oltem> Item where to move
Return value	<ISuccess> True if success

Move the item one position down.

Type	Standard
Parameters	None
Return value	<ISuccess> True if success

Move the item one position up.

Type	Standard
Parameters	None
Return value	<ISuccess> True if success

Selects an item on the TTreeViewMod.

Type	Standard
Parameters	[<IFirstVisible>] If True makes the item the first visible item. By default true
Return value	NIL

Allows to sort all child elements of the item

Type	Standard
Parameters	[<bSort>] Code-block for sorting, with the same operation as the Asort() function of Harbour [<IRecursive>] If True the sorting will be done on all lower branches. By default

	false
Return value	NIL

1.6.6 System

1.6.6.1 TImageList

Class to manipulate the 'Image List' Windows object type.

Description:

The TImageList class allows to manage the typical "Image List" provided by the Windows API.

You can add bitmap images or icons from the resource or an external files. The image list are used in several controls, those that are capable to show images, like TBtnBmp or TTreeView.

Hierarchy	Inherits from TWinObject
File name	\source\ImageList.prg

1.6.6.1.1 TImageList:Properties

read Only
 Assignable
 Design assignable
 Run-time assignable

Scope	Name	Type	Initial value
<input type="checkbox"/>	nHeight	Numeric	1
<input type="checkbox"/>	nWidth	Numeric	1
<input type="checkbox"/>	oParent	Object	NIL

1.6.6.1.1.1 TImageList:nHeight

Image height indicated in pixels.

Scope	Assignable only before to add an image in the list
Type	Numeric
Initial value	1

The image height in the TImageList don't need to coincide with the height of every image that will be integrated using the Add method. Is very normal to add a image where its dimensions are bigger than the indicated in the **nHeight** and nWidth properties. In this way the TImageList class will divide the added image in several sub-images as needed, with the height and width specified.

1.6.6.1.1.2 TImageList:nWidth

Images width specified in pixels.

Scope	Assignable only before to add an image in the list
Type	Numeric
Initial value	1

The image width in the TImageList don't need to coincide with the width of every image that will be integrated using the Add method. Is very normal to add a image where its dimensions are bigger than the indicated in the nHeight and nWidth properties. In this way the TImageList class will divide the added image in several sub-images as needed, with the height and width specified.

1.6.6.1.1.3 TImageList:oParent

TImageList proprietary form or control.

Scope	Assignable only before to add an image in the list
Type	Object
Initial value	NIL

Normally this data is passed as parameter in the Create constructor.

1.6.6.1.2 TImageList:Methods

■ Constructor ■ Standard

Typ	Name
■	Add
■	AddIcon
■	Create
■	Destroy
■	GetBitmap
■	GetIcon
■	GetIconSize
■	GetImageCount
■	New
■	Remove
■	RemoveAll
■	Replacelcon
■	SetIconSize

1.6.6.1.2.1 TImageList:Add

Adds an bitmap image type to the Image List.

Type	Standard
Parameters	<xImage> Image name. It can be a resource name, a file name or an existing bitmap handle [<IMasked>] If it is .T. the image will be converted to a mask format: its colors will be modified to different gray tones.
Return value	<nImage> Index from the recently added image in the Image List

If it is the first image that is added to the TImageList and the nWidth and nHeight properties have a value=1, then the nWidth and nHeight dimensions are adjusted from the dimensions of this first image.

1.6.6.1.2.2 TImageList:AddIcon

Adds an Icon type image to the Image List. Accepts the following file extensions: ICO, DLL, EXE.

Type	Standard
Parameters	<clcon> Icon name. It can be a resource name or a file name. [<nIcon>] Icon number when loaded from a file. By default first icon in file
Return value	<nImage> Index from the recently added image in the Image List

1.6.6.1.2.3 TImageList:Create

Class constructor.

Type	Constructor
Parameters	<oParent> Container object. It can be a form or a control [<nWidth>] Default image width [<nHeigh>] Default image height
Return	<Self>

value	Self reference
--------------	----------------

1.6.6.1.2.4 TImageList:Destroy

Destroys the Image List and releases all the memory used by it.

Type	Standard
Parameters	None
Return value	Nil

1.6.6.1.2.5 TImageList:GetBitmap

Gets a bitmap from the Image List.

Type	Standard
Parameters	<nBitmap> Bitmap index in the TImageList
Return value	<hIcon> Bitmap handle

The returned value is simply an internal Windows handle. To convert it in a TBitmap object you should create an object passing as parameter this handle.

Is the programmers responsibility to destroy the returned handle. When the TImageList is destroyed, it does not destroy this handle neither.

1.6.6.1.2.6 TImageList:GetIcon

Gets an icon from the Image List.

Type	Standard
Parameters	<nIcon> Icon index in the Image List
Return value	<hIcon> Icon Handle

The returned value is simply an internal Windows handle. To convert it in a TIcon object you should create an object passing as parameter this handle.

Is the programmers responsibility to destroy the returned handle. When the TImageList is destroyed, it does not destroy this handle neither.

1.6.6.1.2.7 TImageList:GetIconSize

Gets the dimensions of every image in the TImageList.

Type	Standard
Parameters	None
Return value	<{ nWidth, nHeight }> Array with the dimensions of every image

1.6.6.1.2.8 TImageList:GetImageCount

Gets the number of images in the TImageList

Type	Standard
Parameters	None
Return value	<nTotal> Total number of images

1.6.6.1.2.9 TImageList:Remove

Removes an image from the Image List.

Type	Standard
Parameters	<nImage> Image index in the TImageList
Return value	<ISuccess> .T. if the operation is successful

1.6.6.1.2.10 TImageList:RemoveAll

Removes all the images from the Image List.

Type	Standard
Parameters	None
Return value	<ISuccess> .T. if the operation is successful

1.6.6.1.2.11 TImageList:Replacelcon

Replaces an icon or image in the Image List.

Type	Standard
Parameters	<nIndex> Icon or image index in the TImageList <hIcon> New icon or image Handle
Return value	<ISuccess> .T. if the operation is successful

1.6.6.1.2.12 TImageList:SetIconSize

Establishes the dimensions of every image in the TImageList.

Type	Standard
Parameters	<nWidth> New width for every image <nHeight> New height for every image
Return value	<ISuccess> .T. if the operation is successful

Note: This method destroys any image that could be in the TImageList.

1.6.6.2 TNotifyIcon

This class manages the icons in the notification area (SystemTray).

Description:

The TNotifyIcon allows to manage the icons in the notification area (SystemTray).

Hierarchy Inherits from TWinObject
File name \source\NotifyIcon.prg

1.6.6.2.1 TNotifyIcon:Properties

■ read Only ■ Assignable ■ Design assignable ■ Run-time assignable

Scope	Name	Type	Initial value
■	cToolTip	Character	""
■	IVisible	Logic	.F.
■	nImage	Numeric	0

■	oImageList	Object	
■	oMenu	Object	
■	oParent	Object	

1.6.6.2.1.1 TNotifyIcon:cToolTip

Icon description or any text indicating the application description.

Scope	Assignable
Type	Character
Initial value	""

1.6.6.2.1.2 TNotifyIcon:IVisible

If it is `.F.` it will not show the notification icon in the task bar. When this property is `.T.` it will show the icon in the SystemTray having the same effect than using the `Show` method.

Scope	Assignable
Type	Logic
Initial value	<code>.F.</code>

1.6.6.2.1.3 TNotifyIcon:nImage

Index icon to be shown.

Scope	Assignable
Type	Numeric
Initial value	0

1.6.6.2.1.4 TNotifyIcon:oImageList

TImageList object with the icon list that the control can show.

Scope	Assignable
Type	Object
Initial value	

1.6.6.2.1.5 TNotifyIcon:oMenu

TMenu object associated to the icon.

Scope	Assignable
Type	Object
Initial value	

1.6.6.2.1.6 TNotifyIcon:oParent

Form where the control belongs to.

Scope	read Only
Type	Object
Initial value	

1.6.6.2.2 TNotifyIcon:Methods

■ Constructor ■ Standard

Typ	Name
■	AddIcon
■	Create
■	Destroy
■	Hide
■	HideBalloon
■	New
■	Refresh
■	SetIcon
■	SetMenu
■	SetToolTip
■	Show
■	ShowBalloon

1.6.6.2.2.1 TNotifyIcon:AddIcon

Adds an icon to the image list.

Type	Standard
Parameters	<clcon> File name or icon resource
Return	<nPos>

value	Index icon in the image list
--------------	------------------------------

1.6.6.2.2.2 TNotifyIcon:Create

Creates the object with the properties and events defined.

Type	Constructor
Parameters	<oParent> Form where the control belongs to
Return value	Self reference (Self)

1.6.6.2.2.3 TNotifyIcon:Destroy

Deleted the icon from the notification area. Destroys the TImageList if it belongs to the control and releases system resources.

Type	Standard
Parameters	None
Return value	NIL

1.6.6.2.2.4 TNotifyIcon:Hide

Hides the icon in the notification area.

Type	Standard
Parameters	None
Return value	NIL

1.6.6.2.2.5 TNotifyIcon:HideBalloon

Hides the balloon window on the notification area showed with the method ShowBalloon.

Type	Standard
Parameters	None
Return value	NIL
See also	ShowBalloon

1.6.6.2.2.6 TNotifyIcon:New

Creates the object and prepares it to configure it..

Type	Constructor
Parameters	<oParent> Form where the control belongs to
Return value	Self reference(Self)

1.6.6.2.2.7 TNotifyIcon:Refresh

Refreshes the icon when its configuration changes.

Type	Standard
Parameters	None
Return value	NIL

1.6.6.2.2.8 TNotifyIcon:SetIcon

Establishes and refreshes the icon that is shown in the notification area. It is equivalent to assign the nImageIndex property.

Type	Standard
Parameters	<nIndex> Icon index in the image list
Return value	NIL

1.6.6.2.2.9 TNotifyIcon:SetMenu

Establishes the menu that displays the icon. It is equivalent to assign the oMenu property.

Type	Standard
Parameters	<oMenu> Menu assigned to the control
Return value	NIL

1.6.6.2.2.10 TNotifyIcon:SetToolTip

Establishes and refreshes the Tooltip that shows the icon in the notification area. It is equivalent to assign the cToolTip property.

Type	Standard
Parameters	<cToolTip> Icon description or any text that describes the application
Return value	NIL

1.6.6.2.2.11 TNotifyIcon:Show

Shows the icon in the notification area.

Type	Standard
Parameters	None
Return value	NIL

1.6.6.2.2.12 TNotifyIcon:ShowBalloon

Shows a balloon window on the notification area.

Type	Standard
Parameters	<cMessage> Message to show <cTitle> Window title <nFlags> Type of icon to show on the balloon window: <ul style="list-style-type: none">• NIIF_NONE (0)• NIIF_INFO (1)• NIIF_WARNING (2)• NIIF_ERROR (3) Defined in Shellapi.api <nTimeOut> Display timeout in <u>milliseconds</u> . Only on XP/W2k, ignored in Vista and above. Minimum 10 seconds, maximum 30 seconds.
Return value	NIL
See also	HideBallon

1.6.6.2.3 TNotifyIcon:Events

Name
OnBalloonClick
OnBalloonHide
OnBalloonShow
OnBalloonTimeout
OnDbtClick
OnLButtonDown
OnLButtonUp
OnMouseMove
OnRButtonUp
OnRButtonDbtClick
OnRButtonDown

1.6.6.2.3.1 TNotifyIcon:OnBalloonClick

Event that is triggered when the user clicks on the balloon window closing it.

Parameters	<oSender> Reference to the object that triggers the event
Return value	NIL

1.6.6.2.3.2 TNotifyIcon:OnBalloonHide

Event that is triggered when the balloon window hides.

Parameters	<oSender> Reference to the object that triggers the event
Return value	NIL

1.6.6.2.3.3 TNotifyIcon:OnBalloonShow

Event that is triggered when the balloon window shows.

Parameters	<oSender> Reference to the object that triggers the event
-------------------	---

Return value	NIL
---------------------	-----

1.6.6.2.3.4 TNotifyIcon:OnBalloonTimeout

Event that is triggered when the balloon window closes due a timeout.

Parameters	<oSender> Reference to the object that triggers the event
Return value	NIL

1.6.6.2.3.5 TNotifyIcon:OnDblClick

Event that is triggered when the user double clicks the mouse left button. If there is a menu assigned to the oMenu property and the menu has defined an action, this actions is executed. Otherwise, this event is triggered.

Parameters	<oSender> Reference to the object that triggers the event < nX, nY > Mouse coordinates
Return value	NIL

1.6.6.2.3.6 TNotifyIcon:OnLButtonDown

Event that is triggered when th user click the left mouse button.

Parameters	<oSender> Reference to the object that triggers the event < nX, nY > Mouse coordinates
Return value	NIL

1.6.6.2.3.7 TNotifyIcon:OnLButtonUp

Event that is triggered when the user releases the mouse left button.

Parameters	<oSender>
-------------------	------------------------

	Reference to the object that triggers the event < nX, nY> Mouse coordinates
Return value	NIL

1.6.6.2.3.8 TNotifyIcon:OnMouseMove

Event that is produced when the mouse pointer is moved over the control.

Parameters	<oSender> Reference to the object that triggers the event < nX, nY> Mouse coordinates
Return value	NIL

1.6.6.2.3.9 TNotifyIcon:OnRButtonUp

Event that is triggered when the user releases the mouse right button. If the event is not assigned or if returns NIL, it will show the menu assigned to the oMenu property.

Parameters	<oSender> Reference to the object that triggers the event < nX, nY> Mouse coordinates
Return value	NIL

1.6.6.2.3.10 TNotifyIcon:OnRButtonDbClick

Event that is triggered when the user double-clicks the mouse right button.

Parameters	<oSender> Reference to the object that triggers the event < nX, nY> Mouse coordinates
Return value	NIL

1.6.6.2.3.11 TNotifyIcon:OnRButtonDown

Event that is triggered when the user clicks the mouse right button.

Parameters	<oSender> Reference to the object that triggers the event <nX, nY> Mouse coordinates
Return value	NIL

1.6.6.3 TProcess

Class to execute external processes or applications.

Note: The use of the class TRunProcess is recommended instead of this one (Only on Enterprise version)

Hierarchy	Inherits from TComponent
File name	\source\Process.prg

1.6.6.3.1 TProcessProperties

read Only
 Assignable
 Design assignable
 Run-time assignable

Scope	Name	Type	Processtial value
<input type="checkbox"/>	cName	Character	""
<input type="checkbox"/>	IWait	Logical	.F.

1.6.6.3.1.1 TProcess:cName

Name and path of the process or application to launch.

Scope	Standard (before calling Run method)
Type	Character
Initial value	""

1.6.6.3.1.2 TProcess:IWait

Indicates if the primary process should wait to the secondary process to finish.

Scope	Standard (before calling Run method)
Type	Logical

Initial value .F.

1.6.6.3.2 TProcessMethods

■ Constructor ■ Standard ■ Only after Create()

Typ Name

e

- IsRunning
- Run
- Stop

1.6.6.3.2.1 TProcess:IsRunning

Return true if the process is still executing.

Type	Standard
Parameters	None
Return value	Logical

1.6.6.3.2.2 TProcess:Run

Runs the process.

Type	Standard
Parameters	[<nShowMode>]: Display mode. By default smNORMAL. Possible values: smNORMAL: normal form smMAXIMIZE: maximized form ftMINIMIZE: minimized form
Return value	<ISuccess> True if success

1.6.6.3.2.3 TProcess:Stop

Stops the running process.

Type	Standard
Parameters	None

Return value	<ISuccess> True if success
---------------------	-------------------------------

1.6.6.4 TRegistry

This class encapsulated the system registry management.

Hierarchy	Inherits from TComponent
File Name	\source\Registry.prg

1.6.6.4.1 TRegistry:Properties

read Only
 Assignable
 Design assignable
 Run-time assignable

Scope	Name	Type	Initial value
<input type="checkbox"/>	hCurrentKey	Numeric	0
<input type="checkbox"/>	nLastError	Numeric	0
<input type="checkbox"/>	Root	Numeric	HKEY_CURRENT_USER

1.6.6.4.1.1 TRegistry:Root

Indicates the registry key where it will work.

Scope	Assignable
Type	Numeric
Initial value	HKEY_CURRENT_USER
Possible values	HKEY_CURRENT_USER, HKEY_LOCALMACHINE, HKEY_CLASSES_ROOT, HKEY_USERS, HKEY_PERFORMANCE_DATA, HKEY_CURRENT_CONFIG and HKEY_DYN_DATA

You can find the possible values definitions in the **WinReg.api** file in the include directory. For more information about the windows registry management, see the Windows API documentation in <http://msdn.microsoft.com>.

1.6.6.4.1.2 TRegistry:hCurrentKey

Indicates the current open registry key.

Scope	read Only
Type	Numeric

Initial value 0

1.6.6.4.1.3 TRegistry:nLastError

Error code (if any) after the call to any method.

Scope	read Only
Type	Numeric
Initial value	0

1.6.6.4.2 TRegistry:Methods

■ Constructor ■ Standard

Typ	Name
■	CloseKey
■	Connect
■	CreateKey
■	DeleteKey
■	DeleteValue
■	Destroy
■	Disconnect
■	EnumKeys
■	EnumValues
■	Flush
■	GetBinary
■	GetDate
■	GetLogical
■	GetNumeric
■	GetString
■	OpenKey
■	SetDate
■	SetLogical
■	SetNumeric
■	SetString
■	SetValue

1.6.6.4.2.1 TRegistry:CloseKey

Closes the current key and updates nLastError .

Type	Standard
Parameters	None
Return value	<nError> Error code

1.6.6.4.2.2 TRegistry:Connect

Connects with a remote computer's registry.

Type	Standard
Parameters	<cComputer> Remote computer's name. <Key> Key to be connected to. It can be HKEY_LOCAL_MACHINE or HKEY_USERS
Return value	<nError> Error code

1.6.6.4.2.3 TRegistry:CreateKey

Creates a new key with the options and permissions specified and updates nLastError.

Type	Standard
Parameters	<cKey> Key to be created. <cClass> Key class. Default: "". <nOptions> Create key options. Default: 0. <nAccess> Key access permissions. Default: KEY_ALL_ACCESS
Return value	<nError> Error code

1.6.6.4.2.4 TRegistry>DeleteKey

Deletes a key and updates nLastError. In Windows 9X deletes also the sub keys and in Windows NT/XP they must be deleted recursively.

Type	Standard
Parameters	<cKey> Key to be deleted.
Return value	<nError> Error code

1.6.6.4.2.5 TRegistry:DeleteValue

Deletes a value and updates nLastError.

Type	Standard
Parameters	<cValue> Value to be deleted
Return value	<nError> Error code

1.6.6.4.2.6 TRegistry:Destroy

Close the current key and release the system resources.

Type	Standard
Parameters	None
Return value	NIL

1.6.6.4.2.7 TRegistry:Disconnect

Disconnects from a remote computer's registry.

Type	Standard
Parameters	None
Return value	<nError> Error code

1.6.6.4.2.8 TRegistry:EnumKeys

Lists the keys that belongs to an previously open key and updates nLastError.

Type	Standard
Parameters	None

Return value	<aKeys> Key list
---------------------	----------------------------------

1.6.6.4.2.9 TRegistry:EnumValues

Lists the values that belong to an previously open key and updates nLastError.

Type	Standard
Parameters	None
Return value	<aValues> Value list

1.6.6.4.2.10 TRegistry:Flush

Commits the registry content to disk. It is used when connecting to remote registry and want to force its update. On error it will return the error number and the property nLastError will be updated.

Type	Standar
Paremeters	None
Return value	<nResult> Return value. See also RegFlushKey() API documentation

1.6.6.4.2.11 TRegistry:GetBinary

Returns the value from and binary entry in character format, using the default value if there is not an entry and updating nLastError.

Type	Standard
Parameters	<cKey> Key name <cDefault> Default value
Return value	<nError> Error code

1.6.6.4.2.12 TRegistry:GetDate

Returns the entry value in date format, using the default value if there is not an entry and updating nLastError.

Type	Standard
Parameters	<cKey> Key name <dDefault> Default value
Return value	<nError> Error code

1.6.6.4.2.13 TRegistry:GetLogical

Returns the entry value in logic format, using the default value if there is not an entry and updating nLastError

Type	Standard
Parameters	<cKey> Key name <lDefault> Default value
Return value	<nError> Error code

1.6.6.4.2.14 TRegistry:GetNumeric

Returns the entry value in numeric format, using the default value if there is not an entry and updating nLastError.

Type	Standard
Parameters	<cKey> Key name <nDefault> Default value
Return value	<nError> Error code

1.6.6.4.2.15 TRegistry:GetString

Returns the entry value in character format, using the default value if there is not an entry and updating nLastError.

Type	Standard
Parameters	<cKey> Key name <cDefault> Default value
Return value	<nError> Error code

1.6.6.4.2.16 TRegistry:OpenKey

Opens the indicated key with the options and permissions specified and updates nLastError. After you work with the key you must close it always with CloseKey before to open another key to work.

Type	Standard
Parameters	<cKey> Key to be open <nAccess> Access key permissions. Default: KEY_ALL_ACCESS
Return value	<nError> Error code

1.6.6.4.2.17 TRegistry:SetDate

Establishes the entry value in date format and updates nLastError.

Type	Standard
Parameters	<cKey> Key name <dDate> Default value
Return value	<nError> Error code

1.6.6.4.2.18 TRegistry:SetLogical

Establishes the entry value in logic format and updates nLastError.

Type	Standard
Parameters	<cKey> Key name <lValue> Default value

Return value	<nError> Error code
---------------------	-------------------------------------

1.6.6.4.2.19 TRegistry:SetNumeric

Establishes the entry value in numeric format and updates nLastError.

Type	Standard
Parameters	<cKey> Key name <nValue> Default value
Return value	<nError> Error code

1.6.6.4.2.20 TRegistry:SetString

Establishes the entry value in character format and updates nLastError.

Type	Standard
Parameters	<cKey> Key name <cValue> Default value
Return value	<nError> Error code

1.6.6.4.2.21 TRegistry:SetValue

Establishes the entry value in character format and updates nLastError.

Type	Standard
Parameters	<cKey> Key name <uValue> key value
Return value	<nError> Error code

1.6.6.5 TRunProcess

Class for asynchronous execution (in a second thread) of any application or command that can capture several events to completely control the entire process. In the case of console application is also possible to use STDIN through the Write() method or capture the event STDOUT with OnRead().

Note:

This class is descendant from TMTObject class which is not documented since is a internal class. The next class in his hierarchy is TWinObject.

Hierarchy	Descendent of TMTObject
File name	\\source\\Enterprise\\RunProcess.prg

1.6.6.5.1 TRunProcess:Properties

Read only
 Assignable
 Design assignable
 Run-time assignable

Scope	Name	Type	Initial value
<input type="checkbox"/>	cCommandLine	Character	""
<input type="checkbox"/>	cErrorMsg	Character	""
<input type="checkbox"/>	cWorkingDir	Character	""
<input type="checkbox"/>	hProcess	Numeric	0
<input type="checkbox"/>	hThread	Numeric	0
<input type="checkbox"/>	nShowMode	Numeric	SW_SHOW

1.6.6.5.1.1 TRunProcess:cCommandLine

Command or application to execute.

Scope	Design assignable
Type	Character
Initial value	""

1.6.6.5.1.2 TRunProcess:cErrorMsg

Execution error description.

Scope	Read only
Type	Character
Initial value	""

1.6.6.5.1.3 TRunProcess:cWorkingDir

Application directory.

Scope	Design assignable
Type	Character
Initial value	""

1.6.6.5.1.4 TRunProcess:hProcess

Internal process identifier.

Scope	Read only
Type	Numeric
Initial value	0

1.6.6.5.1.5 TRunProcess:hThread

Internal thread identifier.

Scope	Read only
Type	Numeric
Initial value	0

1.6.6.5.1.6 TRunProcess:nShowMode

Application show mode.

Scope	Design assignable	
Type	Numeric	
Initial value	SW_SHOW	
Possible values	SW_HIDE	0
	SW_SHOWNORMAL	1
	SW_NORMAL	1
	SW_SHOWMINIMIZED	2
	SW_SHOWMAXIMIZED	3
	SW_MAXIMIZE	3
	SW_SHOWNOACTIVATE	4
	SW_SHOW	5
	SW_MINIMIZE	6
	SW_SHOWMINNOACTIVE	7
	SW_SHOWNA	8

SW_RESTORE	9
SW_SHOWDEFAULT	10
SW_FORCEMINIMIZE	11
SW_MAX	11

1.6.6.5.2 TRunProcess:Methods

■ Constructor ■ Standard

Type	Name
■	End
■	IsRunning
■	Kill
■	Run
■	Write

1.6.6.5.2.1 TRunProcess:Run

Runs the process.

Type	Standard
Parameters	[<cCommandLine>] Command line. By default cCommandLine property [<cWorkingDir>] Working directory. By default cWorkingDir property. [<nShowMode>] Show mode. By default nShowMode property.
Return value	<ISuccess> True if success

1.6.6.5.2.2 TRunProcess:End

Forces application end and destroys object.

Type	Standard
Parameters	None
Return value	NIL

1.6.6.5.2.3 TRunProcess:IsRunning

Returns true if the process is running.

Type	Standard
Parameters	None
Return value	Logical

1.6.6.5.2.4 TRunProcess:Kill

Forces program termination.

Type	Standard
Parameters	None
Return value	<nExitCode> Exit code

1.6.6.5.2.5 TRunProcess:Write

Sends a string to the application through STDIN.

Type	Standard
Parameters	[<cData>] String to send
Return value	NIL

1.6.6.5.3 TRunProcess:Events

Name
OnEnd
OnRead
OnStart

1.6.6.5.3.1 TRunProcess:OnEnd

Event fired when the application ends.

Parameters	<oSender> Reference to the object that triggers the event
Return value	NIL

1.6.6.5.3.2 TRunProcess:OnRead

Event fired when it receives information through STDOUT from the application.

Parameters	<oSender> Reference to the object that triggers the event <cData> String received
Return value	NIL

1.6.6.5.3.3 TRunProcess:OnStart

Event fired when the application starts.

Parameters	<oSender> Reference to the object that triggers the event
Return value	NIL

1.6.6.6 TSharedModule

This class allows you to create component containers that can then be used directly on any form using the **TSharedModule** control located on the **'System'** tab.

In order to use a **TSharedModule** component you must first create a special type of form called 'SharedModule' that you can locate under the menu option: [File->New->SharedModule](#). This is a special type of form in which only 'component' (non-visual) type controls can be deposited. Its only property is cClassname that indicates the name that we will use to reference this container.

On all forms in which you want to access any of the controls in the container, you must only include a **TSharedModule** control and from then on, all controls included in it can be accessed by any control of your form.

Important note:

If you modify the cClassname property on the special type of form, all forms that use that **SharedModule** and are not instantiated in the IDE will not automatically update that property and therefore fail when you attempt to load them, causing the loss of that component in the form. To avoid this problem you can either load any form using the SharedModule form or manually modify the XFM files.

Hierarchy	TComponent descendant
File	\source\SharedModule.prg

1.6.6.6.1 TSharedModule:Properties

■ Read only ■ Assignable ■ Design assignable ■ Run-time assignable

Scope	Name	Type	Initial value
■	cClassname	Character	""

1.6.6.6.1.1 TSharedModule:cClassname

Name of the class to be used as a reference for its location and mapping in **TSharedModule** components. For more information see introduction of this class.

Scope	Assignable
Type	Character
Initial value	""

1.6.6.6.2 TSharedModule:Events

Name
OnCreate
OnInitialize

1.6.6.6.2.1 TSharedModule:OnCreate

Event that occurs when the control is created. Note that at that time its child component controls are not yet created.

Parameters	<oSender>: Reference to the object that trigger the event
Return value	NIL

1.6.6.6.2.2 TSharedModule:OnInitialize

Event triggered when initiating the control and after all its child component controls are created.

Parameters	<oSender>: Reference to the object that trigger the event
Return value	NIL

1.6.6.7 TSysImageList

The TSystemImageList (or Shell Icon Cache) lets us access the icon list used by the system Shell and use them in our applications.

This list of icons is the one used by Windows Explorer and other applications to show icons which identify system objects, applications and documents.

Description:

The TSysImageList class allows to encapsulate the standard Windows image list.

Hierarchy	Inherits from TWinObject
File name	\source\SysImageList.prg

1.6.6.7.1 TSysImageList:Properties

■ read Only ■ Assignable ■ Design assignable ■ Run-time assignable

Scope	Name	Type	Initial value
■	ILargeIcons	Logic	.T.
■	IUseFileAttributes	Logic	.T.
■	IOverlay	Logic	.F.
■	ISelected	Logic	.F.
■	IShellSize	Logic	.F.

1.6.6.7.1.1 TSysImageList:ILargeIcons

Returns the handle to the system image list that contains small or large icon images.

Scope	Assignable
Type	Logic
Initial value	.T.

1.6.6.7.1.2 TSysImageList:IUseFileAttributes

Control if the image should be retrieved from the cache or the disk. This property is handy to avoid the error message when accessing to disk units when they are not ready or not connected resources.

Scope	Assignable
Type	Logic
Initial value	.T.

1.6.6.7.1.3 TSysImageList:IOverlay

Adds the link overlay to the icon.

Scope	Assignable
Type	Logic
Initial value	.F.

1.6.6.7.1.4 TSysImageList:ISelected

Blends the icon with the system highlight color.

Scope	Assignable
Type	Logic
Initial value	.F.

1.6.6.7.1.5 TSysImageList:IShellSize

Retrieves a Shell-sized icon.

Scope	Assignable
Type	Logic
Initial value	.F.

1.6.6.7.2 TSysImageList:Methods

■ Constructor ■ Standard

Typ	Name
■	Create
■	GetBitmap
■	GetIcon
■	GetIconIndex
■	GetIconName
■	GetIconType

1.6.6.7.2.1 TSysImageList:Create

Creates the object that allows the access to the system image list.

Type	Constructor
Parameters	None
Return value	Self reference (Self)

1.6.6.7.2.2 TSysImageList:GetBitmap

Returns the related bitmap of a file type. Each handle returned from this method must be destroyed with DeleteObject..

Type	Standard
Parameters	<nIndex> Image index
Return value	<hImage> Bitmap handle

1.6.6.7.2.3 TSysImageList:GetIcon

Returns the related icon of a file type. Each handle returned from this method must be destroyed with DestroyIcon.

Type	Standard
Parameters	<cFilename> Icon filename
Return value	<hIcon> Icon handle

1.6.6.7.2.4 TSysImageList:GetIconIndex

Return the icon index related to a file type.

Type	Standard
Parameters	<cFilename> File name
Return value	<nIndex> Position

1.6.6.7.2.5 TSysImageList:GetIconName

Returns the string that contains the name of the file as it appears in the Shell or the file name of the file that contains the icon representing the file.

Type	Standard
Parameters	<cFilename> File name
Return value	<cName> File name

1.6.6.7.2.6 TSysImageList:GetIconType

Returns the string that contains the name of the file as it appears in the Shell or the file name of the file that contains the icon representing the file.

Type	Standard
Parameters	<cFilename> File name
Return value	<cType> File type

1.6.6.8 TTimer

This class represents a TIMER Windows object.

Description:

This class represents a TIMER Windows object.

It is not needed to create a window to make it work.

Hierarchy	Inherits from TWinObject
File name	\source\Timer.prg

1.6.6.8.1 TTimer:Properties

■ read Only ■ Assignable ■ Design assignable ■ Run-time assignable

Scope	Name	Type	Initial value
■	Enabled	Logic	.F.
■	nInterval	Numeric	1000
■	oParent	Object	

1.6.6.8.1.1 TTimer:Enabled

Enables or disables the timer.

Scope	Assignable
Type	Logic
Initial value	.F.

1.6.6.8.1.2 TTimer:nInterval

Time interval specified in milliseconds.

Scope	Assignable
Type	Numeric
Initial value	1000

1.6.6.8.1.3 TTimer:oParent

Form where the control belongs to.

Scope	read Only
Type	Object
Initial value	

1.6.6.8.2 TTimer:Methods

■ Constructor ■ Standard

Typ	Name
■	Create
■	Destroy
■	Disable
■	Enable
■	New
■	SetInterval
■	Stop

1.6.6.8.2.1 TTimer:Create

Creates and activates a timer.

Type	Constructor
Parameters	<oParent> Form where the control belongs to <nInterval> Time interval specified in milliseconds <bOnTimer> Action to execute in every time interval
Return value	Self reference (self)

1.6.6.8.2.2 TTimer:Destroy

Disables and destroys the timer.

Type	Standard
Parameters	None
Return value	NIL

1.6.6.8.2.3 TTimer:Disable

Disable the timer.

Type	Standard
Parameters	None
Return value	NIL

1.6.6.8.2.4 TTimer:Enable

Activates the timer.

Type	Standard
Parameters	None
Return value	NIL

1.6.6.8.2.5 TTimer:New

Creates and activates a timer.

Type	Constructor
Parameters	<oParent> Form where the control belongs to
Return value	Self reference (self)

1.6.6.8.2.6 TTimer:SetInterval

Sets the time interval. It is equivalent to assign the nInterval property.

Type	Standard
Parameters	<nInterval> Time specified in milliseconds
Return value	Self reference (self)

1.6.6.8.2.7 TTimer:Stop

Stops the timer.

Type	Standard
Parameters	None
Return value	NIL

1.6.6.8.3 TTimer:Events

Name	
OnTimer	

1.6.6.8.3.1 TTimer:OnTimer

Event that is triggered every time interval.

Parameters	<oSender> Reference to the object that triggers the event
Return	NIL

value

1.6.6.9 TWinService

This control allows you to create and manipulate Windows services. The **TWInService** control located on the '**System**' tab allows you to manipulate any system service. To do this, simply use two properties: `cName` and `cDisplayName`, the latter being only necessary if you are creating a new service.

If you want to create an application with Xailer that can be used as a Windows service you must also use the **TWInService** control, but the most important thing is to capture the `OnRun` event that occurs every time there is a state change in the service. In this event you must include the code that you want to run the service on a recurring basis. It is very important that you make calls to **ProcessMessages** leaving a short pause to allow the system to not lock.

```
oWinService:OnRun := { |oSender| ServiceMain( oSender ) }
...
...
STATIC PROCEDURE ServiceMain( oSender )

    LogFile( "Service started." )

    DO WHILE oSender:nStatus == ssRUNNING
        /*
        Su código
        */
        ProcessMessages( 1000 )
    ENDDO

    LogFile( "Service stopped." )

RETURN
```

Important note:

In order to perform any operations of manipulating Windows services it is important that the application be run as '*administrator*'. You can control whether your application is raised as an administrator by using the **IsElevated()** function.

For further information consult the samples at `\samples\Winservice`. On that sample, the same application is used to manipulate and create a new service.

Hierarchy File TComponent descendant
\source\Winservice.prg

1.6.6.9.1 TWinService:Properties

■ Read only ■ Assignable ■ Design assignable ■ Run-time assignable

Scope	Name	Type	Initial value
■	<code>cDescription</code>	Character	""
■	<code>cDisplayName</code>	Character	""
■	<code>cName</code>	Character	""
■	<code>nStatus</code>	Numeric	ssSTOPPED

1.6.6.9.1.1 TWinService:cDescription

Service description.

Scope	Assignable
Type	Character
Initial value	""

1.6.6.9.1.2 TWinService:cDisplayName

Service short display name.

Scope	Assignable
Type	Character
Initial value	""

1.6.6.9.1.3 TWinService:cName

Service name.

Scope	Assignable
Type	Character
Initial value	""

1.6.6.9.1.4 TWinService:nStatus

Service status.

Scope	Assignable
Type	Numeric
Initial value	ssSTOPPED
Valores posibles	ssSTOPPED, ssSTART_PENDING, ssSTOP_PENDING, ssRUNNING, ssCONTINUE_PENDING, ssPAUSE_PENDING, ssPAUSED

1.6.6.9.2 TWinService:Methods

■ Constructor ■ Standard

Type	Name
■	Install

■	IsInstalled
■	IsRunning
■	Run
■	Start
■	Stop
■	Uninstall

1.6.6.9.2.1 TWinService:Install

Installs the service.

Type	Standard
Parameters	<cName> Service name <cDisplayName> Service description
Return value	<ISuccess> True if success

1.6.6.9.2.2 TWinService:IsInstalled

True if service is installed.

Type	Standard
Parameters	None
Return value	<IValue>

1.6.6.9.2.3 TWinService:IsRunning

True if service is running.

Type	Standard
Parameters	None
Return value	<IValue>

1.6.6.9.2.4 TWinService:Run

Executes the service as a process itself. This method is only used when our own application is going to behave as a service. It is absolutely necessary to have the OnRun event defined to work correctly. For more information, see the introduction of this control.

Type	Standard
Parameters	None
Return value	<ISuccess> True if success

1.6.6.9.2.5 TWinService:Start

Starts the service.

Type	Standard
Parameters	None
Return value	<ISuccess> True if success

1.6.6.9.2.6 TWinService:Stop

Stops the service.

Type	Standard
Parameters	None
Return value	<ISuccess> True if success

1.6.6.9.2.7 TWinService:Uninstall

Uninstalls the service.

Type	Standard
Parameters	None
Return value	<ISuccess> True if success

1.6.6.9.3 TWinService:Events

Name
OnPowerEvent
OnRun
OnShutDown

1.6.6.9.3.1 TWinService:OnPowerEvent

When the service is running as a process itself, this event occurs every time the computer is turned on.

Parameters:	<oSender>: Reference to the object that triggers the event
	<nEventType>: DBT_DEVICEARRIVAL DBT_DEVICEREMOVECOMPLETE DBT_DEVICEQUERYREMOVE DBT_DEVICEQUERYREMOVEFAILED DBT_DEVICEREMOVEPENDING DBT_CUSTOMEVENT
	For further information follow this link.
Return value:	NIL

1.6.6.9.3.2 TWinService:OnRun

When the service is running as a process itself, this event occurs every time the service state changes. For this event to occur it is necessary to have the service started with the Run method. For more information, see the introduction of this control.

Parameters:	<oSender>: Reference to the object that triggers the event
Return value:	NIL

1.6.6.9.3.3 TWinService:OnShutDown

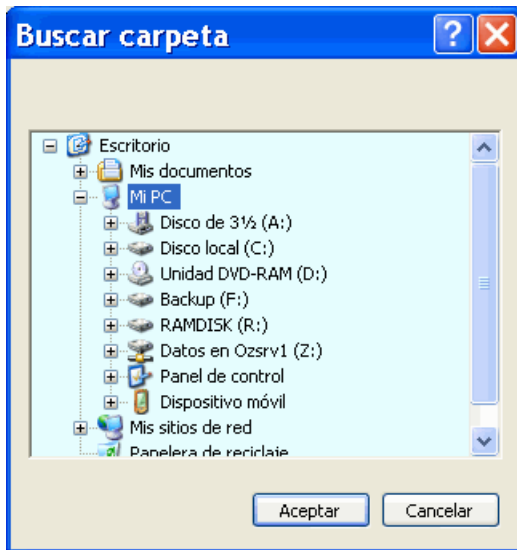
When the service is running as a process itself, this event occurs every time the computer is turned off.

Parameters:	<oSender>: Reference to the object that triggers the event
Return value:	NIL

1.6.7 Dialogs

1.6.7.1 TBrowseForFolder

System dialog to select folders and other components.



For more information, please check the SHBrowseForFolder function in the Windows API.

Hierarchy	Inherits from TComponent
See also	TFileOpenDlg, TFileSaveDlg
File name	\source\BrowseForFolder.prg

1.6.7.1.1 TBrowseForFolder:Properties

■ read Only ■ Design assignable ■ Design Design assignable ■ Run-time Design assignable

Scope	Name	Type	Initial value
■	cOkBtnText	Character	""
■	cRoot	Character	""
■	cStatusText	Character	""
■	cText	Character	""
■	cTitle	Character	""
■	IBrowseComputer	Logic	.F.
■	ICenter	Logic	.F.
■	IDontGoBelowDo main	Logic	.F.
■	IEditBox	Logic	.F.
■	IExtendedUI	Logic	.F.
■	IForceCreate	Logic	.F.
■	IIncludeFiles	Logic	.F.
■	IIncludeURLS	Logic	.F.
■	INewDlgStyle	Logic	.T.

□	INoNewFolderButton	Logic	.F.
□	IReturnFSAncestors	Logic	.F.
□	IReturnFSDirs	Logic	.F.
□	IShareable	Logic	.F.
□	IToolTips	Logic	.F.
□	IValidate	Logic	.F.
□	nIconIndex	Numeric	0
□	nRoot	Numeric	ssfNONE

All the properties must be assigned before to call the methods that show the dialog. Not all the properties are combinable. For more information, please check the SHBrowseForFolder function in the Windows API.

1.6.7.1.1.1 TBrowseForFolderDlg:cOkBtnText

Text to be displayed in the "Accept" button.

Scope:	Design assignable
Type:	Character
Initial value:	""

1.6.7.1.1.2 TBrowseForFolderDlg:cText

Indicates the dialog title.

Scope:	Design assignable
Type:	Character
Initial value:	""

1.6.7.1.1.3 TBrowseForFolderDlg:cRoot

Indicates the base directory where the dialog will start the selection. Also indicates the directory selected by the user after executing the Run method.

Scope:	Design assignable
Type:	Character
Initial value:	""

1.6.7.1.1.4 TBrowseForFolderDlg:cStatusText

Indicates the dialog status message.

Scope:	Design assignable
Type:	Character
Initial value:	""

1.6.7.1.1.5 TBrowseForFolderDlg:cTitle

Indicates the text to be displayer above the selection tree.

Scope:	Design assignable
Type:	Character
Initial value:	""

1.6.7.1.1.6 TBrowseForFolderDlg:lBrowseComputer

Shows computers.

Scope:	Design assignable
Type:	Logic
Initial value:	.F.

1.6.7.1.1.7 TBrowseForFolderDlg:lCenter

If true the dialog will be centered on the screen.

Scope:	Design assignable on design time
Type:	Logic
Initial value:	.F.

1.6.7.1.1.8 TBrowseForFolderDlg:lDontGoBelowDomain

Hides the folders that don't below to the domain.

Scope:	Design assignable
Type:	Logic
Initial value:	.F.

1.6.7.1.1.9 TBrowseForFolderDlg:IEditBox

Includes a TEdit control to type the selection.

Scope:	Design assignable
Type:	Logic
Initial value:	.F.

1.6.7.1.1.10 TBrowseForFolderDlg:IExtendedUI

Shows the extended dialog. It is equivalent to activate IEditBox and INewDlgStyle

Scope:	Design assignable
Type:	Logic
Initial value:	.F.

1.6.7.1.1.11 TBrowseForFolderDlg:IIncludeFiles

Allows to select files.

Scope:	Design assignable
Type:	Logic
Initial value:	.F.

1.6.7.1.1.12 TBrowseForFolderDlg:IIncludeURLS

Includes URLS. It must be used with IIncludeFiles and IExtendedUI

Scope:	Design assignable
Type:	Logic
Initial value:	.F.

1.6.7.1.1.13 TBrowseForFolderDlg:IForceCreate

Force to create the folder in the system when they don't exist or have been deleted.

Scope:	Design assignable
Type:	Logic
Initial value:	.F.

1.6.7.1.1.14 TBrowseForFolder:INewDlgStyle

Dialog with Drag & Drop support, menus, etc. according to the operating system version. On Windows 98 operating system Xailer includes a button to create directories.

Scope:	Design assignable
Type:	Logic
Initial value:	.T.

1.6.7.1.1.15 TBrowseForFolderDlg:INoNewFolderButton

It does not show the "New Folder" button. It works only in Windows XP or beyond.

Scope:	Design assignable
Type:	Logic
Initial value:	.F.

1.6.7.1.1.16 TBrowseForFolderDlg:IReturnFSAncestors

Allows to see only sub folders that below to the system folders.

Scope:	Design assignable
Type:	Logic
Initial value:	.F.

1.6.7.1.1.17 TBrowseForFolderDlg:IReturnFSDirs

Allows to see only system folders..

Scope:	Design assignable
Type:	Logic

Initial value:	.F.
-----------------------	-----

1.6.7.1.1.18 TBrowseForFolderDlg:IShareable

Shows shared resources. It must be used with INewDlgStyle.

Scope:	Design assignable
Type:	Logic
Initial value:	.F.

1.6.7.1.1.19 TBrowseForFolderDlg:ITooltips

Shows tool tips. It must be used with INewDlgStyle. It works only with Windows XP or beyond.

Scope:	Design assignable
Type:	Logic
Initial value:	.F.

1.6.7.1.1.20 TBrowseForFolderDlg:IValidate

Activates the OnValidate event a non valid name is introduced in the TEdit control dialog. It must be used with IEditBox.

Scope:	Design assignable
Type:	Logic
Initial value:	.F.

1.6.7.1.1.21 TBrowseForFolderDlg:nIconIndex

Index image associated with the selected file, folder or object in the system Imagelist.

Scope:	Design assignable
Type:	Numeric
Initial value:	0

See also: TSysImageList.

1.6.7.1.1.22 TBrowseForFolderDlg:nRoot

Selects one standard system folder. For example, `ssfPERSONAL` is equivalent to "C:\My Documents", according to the operating system version. This property allows to access the standard folders through its identifier instead of its name.

Scope:	Design assignable
Type:	Numeric
Initial value:	<code>ssfNONE</code>
Possible values:	<code>ssfNONE, ssfALTSTARTUP, ssfAPPDATA, ssfBITBUCKET, ssfCOMMONALTSTARTUP, ssfCOMMONAPPDATA, ssfCOMMONDESKTOPDIR, ssfCOMMONFAVORITES, ssfCOMMONPROGRAMS, ssfCOMMONSTARTMENU, ssfCOMMONSTARTUP, ssfCONTROLS, ssfCOOKIES, ssfDESKTOP, ssfDESKTOPDIRECTORY, ssfDRIVES, ssfFAVORITES, ssfFONTS, ssfHISTORY, ssfINTERNETCACHE, ssfLOCALAPPDATA, ssfMYPICTURES, ssfNETHOOD, ssfNETWORK, ssfPERSONAL, ssfPRINTERS, ssfPRINTHOOD, ssfPROFILE, ssfPROGRAMFILES, ssfPROGRAMS, ssfRECENT, ssfSENDTO, ssfSTARTMENU, ssfSTARTUP, ssfSYSTEM, ssfTEMPLATES, ssfWINDOWS</code>

For more information, please check the `SHBrowseForFolder` function from the Windows API.

1.6.7.1.2 TBrowseForFolderDlg:Methods

■ Constructor ■ Standard ■ Only after Create()

Typ	Name
■	<code>BrowseComputers</code>
■	<code>BrowseFile</code>
■	<code>BrowseFolder</code>
■	<code>BrowsePrinters</code>
■	<code>Execute</code>
■	<code>Run</code>

1.6.7.1.2.1 TBrowseForFolderDlg:Run

Type	Standard
Parameters	<code>[<nCSIDL>]</code>

	Constant that identifies an standard folder from the system. Optional
Return value	ISuccess .T. if an element has been selected

Description:

Activates the BrowseForFolder dialog.

1.6.7.1.2.2 TBrowseForFolderDlg:BrowseComputers

Type	Standard
Parameters	None
Return value	ISuccess .T. if an element has been selected

Description:

Configures and activates the dialog to show only computers.

1.6.7.1.2.3 TBrowseForFolderDlg:BrowseFile

Type	Standard
Parameters	[<nCSIDL>] Constant that identifies an standard folder from the system. Optional
Return value	ISuccess .T. if an element has been selected

Description:

Configures and activates the dialog to show files and folders.

1.6.7.1.2.4 TBrowseForFolderDlg:BrowseFolder

Type	Standard
Parameters	[<nCSIDL>] Constant that identifies an standard folder from the system. Optional
Return value	ISuccess .T. if an element has been selected

Description:

Configures and activates the dialog to show folders.

1.6.7.1.2.5 TBrowseForFolderDlg:BrowsePrinters

Type	Standard
Parameters	None
Return value	ISuccess .T. if an element has been selected

Description:

Configures and activates the dialog to show only printers.

1.6.7.1.3 TBrowseForFolderDlg:Events

Name	
	OnChange
	OnInitialized
	OnValidate

1.6.7.1.3.1 TBrowseForFolderDlg:OnChange

Event produced when the selected element is changed.

Parameters	<oSender> : Object that triggers the event.
:	<cFolder> : Selected element's name
Return value:	ISuccess If it returns .T., the dialog allows to select the highlighted element.

Description:

This event is triggered every time that the element selected is changed and allows to disable its selection through its return value.

1.6.7.1.3.2 TBrowseForFolderDlg:OnInitialized

Event produced when the dialog is initialized.

Parameters	<oSender> : Object that triggers the event.
:	<hWnd> : Dialog handle
Return	NIL

value:

Description:

This event is triggered when the dialog is initialized and allows its customization.

1.6.7.1.3.3 TBrowseForFolderDlg:OnValidate

Event to validate a name introduced by the user.

Parameters	<oSender> : Object that triggers the event.
:	<cText> : TEdit dialog control text.
Return value:	ISuccess If return .T. the dialog is still open.

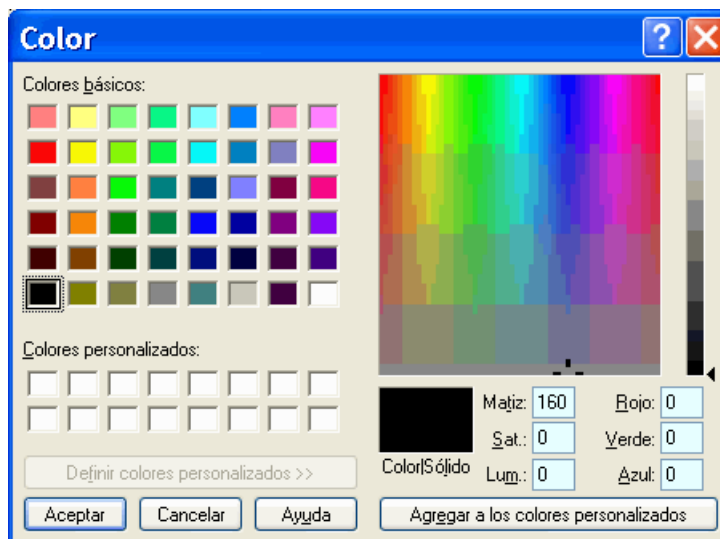
Description:

This event is triggered when the user type a not valid name in the TEdit dialog control.

See also the IEditBox property.

1.6.7.2 TChooseColorDlg

Class to manage the standard color selection Windows dialog.



Description:

The TChooseColorDlg class provides access to the standard color selection Windows dialog.

Hierarchy Inherits from TComponent
See also TChooseFontDlg
File Name \source\ChooseColor.prg

1.6.7.2.1 TChooseColorDlg:Properties

■ read Only ■ Assignable ■ Design assignable ■ Run-time assignable

Scope	Name	Type	Initial value
■	aCustomColors	Array	{16}
■	cHtmlColor	Character	""
■	HelpID	Numeric or character	NIL
■	IAnyColor	Logic	.T.
■	ICenter	Logic	.F.
■	IFullOpen	Logic	.F.
■	IPreventFullOpen	Logic	.F.
■	IRGBInit	Logic	.T.
■	IShowHelp	Logic	.F.
■	ISolidColor	Logic	.F.
■	nColor	Numeric	0
■	nLastError	Numeric	0

1.6.7.2.1.1 TChooseColorDlg:aCustomColors

Array with 16 color that indicates the custom colors to be shown in the dialog.

Scope	Assignable
Type	Array
Initial value	{ 16 } (All the values are numeric)

1.6.7.2.1.2 TChooseColorDlg:cHtmlColor

nColor property in HTML format.

Scope	Read only
Type	Character
Initial value	""

1.6.7.2.1.3 TChooseColorDlg:HelpID

Identifier for external Help system request when the push button is pressed.

Scope:	Assignable
Type:	Numeric or character
Initial value:	Nil

For further information consult the THelp class and the OnHelp and OnHelpClick events from TForm class.

1.6.7.2.1.4 TChooseColorDlg:IAnyColor

Shows all the possible basic colors.

Scope	Design assignable
Type	Logic
Initial value	.T.

1.6.7.2.1.5 TChooseColorDlg:ICenter

If true the dialog will be centered on the screen.

Scope	Design assignable
Type	Logic
Initial value	.F.

1.6.7.2.1.6 TChooseColorDlg:IFullOpen

Opens the dialog showing the custom color palette.

Scope	Design assignable
Type	Logic
Initial value	.F.

1.6.7.2.1.7 TChooseColorDlg:IPreventFullOpen

Avoids to show the custom colors.

Scope	Design assignable
--------------	-------------------

Type	Logic
Initial value	.F.

1.6.7.2.1.8 TChooseColorDlg:IRGBInit

Selects the nColor color property to activate the dialog.

Scope	Design assignable
Type	Logic
Initial value	.T.

1.6.7.2.1.9 TChooseColorDlg:IShowHelp

Shows the help icon (a question mark) in the dialog title.

Scope	Design assignable
Type	Logic
Initial value	.T.

1.6.7.2.1.10 TChooseColorDlg:ISolidColor

Shows only the basic solid colors.

Scope	Design assignable
Type	Logic
Initial value	.F.

1.6.7.2.1.11 TChooseColorDlg:nColor

Initial color that will be shown when the dialog is activated if the IRGBInit property is active. If the dialog is closed without cancel, the system will be updated with the selected color. This property also holds the value selected by the user on the dialog.

Scope	Design assignable
Type	Numeric
Initial value	0

1.6.7.2.1.12 TChooseColorDlg:nLastError

Indicates the error code, if any, during the dialog execution/creation.

Scope	readOnly
Type	Numeric
Initial value	0

1.6.7.2.2 TChooseColorDlg:Methods

■ Constructor ■ Standard ■ Only after Create()

Type	Name
■	Execute Run

1.6.7.2.2.1 TChooseColorDlg:Run

Sows the color selector dialog. The color selected by the user is set on the nColor property.

Type	Standard
Parameters	None
Return value	<ISuccess>: .T. if the dialog is closed with the "Accept" button.

1.6.7.2.3 TChooseColorDlg:Events

Name
OnClose
OnHelp
OnShow

1.6.7.2.3.1 TChooseColorDlg:OnClose

Event that is produced when the dialog is closed.

Parameters	<oSender>:
:	Reference to the object that triggers the event
Return value:	Nil

1.6.7.2.3.2 TChooseColorDlg:OnHelp

Event that is produced when the help button is pushed.

Parameters	<oSender> : A reference to the object that fired the request <HelpId> : HelpId property value of the control or form that request the help <nPosX> Mouse X coordinate when the request proceed from a mouse click <nPosY> Mouse Y coordinate when the request proceed from a mouse click
Return value:	NIL

The help system works as follows:

- The dialog that receives the help request checks that its property HelpId has a value. If Nil, then the request will be sent to its parent. If not Nil, then calls the help system of its container form. If on the form OnHelp event is assigned, then is triggered. If is not assigned then the TApplication help systems is called.
- If the TApplication OnHelp event is assigned, then is triggered. If is not assigned and its property oHelp has a valid THelp object, then it shows the help from the topic HelpId of the control or form that request the help.
- The **oSender** parameter of TForm or TApplication **OnHelp** event is the control that request the help whose HelpId had a value.

1.6.7.2.3.3 TChooseColorDlg:OnShow

Event that is produced when the dialog is shown.

Parameters	<oSender> : Reference to the object that triggers the event
Return value:	Nil

1.6.7.2.4 TChooseColorDlg:Functions

Name
ChooseColorDlg
RGBInverse
RGBToHTML

1.6.7.2.4.1 ChooseColorDlg

Function that encapsulates the process to create and configure the TChooseColorDlg object when it is not needed to personalize the dialog and it is enough to use it with the default options.

Parameters	<oParent> : Parent form <nColor> : Initial dialog's color, or the selected value.
Return value	<ISuccess> : .T. if the dialog is closed with the "Accept" button.

1.6.7.2.4.2 RGBInverse

Converts a RGB color to its inverse color.

Parameters	<nColor> : Standard RGB Windows color
Return value	<nInverseColor> : Inverse color in RGB format

1.6.7.2.4.3 RGBToHTML

Converts a color from a RGB to HTML format.

Parameters	<nColor> : Standard RGB Windows color
Return value	<cColor> : Color description in HTML format

1.6.7.3 TChooseFontDlg

This class encapsulates the common ChooseFont dialog.



Description:

The TChooseFontDlg provides access to the standard Windows font selection dialog.

Hierarchy Inherits from TComponent
See also TChooseFontDlg
File name: \source\ChooseFont.prg

1.6.7.3.1 TChooseFontDlg:Properties

■ read Only ■ Assignable ■ Design assignable ■ Run-time assignable

Scope	Name	Type	Initial value
■	cName	Character	""
■	HelpID	Numeric or character	NIL
■	IAllFonts	Logic	.F.
■	IBold	Logic	.F.
■	ICenter	Logic	.F.
■	IEffects	Logic	.F.
■	IFixedPitch	Logic	.F.
■	IForceFontExist	Logic	.F.
■	IItalic	Logic	.F.
■	INoFaceSel	Logic	.F.
■	INoScript	Logic	.F.
■	INoSimulations	Logic	.F.
■	INoSizeSel	Logic	.F.
■	INoStyleSel	Logic	.F.

■	INoVectorFonts	Logic	.F.
■	INoVertFonts	Logic	.F.
■	IPrinterFonts	Logic	.F.
■	IScalableOnly	Logic	.F.
■	IScreenFonts	Logic	.F.
■	IScriptsOnly	Logic	.F.
■	IShowHelp	Logic	.F.
■	IStrikeOut	Logic	.F.
■	ITrueTypeOnly	Logic	.F.
■	IUnderline	Logic	.F.
■	IWysiwyg	Logic	.F.
■	nCharSet	Numeric	0
■	nColor	Numeric	0
■	nHeight	Numeric	0
■	nLastError	Numeric	0
■	nSize	Numeric	0
■	nSizeMax	Numeric	0
■	nSizeMin	Numeric	0
■	nStyle	Numeric	0
■	nWeight	Numeric	FW_NORMAL
■	nWidth	Numeric	0

1.6.7.3.1.1 TChooseFontDlg:cName

Selected font name. If the dialog is close with 'Accept, it will return the selected front.

Scope	Assignable
Type	Character
Initial value	""

Description:

Long font description.

1.6.7.3.1.2 TChooseFontDlg:HelpID

Identifier for external Help system request when the push button is pressed.

Scope:	Assignable
Type:	Numeric or character
Initial value:	Nil

For further information consult the THelp class and the OnHelp and OnHelpClick events from TForm class.

1.6.7.3.1.3 TChooseFontDlg:IAIIFonts

Allows to select printer and screen fonts.

Scope	Design assignable
Type	Logic
Initial value	.T.

1.6.7.3.1.4 TChooseFontDlg:IBold

Preselects bold style. Returns the bold style from the selected font.

Scope	Design assignable
Type	Logic
Initial value	.F.

1.6.7.3.1.5 TChooseFontDlg:ICenter

Shows the dialog centered.

Scope	Design assignable
Type	Logic
Initial value	.T.

1.6.7.3.1.6 TChooseFontDlg:IEffects

Allows to select font effects like strikethrough and underline.

Scope	Design assignable
Type	Logic
Initial value	.F.

1.6.7.3.1.7 TChooseFontDlg:IFixedPitch

Allows to select only fixed pitch fonts.

Scope	Design assignable
Type	Logic
Initial value	.F.

1.6.7.3.1.8 TChooseFontDlg:IForceFontExist

Shows and error if the preselected font does not exist.

Scope	Design assignable
Type	Logic
Initial value	.F.

1.6.7.3.1.9 TChooseFontDlg:Italic

Selects the Italian style. Returns the Italian style from the selected font.

Scope	Design assignable
Type	Logic
Initial value	.F.

1.6.7.3.1.10 TChooseFontDlg:INoFaceSel

It does not show the initial font name.

Scope	Design assignable
Type	Logic
Initial value	.F.

1.6.7.3.1.11 TChooseFontDlg:INoScript

Disable the char set selection.

Scope	Design assignable
Type	Logic
Initial value	.F.

1.6.7.3.1.12 TChooseFontDlg!NoSimulations

Enables/Disables to show GDI simulated fonts.

Scope	Design assignable
Type	Logic
Initial value	.F.

1.6.7.3.1.13 TChooseFontDlg!NoSizeSel

Enables/Disables to show the initial font size.

Scope	Design assignable
Type	Logic
Initial value	.F.

1.6.7.3.1.14 TChooseFontDlg!NoStyleSel

Enables/Disables to show the initial font style.

Scope	Design assignable
Type	Logic
Initial value	.F.

1.6.7.3.1.15 TChooseFontDlg!NoVectorFonts

Disables vector fonts selection.

Scope	Design assignable
Type	Logic
Initial value	.F.

1.6.7.3.1.16 TChooseFontDlg!NoVertFonts

Disable vertical fonts.

Scope	Design assignable
Type	Logic
Initial value	.F.

1.6.7.3.1.17 TChooseFontDlg:IPrinterFonts

It shows only printer fonts.

Scope	Design assignable
Type	Logic
Initial value	.F.

1.6.7.3.1.18 TChooseFontDlg:IScalableOnly

Shows only scalable fonts.

Scope	Design assignable
Type	Logic
Initial value	F.

1.6.7.3.1.19 TChooseFontDlg:IScreenFonts

Shows only screen fonts.

Scope	Design assignable
Type	Logic
Initial value	.T.

1.6.7.3.1.20 TChooseFontDlg:IScriptsOnly

Allows OEM and ANSI fonts.

Scope	Design assignable
Type	Logic
Initial value	.T.

1.6.7.3.1.21 TChooseFontDlg:IShowHelp

Shows the help icon (a question mark) in the dialog title.

Scope	Design assignable
Type	Logic
Initial value	.T.

1.6.7.3.1.22 TChooseFontDlg:IStrikeOut

Selects the strikethrough style. Returns the strikethrough font style from the selected font.

Scope	Design assignable
Type	Logic
Initial value	.F.

1.6.7.3.1.23 TChooseFontDlg:ITrueTypeOnly

Allows only True Type fonts.

Scope	Design assignable
Type	Logic
Initial value	.F.

1.6.7.3.1.24 TChooseFontDlg:IUnderline

Selects the underline style. Returns the underline style from the selected font.

Scope	Design assignable
Type	Logic
Initial value	.F.

1.6.7.3.1.25 TChooseFontDlg:IWysiwyg

Shows only screen and printer fonts. (What you see is what you get)

Scope	Design assignable
Type	Logic
Initial value	.F.

1.6.7.3.1.26 TChooseFontDlg:nCharSet

Selects the font character set.

Scope	Design assignable
Type	Numeric
Initial value	csDEFAULT
Possible values:	csANSI, csDEFAULT, csSYMBOL, csSHIFTJIS, csGB2312, csHANGEUL, csCHINESEBIG5, csOEM

1.6.7.3.1.27 TChooseFontDlg:nColor

Initial or selected dialog color.

Scope	Assignable
Type	Numeric
Initial value	0

Consult the appendix for the list of available colors

1.6.7.3.1.28 TChooseFontDlg:nHeight

Selects the font height.

Scope	Assignable
Type	Numeric
Initial value	0

1.6.7.3.1.29 TChooseFontDlg:nLastError

Returns the error code (if any) for the last dialog creation or execution.

Scope	read Only
Type	Numeric
Initial value	0

1.6.7.3.1.30 TChooseFontDlg:nSize

Determines the font size.

Scope	Assignable
Type	Numeric
Initial value	0

1.6.7.3.1.31 TChooseFontDlg:nSizeMax

Indicates the maximum font size available.

Scope	Assignable
Type	Numeric
Initial value	0

1.6.7.3.1.32 TChooseFontDlg:nSizeMin

Indicates the minimum font size available.

Scope	Assignable
Type	Numeric
Initial value	0

1.6.7.3.1.33 TChooseFontDlg:nStyle

Indicates the font type selected.

Scope	Assignable
Type	Numeric
Initial value	

1.6.7.3.1.34 TChooseFontDlg:nWeight

Indicates the font weight.

Scope	Assignable
Type	Numeric
Initial value	FW_NORMAL

1.6.7.3.1.35 TChooseFontDlg:nWidth

Indicates the font width.

Scope	Assignable
Type	Numeric
Initial value	0

1.6.7.3.2 TChooseFontDlg:Methods

■ Constructor ■ Standard ■ Only after Create()

Typ	Name
■	CreateFont
■	Execute Run
■	SetFont

1.6.7.3.2.1 TChooseFontDlg>CreateFont

Returns and font object from the dialog properties. You can select a personalized font in the dialog and then create automatically a font using this method.

Type	Standard
Parameters	None
Return value	<oFont>

1.6.7.3.2.2 TChooseFontDlg:Run

Shows a font selection dialog.

Type	Standard
Parameters	None
Return value	<IOK> Returns .T. if the dialog was closed with 'Accept'.

1.6.7.3.2.3 TChooseFont:SetFont

Initializes the dialog properties with the oFont object properties.

Type	Standard
Parameters	<oFont>: TFont object
Return value	NIL

1.6.7.3.3 TChooseFontDlg:Events

Name	
OnClose	
OnHelp	
OnShow	

1.6.7.3.3.1 TChooseFontDlg:OnClose

Event that is produced when the dialog is closed.

Parameters	<oSender>:
:	Reference to the object that triggers the event
Return value:	Nil

1.6.7.3.3.2 TChooseFontDlg:OnHelp

Event that is produced when the help button is pushed.

Parameters	<oSender>:
:	A reference to the object that fired the request
	<HelpId>: HelpId property value of the control or form that request the help
	<nPosX> Mouse X coordinate when the request proceed from a mouse click
	<nPosY> Mouse Y coordinate when the request proceed from a mouse click
Return value:	NIL

The help system works as follows:

- The dialog that receives the help request checks that its property `HelpId` has a value. If `Nil`, then the request will be sent to its parent. If not `Nil`, then calls the help system of its container form. If on the form `OnHelp` event is assigned, then is triggered. If is not assigned then the `TApplication` help systems is called.
- If the `TApplication` `OnHelp` event is assigned, then is triggered. If is not assigned and its property `oHelp` has a valid `THelp` object, then it shows the help from the topic `HelpId` of the control or form that request the help.
- The **`oSender`** parameter of `TForm` or `TApplication` **`OnHelp`** event is the control that request the help whose `HelpId` had a value.

1.6.7.3.3.3 TChooseFontDlg:OnShow

Event that is produced when the dialog is shown.

Parameters	<oSender> :
:	Reference to the object that triggers the event
Return value:	Nil

1.6.7.3.4 TChooseFontDlg:Functions

Name
ChooseFontDlg
CreateFontDlg

1.6.7.3.4.1 ChooseFontDlg

Function that encapsulates the process to create and configure the `TChooseFontDlg` object when it is not needed to personalize the dialog and it is enough to use it with the default options.

Parameters	<oParent> : Form <cName> : Initial font name or font name selected <nHeight> : Initial font height or selected value <nColor> : Initial dialog color of selected value <IScreen> : Shows the screen fonts <IPrinter> : Shows the printer fonts <nCharSet> : Font char set
Return Value	<IOK> : If the dialog was closed with the Accept button.

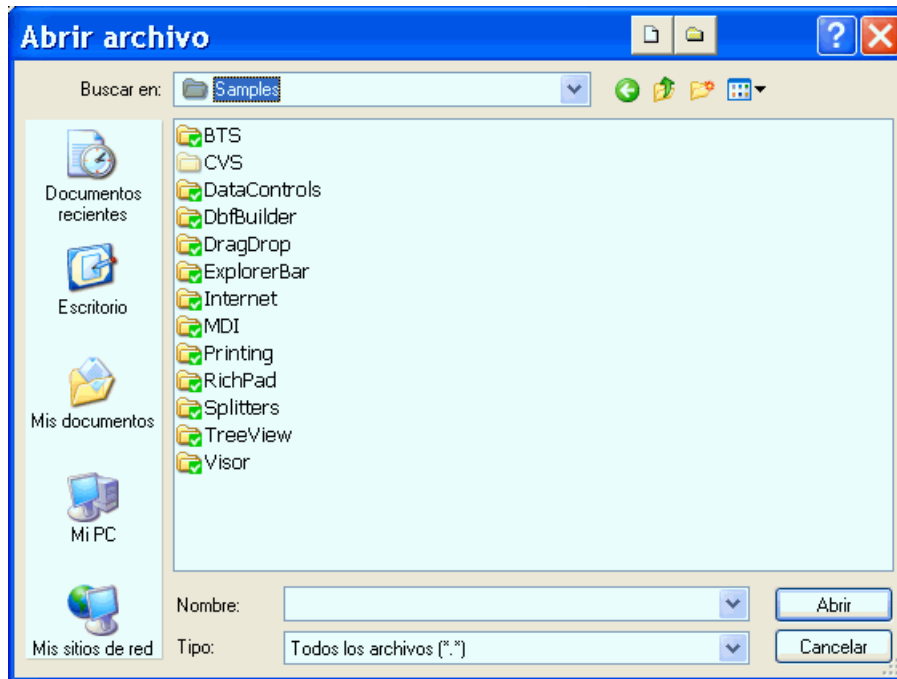
1.6.7.3.4.2 CreateFontDlg

Function that encapsulates the process to create and configure the TChooseFontDlg object when it is not needed to personalize the dialog and it is enough to use it with the default options. It returns a Tfont object already instantiated and ready to use.

Parameters	<oParent>: Parent form <cName>: Initial font name or font name selected <nHeight>: Initial font height or selected value <nColor>: Initial dialog color of selected value <IScreen>: Shows the screen fonts <IPrinter>: Shows the printer fonts <nCharSet>: Font char set
Return value	<oFont>: TFont object created and based in the selected dialog data, or NIL if the dialog is closed with Cancel .

1.6.7.4 TFileOpenDlg

Class to manage the standard file open Windows dialog.



Description:

The TFileOpenDlg class allows access to the standard file open Windows dialog.

You can use a fast access to this dialog using the function FileOpenDlg().

Inherited classes:

The class **TFileOpenImageDlg** is an inherited class from TFileOpenDlg specialized on image handling since includes a file preview and a image information display. The properties cTitle and cFilter are autonomously initialized for the kind of files to show and the unique new property is IShowImageInfo (by default .T.) that shows image information of the file been sought.

Hierarchy	Inherits from TComponent
See also	TFileSaveDlg
File name	\source\CommonDlg.prg

1.6.7.4.1 TFileOpenDlg:Properties

■ read Only ■ Assignable ■ Design assignable (Before to execute the Run method) ■ Run-time assignable

Scope	Name	Type	Initial value
■	cDefaultExt	Character	""
■	cFilename	Character	""
■	cFilenameOnly	Character	""
■	cFilePath	Character	""

■	cFilter	Character	"All Files (*.*) *.*"
■	cFullFilename	Character	""
■	cInitialDir	Character	""
■	cTitle	Character	"Open File"
■	HelpID	Numeric or character	NIL
■	IAllowMultiSelect	Logic	.F.
■	ICenter	Logic	.F.
■	ICreatePrompt	Logic	.F.
■	IDontAddToRece nt	Logic	.F.
■	IEnableSizing	Logic	.T.
■	IFileMustExist	Logic	.T.
■	IForceShowHidde n	Logic	.F.
■	IHideReadOnly	Logic	.T.
■	INoChangeDir	Logic	.T.
■	INoDereferenceLi nks	Logic	.F.
■	INoNetworkButton	Logic	.F.
■	INoValidate	Logic	.F.
■	IPathMustExist	Logic	.T.
■	IReadOnly	Logic	.F.
■	IShareAware	Logic	.F.
■	nFilterIndex	Numeric	1

1.6.7.4.1.1 TFileOpenDlg:cDefaultExt

Default filename extension to be open.

Scope	Design assignable (before to execute the Run method)
Type	Character
Initial value	""

1.6.7.4.1.2 TFileOpenDlg:cFilename

Default filename to be open.

After the Run method is executed, the property takes the selected filename.

Scope	Design assignable (before to execute the Run method)
Type	Character
Initial value	""

1.6.7.4.1.3 TFileOpenDlg:cFilenameOnly

Indicates the selected filename, excluding the directory (path). For example:

```
"fileopendlg.txt"
```

After the Run method is executed, the property takes the selected filename.

Scope	Design assignable (before to execute the Run method)
Type	Character
Initial value	""

1.6.7.4.1.4 TFileOpenDlg:cFilter

Mask with the possible filenames and extensions types that can be selected.

Scope	Design assignable (before to execute the Run method)
Type	Character
Initial value	"All Files (*.*) *.*"

Examples:

```
"Rich Text Format (*.rtf)|*.rtf|" + ;
"Only Text Format (*.txt)|*.txt|" + ;
"All files (*.*)|*.*"
```

As you can see, it is needed to indicate the extension description first and then, using the 'pipe' separator, you need to specify the file extension. In the case that you need to specify more than one filter, you can type it and separate it with the 'pipe' symbol.

The nFilterIndex property, allows to indicate the initial filter that will be shown for the FileOpen dialog.

1.6.7.4.1.5 TFileOpenDlg:cFilePath

Indicates the selected file directory (path), excluding the filename. For example:

```
"c:\xailer\doc\"
```

Scope	Design assignable (before to execute the Run method)
--------------	--

Type	Character
Initial value	""

1.6.7.4.1.6 TFileOpenDlg:cFullFilename

Default filename to be open, with the path included. For example:

```
"c:\xailer\doc\fileopendlg.txt"
```

After the Run method is executed, the property takes the selected filename.

Scope	Design assignable (before to execute the Run method)
Type	Character
Initial value	""

1.6.7.4.1.7 TFileOpenDlg:cInitialDir

Default directory where the FileOpen dialog will be used.

Scope	Design assignable (before to execute the Run method)
Type	Character
Initial value	""

1.6.7.4.1.8 TFileOpenDlg:cTitle

OpenFile Dialog's title.

Scope	Design assignable (before to execute the Run method)
Type	Character
Initial value	"Open File"

1.6.7.4.1.9 TFileOpenDlg:HelpID

Identifier for external Help system request when the push button is pressed.

Scope:	Assignable
Type:	Numeric or character

Initial value: Nil

For further information consult the THelp class and the OnHelp and OnHelpClick events from TForm class.

1.6.7.4.1.10 TFileOpenDlg:IAAllowMultiSelect

Allows to select multiple files.

Scope	Design assignable (before to execute the Run method)
Type	Logic
Initial value	.F.

When this property is .T., the cFilename, cFullFilename and cFilenameOnly will become character arrays and they will take all the files selected in the dialog.

1.6.7.4.1.11 TFileOpenDlg:ICenter

If true the dialog will be centered on the screen.

Scope	Design assignable (before to execute the Run method)
Type	Logic
Initial value	.F.

1.6.7.4.1.12 TFileOpenDlg:ICreatePrompt

If a filename that does not exist is selected, it asks to the user if it can be created.

Scope	Design assignable (before to execute the Run method)
Type	Logic
Initial value	.F.

1.6.7.4.1.13 TFileOpenDlg:IDontAddToRecent

It does not add the filename to the recent document list.

Scope	Design assignable (before to execute the Run method)
--------------	--

Type	Logic
Initial value	.F.

1.6.7.4.1.14 TFileOpenDlg:IEnableSizing

Allows to change the dialog's window size.

Scope	Design assignable (before to execute the Run method)
Type	Logic
Initial value	.T.

1.6.7.4.1.15 TFileOpenDlg:IFileMustExist

It allows to select only existing files.

Scope	Design assignable (before to execute the Run method)
Type	Logic
Initial value	.T.

1.6.7.4.1.16 TFileOpenDlg:IForceShowHidden

Shows the hidden and system files.

Scope	Design assignable (before to execute the Run method)
Type	Logic
Initial value	.F.

1.6.7.4.1.17 TFileOpenDlg:IHideReadOnly

Hides the read only files checkbox.

Scope	Design assignable (before to execute the Run method)
Type	Logic
Initial value	.T.

1.6.7.4.1.18 TFileOpenDlg:INoChangeDir

Restores the directory if the user changed it when he was selecting the file. This is an obsolete style in Windows 2K/XP.

Scope	Design assignable (before to execute the Run method)
Type	Logic
Initial value	.T.

1.6.7.4.1.19 TFileOpenDlg:INoDereferenceLinks

When an shortcut is selected, return the filename and path from the LNK file or to the shortcut reference.

Scope	Design assignable (before to execute the Run method)
Type	Logic
Initial value	.F.

1.6.7.4.1.20 TFileOpenDlg:INoNetworkButton

Hides the "Network" button in the dialog.

Scope	Design assignable (before to execute the Run method)
Type	Logic
Initial value	.F.

1.6.7.4.1.21 TFileOpenDlg:INoValidate

Allows to type non valid characters in the file name.

Scope	Design assignable (before to execute the Run method)
Type	Logic
Initial value	.F.

1.6.7.4.1.22 TFileOpenDlg:IPathMustExist

Allows to the user to use only valid filenames and paths.

Scope	Design assignable (before to execute the Run method)
Type	Logic
Initial value	.T.

1.6.7.4.1.23 TFileOpenDlg:IReadOnly

Checks the read only files checkbox.

Scope	Design assignable (before to execute the Run method)
Type	Logic
Initial value	.F.

1.6.7.4.1.24 TFileOpenDlg:IShareAware

If the dialogs fails due a network error, the error is ignored.

Scope	Design assignable (before to execute the Run method)
Type	Logic
Initial value	.F.

1.6.7.4.1.25 TFileOpenDlg:nFilterIndex

Default filter to e used when the FileOpen dialog is open.

Scope	Design assignable (before to execute the Run method)
Type	Numeric
Initial value	1

See also the cFilter property.

1.6.7.4.2 TFileOpenDlg:Methods

■ Constructor ■ Standard ■ Only after Create()

Typ	Name
■	Execute Run

1.6.7.4.2.1 TFileOpenDlg:Run

Shows the OpenFile dialog.

Type	Standard
Parameters	None
Return value	<ISuccess> Returns .T. if it is closed with the "OK" button. Otherwise, it will be .F.

1.6.7.4.3 TFileOpenDlg:Events

Name
OnClose
OnHelp
OnShow

1.6.7.4.3.1 TFileOpenDlg:OnClose

Event that is produced when the dialog is closed.

Parameters	<oSender>:
:	Reference to the object that triggers the event
Return value:	Nil

1.6.7.4.3.2 TFileOpenDlg:OnHelp

Event that is produced when the help button is pushed.

Parameters	<oSender>:
:	A reference to the object that fired the request
	<HelpId>: HelpId property value of the control or form that request the help
	<nPosX> Mouse X coordinate when the request proceed

	from a mouse click <nPosY> Mouse Y coordinate when the request proceed from a mouse click
Return value:	NIL

The help system works as follows:

- The dialog that receives the help request checks that its property HelpId has a value. If Nil, then the request will be sent to its parent. If not Nil, then calls the help system of its container form. If on the form OnHelp event is assigned, then is triggered. If is not assigned then the TApplication help systems is called.
- If the TApplication OnHelp event is assigned, then is triggered. If is not assigned and its property oHelp has a valid THelp object, then it shows the help from the topic HelpId of the control or form that request the help.
- The **oSender** parameter of TForm or TApplication **OnHelp** event is the control that request the help whose HelpId had a value.

1.6.7.4.3.3 TFileOpenDlg:OnShow

Event that is produced when the dialog is shown.

Parameters	<oSender>: Reference to the object that triggers the event
Return value:	Nil

1.6.7.4.4 TFileOpenDlg:Functions

Name	
FileOpenImageDlg	
FileOpenDlg	

1.6.7.4.4.1 FileOpenImageDlg

Quick function to be used in the TFileOpenImageDlg

Parameters	<oParent> Proprietary form [<cTitle>] Dialog title. Default: cTitle property value. [<cInitialDir>] Initial directory. Default: cInitialDir property value. [<cFilter>] Filter to be used. Default: the cFilter property
-------------------	---

	value. [<nFilter>] Filter selected. Default: The nFilterIndex property value.
Return value	<cFile> Filename selected

1.6.7.4.4.2 FileOpenDlg

Quick function to be used in the TFileOpenDlg

Parameters	<oParent> Proprietary form [<cTitle>] Dialog title. Default: cTitle property value. [<cInitialDir>] Initial directory. Default: cInitialDir property value. [<cFilter>] Filter to be used. Default: the cFilter property value. [<nFilter>] Filter selected. Default: The nFilterIndex property value.
Return value	<cFile> Filename selected

1.6.7.5 TFileOperationDlg

This class encapsulates the common dialog to copy, move, rename or delete files.

Hierarchy	Inherits from TComponent
See also	TProgressDlg
File name	\source\FileOperation.prg

1.6.7.5.1 TFileOperationDlg:Properties

■ read Only ■ Assignable ■ Design assignable ■ Run-time assignable

Scope	Name	Type	Initial value
■	aFrom	Array	{}
■	aRenamed	Array	{}
■	aTo	Array	{}
■	cText	Logic	""
■	IAborted	Logic	.F.
■	IAllowUndo	Logic	.F.

■	IAutoRename	Logic	.F.
■	IFilesOnly	Logic	.F.
■	INoConfirmation	Logic	.F.
■	INoConfirmMkDir	Logic	.F.
■	INoCopySecurity Atrib	Logic	.F.
■	INoErrorUI	Logic	.F.
■	ISilent	Logic	.F.
■	ISimpleProgress	Logic	.F.
■	IWantMapping	Logic	.F.
■	nAction	Numeric	foNONE

1.6.7.5.1.1 TFileOperationDlg:aFrom

Filename or origin directory or origin files.

Scope	Design assignable (before to execute the Run method)
Type	Array
Initial value	{}

1.6.7.5.1.2 TFileOperationDlg:aRenamed

Contains an array with the list of renamed files when has been a name collision. Every element is an array that contains the original name and the new name.

Scope	read Only
Type	Array
Initial value	{}

To have the array with the list of renamed files, it is needed to activate the IAutoRename and IWantMapping properties.

1.6.7.5.1.3 TFileOperationDlg:aTo

Filename or target directory or list of target files.

Scope	Design assignable (before to execute the Run method)
Type	Array
Initial value	{}

1.6.7.5.1.4 TFileOperationDlg:cText

Dialog's windows title.

Scope	Design assignable (before to execute the Run method)
Type	Character
Initial value	""

1.6.7.5.1.5 TFileOperationDlg:IAborted

Indicates if the operation was canceled by the user before it finishes.

Scope	read Only
Type	Logic
Initial value	.F.

1.6.7.5.1.6 TFileOperationDlg:IAllowUndo

Keeps the information to undo the operation, if possible.

Scope	Design assignable (before to execute the Run method)
Type	Logic
Initial value	.F.

1.6.7.5.1.7 TFileOperationDlg:IAutoRename

Renames automatically the target files in case that exist one with the same name.

Scope	Design assignable (before to execute the Run method)
Type	Logic
Initial value	.F.

See also the `aRenamed` property.

1.6.7.5.1.8 TFileOperationDlg:IFilesOnly

Executes file operations over files only when it is used *.* mask as origin.

Scope	Design assignable (before to execute the Run method)
Type	Logic
Initial value	.F.

1.6.7.5.1.9 TFileOperationDlg:INoConfirmation

Assumes the answer "Yes" if any dialog is shown during the operation.

Scope	Design assignable (before to execute the Run method)
Type	Logic
Initial value	.F.

1.6.7.5.1.10 TFileOperationDlg:INoConfirmMkDir

It does not confirm the target directory creation, if is needed to create the target directory.

Scope	Design assignable (before to execute the Run method)
Type	Logic
Initial value	.F.

1.6.7.5.1.11 TFileOperationDlg:INoCopySecurityAttrib

Do not copy files security attribs.

Scope	Design assignable (before to execute the Run method)
Type	Logic
Initial value	.F.

1.6.7.5.1.12 TFileOperationDlg:INoErrorUI

In case of error, it will not display any message.

Scope	Design assignable (before to execute the Run method)
Type	Logic
Initial value	.F.

1.6.7.5.1.13 TFileOperationDlg:ISilent

It will not display any dialog in the process.

Scope	Design assignable (before to execute the Run method)
Type	Logic
Initial value	.F.

1.6.7.5.1.14 TFileOperationDlg:ISimpleProgress

It will show the progress bar only, without the file names that is operating.

Scope	Design assignable (before to execute the Run method)
Type	Logic
Initial value	.F.

1.6.7.5.1.15 TFileOperationDlg:IWantMapping

Returns file list that have been renamed because there are other files with the same name in the target.

Scope	Design assignable (before to execute the Run method)
Type	Logic
Initial value	.F.

It must be used together with the IAutoRename property.

1.6.7.5.1.16 TFileOperationDlg:nAction

Indicates the operation type that will be executed.

Scope	Design assignable (before to execute the Run method)
Type	Numeric
Initial value	foNONE
Possible values	foNONE, foMOVE, foCOPY, foDELETE, foRENAME

1.6.7.5.2 TFileOperationDlg:Methods

■ Constructor ■ Standard ■ Only after Create()

Type	Name
■	Execute Run

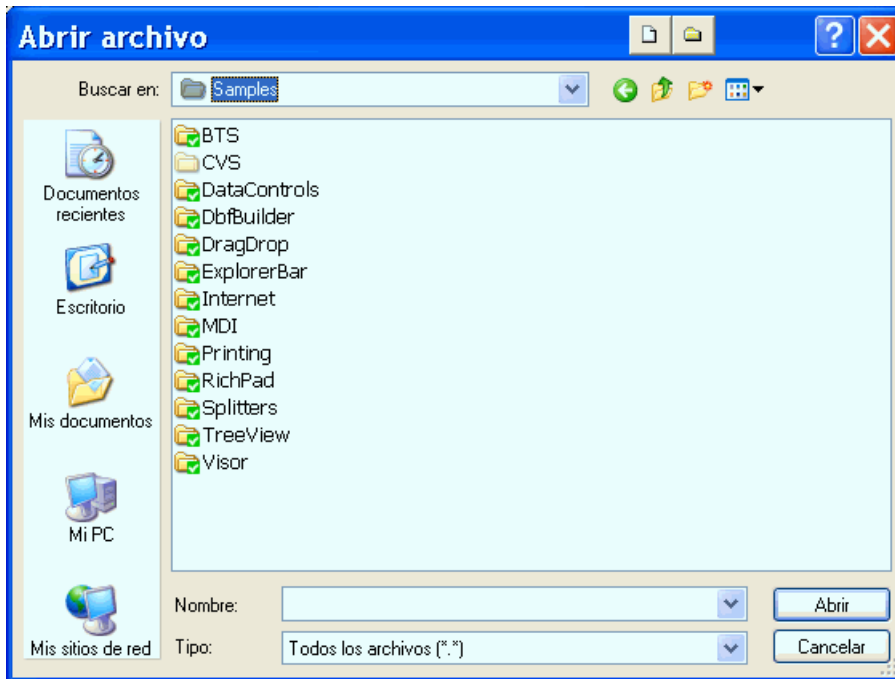
1.6.7.5.2.1 TFileOperationDlg:Run

Activates the FileOperation dialog and executes the action indicates in nAction.

Type	Standard
Parameters	None
Return value	<ISuccess> .T. if the operation has been successful

1.6.7.6 TFileSaveDlg

Class to manage the standard file selector Windows dialog.



Description:

The TFileSaveDlg class allows to access to the standard file selector Windows dialog, used to create or save files.

Inherited classes:

The class **TFileSaveImageDlg** is an inherited class from TFileSaveDlg specialized on image handling since includes a file preview and a image information display.

Hierarchy Inherits from TFileOpenDlg
See also TFileOpenDlg
File name \source\CommonDlg.prg

1.6.7.6.1 TFileSaveDlg:Properties

■ read Only ■ Assignable ■ Design assignable ■ Run-time assignable

Scope	Name	Type	Initial value
■	cDefaultExt	Character	"Save File"
■	cTitle	Character	"Save File"
■	INoTestFileCreate	Logic	.F.
■	IOverWritePrompt	Logic	.T.

1.6.7.6.1.1 TFileSaveDlg:cDefaultExt

Default filename extension to be saved.

Scope	Design assignable (before to execute the Run method)
Type	Character
Initial value	""

1.6.7.6.1.2 TFileSaveDlg:cTitle

Dialog's title.

Scope	Design assignable (before to execute the Run method)
Type	Character
Initial value	"Save File"

1.6.7.6.1.3 TFileSaveDlg:lNoTestFileCreate

Indicates if the file will not be create before to close the dialog.

Scope	Design assignable (before to execute the Run method)
Type	Logic
Initial value	.F.

1.6.7.6.1.4 TFileSaveDlg:lOverWritePrompt

If the file exists, it will ask to the user if he wants to overwrite it or not.

Scope	Design assignable (before to execute the Run method)
Type	Logic
Initial value	.T.

1.6.7.7 TFileSaveDlg:Functions

Name
FileSavelImageDlg
FileSaveDlg

1.6.7.7.1 FileSavelImageDlg

Quick function to be used in the TFileSavelImageDlg

Parameters	<oParent> Proprietary form [<cTitle>] Dialog title. Default: cTitle property value. [<cInitialDir>] Initial directory. Default: cInitialDir property value. [<cFilter>] Filter to be used. Default: the cFilter property value. [<nFilter>] Filter selected. Default: The nFilterIndex property value.
Return value	<cFile> Filename selected

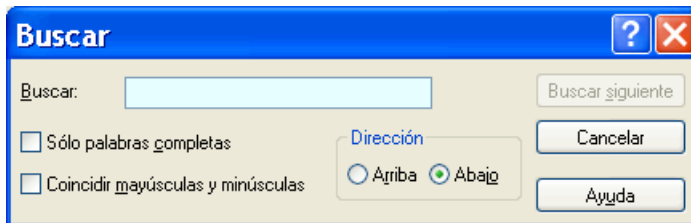
1.6.7.7.2 FileSaveDlg

Quick function to be used in the TFileSaveDlg

Parameters	<oParent> Proprietary form [<cTitle>] Dialog title. Default: cTitle property value. [<cInitialDir>] Initial directory. Default: cInitialDir property value. [<cFilter>] Filter to be used. Default: the cFilter property value. [<nFilter>] Filter selected. Default: The nFilterIndex property value.
Return value	<cFile> Filename selected

1.6.7.8 TFindDlg

This class encapsulates the common FindText Windows dialog.



Description:

The TFindDlg class allows to manage the common FindText Windows dialog.

Hierarchy	Inherits from TComponent
File name	\source\FindDlg.prg

1.6.7.8.1 TFindDlg:Properties

■ read Only ■ Assignable ■ Design assignable ■ Run-time assignable

Scope	Name	Type	Initial value
■	cText	Character	""
■	HelpID	Numeric or character	NIL
■	ICenter	Logic	.T.
■	IDown	Logic	.T.
■	IMatchCase	Logic	.F.
■	INoCase	Logic	.F.
■	INoUpDown	Logic	.F.
■	INoWholeWord	Logic	.F.
■	IShowHelp	Logic	.T.
■	IWholeWord	Logic	.F.

1.6.7.8.1.1 TFindDlg:cText

Initial text to search.

Scope	Assignable
Type	Character
Initial value	""

1.6.7.8.1.2 TFindDlg:HelpID

Identifier for external Help system request when the push button is pressed.

Scope:	Assignable
Type:	Numeric or character
Initial value:	Nil

For further information consult the THelp class and the OnHelp and OnHelpClick events from TForm class.

1.6.7.8.1.3 TFindDlg:ICenter

Shows the dialog centered.

Scope	Design assignable
Type	Logic
Initial value	.T.

1.6.7.8.1.4 TFindDlg:IDown

Initial search direction.

Scope	Assignable
Type	Logic
Initial value	.T.
Possible values	<.T.> Indicates to the end <.F.> Indicates to the beginning.

1.6.7.8.1.5 TFindDlg:IMatchCase

Case insensitive.

Scope	Assignable
Type	Logic
Initial value	.F.

1.6.7.8.1.6 TFindDlg:INoCase

Hides the possibility of case sensitive.

Scope	Assignable
Type	Logic
Initial value	.F.

1.6.7.8.1.7 TFindDlg:INoUpDown

Hides the select buttons from the search direction in the dialog.

Scope	Assignable
Type	Logic
Initial value	.F.

1.6.7.8.1.8 TFindDlg:INoWholeWord

Hides the possibility to select complete words.

Scope	Assignable
Type	Logic
Initial value	.F.

1.6.7.8.1.9 TFindDlg:IShowHelp

Shows the help icon (a question mark) in the dialog's title.

Scope	Assignable
Type	Logic
Initial value	.T.

1.6.7.8.1.10 TFindDlg:IWholeWord

Searches only completed words.

Scope	Assignable
Type	Logic
Initial value	.F.

1.6.7.8.2 TFindDlg:Methods

■ Constructor ■ Standard

Type	Name
■	Execute / Run

1.6.7.8.2.1 TFindDlg:Run

Activates and shows the FindText Windows dialog.

Type	Standard
Parameters	None
Return value	<ISuccess> .T. if the dialog was executed successfully.

1.6.7.8.3 TFindDlg:Events

Name
OnFind
OnHelp
OnClose

1.6.7.8.3.1 TFindDlg:OnFind

Events that is produced every time that the "Search" button is pushed in the dialog.

Type	Standard
Parameters	<oSender> Reference to the object that triggers the event. <cText> Text to search <lDown> Search to the beginning or to the end <lWholeWord> Search complete words <lMatchCase> Case insensitive
Return value	NIL

1.6.7.8.3.2 TFindDlg:OnHelp

Event that is produced when the help button is pushed.

Parameters	<oSender> : A reference to the object that fired the request <HelpId> : HelpId property value of the control or form that request the help <nPosX> Mouse X coordinate when the request proceed from a mouse click <nPosY> Mouse Y coordinate when the request proceed from a mouse click
Return value:	NIL

The help system works as follows:

- The dialog that receives the help request checks that its property HelpId has a value. If Nil, then the request will be sent to its parent. If not Nil, then calls the help system of its container form. If on the form OnHelp event is assigned, then is triggered. If is not assigned then the TApplication help systems is called.
- If the TApplication OnHelp event is assigned, then is triggered. If is not assigned and its property oHelp has a valid THelp object, then it shows the help from the topic HelpId of the control or form that request the help.
- The **oSender** parameter of TForm or TApplication **OnHelp** event is the control that request the help whose HelpId had a value.

1.6.7.8.3.3 TFindDlg:OnClose

Event that is produced when the dialog is closed.

Type	Standard
Parameters	<oSender> Reference to the object that triggers the event.
Return value	NIL

1.6.7.9 TFindFile

This class encapsulates the standard file search Windows functions.

Description:

The TFindFile class allows to manipulate the standard file search Windows functions.

Hierarchy Inherits from **TWinObject**
File Name \source\Findfile.prg

1.6.7.9.1 TFindFile:Properties

■ read Only ■ Assignable ■ Design assignable ■ Run-time assignable

Scope	Name	Type	Initial value
■	cCreationTime	Character	"00:00:00"
■	cFileMask	Character	"*.*"
■	cFileName	Character	""
■	cFullPath	Character	""
■	cLastAccessTime	Character	"00:00:00"
■	cLastWriteTime	Character	"00:00:00"
■	cShortFileName	Character	""
■	cShortPath	Character	""
■	dCreationDate	Date	CToD("")
■	dLastAccessDate	Date	CToD("")
■	dLastWriteDate	Date	CToD("")
■	IArchive	Logic	.F.
■	ICompressed	Logic	.F.
■	IDirectory	Logic	.F.
■	IEncrypted	Logic	.F.
■	IHidden	Logic	.F.
■	INormal	Logic	.F.
■	IOffline	Logic	.F.
■	IReadOnly	Logic	.F.
■	ISystem	Logic	.F.
■	ITemporary	Logic	.F.
■	nFileSize	Numeric	0

1.6.7.9.1.1 TFindFile:cCreationTime

File creation time.

Scope	read Only
Type	Character
Initial value	"00:00:00"

1.6.7.9.1.2 TFindFile:cFileMask

Mask indicating the files to search.

Scope	Assignable
Type	Character
Initial value	"*.*"

1.6.7.9.1.3 TFindFile:cFileName

Indicates the file name.

Scope	read Only
Type	Character
Initial value	""

1.6.7.9.1.4 TFindFile:cFullPath

Indicates the file's full path.

Scope	read Only
Type	Character
Initial value	""

1.6.7.9.1.5 TFindFile:cLastAccessTime

Indicates the time for the last access to the file.

Scope	read Only
Type	Character
Initial value	"00:00:00"

1.6.7.9.1.6 TFindFile:cLastWriteTime

Indicate the last update to the file.

Scope	read Only
Type	Character
Initial value	"00:00:00"

1.6.7.9.1.7 TFindFile:cShortFileName

Indicates the short file name.

Scope	read Only
Type	Character
Initial value	""

1.6.7.9.1.8 TFindFile:cShortPath

Indicates the file pat in short format.

Scope	read Only
Type	Character
Initial value	""

1.6.7.9.1.9 TFindFile:dCreationDate

Indicates the file creation date.

Scope	read Only
Type	Date
Initial value	CToD("")

1.6.7.9.1.10 TFindFile:dLastAccessDate

Indicates the last access date to the file.

Scope	read Only
Type	Date
Initial value	CToD("")

1.6.7.9.1.11 TFindFile:dLastWriteDate

Indicates the last update date to the file.

Scope	readOnly
Type	Date
Initial value	CToD("")

1.6.7.9.1.12 TFindFile:IArchive

Indicates if the file attribute is activated.

Scope	readOnly
Type	Logic
Initial value	.F.

1.6.7.9.1.13 TFindFile:ICompressed

Indicates if the file is compressed.

Scope	readOnly
Type	Logic
Initial value	.F.

1.6.7.9.1.14 TFindFile:IDirectory

Indicates if the file is a directory.

Scope	readOnly
Type	Logic
Initial value	.F.

1.6.7.9.1.15 TFindFile:IEncrypted

The file is encrypted. If it is a directory, all its content is encrypted as well.

Scope	readOnly
Type	Logic
Initial value	.F.

1.6.7.9.1.16 TFindFile:IHidden

Indicates if the file is hidden.

Scope	read Only
Type	Logic
Initial value	.F.

1.6.7.9.1.17 TFindFile:INormal

Indicates if the normal attribute is activated.

Scope	read Only
Type	Logic
Initial value	.F.

1.6.7.9.1.18 TFindFile:IOffline

The file is not available.

Scope	read Only
Type	Logic
Initial value	.F.

1.6.7.9.1.19 TFindFile:IReadOnly

Indicates if is read only file.

Scope	read Only
Type	Logic
Initial value	.F.

1.6.7.9.1.20 TFindFile:ISystem

The file is a system file or is being used by the system.

Scope	read Only
Type	Logic
Initial value	.F.

1.6.7.9.1.21 TFindFile:ITemporary

Indicates if the file is temporal.

Scope	read Only
Type	Logic
Initial value	.F.

1.6.7.9.1.22 TFindFile:nFileSize

Indicates the file size.

Scope	read Only
Type	Numeric
Initial value	0

1.6.7.9.2 TFindFile:Methods

■ Constructor ■ Standard

Typ	Name
■	Create
■	Destroy
■	Find
■	FindNext

1.6.7.9.2.1 TFindFile:Create

Creates the object with the specified file mask.

Type	Constructor
Parameters	<cFileMask> Mask indicating the files to search

Return value	Self Reference (Self)
---------------------	-----------------------

1.6.7.9.2.2 TFindFile:Destroy

Finalizes the current file search, releases the resources and triggers the OnStop event.

Type	Standard
Parameters	None
Return value	<ISuccess> .T. if the file search ended successfully

1.6.7.9.2.3 TFindFile:Find

Starts file search operation according with the specified mask and triggers the OnStart event. If a file with the specified criteria is found, it triggers the OnFind event.

Type	Standard
Parameters	None
Return value	<ISuccess> .T. if a file with the specified criteria is found

1.6.7.9.2.4 TFindFile:FindNext

Continues the search started with Find. If a file is found, the OnFind event is triggered.

Type	Standard
Parameters	None
Return value	<ISuccess> .T. if a file with the specified criteria is found

1.6.7.9.3 TFindFile:Events

Name
OnFind
OnStart
OnStop

1.6.7.9.3.1 TFindFile:OnStart

Event that is triggered when starts the file search operation.

Parameters	<oSender> Reference to the object that triggers the event
Return value	NIL

1.6.7.9.3.2 TFindFile:OnFind

Event that is triggered every time that a file with the search criteria is found.

Parameters	<oSender> Reference to the object that triggers the event
Return value	NIL

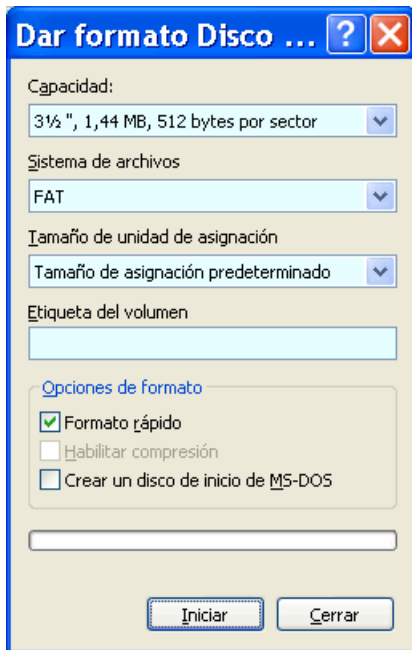
1.6.7.9.3.3 TFindFile:OnStop

Event that is triggered when finalizes the file search operation.

Parameters	<oSender> Reference to the object that triggers the event
Return value	NIL

1.6.7.10 TFormatDriveDlg

This class encapsulates the common FormatDrive Windows dialog.



Description:

This class allows to manipulate the FormatDrive Windows dialog.

Hierarchy Inherits from TComponent
File name \source\FormatDriveDlg.prg

1.6.7.10.1 TFormatDriveDlg:Properties

■ read Only ■ Assignable ■ Design assignable ■ Run-time assignable

Scope	Name	Type	Initial value
■	cDrive	Character	""
■	cLabel	Character	""
■	nType	Númeric	shfQUICK
■	nLastError	Numeric	0

1.6.7.10.1.1 TFormatDriveDlg:cDrive

Indicates the disk drive to be formatted.

Scope	Assignable
Type	Character
Initial value	""

1.6.7.10.1.2 TFormatDriveDlg:cLabel

Indicates the volume label for the disk to be formatted.

Scope	Assignable
Type	Character
Initial value	""

1.6.7.10.1.3 TFormatDriveDlg:nType

Format type.

Scope	Assignable
Type	Numeric
Initial Value	shfQUICK
Possible values	shfQUICK, shfFULL, shfSYSONLY

1.6.7.10.1.4 TFormatDriveDlg:nLastError

Indicates if an error has been produced when the dialog was shown or during the format process.

Scope	read Only
Type	Numeric
Initial value	0.

1.6.7.10.2 TFormatDriveDlg:Methods

■ Constructor ■ Standard

Typ	Name
■	Execute Run

1.6.7.10.2.1 TFormatDriveDlg:Run

Class constructor.

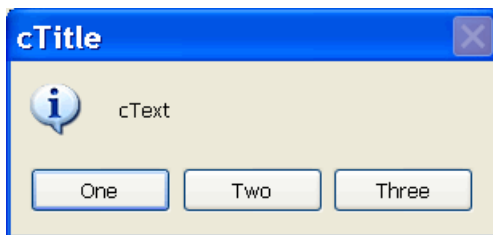
Type	Standard
-------------	----------

Parameters	None
Return value	<ISuccess> Returns .T. if the dialog was closed. Otherwise, it returns .F.

1.6.7.11 TMessageBox

Class to manage the standard 'MessageBox' Windows Dialog. The basic functions to show messages like MsgBox, MsgBoxAlert, MsgBoxYesNo and others are based in this standard Windows component.

Trough this class you can customize the dialogs and you can define the image to be shown.



Hierarchy	Inherits from TWinObject
File name	\source\MessageBox.prg

1.6.7.11.1 TMessageBox:Properties

■ read Only ■ Assignable ■ Design assignable ■ Run-time assignable

Scope	Name	Type	Initial value
■	HelpId	Numeric or Character	NIL
■	aButtons	Array	{}
■	cSoundFile	Character	""
■	cText	Character	""
■	cTitle	Character	""
■	IAppModal	Logic	.T.
■	ICenter	Logic	.F.
■	IHelpButton	Logic	.F.
■	IRigthText	Logic	.F.
■	ISetForeGround	Logic	.F.
■	ITaskModal	Logic	.F.
■	ITopMost	Logic	.F.
■	nDefaultButton	Numeric	mbDEFBUTTON1
■	nResult	Numeric	mrNONE
■	nStyle	Numeric	mbOK

■	nTimeout	Numeric	0
■	nTimeoutResult	Numeric	mrNONE
■	olcon	Object	NIL

1.6.7.11.1.1 TMessageBox:HelpId

Help identifier.

Scope:	Design assignable
Type:	Numeric o Character
Initial value:	Nil

This property allows to identify a control with a unique identifier in the help system. For more information, see also the THelp property.

1.6.7.11.1.2 TMessageBox:aButtons

List with the button text to show in the dialog.

Scope:	Design assignable
Type:	Array
Initial value:	{}

1.6.7.11.1.3 TMessageBox:cSoundFile

Filename and path for the sound to be play when the message is shown.

Scope:	Design assignable
Type:	Character
Initial value:	""

1.6.7.11.1.4 TMessageBox:cText

Message to show. If the timeout is activated with the nTimeout property and it is included the text "%i", it will show the remaining time.

Scope:	Design assignable
Type:	Character
Initial value:	""

1.6.7.11.1.5 TMessageBox:cTitle

Indicates the dialog title.

Scope:	Design assignable
Type:	Character
Initial value:	""

1.6.7.11.1.6 TMessageBox:IAppModal

Indicates the style that avoids to go back to the parent windows until the user answers the dialog.

Scope:	Design assignable
Type:	Logic
Initial value:	.T.

1.6.7.11.1.7 TMessageBox:ICenter

If it is .T. the dialog will be shown centered in the screen.

Scope:	Design assignable
Type:	Logic
Initial value:	.F.

1.6.7.11.1.8 TMessageBox:IHelpButton

If it is .T. the dialog will show a Help button. The HelpId property indicates the help element to be shown.

Scope:	Design assignable
Type:	Logic
Initial value:	.F.

1.6.7.11.1.9 TMessageBox:IRightText

If it is `.T.` the dialog text will be aligned to the right. Only available in Win XP.

Scope:	Design assignable
Type:	Logic
Initial value:	<code>.F.</code>

1.6.7.11.1.10 TMessageBox:ISetForeGround

Shows the dialog always in foreground.

Scope:	Design assignable
Type:	Logic
Initial value:	<code>.F.</code>

1.6.7.11.1.11 TMessageBox:ITaskModal

Same as `IAppModal` but disable all the application windows until the user answers the dialog.

Scope:	Design assignable
Type:	Logic
Initial value:	<code>.F.</code>

1.6.7.11.1.12 TMessageBox:ITopMost

The dialog is created with the Topmost style. It means that will be shown in foreground even if it is not activated.

Scope:	Design assignable
Type:	Logic
Initial value:	<code>.F.</code>

1.6.7.11.1.13 TMessageBox:nDefaultButton

Establishes the button with the focus.

Scope:	Design assignable
Type:	Numeric
Possible values:	mbDEFBUTTON1,mbDEFBUTTON2,mbDEFBUTTON3,mbDEFBUTTON4
Initial value:	mbDEFBUTTON1

1.6.7.11.1.14 TMessageBox:nResult

Indicates the dialog's return value.

Scope:	read Only
Type:	Numeric
Possible values:	mrNONE, mrOK, mrCANCEL, mrABORT, mrRETRY, MrIGNORE, mrYES, mrNO, mrTRYAGAIN, mrCONTINUE
Initial value:	mrNONE

1.6.7.11.1.15 TMessageBox:nStyle

Button combination used by the dialog.

Scope:	Design assignable
Type:	Numeric
Possible values:	-mbOK, MB_OKCANCEL, mbABORTRETRYIGNORE, -mbYESNOCANCEL, mbYESNO, mbRETRYCANCEL, -mbCANCELTRYCONTINUE (Only available in Win XP)
Initial value:	mbOK

1.6.7.11.1.16 TMessageBox:nTimeOut

Time to wait before to close the message automatically. If you need that the dialog shows the remaining time, see the cText property.

Scope:	Design assignable
---------------	-------------------

Type:	Numeric
Initial value:	0

1.6.7.11.1.17 TMessageBox:nTimeOutResult

Default value that returns the dialog if there is a time out and the user has not selected any option.

Scope:	Design assignable
Type:	Numeric
Possible values:	mrNONE, mrOK, mrCANCEL, mrABORT, mrRETRY, MrIGNORE, mrYES, mrNO, mrTRYAGAIN, mrCONTINUE
Initial value:	mrNONE

1.6.7.11.1.18 TMessageBox:olcon

Icon to show.

Scope:	Design assignable
Type:	Object
Possible values:	- File name, Loads an icon from a file in disk - Resource nam, shows an icon from the resource file - The constant mb<Name>, shows an standard icon.
Initial value:	NIL

1.6.7.11.2 TMessageBox Methods

■ Constructor ■ Standard ■ Only after Create()

Typ Name
■ Beep
■ Run Execute

1.6.7.11.2.1 TMessageBox:Beep

Produces a beep.

Type	Standard
-------------	----------

Parameters	<nIcon> Icon that indicates the sound type to play: MB_ICONASTERISK MB_ICONEXCLAMATION MB_ICONHAND MB_ICONQUESTION MB_OK
Return value	<ISuccess> .T. if it executed correctly

1.6.7.11.2.2 TMessageBox:Run

Activates and shows the MessageBox dialog.

Type	Standard
Parameters	None
Return value	<ISuccess> .T. if it executed correctly

1.6.7.11.3 TMessageBox Events

Name	
OnShow	

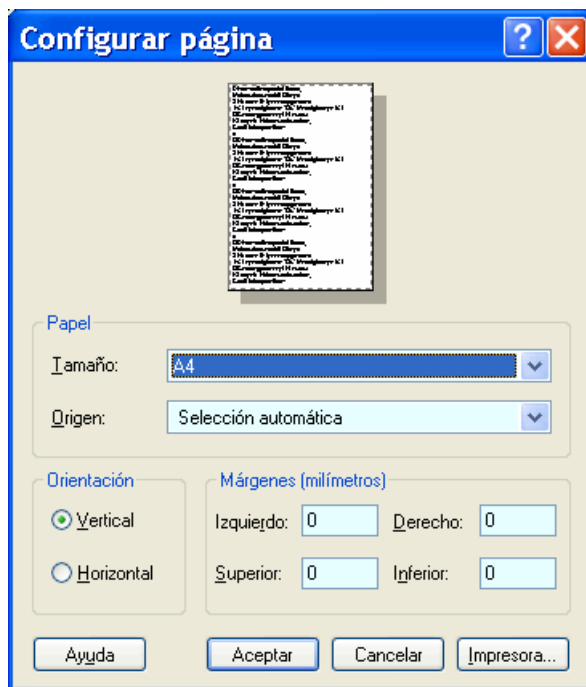
1.6.7.11.3.1 TMeessageBox:OnShow

Evento that is triggered when the dialog is shown.

Parameters	<oSender> :
:	Reference to the object that triggers the event
Return value:	NIL

1.6.7.12 TPageSetupDlg

Class to manage the standard Windows dialog to format print pages.



Description:

The TPageSetupDlg class allows to mane the access to the standard Windows dialog to format the print pages.

Hierarchy	Inherits from TComponent
File Name	\source\PageDlg.prg

1.6.7.12.1 TPageSetupDlg:Properties

■ read Only ■ Assignable ■ Design assignable ■ Run-time assignable

Scope	Name	Type	Initial value
■	aMargins	Array	{ 0, 0, 0, 0 }
■	aMinMargins	Array	{ 0, 0, 0, 0 }
■	aPaperSize	Array	{ 0, 0 }
■	HelpID	Numeric or character	NIL
■	ICenter	Logic	.F.
■	IDefMinMargins	Logic	.F.
■	IInches	Logic	.F.
■	IMargins	Logic	.T.
■	IMinMargins	Logic	.F.
■	INoMargins	Logic	.F.

<input type="checkbox"/>	INoNetwork	Logic	.F.
<input type="checkbox"/>	INoOrientation	Logic	.F.
<input type="checkbox"/>	INoPageDraw	Logic	.F.
<input type="checkbox"/>	INoPaper	Logic	.F.
<input type="checkbox"/>	INoPrinter	Logic	.F.
<input type="checkbox"/>	INoWarning	Logic	.F.
<input type="checkbox"/>	IShowHelp	Logic	.F.
<input type="checkbox"/>	nLastError	Numeric	0
<input type="checkbox"/>	oParent	Object	NIL

1.6.7.12.1.1 TPageSetupDlg:aMargins

Default paper margins when the dialog is open or the selected margins after the dialog is executed.

Scope	Assignable
Type	Array
Initial value	{ 0, 0, 0, 0 }

The coordinates always are indicated in millimeters, unless there is a change in the `lInInches` property.

The array values correspond to the left, right, upper and lower margins.

1.6.7.12.1.2 TPageSetupDlg:aMinMargins

Minimum default paper margins when the dialog is open or the minimum selected margins after the dialog is executed.

Scope	Assignable
Type	Array
Initial value	{ 0, 0, 0, 0 }

The coordinates always are indicated in millimeters, unless there is a change in the `lInInches` property.

The array values correspond to the left, right, upper and lower margins.

1.6.7.12.1.3 TPageSetupDlg:aPaperSize

Default paper size when the dialog is open or the selected value after the dialog is executed.

Scope	Assignable
--------------	------------

Type	Array
Initial value	{ 0, 0 }

The coordinates always are indicated in millimeters, unless there is a change in the `lnInches` property.

The array values correspond to the left, right, upper and lower margins.

1.6.7.12.1.4 TPageSetupDlg:HelpID

Identifier for external Help system request when the push button is pressed.

Scope:	Assignable
Type:	Numeric or character
Initial value:	Nil

For further information consult the `THelp` class and the `OnHelp` and `OnHelpClick` events from `TForm` class.

1.6.7.12.1.5 TPageSetupDlg:ICenter

Centers the dialog in the screen.

Scope	Design assignable (Before to execute the Run method)
Type	Logic
Initial value	.F.

1.6.7.12.1.6 TPageSetupDlg:IDefMinMargins

Establishes as minimum margins the margins indicated in `aMinMargins`.

Scope	Design assignable (Before to execute the Run method)
Type	Logic
Initial value	.F.

1.6.7.12.1.7 TPageSetupDlg:InInches

Indicates that the measure units are described in inches. By default the measure units are millimeters.

Scope	Design assignable (Before to execute the Run method)
Type	Logic
Initial value	.F.

1.6.7.12.1.8 TPageSetupDlg:IMargins

Establishes the margins the indicated in aMargins.

Scope	Design assignable (Before to execute the Run method)
Type	Logic
Initial value	.T.

1.6.7.12.1.9 TPageSetupDlg:IMinMargins

Avoids to accept lower values for the margins specified in aMinMargins.

Scope	Design assignable (Before to execute the Run method)
Type	Logic
Initial value	.F.

1.6.7.12.1.10 TPageSetupDlg:INoMargins

Does not allow to change the margin values.

Scope	Design assignable (Before to execute the Run method)
Type	Logic
Initial value	.F.

1.6.7.12.1.11 TPageSetupDlg!NoNetwork

Hides the "Network" button in the dialog.

Scope	Design assignable (Before to execute the Run method)
Type	Logic
Initial value	.F.

1.6.7.12.1.12 TPageSetupDlg!NoOrientation

Does not allow to change the printer orientation.

Scope	Design assignable (Before to execute the Run method)
Type	Logic
Initial value	.F.

1.6.7.12.1.13 TPageSetupDlg!NoPageDraw

Does not draw a draft page in the dialog.

Scope	Design assignable (Before to execute the Run method)
Type	Logic
Initial value	.F.

1.6.7.12.1.14 TPageSetupDlg!NoPaper

Does not allow to change the paper size and paper origin.

Scope	Design assignable (Before to execute the Run method)
Type	Logic
Initial value	.F.

1.6.7.12.1.15 TPageSetupDlg:INoPrinter

Hides the "Printer" button in the dialog and avoids to change other printer configuration settings from tis dialog.

Scope	Design assignable (Before to execute the Run method)
Type	Logic
Initial value	.F.

1.6.7.12.1.16 TPageSetupDlg:INoWarning

It does not show warnings if there are not printers installed when the dialog is called.

Scope	Design assignable (Before to execute the Run method)
Type	Logic
Initial value	.F.

1.6.7.12.1.17 TPageSetupDlg:IShowHelp

Shows the help icon (a question mark) in the dialog's title.

Scope	Design assignable (Before to execute the Run method)
Type	Logic
Initial value	.T.

1.6.7.12.1.18 TPageSetupDlg:nLastError

Informs the error, if any, when the dialog was shown or during the dialog execution.

Scope	read Only
Type	Numeric
Initial value	0

1.6.7.12.1.19 TPageSetupDlg:oParent

Control's parent form.

Scope	Design assignable (Before to execute the Run method)
Type	Object
Initial value	NIL

1.6.7.12.2 TPageSetupDlg:Methods

■ Constructor ■ Standard ■ Only after Create()

Type	Name
■	Run Execute

1.6.7.12.2.1 TPageSetupDlg:Run

Activates and shows the dialog.

Type	Only after Create()
Parameters	None
Return value	<ISuccess> .T. if the dialog was closed with the "OK" button.

1.6.7.12.3 TPageSetupDlg:Events

Name
OnClose
OnHelp
OnShow

1.6.7.12.3.1 TPageSetupDlg:OnClose

Event that is produced when the dialog is closed.

Parameters	<oSender>:
:	Reference to the object that triggers the event
Return value:	Nil

1.6.7.12.3.2 TPageSetupDlg:OnHelp

Event that is produced when the help button is pushed.

Parameters	<oSender>: A reference to the object that fired the request <HelpId>: HelpId property value of the control or form that request the help <nPosX> Mouse X coordinate when the request proceed from a mouse click <nPosY> Mouse Y coordinate when the request proceed from a mouse click
Return value:	NIL

The help system works as follows:

- The dialog that receives the help request checks that its property HelpId has a value. If Nil, then the request will be sent to its parent. If not Nil, then calls the help system of its container form. If on the form OnHelp event is assigned, then is triggered. If is not assigned then the TApplication help systems is called.
- If the TApplication OnHelp event is assigned, then is triggered. If is not assigned and its property oHelp has a valid THelp object, then it shows the help from the topic HelpId of the control or form that request the help.
- The **oSender** parameter of TForm or TApplication **OnHelp** event is the control that request the help whose HelpId had a value.

1.6.7.12.3.3 TPageSetupDlg:OnShow

Event that is produced when the dialog is shown.

Parameters	<oSender>: Reference to the object that triggers the event
Return value:	Nil

1.6.7.13 TPickIconDlg

Class to manage the standard Windows dialog to select icons.



Description:

The TPickIconDlg class allows access to the standard Windows dialog to select icons.

Hierarchy	Inherits from TComponent
File name	\source\PickIconDlg.prg

1.6.7.13.1 TPickIconDlg:Properties

■ read Only ■ Assignable ■ Design assignable ■ Run-time assignable

Scope	Name	Type	Initial value
■	cFileName	Character	""
■	nCount	Numeric	0
■	nIndex	Numeric	1
■	oParent	Object	NIL

1.6.7.13.1.1 TPickIconDlg:cFileName

File name that contains the icon to be selected.

Scope	Design assignable (Before to execute the Run method)
Type	Character
Initial value	""

1.6.7.13.1.2 TPickIconDlg:nCount

Number of icons in the file.

Scope	read Only
Type	Numeric
Initial value	0

1.6.7.13.1.3 TPickIconDlg:nIndex

Index for the icon that appears preselected when the dialog is open. When the dialog is closed, it has the selected icon.

Scope	Dessign assignable (Before to execute the Run method)
Type	Numeric
Initial value	0

1.6.7.13.1.4 TPickIconDlg:oParent

Parent form where the control belongs to.

Scope	Dessign assignable (Before to execute the Run method)
Type	Object
Initial value	NIL

1.6.7.13.2 TPickIconDlg:Methods

Constructor Standard Only after create()

Typ	Name
<input type="checkbox"/>	Run

1.6.7.13.2.1 TPickIconDlg:Run

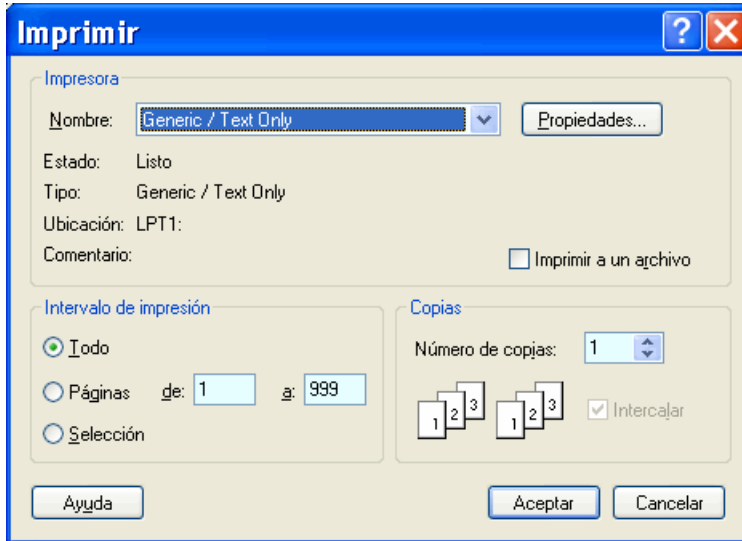
Activates and shows the dialog.

Type	Only after Create
-------------	-------------------

Parameters	None
Return value	<ISuccess> .T. if the dialog was closed with the "OK" button.

1.6.7.14 TPrintDlg

Class to manage the standard Windows dialog to select printer and print.



Description:

The TPrintDlg class allows access to the standard Windows Dialog to select printer and print.

Hierarchy	Inherits from TComponent
File Name	\source\PrintDlg.prg

1.6.7.14.1 TPrintDlg:Properties

■ read Only ■ Assignable ■ Design assignable ■ Run-time assignable

Scope	Name	Type	Initial value
■	hDC	Numeric	0
■	HelpID	Numeric or character	NIL
■	IAllPages	Logic	.F.
■	ICenter	Logic	.F.
■	ICollate	Logic	.F.
■	IHidePrintToFile	Logic	.F.
■	INoPageNums	Logic	.F.
■	INoSelection	Logic	.F.
■	INoWarning	Logic	.F.

■	IPageNums	Logic	.F.
■	IPrintSetup	Logic	.F.
■	IPrintToFile	Logic	.F.
■	IReturnIC	Logic	.F.
■	IReturnDC	Logic	.F.
■	ISelection	Logic	.F.
■	IShowHelp	Logic	.F.
■	nCopies	Numeric	1
■	nFromPage	Numeric	0
■	nLastError	Numeric	0
■	nMaxPage	Numeric	0
■	nMinPage	Numeric	0
■	nToPage	Numeric	0
■	oParent	Object	NIL

1.6.7.14.1.1 TPrintDlg:hDC

Printer device handle. See also the IReturnDC and IReturnIC properties.

Scope	read Only
Type	Numeric
Initial value	0

A device handle is the tool that is used by the Windows API to paint in any device, for example a printer.

1.6.7.14.1.2 TPrintDlg:HelpID

Identifier for external Help system request when the push button is pressed.

Scope:	Assignable
Type:	Numeric or character
Initial value:	Nil

For further information consult the THelp class and the OnHelp and OnHelpClick events from TForm class.

1.6.7.14.1.3 TPrintDlg:lAllPages

Selects the option "All" for the printer range when the dialog is open or the selected value after the dialog is executed.

Scope	Assignable
--------------	------------

Type	Logic
Initial value	.T.

1.6.7.14.1.4 TPrintDlg:ICenter

Centers the dialog in the screen.

Scope	Design assignable (Before to execute the Run method)
Type	Logic
Initial value	.F.

1.6.7.14.1.5 TPrintDlg:ICollate

Selects the "Collate" option when the dialog is open or the selected value after the dialog is executed.

Scope	Assignable
Type	Logic
Initial value	.F.

1.6.7.14.1.6 TPrintDlg:IHidePrintToFile

Hides the button to select the print to file option.

Scope	Assignable
Type	Logic
Initial value	.F.

1.6.7.14.1.7 TPrintDlg:INoPageNums

Disables the "Pages" option in the printer range.

Scope	Assignable
Type	Logic
Initial value	.F.

1.6.7.14.1.8 TPrintDlg:INoSelection

Disables the "Select" option in the printer range.

Scope	Assignable
Type	Logic
Initial value	.F.

1.6.7.14.1.9 TPrintDlg:INoWarning

Does not show any warning if there are not installed printers when the dialog is called.

Scope	Assignable
Type	Logic
Initial value	.F.

1.6.7.14.1.10 TPrintDlg:IPageNums

Selects the "Page" option for the printer range when the dialog is open or the selected value after the dialog is executed.

Scope	Assignable
Type	Logic
Initial value	.F.

1.6.7.14.1.11 TPrintDlg:IPrintSetup

Shows the "Printer Setup" dialog instead the printout.

Scope	Assignable
Type	Logic
Initial value	.F.

1.6.7.14.1.12 TPrintDlg:IPrintToFile

Selects the option "Print to file" when the dialog is open or the selected value after the dialog is executed.

Scope	Assignable
--------------	------------

Type	Logic
Initial value	.F.

1.6.7.14.1.13 TPrintDlg:IReturnDC

Returns the device handle in the hDC property with the selected printer options.

Scope	Assignable
Type	Logic
Initial value	.F.

1.6.7.14.1.14 TPrintDlg:IReturnIC

Returns the device handle in the hDC property for the selected options.

Scope	Assignable
Type	Logic
Initial value	.F.

1.6.7.14.1.15 TPrintDlg:ISelection

Selects the "Range" or "Selection" option for the printer range when the dialog is open or the selected value after the dialog is executed.

Scope	Assignable
Type	Logic
Initial value	.F.

1.6.7.14.1.16 TPrintDlg:IShowHelp

Shows the help icon (a question mark) in the dialog's title.

Scope	Assignable
Type	Logic
Initial value	.T.

1.6.7.14.1.17 TPrintDlg:nCopies

Default number of copies when the dialog is open or the selected value after the dialog is executed.

Scope	Assignable
Type	Numeric
Initial value	1

1.6.7.14.1.18 TPrintDlg:nFromPage

Default starting page when the dialog is open or the selected value after the dialog is executed.

Scope	Assignable
Type	Numeric
Initial value	0

1.6.7.14.1.19 TPrintDlg:nLastError

Indicates the error, if any, when the dialog is shown or during the dialog's operation.

Scope	read Only
Type	Numeric
Initial value	0

1.6.7.14.1.20 TPrintDlg:nMaxPage

Maximum value that can be used as end page.

Scope	Assignable
Type	Numeric
Initial value	0

1.6.7.14.1.21 TPrintDlg:nMinPage

Minimum value that can be used as starting page.

Scope	Assignable
Type	Numeric

Initial value 0

1.6.7.14.1.22 TPrintDlg:nToPage

Default end page when the dialog is open or the selected value is selected after the dialog is executed.

Scope	Assignable
Type	Numeric
Initial value	0

1.6.7.14.1.23 TPrintDlg:oParent

Control's form.

Scope	Design assignable (Before to execute the Run method)
Type	Object
Initial value	NIL

1.6.7.14.2 TPrintDlg:Methods

Constructor Standard Only after Create()

Typ	Name
<input type="checkbox"/>	Run Execute

1.6.7.14.2.1 TPrintDlg:Run

Activates and shows the dialog.

Type	Only after create
Parameters	None
Return value	<ISuccess> .T. if the user closes the dialog with the "OK" button.

1.6.7.14.3 TPrintDlg:Events

Name
OnClose
OnHelp
OnShow

1.6.7.14.3.1 TPrintDlg:OnClose

Event that is produced when the dialog is closed.

Parameters	<oSender> :
:	Reference to the object that triggers the event
Return value:	Nil

1.6.7.14.3.2 TPrintDlg:OnHelp

Event that is produced when the help button is pushed.

Parameters	<oSender> :
:	A reference to the object that fired the request
	<HelpId> :
	HelpId property value of the control or form that request the help
	<nPosX> :
	Mouse X coordinate when the request proceed from a mouse click
	<nPosY> :
	Mouse Y coordinate when the request proceed from a mouse click
Return value:	NIL

The help system works as follows:

- The dialog that receives the help request checks that its property HelpId has a value. If Nil, then the request will be sent to its parent. If not Nil, then calls the help system of its container form. If on the form OnHelp event is assigned, then is triggered. If is not assigned then the TApplication help systems is called.
- If the TApplication OnHelp event is assigned, then is triggered. If is not assigned and its property oHelp has a valid THelp object, then it shows the help from the topic HelpId of the control or form that request the help.
- The **oSender** parameter of TForm or TApplication **OnHelp** event is the control that request the help whose HelpId had a value.

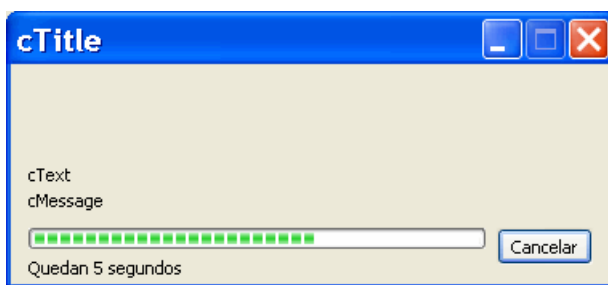
1.6.7.14.3.3 TPrintDlg:OnShow

Event that is produced when the dialog is shown.

Parameters	<oSender>:
:	Reference to the object that triggers the event
Return value:	Nil

1.6.7.15 TProgressDlg

This class encapsulates the common ProgressDlg Windows dialog.

**Description:**

It shows a progress dialog that normally includes an AVI video type. The dialog is **only** shown when the process takes more than a couple of seconds.

Hierarchy	Inherits from TComponent
File name	\source\ProgressDlg.prg

1.6.7.15.1 TProgressDlg:Properties

■ read Only ■ Assignable ■ Design assignable ■ Run-time assignable

Scope	Name	Type	Initial value
■	cMessage	Character	""
■	cMessageCancel	Character	""
■	cText	Character	""
■	cTitle	Character	""
■	IAutoTime	Logic	.F.
■	ICenter	Logic	.T.
■	ICompactPath	Logic	.F.
■	IModal	Logic	.T.
■	INoCancel	Logic	.F.
■	INoMinimize	Logic	.F.

❑	INoTime	Logic	.F.
❑	INoProgress	Logic	.F.
■	nCommonAVI	Numeric	aviNONE
■	nMax	Numeric	100
■	nResourceAVI	Numeric	aviNONE
■	nValue	Numeric	0

1.6.7.15.1.1 TProgressDlg:cMessage

Informative text that describes the process status.

Scope	Assignable
Type	Character
Initial value	""

1.6.7.15.1.2 TProgressDlg:cMessageCancel

Message shown when the process is canceled.

Scope	Assignable
Type	Character
Initial value	""

1.6.7.15.1.3 TProgressDlg:cText

Informative text that describes the process that is executing.

Scope	Assignable
Type	Character
Initial value	""

1.6.7.15.1.4 TProgressDlg:cTitle

Dialog title.

Scope	Assignable
Type	Character
Initial value	""

1.6.7.15.1.5 TProgressDlg:IAutoTime

Calculates and shows the estimated process time.

Scope	Design assignable
Type	Logic
Initial value	.F.

1.6.7.15.1.6 TProgressDlg:ICenter

Shows the dialog centered

Scope	Design assignable
Type	Logic
Initial value	.T.

1.6.7.15.1.7 TProgressDlg:ICompactPath

In case that the cText and cMessage properties have a long file name, it adjusted to show it correctly.

Scope	Design assignable
Type	Logic
Initial value	.F.

1.6.7.15.1.8 TProgressDlg:IModal

If .T. the dialog is modal, it means that the control will not back to the application until the dialog is closed.

Scope	Design assignable
Type	Logic
Initial value	.T.

1.6.7.15.1.9 TProgressDlg:INoCancel

Prevents to close the dialog while a there is a process running.

Scope	Design assignable
--------------	-------------------

Type	Logic
Initial value	.F.

1.6.7.15.1.10 TProgressDlg:INoMinimize

Prevents to minimize the dialog while a process is running.

Scope	Design assignable
Type	Logic
Initial value	.F.

1.6.7.15.1.11 TProgressDlg:INoTime

It does not show the default text together with the estimated time.

Scope	Design assignable
Type	Logic
Initial value	.F.

1.6.7.15.1.12 TProgressDlg:INoProgress

Does not show the progress bar.

Scope	Design assignable
Type	Logic
Initial value	.F.

1.6.7.15.1.13 TProgressDlg:nCommonAVI

Standard Windows animation identifier. The animations with the aviXP prefix are Windows XP specific.

Scope	Assignable
Type	Numeric
Initial value	aviNONE
Possible values	aviNONE, aviFIND, aviFINDFILES, aviFINDCOMPUTER, aviCOPYFILES, aviCOPYFILE, aviRECYCLEFILE, aviDELETEFILE, aviXPINTERNETSEARCH, aviXPDOWNLOADFILE

1.6.7.15.1.14 TProgressDlg:nMax

Maximum range value indicated in the progress bar. The default value is 100.

Scope	Assignable
Type	Numeric
Initial value	100

1.6.7.15.1.15 TProgressDlg:nResourceAVI

Animation identifier linked in the application resources.

Scope	Assignable
Type	Numeric
Initial value	aviNONE

1.6.7.15.1.16 TProgressDlg:nValue

Current progress bar position.

Scope	Assignable
Type	Numeric
Initial value	0

1.6.7.15.2 TProgressDlg:Methods

■ Constructor ■ Standard

Typ	Name
■	Create
■	Destroy
■	Execute
■	IsCancel
■	New
■	Reset
■	Run
■	Stop

1.6.7.15.2.1 TProgressDlg:Create

Class constructor.

Type	Constructor
Parameters	<oParent> Reference to the parent form
Return value	Self Reference (Self)

1.6.7.15.2.2 TProgressDlg:Destroy

Class destroyer.

Type	Standard
Parameters	None
Return value	NIL

1.6.7.15.2.3 TProgressDlg:Execute

Activates and shows the dialog and triggers the OnStart event.

Type	Standard
Parameters	None
Return value	<ISuccess> .T. if the dialog was executed successfully.

1.6.7.15.2.4 TProgressDlg:IsCancel

Checks if the user has closed the dialog.

Type	Standard
Parameters	None
Return value	<ICancel> .T. if the process has been canceled

1.6.7.15.2.5 TProgressDlg:New

Class constructor.

Type	Constructor
Parameters	<oParent> Reference to the parent form
Return value	Self reference (Self)

1.6.7.15.2.6 TProgressDlg:Reset

If the process will not start immediately after the dialog is activated and if the estimated time will be calculated automatically (see the IAutoTime property), you should call this method to recalculate it.

Type	Standard
Parameters	None
Return value	<ISuccess> .T. if the method was executed correctly.

1.6.7.15.2.7 TProgressDlg:Stop

Finishes the dialog and triggers the OnStop event.

Type	Standard
Parameters	None
Return value	<ISuccess> .T. if the dialog was closed correctly

1.6.7.15.3 TProgressDlg:Events

Name
OnCancel
OnClose
OnReset
OnShow

1.6.7.15.3.1 TProgressDlg:OnCancel

Event that is triggered when the process is canceled.

Parameters	<oSender> Reference to the object that triggers the event
Return value	NIL

1.6.7.15.3.2 TProgressDlg:OnClose

Event that is triggered when the dialog is closed.

Parameters	<oSender> Reference to the object that triggers the event
Return value	NIL

1.6.7.15.3.3 TProgressDlg:OnReset

Event that is triggered when the process is reset.

Parameters	<oSender> Reference to the object that triggers the event
Return value	NIL

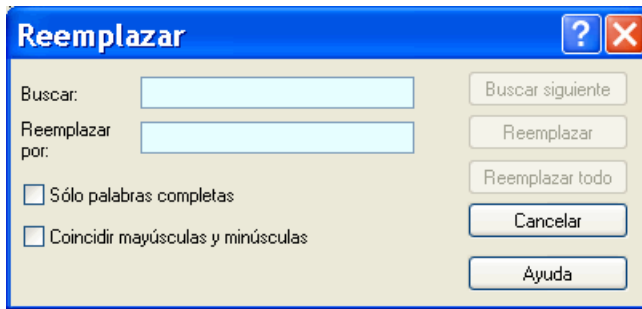
1.6.7.15.3.4 TProgressDlg:OnShow

Event that is triggered when the dialog is shown.

Parameters	<oSender> Reference to the object that triggers the event
Return value	NIL

1.6.7.16 TReplaceDlg

This class encapsulates the common ReplaceText Windows dialog.



Description:

The TReplaceDlg class allows to manipulate the common ReplaceText Windows dialog.

Hierarchy Inherits from TComponent
File name \source\ReplaceDlg.prg

1.6.7.16.1 TReplaceDlg:Properties

■ read Only ■ Assignable ■ Design assignable ■ Run-time assignable

Scope	Name	Type	Initial value
■	cReplace	Character	""
■	cText	Character	""
■	HelpID	Numeric or character	NIL
■	ICenter	Logic	.F.
■	IMatchCase	Logic	.F.
■	INoCase	Logic	.F.
■	INoWholeWord	Logic	.F.
■	IShowHelp	Logic	.T.
■	IWholeWord	Logic	.F.

1.6.7.16.1.1 TReplaceDlg:cReplace

Initial test to replace.

Scope	Assignable
Type	Character
Initial value	""

1.6.7.16.1.2 TReplaceDlg:cText

Initial text to search.

Scope	Assignable
Type	Character
Initial value	""

1.6.7.16.1.3 TReplaceDlg:HelpID

Identifier for external Help system request when the push button is pressed.

Scope:	Assignable
Type:	Numeric or character
Initial value:	Nil

For further information consult the THelp class and the OnHelp and OnHelpClick events from TForm class.

1.6.7.16.1.4 TReplaceDlg:ICenter

Displays the dialog centered.

Scope	Design assignable
Type	Logic
Initial value	.F.

1.6.7.16.1.5 TReplaceDlg:IMatchCase

No case sensitivity.

Scope	Assignable
Type	Logic
Initial value	.F.

1.6.7.16.1.6 TReplaceDlg:INoCase

Hides the possibility of case sensitivity.

Scope	Assignable
--------------	------------

Type	Logic
Initial value	.F.

1.6.7.16.1.7 TReplaceDlg:INoWholeWord

Hides the possibility of select complete words.

Scope	Assignable
Type	Logic
Initial value	.F.

1.6.7.16.1.8 TReplaceDlg:IShowHelp

Shows the help icon (a question mark) in the dialog's title.

Scope	Assignable
Type	Logic
Initial value	.T.

1.6.7.16.1.9 TReplaceDlg:IWholeWord

Searches only complete words.

Scope	Assignable
Type	Logic
Initial value	.F.

1.6.7.16.2 TReplaceDlg:Methods

Constructor Standard

Type	Name
<input type="checkbox"/>	Run Execute

1.6.7.16.2.1 TReplaceDlg:Run

Activates and shows the ReplaceText dialog.

Type	Standard
-------------	----------

Parameters	None
Return value	<ISuccess> .T. was executed successfully. Otherwise, .F.

1.6.7.16.3 TReplaceDlg:Events

Name	
OnClose	
OnFind	
OnHelp	
OnReplace	
OnReplaceAll	

1.6.7.16.3.1 TReplaceDlg:OnClose

Event that is triggered when the dialog is closed.

Parameters	<oSender> Reference to the object that triggers the event.
Return value	NIL

1.6.7.16.3.2 TReplaceDlg:OnHelp

Event that is produced when the help button is pushed.

Parameters	<oSender>: : A reference to the object that fired the request <HelpId>: HelpId property value of the control or form that request the help <nPosX> Mouse X coordinate when the request proceed from a mouse click <nPosY> Mouse Y coordinate when the request proceed from a mouse click
Return value:	NIL

The help system works as follows:

- The dialog that receives the help request checks that its property HelpId has a value. If Nil, then the request will be sent to its parent. If not Nil, then calls the help system of its container form. If on the form OnHelp event is assigned, then is triggered. If is not assigned then the TApplication

help systems is called.

- If the TApplication OnHelp event is assigned, then is triggered. If is not assigned and its property oHelp has a valid THelp object, then it shows the help from the topic HelpId of the control or form that request the help.
- The **oSender** parameter of TForm or TApplication **OnHelp** event is the control that request the help whose HelpId had a value.

1.6.7.16.3.3 TReplaceDlg:OnFind

Event that is triggered every time that the user clicks the "Search" button from the dialog.

Parameters	<oSender> Reference to the object that triggers the event. <cText> Text to search <IWholeWord> Search complete words. <IMatchCase> No case sensitivity
Return value	NIL

1.6.7.16.3.4 TReplaceDlg:OnReplace

Event that is triggered every time that the user clicks the "Replace" button from the dialog.

Parameters	<oSender> Reference to the object that triggers the event. <cText> Text to search <cReplace> Text to replace cText <IWholeWord> Search complete words. <IMatchCase> No case sensitivity
Return value	NIL

1.6.7.16.3.5 TReplaceDlg:OnReplaceAll

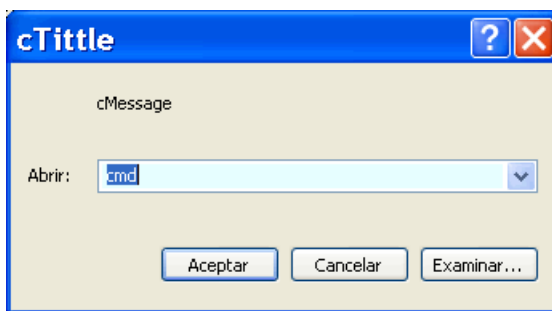
Event that is triggered when the user clicks the "Replace all" button from the dialog.

Parameters	<oSender>
-------------------	------------------------

	Reference to the object that triggers the event. <cText> Text to search <cReplace> Text to replace cText <IWholeWord> Search complete words. <IMatchCase> No case sensitivity
Return value	NIL

1.6.7.17 TRunFileDialog

Class to manage the standard Windows dialog to execute programs.



Description:

The TRunFileDialog allows access to the standard Windows dialog to execute programs.

Hierarchy	Inherits from TWinObject
File name	\source\RunFileDialog.prg

1.6.7.17.1 TRunFileDialog:Properties

Scop e	Name	Type	Initial value
<input type="checkbox"/>	cDirectory	Character	""
<input type="checkbox"/>	cMessage	Character	""
<input type="checkbox"/>	cTitle	Character	""
<input type="checkbox"/>	INoBrowse	Logic	.F.
<input type="checkbox"/>	ICalcDirectory	Logic	.T.
<input type="checkbox"/>	INoDefault	Logic	.F.
<input type="checkbox"/>	INoLabel	Logic	.F.
<input type="checkbox"/>	INoSeparateMem	Logic	.F.
<input type="checkbox"/>	olcon	Object	NIL

1.6.7.17.1.1 TRunFileDialog:cDirectory

Default folder or directory to search for the file to be executed.

Scope	Design assignable (before Run or Execute)
Type	Character
Initial value	""

1.6.7.17.1.2 TRunFileDialog:cMessage

Informative dialog text.

Scope	Design assignable (before Run or Execute)
Type	Character
Initial value	""

1.6.7.17.1.3 TRunFileDialog:cTitle

Indicates the dialog's title.

Scope	Design assignable (before Run or Execute)
Type	Character
Initial value	""

1.6.7.17.1.4 TRunFileDialog:IHideBrowse

Hides the "Browse" button.

Scope	Design assignable (before Run or Execute)
Type	Logic
Initial value	.F.

1.6.7.17.1.5 TRunFileDialog:IUseDefaultDirectory

Gets the default directory from cDirectory.

Scope	Design assignable (before Run or Execute)
--------------	---

Type	Logic
Initial value	.T.

1.6.7.17.1.6 TRunFileDialog:INoDefault

It does not show the last selected file in the dialog.

Scope	Design assignable (before Run or Execute)
Type	Logic
Initial value	.F.

1.6.7.17.1.7 TRunFileDialog:INoLabel

It does not show the message in the dialog.

Scope	Design assignable (before Run or Execute)
Type	Logic
Initial value	.F.

1.6.7.17.1.8 TRunFileDialog:INoSeparateMem

It does not show separate mark to execute in a separate memory space. This property only has effect in WinNT or beyond.

Scope	Design assignable (before Run or Execute)
Type	Logic
Initial value	.F.

1.6.7.17.1.9 TRunFileDialog:olcon

Tlcon object to personalize the dialog.

Scope	Design assignable (before Run or Execute)
Type	Object
Initial value	NIL

1.6.7.17.2 TRunFileDlg:Methods

Constructor
 Standard
 Only after Create()

Typ Name
<input checked="" type="checkbox"/> Run Execute

1.6.7.17.2.1 TRunFileDlg:Run

Activates and shows the RunFileDlg dialog.

Type	Standard
Parameters	None
Return value	<ISuccess> .T. if the dialog was executed successfully

1.6.7.17.3 TRunFileDlg:Events

Name
OnExecute

1.6.7.17.3.1 TRunFileDlg:OnExecute

Event that is produced after a file has been selected and determines if the file will be executed or not.

Parameters	<oSender> Object that triggers the event (Self). <cFile> Selected file <cDirectory> File path <nShow> Application mode: smNORMAL: Normal Window smMAXIMIZE: Maximized window smMINIMIZE: Minimized window
Return value:	<nAction> RF_OK: Executed the action RF_CANCEL: The action is not executed and the dialog is closed RF_RETRY: The action is not executed and the dialog continues.

1.6.8 Others

1.6.8.1 TInplaceEdit

This class represents a Windows Edit control when it is embedded inside other control (for example, a combo box).

Description:

This class represents a Windows Edit control when it is embedded inside other control (for example, a combo box).

It is not directly used.

Hierarchy Inherits from TControl
File name \source\InplaceEdit.prg

1.6.8.1.1 TInplaceEdit:Methods

■ Constructor ■ Standard

Typ	Name
■	Create

1.6.8.1.1.1 TInplaceEdit:Create

Class constructor. Creates the object and assigns the control handle to it.

Type	Constructor
Parameters	<oParent> Control parent form <Handle> Embedded control handle
Return value	<Self> Self reference

1.7 DataControls

Enter topic text here.

1.7.1 DataControls introduction

The data controls are not more than specialized controls to be used with databases, either DBF files or SQL engines. Every DataControl control type is an inherit Xailer's control with added functionality to interact with data bases.

The DataControls will be used together with the data management components TDataSource and

TDataSet.

A TDataSource component will provide all the needed functionality to connect with the data origin. In the case of DBF files, its functionality is minimized due it will only have properties to indicate the RDD type and the root directory where are located all the files to be open. However, in other data types like the ODBC access, the TDataSource object encapsulates more information like the name, connection code or the DNS (Data Source Name).

The TDataSet component allows to manage records/rows data collections. In the case of DBF files, the TDataSet component its like a table. However, in ODBC database connections it is like a SELECT command. Through the TDataSet it is possible to navigate for all the records/rows collection, add new records, delete them and any other typical database management operation.

In the traditional Xbase system, its needed to use only one instruction USE (cFile) to open a database file. In Xailer, using the DataControls, it is needed to instantiate at least 2 objects: one for the datasource that indicates the data origin and other for the DataSet table to be open.

The TDataSet class allows to manage any DBF table using its properties and methods. You don't need to remember its alias name. You don't need to use a Select() operation neither. For the other side, all the basic database functions like DbGoto(), Recno(), etc. have their equivalent in the TDataSet as a method. You will find that in several cases, Xailer offers added functionality in those basic functions.

The easiest way to use the TDataSource and TDataSet components is putting them in a form. In future Xailer releases it might be that we include an especial form named DataModule that is a components container that will be visible inside the application. In this way all the components that are in the DataModule can be accessible from any other form.

The TDataSource and TDataSet can be used directly in the programming code:

```
oDataSource := TDataSource():New( oForm, cConnect )
```

To add a table in the TDataSource :

```
oDataSource.NewDataSet( cName, cProcess )
```

Where cName is the table name and cProcess is an process optional name.

In the case of DBF files, cName corresponds with the table name, but the root directory that indicates the table's origin, is specified in the TDataSource **cConnect** property, and this directory is located in "." (the same directory where the application is located). Remember that is not the current directory, but the directory where the application is being executed. Then, if you want to access a table like 'clients.dbf' located in the App\Data directory (where App is the application directory), then the configuration would be:

```
oDataSource:cConnect := ".\Data"  
oDataSet:cName := "Clients.dbf"
```

The **cProcess** property indicates the process linked to the table. The TDataSource contains all the TDataSet that have been open through it, but if you need you can differentiate between the TDataSet inside the same TDataSource establishing process names in the moment you create the TDataSource. For example, you can have a unique TDataSource for all the application, and then multiple processes for the different operations. For example, the process 'Invoices' needs to open 5 Datasets. To assign the same process to the 5 Datasets you can do it with the **cProcess** parameter or you can establish a process name that is going to open the new Datasets with the

DataSource:NewProcess('Invoices') method. In the moment that you have incorporated several TDataSet in the same process, you can manage them like open, close and destroy block Dataset operations. See also the TDataSource OpenProcess(), CloseProcess() and DelProcess() methods for more information.

The TDataSet class covers all the typical database operations. Its properties and methods are very similar to the DBF functions to manage DBF files. One of the most important TDataSet properties that make it atypical, allows to access the table fields by its name, like a class' property. For example, to access the 'Code' field, you can use the following expression: '**oDataSet:Code**'

Every TDataSet field can also be treated like an TDataField class object. The TDataField class covers all the properties regarding tables fields or resulting columns from an Select operation: like size, type, name, etc. To have access to an TDataSet object field, you can use the **oFieldByName**(cName) method or you can access directly to the object name array **aFields**.

The TDataSet class includes big improvements in the traditional DBF tables from any Xbase languages like:

- **Triggers**

The triggers are events that are produced in typical database maintain operations that the programmer can capture to execute some code every time that is produced certain operation, like an append operation, for example. There are triggers for append, edit, open and close operations. The append, delete and edit operations have an additional trigger **before** the operation is executed and if it returns .F. it allows to cancel the append, delete or edit operation.

- **Calculated fields**

The calculated fields are the result of an expression that might involve other fields form the table. A calculated field can be 'Name + " " + LastName and we can define it in the following calculated field:

```
oDataSet:AddCalcField( "FullName", ;
                    { |oField| oField:DataSet:Name + ;
                      oField:DataSet:LastName }
```

It might sound estrange, but the calculated fields can be even saved and can overload the **OnSetValue** event from the TDataField object. In this event you can make the assignments you want in the original fields.

- **Simulated members to acces fields:**

Is possible to access any DataSet field value using its name as a member of the class. For example:

```
oDataSet:Code := 1
```

You can also access the same way to calculated fields, for example:

```
Msginfo( oDataSet:FullName )
```

Is also possible to access any DataSet TDatafield object using its name as a member of the class with the parameter **dsOBJECT**. For example:

```
Msginfo( oDataSet:Code( dsOBJECT ):Type )
```


- **Edit buffer:**

Any Xbase programmer is familiar with the local or private variables to allow the data input or to make append or edit operations. To do that, you need to create all the temporary variables first, after that you need to assign an initial value (like blanks or the current field value, in the case of an edit operation). and then if the user has confirmed the operation, you need to save the values from those variables to the fields in the table. There is a lot of repetitive code.

Later, with the OOP engines in the Clipper environment, some classes were developed to manage DBF tables. The main feature they offered was the possibility to avoid the temporary variables, creating DATAs with the same field name but they were pointing to a **buffer** instead of point directly to the table. It was needed to use the **Load()** method, available in the same class, to update this **buffer**. And it was another method available (**Save()**) to save the information from the buffer to the table.

The Xailer's TDataset improves this scheme adding a system where the DATA with field names can behave either, like data coming from the table without the need of a **Load()** operation, or like a **buffers** that allow the append and edit operations through them. To be able to provide this functionality, all the append and edit operation are executed in 2 steps. The first step indicates that the TDataset is in edit mode through and append operation (**AdNew()** method) or an edit operation (**Edit()** method). In that moment, the **buffers** take the current field value or they become blank (in the case of new records) and the DATA with field names don't point directly to the tables, but they point to the **buffers** instead. The second step specifies how to exit from the edit mode. And it can be either, calling the **Update()** method to save all the changes, or calling to the **Cancel()** method to cancel all the changes. In that moment, the DATA with field names will point again to the table fields.

Important note:

When you are in **AddNew** mode, independently of the Recno position in the dataset, the FieldGet method and the field access through its name, by default, returns the internal buffer value used for the append but not its field value since its record position. You must use the dsFIELD parameter to access to the field value on those cases.

```
oDataSet:Code( dsFIELD ) // Retrieves the field value from the table
```

In this moment we have reviewed only the non visual components for database access. And they add a lot of functionality, but they will be limited if they are not used together with the DataControls. The DataControls are a visual control collection that add database management functionality. The main **DataControl** advantages are:

- They don't need to create any temporary variable
- They don't need to assign for append or edit operations any **'buffer'** variable
- They don't need to execute multiple 'replace' operations to save information.
- They don't need to update controls manually (refresh) when the TDataset position changes. It is done automatically.
- The controls will be read only as far as the TDataset is not in edit mode. When the TDataset enters to edit mode, the DataControls can be edited as any other control type.

The DataControl allows to make a complete table maintenance only establishing properties and

without the need to write a single line of code.

To offer all this important functionality, all the DataControls add some new properties in reference to their proceeding controls and they are:

- The TDataset object
- The TDataField object that indicates the field to be used. This field can be a calculated field.

These two properties can be assigned during design time. If the TDataset is open, Xailer will show all the possible fields to be linked in the **oDataField** property, or if you prefer, you can input the field name manually as a label.

Note that the **oDataField** property can contain directly a label with the filename instead a TDataField object. When the TDataset depending **DataControl** is open and it confirms that exist a field with that label, then the oDataFields value will have a reference to an TDataField object instead a label.

Once established the relations among the table DataControls with its TDataset object, it is only needed one instruction to enters to edit mode: '**oDataSet:Edit()**' and other to save all the changes: '**oDataSet:Update()**'.

An special DataControl case is the . In this case, the control is not really a control that can be linked to a field table, but it is really a TDataset manager, allowing that TDBBrowse the navigation can be done directly over the TDataset and however, any change in the TDBBrowse affects and updates any other **DataControl** that could be linked with the TDataset. Every TDBBrowse can be linked with a TDataset fields through its **oDataField** property.

However when the TDBBrowse enters to edit mode pressing the <return> key or double clicking on it, it use an internal TGridEdit control that behaves like another **DataControl**. Indeed, when the TDBBrowse enters to edit mode, it calls automatically to the associated TDataset **Edit()** method and therefore the TGridEdit is not read only and can be modified. To finish the edit process, it is needed to call to the **TDataSet:Update()** method from the TDBBrowse **OnPostEdit()** event .

It can be that all this automatic functionality might impact the flexibility to manage complex edit operations and makes the system non usable. We have working very hard to avoid this issue and we think that the flexibility level is very high thanks to the calculated fields in the TDataset and the possibility to modify individually the way that the data are saved and recovered through the TDataField object events **OnGetValue()** and **OnSetValue()**.

For example, lets suppose 4 typical fields in a banking operation: Bank code, branch bank, control digit and account number. We can use the following structure to show it in the browse:

BCBC BBBB CD ANANANANAN

To show it in the browse, we can create a calculated field:

```
oField := oDS:AddCalcField( "FullAccount", ;
    { | oField | oDS:BankCode + " " + oDS:BranchBank
    + " " + + ;
    oDS:ControlDigit + " " + oDS:AccountNumber }
```

And we can use the label 'FullAccount' as field name.

Let's suppose that we need to edit this field directly in the browse. Locally the FullAccount field does not exist. However, we can use the **OnSetValue()** event and we can indicate how we want to save it:

```
oField:OnSetValue := { |oField, Value | Saveit( oField, Value ) }

Function Saveit( oField, Value )

    WITH OBJECT oField:oDataSet
        :BankCode      := Left( Value, 4 )
        :BranchBank    := Substr( Value, 6, 4 )
        :ControlDigit  := Substr( Value, 11, 2 )
        :AccountNumber := Substr( Value, 14, 10 )
    END WITH

RETURN NIL
```

For more information check the DataControls components and controls.

1.7.2 Enhancements on Xailer 7

When DataControls are used, Xailer is able to access the different fields of a table or cursor as if they were members of the TDataSet class itself. An example:

```
WITH OBJECT oDataset
    :Edit()
    :Codigo := 1 // field
    :Nombre := "Ignacio" // field
    :Update()
END WITH
```

When the field name matches an actual member of the dataset itself, access to the field is not possible, as the member of the **TDataSet** class takes precedence. To solve this problem, until now you could either use the TDataSet:FieldPut() method directly or retrieve the TDataField object with the **oFieldByName()** method and then assign the value to it.

As of Xailer 7 we have a more elegant solution to access any field of the **TDataSet** even if its name coincides with a member of the class and consists of using the following instruction: **TDataSet:!
Field**

Notice how the only difference is the **admiration** after the '!'. Also, this way you, as a programmer, will know if you are accessing a field or a member of the class, which will make your code much more readable.

Another important improvement: In Xailer 7 we have made a little nod to NoSQL databases. In NoSQL databases, data schemas are dynamic, that is, there is no fixed table structure and data consistency is less important. In Xailer 7 we wanted to include some of its qualities and now any dataset, even a DBF table, can have the ability to save any field in its table, even if it doesn't even exist in the table. It is only necessary the existence of a field in the database named **MoreData**, which logically should be type BLOB or MEMO.

To access or set the value of this virtual field, you only have to use the syntax: **oDataset:!
Field**.

That is, we must include the character '!' that we have talked about previously. Internally all information is saved in JSON format within that field and therefore the 'MoreData' field can be searched and even edited from your preferred database editor. Our SQLite table editor already supports the JSON field type, so in this case, it is preferable to define the field with that type.

The possibilities of this new system are multiple. As an example:

- Save the full addresses of all your branches in a client table (it is not necessary to create a client-delegation table)
- Save in a table of articles a complete list of all the offers you have by period or quantity
- Avoid having to modify a table due to the need to include a new field

One of the great advantages of this new system is that you can save any type of data in the table, including matrices and Hash objects.

However, there are limitations, which are insurmountable, such as:

- Virtual fields cannot be assigned to the 'oField' property of a datacontrol
- You cannot perform direct searches on virtual fields. You will need to search internally in the 'MoreData' field

1.7.3 Data Sources

1.7.3.1 TDataSource

Class to manage data origins. This class is responsible to make the connections with the data engine. TDtaSource is an abstract class, it means, it does not make any specific job. However is the base for other classes that inherit from it, like CdxDataSource to connect with CDX tables, NtxDataSource to connect with NTX tables, AdsDataSource to connect with Advantage or OdbcDataSource to connect with ODBC engines.

Description:

This class establishes the properties, methods and events that must have any DataSource object created in Xailer. Any other DataSource class type inherits from this class.

The methods, properties and events for this class must be developed in lower classes due several of them are defined in abstract form.

Several of the properties and methods for this class, even being public, are not documented because they are treated with more attention in lower hierarchy classes.

See also 'Introduction to DataControls' for more information.

Hierarchy	Inherits from TComponent
File Name	\source\Datasource.prg
See also	TDataSet, DataControls

1.7.3.1.1 TDataSource:Properties

■ read Only ■ Assignable ■ Design assignable ■ Run-time assignable

Scope	Name	Type	Initial value
■	aDataSet	Array	{}
■	cConnect	Character	""

■	IAbortOnErrors	Logic	.F.
■	IConnected	Logic	.F.
■	IDisplayErrors	Logic	.T.
■	nTimeOut	Numeric	30

1.7.3.1.1.1 TDataSource:aDataSet

TDataSet objects array with all the tables or cursors open. A TDataSet object represents a table (in DBF environment) or a cursor (ODBC).

Scope	Assignable
Type	Array
Initial value	{}

1.7.3.1.1.2 TDataSource:cConnect

String connection with the database. In the case of DBF tables **cConnect** indicates the tables default directory path.

Scope	Assignable
Type	Character
Initial value	""

Important note:

Most of the classes that inherit from this TDataSource base class will modify the value of this property with a more complete expression when the connection is made. If you do not want that this property is altered you should set the value of the IConnected property to true on run-time.

1.7.3.1.1.3 TDataSource:IAbortOnErrors

If it is .T., it will produce an run-time error when there is access data error.

Scope	Assignable
Type	Logic
Initial value	.F.

1.7.3.1.1.4 TDataSource:ICConnected

Connects and disconnects to the database. The .T. value means that the TDataSource is connected.

Scope	Assignable
Type	Logic
Initial value	.F.

1.7.3.1.1.5 TDataSource:IDisplayErrors

If it is .T., it will show in the screen all the error produced in the data access process.

Scope	Assignable
Type	Logic
Initial value	.T.

1.7.3.1.1.6 TDataSource:nTimeOut

Time out for any access or saved data operation.

Scope	Assignable
Type	Numeric
Initial value	30

1.7.3.1.2 TDataSource:Methods

■ Constructor ■ Standard

Type	Name
■	AddDataSet
■	BuildSQLSt
■	CloseProcess
■	Create
■	DelDataSet
■	DelErrors
■	DelProcess
■	EndProcess
■	GetErrors
■	IsError
■	LastError
■	NewProcess

■	OpenProcess
■	QueryMemDataset
■	RunProc
■	RunProcDirect
■	ShowErrorList
■	SqlInsert
■	SqlUpdate

1.7.3.1.2.1 TDataSource:AddDataSet

Adds a new DataSet to the aDataSet array.

Type	Standard
Parameters	<oDataSet> TDataSet object to be included in the array
Return value	NIL

1.7.3.1.2.2 TDataSource:BuildSQLSt

Joins together a SQL sentence with an unlimited number of parameters.

The '?' signs found on the original SQL sentence are replaced by the parameters given. The first '?' found on the SQL sentence is replaced by the first parameter, and so on, until all the '?' signs are processed.

Type	Standard
Parameters	<cSqlSource> Original SQL sentence [<xParam1, ..., xParamN>] [<aParams>] List of parameters to be used. All the parameters should be of type Character, Numeric, Date or Logical.
Return value	<cSqlResult>

Sample:

```
cSelect := "SELECT * FROM CLIENTS WHEEW Name = ? AND Age = ? AND
HIREDATE >= ?"
cResult := ODS:BuildSQLSt( cSelect, "John", 25, Ctod( "01/01/2007" ) )

? cResult // "SELECT * FROM CLIENTS WHEEW Name = 'John' AND Age = 25
AND HIREDATE >= '2007-01-01' "
```

1.7.3.1.2.3 TDataSource:CloseProcess

Closes in one step all the TDataSet object in an specific process.

Type	Standard
Parameters	<cProcess> Process to close
Return value	NIL

1.7.3.1.2.4 TDataSource:Create

Class constructor.

Type	Constructor
Parameters	<oParent> Container form object <cConnect> Connect string for the cConnect property
Return value	NIL

1.7.3.1.2.5 TDataSource:DelDataSet

Removes a DataSet in the aDataSet array.

Type	Standard
Parameters	<oDataSet> TDataSet to remove in the array
Return value	<ISuccess> .T. if the operation is successful

1.7.3.1.2.6 TDataSource:DelErrors

Removes the description for all the errors produced in the object.

Type	Standard
Parameters	None
Return value	NIL

1.7.3.1.2.7 TDataSource:DelProcess

Destroys in one step all the TDataSet objects from an specific process, calling its **End** method and eliminates the DataSet from the aDataSet array.

Type	Standard
Parameters	<cProcess> Process to destroy
Return value	<nClosed> Number of TDataSet destroyed

1.7.3.1.2.8 TDataSource:GetErrors

Returns a multidimensional array with the description of all the errors produced. The array has the following columns:

- Table name or select instruction
- Error description
- Error number
- Procedure when it was produced

Type	Standard
Parameters	NIL
Return value	<aInfo> Array with the error information

1.7.3.1.2.9 TDataSource:IsError

Returns .T. if has been an access error. You must execute the DelErrors method to check again this property in the future.

Type	Standard
Parameters	NIL
Return value	<IError> .T. if there is an error

1.7.3.1.2.10 TDataSource:LastError

Returns the last access error description. You must execute the DelErrors method to check again this property in the future.

Type	Standard
-------------	----------

Parameters	NIL
Return value	<cError> Error description

1.7.3.1.2.11 TDataSource.NewProcess

Creates a new process. This method only modifies the **cProcess** property for the future TDataSet created through this TDataSource.

Type	Standard
Parameters	<cProcess> New process name
Return value	NIL

1.7.3.1.2.12 TDataSource.OpenProcess

Opens in one step all the TDataSet object for an specific process.

Type	Standard
Parameters	<cProcess> Process to open
Return value	NIL

1.7.3.1.2.13 TDataSource.QueryMemDataset

Creates an TMemDataset object through a Select instruction in the database.

Type	Standard
Parameters	<cSelect> SELECT instruction [<cProcess>] Process name. Default value: the value assigned by the method.NewProcess [<oMemDataset>] TMemDataset object that receives the information. Optional.
Return value	<oDataset> TMemDataset object

1.7.3.1.2.14 TDataSource:RunProc

Runs a stored procedure. On this class is declared virtual. Only available with TMariaDBDataSource and TMySQLDataSource.

Type	Standard
Parameters	<oProc> TSqlProc object
Return value	<ISuccess> True if success

1.7.3.1.2.15 TDataSource:RunProcDirect

Run a stored procedure with no need to create a TSqlProc object. This class is declared virtual. Only available with TMariaDBDataSource and TMySQLDataSource.

Type	Standard
Parameters	<cName> Stored procedure name [<aParams> <Value>] Array with parameters input values. It only admits input parameters.
Return value	<Value> Stored procedure return value

1.7.3.1.2.16 TDataSource:ShowErrorList

Shows in a form all the errors produced in the TDataSource.

Type	Standard
Parameters	None
Return value	NIL

1.7.3.1.2.17 TDataSource:SqlInsert

Creates a **INSERT** SQL sentence based on the information given by the first parameter, which is a TExStruct object obtained through the method TDataset:GetRecord and the table name.

Tpe	Standard
Parameters	<oRecord> Record object obtained through TDataset:GetRecord method <cTable>

	Table name [<I ModOnly >] If true only modified fields will be included on the SQL sentence. By default false.
Return value	<c SqlResult >

1.7.3.1.2.18 TDataSource:SqlUpdate

Creates a **UPDATE** SQL sentence based on the information given by the first parameter, which is a TExStruct object obtained through the method TDataset:GetRecord and the table name.

Tpe	Standard
Parameters	<o Record > Record object obtained through TDataset:GetRecord method <c Table > Table name [<I ModOnly >] If true only modified fields will be included on the SQL sentence. By default false.
Return value	<c SqlResult >

1.7.3.1.3 TDataSource:Events

Name
OnConnect
OnConnected
OnCreate
OnDisconnect
OnDisconnected
OnError
OnExecute

1.7.3.1.3.1 TDataSource:OnConnect

Event that is produced when the TDataSource is connected.

Parameters	<o Sender > Object that triggers the event
Return value:	NIL

1.7.3.1.3.2 TDataSource:OnConnected

Event that is produces after the TDataSource is connected.

Parameters	<oSender> Object that triggers the event
Return value:	NIL

1.7.3.1.3.3 TDataSource:OnCreate

Event that is produced after the TDataSource is created.

Parameters	<oSender> Object that triggers the event
Return value:	NIL

1.7.3.1.3.4 TDataSource:OnDisConnect

Event that is produced when the TDataSource is disconnected.

Parameters	<oSender> Object that triggers the event
Return value:	NIL

1.7.3.1.3.5 TDataSource:OnDisConnected

Event that is produced after the TDataSource has been disconnected.

Parameters	<oSender> Object that triggers the event
Return value:	NIL

1.7.3.1.3.6 TDataSource:OnError

Event that is produced when a run-time error is generated due a failure on the Datasource or any of its datasets.

Parameters	<oSender> Object that triggers the event <oError> Error object <oDataset> TDataSet that generated the error. Can be NIL if the error is directly generated by the datasource.
Return value:	<IContinue> If diferent form false, the run-time error is definitely generated.

1.7.3.1.3.7 TDataSource:OnExecute

Event that is produced when a call to the Execute internal method is done. This happens on any consult or update operation on SQL databases.

Parameters	<oSender> Object that triggers the event <cSql> SQL statement
Return value:	NIL

1.7.3.2 TDbfDataSource

Class to manage the DBF files. This is also an abstract class like TDataSource but it adds to its TDataSource class the specific functionality to manage DBF tables.

This class is not designed to be used directly, but it will need to use more specialized classes inherited from TDbfDataSource to connect to CDX tables (CdxDataSource), NTX tables (NtxDataSource) and Advantage tables (AdsDataSource).

Description:

This class establishes the properties, methods and events that must have any specific DataSource object to manage the DBF tables in Xailer.

The methods, properties and events for this class, even when they are public, are not fully documented in this class because they will be treated with more detail in lower hierarchy classes.

There are some existing TDataSource methods and properties that are not accessible to this class, like the IConnected property, because it make no sense to use this properties for this kind of connections.

See also 'Introduction to DataControls' for more information.

Hierarchy	Inherits from TDataSource
File Name	\source\DbfDatasource.prg

See also TDbfDataSet, DataControls

1.7.3.2.1 TDbfDataSource:Properties

■ read Only ■ Assignable ■ Design assignable ■ Run-time assignable

Scope	Name	Type	Initial value
■	cConnect	Character	."
■	cDriver	Character	RddSetDefault()
■	cPassword	Character	""
■	IOptimize	Logical	.T.
■	nLockScheme	Numeric	dIDEFAULT
■	nMemoSize	Numeric	512 / 64
■	nMemoType	Numeric	dmDBT / dmFPT_FLEX
■	nTimeOut	Numeric	5

1.7.3.2.1.1 TDbfDataSource:cConnect

Default table directory. Its initial value is ".", that indicated the initial directory when the application is executed.

Scope	Assignable
Type	Character
Initial value	.

For example, to indicate that the tables are located in the DATA directory based in the location where the application is running:

```
oDataSource:cConnect := ".\DATA"
```

1.7.3.2.1.2 TDbfDataSource:cDriver

RDD driver to be used in the connection. The default value is the selected driver linked to the application through the [x]Harbour SetRddDefault() function.

Scope	Assignable
Type	Character
Initial value	RddSetDefault()

For more RDD information, please check the [x]Harbour documentation.

1.7.3.2.1.3 TDbfDataSource:cPassword

Password string to use for table encryption. The value of this property is assigned automatically on every dataset linked with this datasource. But the dataset has also a property **cPassword** you may modify.

Scope	Assignable
Type	Character
Initial value	""

For more RDD information, please check the [x]Harbour documentation.

1.7.3.2.1.4 TDbfDataSource:lOptimize

Sets filter optimization with indexed databases. For further information consult the xHarbour command SET OPTIMIZE.

Scope	Assignable
Type	Logical
Initial value	.T.

1.7.3.2.1.5 TDbfDataSource:nLockScheme

Selects the locking scheme for shared database access. For further information consult the xHarbour command SET DBFLOCKSCHEME.

Scope	Assignable
Type	Numeric
Initial value	dIDEFAULT
Possible values	dIDEFAULT dICLIP dICL53 dIVFP dICL53EXT dIXHB64

Note: This property can only be used with TDatasets created through a native RDD, TNtxDataSource or TCdxDataSource. The TDataset objects created through TAdsDataSource do not have accesible this property, having only the IAdsLocking property to set the locking scheme.

1.7.3.2.1.6 TDbfDataSource:nTimeOut

Timeout in seconds for any data access or retrieve operation.

Scope	Assignable
Type	Numeric
Initial value	5

1.7.3.2.1.7 TDbfDataSource:nMemoSize

Defines the default block size for memo files. For further information consult the xHarbour command SET MEMOBLOCK.

Scope	Assignable
Type	Numeric
Initial Value	512 (TNtxDataSource) 64 (TCdxDataSource)

Note: The TDataset objects created through TAdsDataSource do not have accesible this property.

1.7.3.2.1.8 TDbfDataSource:nMemoType

Memo fields file system. For further information consult xHarbour RddInfo() function.

Scope	Assignable
Type	Numeric
Initial value	dmDBT (TNtxDataSource) dmFPT_FLEX (TCdxDataSource)
Possible values	dmDBT dmSMT dmFPT_FP dmFPT_SIX dmFPT_FLEX

Note: This property can only be used with TDatasets created through a native RDD, TNtxDataSource o TCdxDataSource. The TDataset objects created through TAdsDataSource do not have accesible this property.

1.7.3.2.2 TDbfDataSource:Methods

■ Constructor ■ Standard

Typ	Name
e	

■	CommitAll
■	CreateTable
■	DefDataExtension
■	DefIdxExtension
■	DefMemoExtension
■	DelTable
■	NewDataSet

1.7.3.2.2.1 TDbfDataSource:CommitAll

Force a save operation for all the open TDbfDataSet in the TDbfDataSource.

Type	Standard
Parameters	None
Return value	NIL

1.7.3.2.2.2 TDbfDataSource:CreateTable

Creates a new table.

Type	Standard
Parameters	<aStructure> Table structure. Uses the same format than xHarbour DbCreate() function [<cTableName>] Table name, by default, a temporary filename
Return value	Object TDbfDataSet

1.7.3.2.2.3 TDbfDataSource:DefDataExtension

Returns the file extension for the data files.

Type	Standard
Parameters	None
Return value	"Dbf"

1.7.3.2.2.4 TDbfDataSource:DefIdxExtension

Returns the index file extension for the DBF files.

Type	Standard
Parameters	None
Return value	"Ntx"

1.7.3.2.2.5 TDbfDataSource:DefMemoExtension

Returns the Memo file extension for the DBF files.

Type	Standard
Parameters	None
Return value	"Dbt"

1.7.3.2.2.6 TDbfDataSource:DelTable

Deletes a table.

Type	Standard
Parameters	<cTableName> Table name
Return value	<ISuccess> True if success

1.7.3.2.2.7 TDbfDataSource:NewDataSet

Creates a new TDbfDataSet for the current TDbfDataSource.

Type	Standard
Parameters	<cTableName> Table name, without the full path, due it is included in the cConnect property. [<cProcess>] Process name. Default value: Name assigned by the NewProcess method to the TDataSource.
Return value	Object TDbfDataSet

1.7.3.3 TNtxDataSource

Class to manage DBF data with DBFNTX RDD original CA-Clipper driver, with NTX index files and DBT memo field extension files.

Description:

This class appears in the components palette, in the DataSets tab with the 'NTX' name. It allows to connect DBF files with Clipper classic NTX index files.

See also 'Introduction to DataControls' for more information.

Hierarchy	Inherits from TDbfDataSource
File Name	\source\NtxDatasource.prg
See also	TDbfDataSet, DataControls

1.7.3.3.1 TNtxDataSource:Properties

read Only
 Assignable
 Design assignable
 Run-time assignable

Scope	Name	Type	Initial value
<input type="checkbox"/>	cDriver	Character	"NTX"

1.7.3.3.1.1 TNtxDataSet:cDriver

RDD driver to be used in the connection. This is a read only property, with a '**DBFNTX**' string value.

Scope	read Only
Type	Character
Initial value	"DBFNTX"

For more information about RDD, please check the [x]Harbour documentation.

1.7.3.3.2 TNtxDataSource:Methods

Constructor
 Standard

Type	Name
<input type="checkbox"/>	DefIdxExtension

1.7.3.3.2.1 TNtxDataSource:DefIdxExtension

Returns the index file extension used by the DBF file.

Type	Standard
Parameters	None
Return value	"Ntx"

1.7.3.4 TCdxDataSource

Class to manage DBF data with DBFCDX RDD original FoxPro driver, with CDX index files and FPT memo field extension files.

Description:

This class appears in the components palette, in the DataSets tab with the 'CDX' name. It allows to connect DBF files with FoxPro classic CDX index files.

See also 'Introduction to DataControls' for more information.

Hierarchy	Inherits from TDbfDataSource
File Name	\source\CdxDatasource.prg
See also	TDbfDataSet, DataControls

1.7.3.4.1 TCdxDataSource:Properties

■ read Only ■ Assignable ■ Design assignable ■ Run-time assignable

Scope	Name	Type	Initial value
■	cDriver	Character	"CDX"
■	IAutoOpen	Logic	.T.
■	nTableType	Numeric	ctDBFSTD

1.7.3.4.1.1 TCdxDataSet:cDriver

RDD driver to be used in the connection. This is a read only property, with a '**DBFCDX**' string value.

Scope	read Only
Type	Character
Initial value	"DBFCDX"

For more information about RDD, please check the [x]Harbour documentation.

1.7.3.4.1.2 TCdxDataSource:IAutoOpen

If true when opening a table (Dataset) it will also be opened its structural index file if present.

Scope	Assignable
Type	Logic
Initial value	.T.

For further information consult the command 'SET AUTOOPEN' from [x]Harbour documentation.

1.7.3.4.1.3 TCdxDataSource:nTableType

Establish the table format for future datasets.

Scope	Assignable
Type	Numeric
Initial	ctDBFSTD
Possible values	ctDBFSTD ctDBFVFP

1.7.3.4.2 TCdxDataSource:Methods

■ Constructor ■ Standard

Typ Name
■ DefIdxExtension

1.7.3.4.2.1 TCdxDataSource:DefIdxExtension

Returns the index file extension used by the DBF file.

Type	Standard
Parameters	None
Return value	"Cdx"

1.7.3.5 TAdsDataSource

Class to manage DBF data types through a local or remote Advantage server.

Description:

This class appears in the component palette, under the DataSet tab "ADS" and allows to make connections with DBF and ADT tables and NTX, CDX or ADI index related files.

To use this DataSource it is needed to install the [Advantage](#) database engine, in both, your development environment and in the final environment where the application will be executed.

The advantage software can be used locally (free of charge!) or remotely, in which case you need to pay for it. The advantage of this database server engine is that allows to migrate easily from one typical DBF environment to a more professional environment based in DBF tables, but they are managed for a dedicated server that ensures a much better integrity levels.

The local Advantage version can be compared to the native DBFNTX and DBFCDX [x]Harbour drivers. However, ADS adds some technology like the optimized filters that are not currently included in the native [x]Harbour drivers. It's possible that in an environment with a lot of users, the native RDD drives will behave better than the local Advantage server. In both cases, we recommend to use the remote Advantage server and much more if the files are stored in a Microsoft server types, due they penalize the shared file access.

The Advantage functionality is still based in DBF tables, but its proprietary ADF format is superior allowing the use of data dictionary, file access through internet, automatic table encryption, transactions, etc.

You can download the local Advantage version directly from:

<http://resolution.extendedsystems.com/ADS/Product+Detail/Advantage+Local+Server/default.htm>

For more information, see also the 'Introducción to DataControls' chapter.

Hierarchy	Inherits from TDbfDataSource
File Name	\source\AdsDatasource.prg
See also	TDbfDataSet, DataControls

1.7.3.5.1 TAdsDataSource:Properties

■ read Only ■ Assignable ■ Design assignable ■ Run-time assignable

Scope	Name	Type	Initial value
■	cConnect	Character	."
■	cDriver	Character	"ADS"
■	cPassword	Character	""
■	cUser	Character	""
■	hConnect	Numérico	0
■	lAdsLocking	Logic	.T.
■	lRightsCheck	Logic	.T.
■	lUseDictionary	Logic	.F.
■	nCharType	Numeric	acANSI
■	nConnectionFlags	Numeric	0
■	nFileType	Numeric	afCDX
■	nServerType	Numeric	asDEFAULT

1.7.3.5.1.1 TAdsDataSource:cConnect

String connection to the database.

Scope	Assignable
Type	Character
Initial value	""

In the case that the connections is made through a dictionary (see also IUseDictionary), it points to the definition dictionary file (with the .ADD extension). In the case of a normal access like any DBF access, the cConnect property will point, as well as the DBF DataSources, to the directory where all the tables are stored and this directory will be a relative path where the application will be executed.

1.7.3.5.1.2 TAdsDataSource:cDriver

RDD driver to be used in the connection. This is a read only property with the value = ADS.

Scope	read Only
Type	Character
Initial value	"ADS"

For more information about how to use the RDD, see also the [x]Harbour documentation.

1.7.3.5.1.3 TAdsDataSource:cPassword

Password to connect to the server.

Scope	Assignable before the connection to the server
Type	Character
Initial value	""

This property is used together with the cUser property to connect remotely to the server.

This property can be used as well as an encrypted code in tables, but in this case it is needed to connect directly to the server, without the dictionary (IUseDictionary property) and the cUser must be empty.

If you need to open the encrypted tables through the dictionary, you will need to use the system offered by Advantage. This system establish a unique password for all the encrypted tables and it will be stored in the same dictionary. For more information, please review the Advantage documentation.

1.7.3.5.1.4 TAdsDataSource:cUser

User name to connect to the server.

Scope	Assignable before the connection to the server
Type	Character
Initial value	""

This property is use together with the cPassword property to connect to a remote server.

To have access to the server thought the dictionary (IUseDictionary property), it is needed to connect to the server indicating a Name and a password. However, to access directly to the tables, this step is not needed.

1.7.3.5.1.5 TAdsDataSource:hConnect

ADS internal handle.

Scope	Read only
Type	Numeric
Initial value	0

1.7.3.5.1.6 TAdsDataSource:lAdsLocking

If it is .T., Advantage will use its own mechanism to block records.

Scope	Assignable before the connection to the server
Type	Logic
Initial value	.T.

If you will not open the tables directly, we recommend to use the Advantage mechanism to block records. in the case of access to remote servers, it allows to open the tables in exclusive mode with a better performance.

1.7.3.5.1.7 TAdsDataSource:lRightsCheck

.T. is the server must control the client rights in every open table.

Scope	Assignable before the connection to the server
Type	Logic
Initial value	.T.

This property has effect only in remote servers. To open a database it is needed to have read and write rights over the table. However, if the read and write operation are done directly for the server, is possible that it might not be needed that the client has those rights due the server has those privileges.

If you set this property to .F. and deleting all the read and write rights over the tables, it is possible that the client has access only to the data through the application, avoiding the non authorized access and even the accidental table deletion from unexperienced users/programmers.

1.7.3.5.1.8 TAdsDataSource:lUseDictionary

It is .T. if the data dictionary must be used to access to the server. In this case the cConnect property points to the file where the dictionary is located (normally an .ADD extension file).

Scope	Assignable before the connection to the server
Type	Logic
Initial value	.T.

If the dictionary is used, it is need to connect to it indicating a name in cUser and a password in cPassword.

1.7.3.5.1.9 TAdsDataSource:nCharType

Character set to be used.

Scope	Assignable before the connection to the server
Type	Numeric
Initial value	acANSI

The possible values are:

- acANSI: ANSI character set.
- acOEM: OEM character set

1.7.3.5.1.10 TAdsDataSource:nConnectionFlags

Advantage server connection flag.

Scope	Assignable before the connection to the server
Type	Numeric
Initial value	0

It corresponds with the fifth [x]Harbour AdsConnect60() parameter, that establishes an additional "Flag" in the server connection. Please review the Advantage documentation, for more information.

1.7.3.5.1.11 TAdsDataSource:nFileType

Origin data type.

Scope	Assignable before the connection to the server
Type	Numeric
Initial value	afNTX

The possible values are:

afNTX	Classic CA-CLIPPER, DBF-DBT with NTX index
afCDX	Classic FORPRO, DBF-FPT with CDX index
afADT	Proprietary Advantage

1.7.3.5.1.12 TAdsDataSource:nServerType

Server data type.

Scope	Assignable before the connection to the server
Type	Numeric
Initial value	asDEFAULT

The possible values are:

asDEFAULT	Default server
asLOCAL	Local server
asREMOTE	Remote server
asAIS	Internet server
asREMOTE_AIS	Remote or Internet server
asANY	Any server

1.7.3.5.2 TAdsDataSource:Methods

■ Constructor ■ Standard

Type	Name
■	BeginTrans
■	CommitTrans
■	DefDataExtension

■	DefIdxExtension
■	QueryArray
■	QueryRow
■	QueryValue
■	RollBackTrans

1.7.3.5.2.1 TAdsDataSource:BeginTrans

Indicates the beginning if a transaction

Type	Standard
Parameters	None
Return value	NIL

A transaction is a multiple operation over one or more tables. Through this and the CommitTrans and RollBackTrans methods it is possible to execute atomic transactions. In other words, transactions that behave like a unique operation. All the operations executed after a BeginTrans operation can be canceled in any moment calling to the RollBackTrans. When the operation has been completed successfully, then it is needed to call the CommitTrans method to indicate to the engine that the operation is correct and can be finalized.

1.7.3.5.2.2 TAdsDataSource:CommitTrans

Indicates the end of a transaction.

Type	Standard
Parameters	None
Return value	NIL

A transaction is a multiple operation over one or more tables. Through this and the CommitTrans and RollBackTrans methods it is possible to execute atomic transactions. In other words, transactions that behave like a unique operation. All the operations executed after a BeginTrans operation can be canceled in any moment calling to the RollBackTrans. When the operation has been completed successfully, then it is needed to call the CommitTrans method to indicate to the engine that the operation is correct and can be finalized.

1.7.3.5.2.3 TAdsDataSource:DefDataExtension

Returns the data file extensions for the current nFileType.

Type	Standard
-------------	----------

Parámetros	None
Return value	"Dbf" o "Adt"

1.7.3.5.2.4 TAdsDataSource:DefIdxExtension

Returns the index file extensions for the current nFileType.

Type	Standard
Parámetros	None
Return value	"Ntx" o "Cdx" o "Adi"

1.7.3.5.2.5 TAdsDataSource:QueryArray

Returns a multidimensional array with all the query information received through a SELECT query in the database.

Type	Standard
Parameters	<cSelect> SELECT instruction [<@aHeaders>] Sets the array with the retrieved column names
Return value	<aData>

1.7.3.5.2.6 TAdsDataSource:QueryRow

Returns an array with the information received through a SELECT query in the database.

Type	Standard
Parameters	<cSelect> SELECT instruction [<@aHeaders>] Sets the array with the retrieved column names
Return value	<aData>

If the select instruction returns more than one row , this method will recover only the first cursor row. The number of elements in the array will be the same as the number of fields in the query.

1.7.3.5.2.7 TAdsDataSource:QueryValue

Returns the result value from the information received through a SELECT instruction in the database.

Type	Standard
Parameters	<cSelect> SELECT instruction
Return value	<Value>

If the select instruction returns more than one row or multiple columns, this method will recover only the first column from the first cursor row.

Example:

```
nTotal := oDataSource:QuerValue( "COUNT(*) FROM Customer " )
```

1.7.3.5.2.8 TAdsDataSource:RollBackTrans

Cancels a transaction started with the BeginTrans method.

Type	Standard
Parameters	None
Return value	NIL

A transaction is a multiple operation over one or more tables. Through this and the CommitTrans and RollBackTrans methods it is possible to execute atomic transactions. In other words, transactions that behave like a unique operation. All the operations executed after a BeginTrans operation can be canceled in any moment calling to the RollBackTrans. When the operation has been completed successfully, then it is needed to call the CommitTrans method to indicate to the engine that the operation is correct and can be finalized.

1.7.3.6 TADODataSource

Class to manage databases through ADO (ActiveX Data Objects). For further information about ADO consult the following link:

http://en.wikipedia.org/wiki/ActiveX_Data_Objects

Description:

This class appears in the component palette with the name 'ADO' and permits to make connections to any database engine that has an ADO client available. Is also possible to access with ADO to any database that has a ODBC client available.

You should install the ADO client libraries on your development environment and on the end user

computers

See also 'Introduction to DataControls' for more information.

Hierarchy Inherits from TDataSource
File Name \source\AdoDataSource.prg
See also TDataSet, DataControls

1.7.3.6.1 TADODataSource:Properties

■ read Only ■ Assignable ■ Design assignable ■ Run-time assignable

Scope	Name	Type	Initial Value
■	cConnect	Character	" "
■	cPassword	Character	""
■	cUser	Character	""
■	IConnectAsync	Logic	.F.
■	IPromptIfError	Logic	.T.
■	nAffectedRows	Numeric	0
■	nAttributes	Numeric	adXactNone
■	nConnTimeOut	Numeric	30
■	nCursorLocation	Numeric	adUseClient
■	nIsolationLevel	Numeric	adXactReadCommitted
■	nMode	Numeric	adModeUnknown
■	oConnection	Object	NIL

1.7.3.6.1.1 TADODataSource:cConnect

String connection to the database. Each engine has its own syntax. Is preferable to use the assistant include it on the Xailer object inspector which will guide to the all process.

Scope	Assignable before the connection to the server
Type	Character
Initial value	""

1.7.3.6.1.2 TADODataSource:cPassword

Password to use to connect to the server. Is possible that the password is already included on the cConnect property, on that case, you should leave this property empty.

With some database engines if you do not specify a password the ADO may ask for one on run-time.

Scope	Assignable before the connection to the server
Type	Character

Initial value	""
----------------------	----

1.7.3.6.1.3 TADODataSource:cUser

Name to access to the database. Is possible that the name is already included on the cConnect property, on that case, you should leave this property empty.

With some database engines if you do not specify a name the ADO may ask for one on run-time.

Scope	Assignable before the connection to the server
Type	Character
Initial value	""

1.7.3.6.1.4 TADODataSource:nAffectedRows

Number of rows affected by the last INSERT, UPDATE or DELETE operation done through the Execute() method.

Scope	Read only
Type	Numeric
Initial value	0

1.7.3.6.1.5 TADODataSource:lConnectAsync

If true the connection will be opened asynchronous.

Scope	Assignable before the connection to the server
Type	Logical
Initial value	.F.

1.7.3.6.1.6 TADODataSource:lPromptIfError

If true and the connections fails, the ADO client connection dialog will be shown.

Scope	Assignable before the connection to the server
Type	Logical
Initial value	.F.

1.7.3.6.1.7 TADODataSource:nAttributes

Attributes for the connection.

Scope	Assignable before the connection to the server
Type	Numeric
Initial value	adXactNone
Possible values	<ul style="list-style-type: none"> • adXactNone: No attributes • adXactAbortRetaining: Performs retaining aborts—that is, calling RollbackTrans automatically starts a new transaction. Not all providers support this. • adXactCommitRetaining: Performs retaining commits—that is, calling CommitTrans automatically starts a new transaction. Not all providers support this.

For further information consult the following link:

<http://msdn.microsoft.com/library/default.asp?url=/library/en-us/ado270/htm/mdcstxactattributeenu.m.asp>

1.7.3.6.1.8 TADODataSource:nConnTimeOut

Connection timeout in seconds.

Scope	Assignable
Type	Numeric
Initial value	30

1.7.3.6.1.9 TADODataSource:nCursorLocation

Indicates the location of the cursor service for further Dataset openings.

Scope	Assignable
Type	Numeric
Initial value	adUseClient
Possible values	adUseClient adUseServer adUseNone

- **adUseServer**: Uses data-provider or driver-supplied cursors. These cursors are sometimes very flexible and allow for additional sensitivity to changes others make to the data source. However, some features of the Microsoft Cursor Service for OLE DB (such as disassociated Recordset objects) cannot be simulated with server-side cursors and these features will be

unavailable with this setting..

- **adUseClient:** Uses client-side cursors supplied by a local cursor library. Local cursor services often will allow many features that driver-supplied cursors may not, so using this setting may provide an advantage with respect to features that will be enabled. For backward compatibility, the synonym `adUseClientBatch` is also supported..
- **adUseNone:** Does not use cursor services

Not all the servers support this values. On that case the Dataset will try to use the best approximate value without generating a run-time error.

For further information consult the following link:

<http://msdn.microsoft.com/library/en-us/ado270/htm/mdprocursorlocation.asp>

1.7.3.6.1.10 TADODataSource:nIsolationLevel

Indicates the level of isolation.

Scope	Assignable
Type	Numeric
Initial value	<code>adXactReadCommitted</code>
Possible values	<code>adXactUnspecified</code> <code>adXactChaos</code> <code>adXactBrowse</code> <code>adXactReadUncommitted</code> <code>adXactCursorStability</code> <code>adXactReadCommitted</code> <code>adXactRepeatableRead</code> <code>adXactIsolated</code> <code>adXactSerializable</code>

- **adXactUnspecified:** Indicates that the provider is using a different isolation level than specified, but that the level cannot be determined.
- **adXactChaos:** Indicates that pending changes from more highly isolated transactions cannot be overwritten.
- **adXactBrowse:** Indicates that from one transaction you can view uncommitted changes in other transactions.
- **adReadUncommitted:** Same as `adXactBrowse`.
- **adXactCursorStability:** Indicates that from one transaction you can view changes in other transactions only after they have been committed.
- **adXactReadCommitted:** Same as `adXactCursorStability`.
- **adXactRepeatableRead:** Indicates that from one transaction you cannot see changes made in other transactions, but that requerying can retrieve new Recordset objects.
- **adXactIsolated:** Indicates that transactions are conducted in isolation of other transactions.
- **adXactSerializable:** Same as `adXactIsolated`.

Not all the databases support this values.

For further information consult the following link:

<http://windowssdk.msdn.microsoft.com/en-gb/library/ms681478.aspx>

1.7.3.6.1.11 TADODataSource:nMode

Indicates the available permissions for modifying data.

Scope	Assignable
Type	Numeric
Initial value	adModeUnknown
Possible values	adModeUnknown adModeRead adModeWrite adModeReadWrite adModeShareDenyRead adModeUnknown adModeShareDenyWrite adModeShareExclusive adModeShareDenyNone

- **adModeUnknown**: Indicates that the permissions have not yet been set or cannot be determined..
- **adModeRead**: Indicates read-only permissions.
- **adModeWrite**: Indicates write-only permissions.
- **adModeReadWrite**: Indicates read/write permissions.
- **adModeShareDenyRead**: Prevents others from opening a connection with read permissions.
- **adModeShareDenyWrite**: Prevents others from opening a connection with write permissions.
- **adModeShareExclusive**: Prevents others from opening a connection.
- **adModeShareDenyNone**: Allows others to open a connection with any permissions. Neither read nor write access can be denied to others.

Not all the databases support this values.

For further information consult the following link:

<http://windowssdk.msdn.microsoft.com/en-gb/library/ms676693.aspx>

1.7.3.6.1.12 TADODataSource:oConnection

ActiveX object that holds the connection.

Scope	Read only
Type	Object
Initial value	NIL

1.7.3.6.2 TADODataSource:Methods

■ Constructor ■ Standard

Type	Name
■	BeginTrans
■	Cancel
■	CommitTrans
■	Connect
■	ConnectTag
■	DelTable
■	Disconnect
■	Execute
■	File
■	GetCatalogs
■	GetTables
■	Properties
■	Query
■	QueryArray
■	QueryReport
■	QueryRow
■	QueryValue
■	RollBackTrans
■	State
■	Table
■	Version

1.7.3.6.2.1 TADODataSource:BeginTrans

Starts a new transaction.

Type	Standard
Parameters	None
Return value	NIL

1.7.3.6.2.2 TADODataSource:Cancel

Cancels execution of a pending asynchronous method call.

Type	Standard
Parameters	None
Return value	NIL

1.7.3.6.2.3 TADODataSource:CommitTrans

Saves any changes and ends the current transaction. It may also start a new transaction.

Type	Standard
Parameters	None
Return value	NIL

1.7.3.6.2.4 TADODataSource:Connect

Connects to the database.

Type	Standard
Parameters	[<cConnect>] String connection.
Return value	<ISuccess> True if success

1.7.3.6.2.5 TAdoDataSource:ConnectTag

Retrieves or changes any property value of current cConnect string.

The connect string cConnect used by ADO is a string of undefined length where all the properties of the connection are defined. The properties are separated to each other by the token ';' and its value is assigned with the operator '='. As a sample:

Provider=MSDASQL;Persist Security Info=False;Data Source=dBASE Files

With this method you can access or change any property of the connect string.

Type	Standard
Parameters	<cProperty> Property to acces or modify

	[<cNewValue>] New value. If NIL it will not be changed.
Return value	<cOldValue> Old value

1.7.3.6.2.6 TADODataSource:DelTable

Deletes a table from the database.

Type	Standard
Parameters	<cTableName> Table name
Return value	<ISuccess> True if success

1.7.3.6.2.7 TADODataSource:Disconnect

Disconnects from the database

Type	Standard
Parameters	None
Return value	<ISuccess> True if success

1.7.3.6.2.8 TADODataSource:Execute

Executes a command on the database.

Type	Standard
Parameters	<cCommand> Command to execute [<cOperation>] Operation description to be shown in case of a run-time error [<nOptions>] Numerical value indicating further attributes to ADO [<@oRS>] Dataset object passed by references in case the command returns a Dataset.
Return value	<ISuccess> True if success

After a call to method Execute, the property nAffectedRows returns the number of affected rows.

nOptions can be a bitmask of one or more CommandTypeEnum or ExecuteOptionEnum values.:

- adCmdText, adCmdTable, adCmdStoredProc, adCmdFile, adCmdTableDirect
- adAsyncExecute, adAsyncFetch, adAsyncFetchNonBlocking, adExecuteNoRecords, adExecuteStream, adExecuteRecord

For further information consult the following link:

<http://windowssdk.msdn.microsoft.com/en-gb/library/ms675023.aspx>

1.7.3.6.2.9 TADODataSource:File

Indicates if table name exists in the database.

Type	Standard
Parameters	<cTableName> Table name
Return value	<ISuccess> True if exists

1.7.3.6.2.10 TADODataSource:GetCatalogs

Returns all the schema catalogs of the database. Not all providers support this method.

Type	Standard
Parameters	None
Return value	<aList> Catalogue list

1.7.3.6.2.11 TADODataSource:GetTables

Return all the tables and views of a database.

Type	Standard
Parameters	<cMask> Mask to use for searching [<IView>] If true the views will be also included. By default .T.
Return value	<aTables> Array with all the view and table names

1.7.3.6.2.12 TADODataSource:Properties

Returns a extended ADO property or a extended ADO properties collection of the database.

Type	Standard
Parameters	[<cProperty>] Property name
Return value	<oProperty aPropertiesCollection> Property or Properties list

Each element of the list is an ADO:Property object wichn includes the following members:

- **Name:** Property name
- **Type:** Property type
- **Value:** Property value
- **Attributes:** Property extended attributes

For further information consult the following link:

<http://msdn.microsoft.com/library/default.asp?url=/library/en-us/ado270/htm/mdobjproperty.asp>

1.7.3.6.2.13 TADODataSource:Query

Creates an TDataset object through a Select instruction in the database.

Type	Standard
Parameters	<cSelect> SELECT instruction [<cProcess>] Process name. Default value: the value assigned by the method.NewProcess
Return value	<oDataset> TDataset object

1.7.3.6.2.14 TADODataSource:QueryArray

Returns a multidimensional array with all the query information received through a SELECT query in the database.

Type	Standard
Parameters	<cSelect> SELECT instruction [<@aHeaders>] Sets the array with the retrieved column names
Return value	<aData>

1.7.3.6.2.15 TADODataSource:QueryReport

Creates an TDataset object through a Select instruction in the database but optimized for being used in reports and no suitable for browses. It creates a read only and move forward dataset.

Type	Standard
Parameters	<cSelect> SELECT instruction [<cProcess>] Process name. Default value: the value assigned by the method.NewProcess
Return value	<oDataset> TDataset object

1.7.3.6.2.16 TADODataSource:QueryRow

Returns an array with the information received through a SELECT query in the database.

Type	Standard
Parameters	<cSelect> SELECT instruction [<@aHeaders>] Sets the array with the retrieved column names [aDefault] Default value in case a error is raised
Return value	<aData>

If the select instruction returns more than one row , this method will recover only the first cursor row. The number of elements in the array will be the same as the number of fields in the query.

1.7.3.6.2.17 TADODataSource:QueryValue

Returns the result value from the information received through a SELECT instruction in the database.

Type	Standard
Parameters	<cSelect> SELECT instruction [uDefault] Default value in case a error is raised
Return value	<Value>

If the select instruction returns more than one row or multiple columns, this method will recover only the first column from the first cursor row.

Example:

```
nTotal := oDataSource:QueryValue( "COUNT(*) FROM Customer" )
```

1.7.3.6.2.18 TADODataSource:RollBackTrans

Cancels any changes made during the current transaction and ends the transaction. It may also start a new transaction..

Type	Standard
Parameters	None
Return value	NIL

1.7.3.6.2.19 TADODataSource:State

Indicates for all applicable objects whether the state of the object is open or closed. If the object is executing an asynchronous method, indicates whether the current state of the object is connecting, executing, or retrieving.

Type	Standard
Parameters	None
Return value	<p><adStateClosed> Indicates that the object is closed.</p> <p><adStateOpen> Indicates that the object is open.</p> <p><adStateConnecting> Indicates that the object is connecting.</p> <p><adStateExecuting> Indicates that the object is executing a command.</p> <p>adStateFetching Indicates that the rows of the object are being retrieved.</p>

1.7.3.6.2.20 TADODataSource:Table

Creates a TDataset object just indicating ist table name on the database.

Type	Standard
Parameters	<p><cTableName> Table name [<cProcess>]</p>

	Process name. Default value: the value assigned by the method.NewProcess
Return value	<oDataset> TDataSet object

1.7.3.6.2.21 TADODataSource:Version

Returns ADO current version.

Type	Standard
Parameters	None
Return value	<cVersionInfo> Version information

1.7.3.7 TMySQLDataSource

Class to manage MySQL databases.

This class gives direct support to MySQL engines. For further information consult the following link:

<http://en.wikipedia.org/wiki/mysql>

You should include the **LibMySQL.lib** library on any project that uses this database. And also you should include the library **LibMySQL.dll** on all the installation programs of the projects that uses this control.

See also 'Introduction to DataControls' for more information..

Hierarchy	Inherits from TDataSource
File Name	\source\EnterpriseMySQL.prg
See also	TDataSet, DataControls

1.7.3.7.1 TMySQLDataSource:Propiedades

Scope	Name	Type	Initial Value
<input type="checkbox"/>	cDataBase	Character	""
<input type="checkbox"/>	cHost	Character	""
<input type="checkbox"/>	cPassword	Character	""
<input type="checkbox"/>	cUser	Character	""
<input type="checkbox"/>	IAutoReconnect	Logical	.F.
<input type="checkbox"/>	nPort	Numeric	3306
<input type="checkbox"/>	nTimeOut	Numeric	1000
<input checked="" type="checkbox"/>	nAffectedRows	Numeric	0

1.7.3.7.1.1 TMySQLDataSource:cDatabase

Database name.

Scope	Assignable before the connection to the server
Type	Character
Initial value	""

1.7.3.7.1.2 TMySQLDataSource:cHost

Host name or IP address.

Scope	Assignable before the connection to the server
Type	Carácter
Initial value	""

1.7.3.7.1.3 TMySQLDataSource:cPassword

Server access password.

Scope	Assignable before the connection to the server
Type	Character
Initial value	""

1.7.3.7.1.4 TMySQLDataSource:cUser

Server access name.

Scope	Assignable before the connection to the server
Type	Character
Initial value	""

1.7.3.7.1.5 TMySQLDataSource:lAutoReconnect

If true the data source will try to reconnect with the server every time the connection is lost.

Scope	Assignable before the connection to the server
--------------	--

Type	Logical
Initial value	.F.

1.7.3.7.1.6 TMySQLDataSource:nPort

IP port to use for the connection to the server.

Scope	Assignable before the connection to the server
Type	Numeric
Initial value	3306

1.7.3.7.1.7 TMySQLDataSource:nTimeOut

Timeout in miliseconds for server connection and any othe query to it.

Scope	Assignable before the connection to the server
Type	Numeric
Initial value	1000

1.7.3.7.1.8 TMySQLDataSource:nAffectedRows

Number of rows affected by the last INSERT, UPDATE or DELETE operation done through the Execute() method.

Scope	Read only
Type	Numeric
Initial value	0

1.7.3.7.2 TMySQLDataSource:Metodos

■ Constructor ■ Standard

Type	Name
■	BeginTrans
■	CommitTrans
■	Connect
■	CreateTable
■	DelTable
■	Disconnect
■	Execute
■	File

■	GetCatalogs
■	GetTables
■	Query
■	QueryArray
■	QueryArrayHash
■	QueryReport
■	QueryRow
■	QueryRowHash
■	QueryValue
■	RollBackTrans
■	SqlNextResult
■	Table

1.7.3.7.2.1 TMySQLDataSource:BeginTrans

Starts a new transaction.

Type	Standard
Parameters	None
Return value	NIL

1.7.3.7.2.2 TMySQLDataSource:CommitTrans

Saves any changes and ends the current transaction. It may also start a new transaction.

Type	Standard
Parameters	None
Return value	NIL

1.7.3.7.2.3 TMySQLDataSource:Connect

Connects to the database.

Type	Standard
Parameters	[<cConnect>] String connection.
Return value	<ISuccess> True if success

1.7.3.7.2.4 TMySQLDataSource:CreateTable

Creates a new table on the database.

Type	Standard
Parameters	<cTable> Table name <aStruct> Table structure. Uses the same format that DBCcreate() xHarbour function, except that if you include an asterisk after the type definition, that field will be considered also part of the primary key. [<cEngine>] Database internal engine. See MySQL documentation.
Return value	<ISuccess> True if success

1.7.3.7.2.5 TMySQLDataSource:DelTable

Deletes a table from the database.

Type	Standard
Parameters	<cTableName> Table name
Return value	<ISuccess> True if success

1.7.3.7.2.6 TMySQLDataSource:Disconnect

Disconnects from the database

Type	Standard
Parameters	None
Return value	<ISuccess> True if success

1.7.3.7.2.7 TMySQLDataSource:Execute

Executes a command on the database.

Type	Standard
Parameters	<cCommand> Command to execute [<cOperation>] Operation description to be shown in case of a run-time error [<@aData>] Array with all the values returned by the instruction in case is a SELECT sentence
Return value	<ISuccess> True if success

After a call to method Execute, the property nAffectedRows returns the number of affected rows.

1.7.3.7.2.8 TMySQLDataSource:File

Indicates if table name exists in the database.

Type	Standard
Parameters	<cTableName> Table name
Return value	<ISuccess> True if exists

1.7.3.7.2.9 TMySQLDataSource:GetCatalogs

Returns all the databases or catalogs on the server.

Type	Standard
Parameters	None
Return value	<aDatabases> Array with all the database names

1.7.3.7.2.10 TMySQLDataSource:GetTables

Return all the tables and views of a database.

Type	Standard
Parameters	<cMask>

	Mask to use for searching [<IView>] If true the views will be also included. By default .T.
Return value	<aTables> Array with all the view and table names

1.7.3.7.2.11 TMySQLDataSource:Query

Creates an TDataset object through a Select instruction in the database.

Type	Standard
Parameters	<cSelect> SELECT instruction [<cProcess>] Process name. Default value: the value assigned by the method.NewProcess
Return value	<oDataset> TDataset object

1.7.3.7.2.12 TMySQLDataSource:QueryArray

Returns a multidimensional array with all the query information received through a SELECT query in the database.

Type	Standard
Parameters	<cSelect> SELECT instruction [<@aHeaders>] Sets the array with the retrieved column names
Return value	<aData>

1.7.3.7.2.13 TMySQLDataSource:QueryArrayHash

Returns a hash with all the query information received through a SELECT query in the database.

Type	Standard
Parameters	<cSelect> SELECT instruction [<@aHeaders>] Sets the array with the retrieved column names
Return	<aData>

value	
-------	--

1.7.3.7.2.14 TMySQLDataSource:QueryReport

Creates an TDataset object through a Select instruction in the database but optimized for being used in reports and no suitable for browses. It creates a read only and move forward dataset.

Type	Standard
Parameters	<cSelect> SELECT instruction [<cProcess>] Process name. Default value: the value assigned by the method.NewProcess
Return value	<oDataset> TDataset object

1.7.3.7.2.15 TMySQLDataSource:QueryRow

Returns an array with the information received through a SELECT query in the database.

Type	Standard
Parameters	<cSelect> SELECT instruction [<@aHeaders>] Sets the array with the retrieved column names [aDefault] Default value in case a error is raised
Return value	<aData>

If the select instruction returns more than one row , this method will recover only the first cursor row. The number of elements in the array will be the same as the number of fields in the query.

1.7.3.7.2.16 TMySQLDataSource:QueryRowHash

Returns a hash with the information received through a SELECT query in the database.

Type	Standard
Parameters	<cSelect> SELECT instruction [<@aHeaders>] Sets the array with the retrieved column names [aDefault] Default value in case a error is raised

Return value	<aData>
---------------------	---------

If the select instruction returns more than one row , this method will recover only the first cursor row.

1.7.3.7.2.17 TMySQLDataSource:QueryValue

Returns the result value from the information received through a SELECT instruction in the database.

Type	Standard
Parameters	<cSelect> SELECT instruction [uDefault] Default value in case a error is raised
Return value	<Value>

If the select instruction returns more than one row or multiple columns, this method will recover only the first column from the first cursor row.

Example:

```
nTotal := oDataSource:QuerValue( "COUNT(*) FROM Customer " )
```

1.7.3.7.2.18 TMySQLDataSource:RollBackTrans

Cancel any changes made during the current transaction and ends the transaction. It may also start a new transaction..

Type	Standard
Parameters	None
Return value	NIL

1.7.3.7.2.19 TMySQLDataSource:SqlNextResult

Retrieves the information of next SELECT statement on a stored procedure called through the Execute method when the stored procedure has more than one SELECT statement.

Type	Standard
Parameters	[<@aData>] Information array on output. If no value given the next result is

	discarded. [<@aHeaders>] Array with field names
Return value	<ISuccess> True if success

1.7.3.7.2.20 TMySQLDataSource:Table

Creates a TDataSet object just indicating ist table name on the database.

Type	Standard
Parameters	<cTableName> Table name [<cProcess>] Process name. Default value: the value assigned by the method.NewProcess
Return value	<oDataset> TDataSet object

1.7.3.8 TMariaDBDataSource

Class to manage **MariaDB** databases.

This class gives direct support to MariaDB engines. For further information consult the following link:

<http://en.wikipedia.org/wiki/MariaDB>

You should include the **LibMariaDB.lib** library on any project that uses this database. And also you should include the library **LibMariaDB.dll** on all the installation programs of the projects that uses this control. This library also uses two Microsoft distributable libraries which are **msvcr100.dll** and **msvcp100.dll** that may not be present on Windows XP systems. The three libraries must installed with your application.

Important note: MariaDB and MySQL are completely compatible, even at the API level. You can use MariaDB to access MySQL databases and viceversa. We recommend to use MariaDB client libraries since they can be included with your applications with no restrictions. **You can not use MariaDB and MySQL datasources at the same time on your applications.**

See also 'Introduction to DataControls' for more information..

Hierarchy	Inherits from TDataSource
File Name	\source\Enterprise\MariaDB.prg
See also	TDataSet, DataControls

1.7.3.8.1 TMariaDBDataSource:Propiedades

■ read Only
 ■ Assignable
 ■ Design assignable
 ■ Run-time assignable

Scope	Name	Type	Initial Value
■	cDataBase	Character	""
■	cHost	Character	""
■	cPassword	Character	""
■	cUser	Character	""
■	IAutoReconnect	Logical	.F.
■	nPort	Numeric	3306
■	nTimeOut	Numeric	1000
■	nAffectedRows	Numeric	0

1.7.3.8.1.1 TMariaDBDataSource:cDatabase

Database name.

Scope	Assignable before the connection to the server
Type	Character
Initial value	""

1.7.3.8.1.2 TMariaDBDataSource:cHost

Host name or IP address.

Scope	Assignable before the connection to the server
Type	Carácter
Initial value	""

1.7.3.8.1.3 TMariaDBDataSource:cPassword

Server access password.

Scope	Assignable before the connection to the server
Type	Character
Initial value	""

1.7.3.8.1.4 TMariaDBDataSource:cUser

Server access name.

Scope	Assignable before the connection to the server
Type	Character
Initial value	""

1.7.3.8.1.5 TMariaDBDataSource:lAutoReconnect

If true the data source will try to reconnect with the server every time the connection is lost.

Scope	Assignable before the connection to the server
Type	Logical
Initial value	.F.

1.7.3.8.1.6 TMariaDBDataSource:nPort

IP port to use for the connection to the server.

Scope	Assignable before the connection to the server
Type	Numeric
Initial value	3306

1.7.3.8.1.7 TMariaDBDataSource:nTimeOut

Timeout in miliseconds for server connection and any othe query to it.

Scope	Assignable before the connection to the server
Type	Numeric
Initial value	1000

1.7.3.8.1.8 TMariaDBDataSource:nAffectedRows

Number of rows affected by the last INSERT , UPDATE or DELETE operation done through the Execute() method.

Scope	Read only
Type	Numeric

Initial value 0

1.7.3.8.2 TMariaDBDataSource:Metodos

■ Constructor ■ Standard

Type	Name
■	BeginTrans
■	CommitTrans
■	Connect
■	CreateTable
■	DelTable
■	Disconnect
■	Execute
■	File
■	GetCatalogs
■	GetTables
■	Query
■	QueryArray
■	QueryArrayHash
■	QueryReport
■	QueryRow
■	QueryRowHash
■	QueryValue
■	RollBackTrans
■	SqlNextResult
■	Table

1.7.3.8.2.1 TMariaDBDataSource:BeginTrans

Starts a new transaction.

Type	Standard
Parameters	None
Return value	NIL

1.7.3.8.2.2 TMariaDBDataSource:CommitTrans

Saves any changes and ends the current transaction. It may also start a new transaction.

Type	Standard
-------------	----------

Parameters	None
Return value	NIL

1.7.3.8.2.3 TMariaDBDataSource:Connect

Connects to the database.

Type	Standard
Parameters	[<cConnect>] String connection.
Return value	<ISuccess> True if success

1.7.3.8.2.4 TMariaDBDataSource:CreateTable

Creates a new table on the database.

Type	Standard
Parameters	<cTable> Table name <aStruct> Table structure. Uses the same format that DBCreate() xHarbour function, except that if you include an asterisk after the type definition, that field will be considered also part of the primary key. [<cEngine>] Database internal engine. See MariaDB documentation.
Return value	<ISuccess> True if success

1.7.3.8.2.5 TMariaDBDataSource:DelTable

Deletes a table from the database.

Type	Standard
Parameters	<cTableName> Table name
Return value	<ISuccess> True if success

1.7.3.8.2.6 TMariaDBDataSource:Disconnect

Disconnects from the database

Type	Standard
Parameters	None
Return value	<ISuccess> True if success

1.7.3.8.2.7 TMariaDBDataSource:Execute

Executes a command on the database.

Type	Standard
Parameters	<cCommand> Command to execute [<cOperation>] Operation description to be shown in case of a run-time error [<@aData>] Array with all the values returned by the instruction in case is a SELECT sentence
Return value	<ISuccess> True if success

After a call to method Execute, the property nAffectedRows returns the number of affected rows.

1.7.3.8.2.8 TMariaDBDataSource:File

Indicates if table name exists in the database.

Type	Standard
Parameters	<cTableName> Table name
Return value	<ISuccess> True if exists

1.7.3.8.2.9 TMariaDBDataSource:GetCatalogs

Returns all the databases or catalogs on the server.

Type	Standard
Parameters	None
Return value	<aDatabases> Array with all the database names

1.7.3.8.2.10 TMariaDBDataSource:GetTables

Return all the tables and views of a database.

Type	Standard
Parameters	<cMask> Mask to use for searching [<IView>] If true the views will be also included. By default .T.
Return value	<aTables> Array with all the view and table names

1.7.3.8.2.11 TMariaDBDataSource:Query

Creates an TDataset object through a Select instruction in the database.

Type	Standard
Parameters	<cSelect> SELECT instruction [<cProcess>] Process name. Default value: the value assigned by the method.NewProcess
Return value	<oDataset> TDataset object

1.7.3.8.2.12 TMariaDBDataSource:QueryArray

Returns a multidimensional array with all the query information received through a SELECT query in the database.

Type	Standard
Parameters	<cSelect>

	SELECT instruction [<@aHeaders>] Sets the array with the retrieved column names
Return value	<aData>

1.7.3.8.2.13 TMariaDBDataSource:QueryArrayHash

Returns a hash with all the query information received through a SELECT query in the database.

Type	Standard
Parameters	<cSelect> SELECT instruction [<@aHeaders>] Sets the array with the retrieved column names
Return value	<aData>

1.7.3.8.2.14 TMariaDBDataSource:QueryReport

Creates an TDataset object through a Select instruction in the database but optimized for being used in reports and no suitable for browses. It creates a read only and move forward dataset.

Type	Standard
Parameters	<cSelect> SELECT instruction [<cProcess>] Process name. Default value: the value assigned by the method.NewProcess
Return value	<oDataset> TDataset object

1.7.3.8.2.15 TMariaDBDataSource:QueryRow

Returns an array with the information received through a SELECT query in the database.

Type	Standard
Parameters	<cSelect> SELECT instruction [<@aHeaders>] Sets the array with the retrieved column names [aDefault] Default value in case a error is raised
Return	<aData>

value

If the select instruction returns more than one row , this method will recover only the first cursor row. The number of elements in the array will be the same as the number of fields in the query.

1.7.3.8.2.16 TMariaDBDataSource:QueryRowHash

Returns a hash with the information received through a SELECT query in the database.

Type	Standard
Parameters	<cSelect> SELECT instruction [<aHeaders>] Sets the array with the retrieved column names [aDefault] Default value in case a error is raised
Return value	<aData>

If the select instruction returns more than one row , this method will recover only the first cursor row.

1.7.3.8.2.17 TMariaDBDataSource:QueryValue

Returns the result value from the information received through a SELECT instruction in the database.

Type	Standard
Parameters	<cSelect> SELECT instruction [uDefault] Default value in case a error is raised
Return value	<Value>

If the select instruction returns more than one row or multiple columns, this method will recover only the first column from the first cursor row.

Example:

```
nTotal := oDataSource:QuerValue( "COUNT(*) FROM Customer " )
```

1.7.3.8.2.18 TMariaDBDataSource:RollBackTrans

Cancels any changes made during the current transaction and ends the transaction. It may also start a new transaction..

Type	Standard
Parameters	None
Return value	NIL

1.7.3.8.2.19 TMariaDBDataSource:SqlNextResult

Retrieves the information of next SELECT statement on a stored procedure called through the Execute method when the stored procedure has more than one SELECT statement.

Type	Standard
Parameters	[<@aData>] Information array on output. If no value given the next result is discarded. [<@aHeaders>] Array with field names
Return value	<ISuccess> True if success

1.7.3.8.2.20 TMariaDBDataSource:Table

Creates a TDataset object just indicating ist table name on the database.

Type	Standard
Parameters	<cTableName> Table name [<cProcess>] Process name. Default value: the value assigned by the method.NewProcess
Return value	<oDataset> TDataset object

1.7.3.9 TSQLiteDataSource

Class to manage SQLite databases.

SQLite is a embedded database engine. The complete engine is inserted on our own Xailer application, so there is no need to deliver any additional software or DLL to make it work. Its use

is completely free for any use, even commercial use. It may be used on LAN's, but obviously when the number of concurrent users increases is preferable to use a independent database engine. For further information consult the following link:

<http://en.wikipedia.org/wiki/SQLite>

Importante note: Encryption is only available on Xailer Enterprise.

You should include the **SQLite.lib** library on any project that uses this database.

The SQLite engine does not have a date or date&time basic type like other engines. Many SQLite programmers use to save dates and times in normal string fields. Though, Xailer uses a better approach, which is the same that uses the rest of the SQL engines, which consists on saving the complete date and time on a float basic data type which only uses 8 bytes. Due that reason, the date&time of SQLite made with Xailer are recognized as simple numeric fields by other SQLite editors. If you do not want this functionality you may avoid it with the property `ldateAsString` that forces that all dates are saved as strings. In order to use the Xailer date format inside your own database triggers we have aggregate some function to SQLite when used in conjunction with Xailer, which are:

- **local_timestamp() -> <cDatetime>**
It returns the local date in string format. SQLite already has a function for this purpose, though, it fails with Windows operating systems on summer time periods.
- **ctod(<cDate>) -> <nDate>**
It converts a date&time value from string to numeric
- **dtoc(<nDate>) -> <cDate>**
It converts a date&time value from numeric to string
- **year(<nDate>|<cDate>) -> <nYear>**
Returns the year of a numeric or string date.
- **month(<nDate>|<cDate>) -> <nMonth>**
Returns the month of a numeric or string date.
- **day(<nDate>|<cDate>) -> <nDay>**
Returns the day of a numeric or string date.
- **quarter(<nDate>|<cDate>) -> <nQuarter>**
Returns the quarter of a numeric or string date.
- **concat(<cString1>, ..., <cStringN>) -> <cString>**
Returns the concatenation of all the parameters passed.
- **lpad(<cString>, <nLen>, <cPadStr>) -> <cString>**
Returns the string **cString**, left-padded with the string **cPadStr** to a length of **len** characters. If **cString** is longer than **nLen**, the return value is shortened to **nLen** characters.
- **rpad(<cString>, <nLen>, <cPadStr>) -> <cString>**
Returns the string **cString**, right-padded with the string **cPadStr** to a length of **len** characters. If **cString** is longer than **nLen**, the return value is shortened to **nLen** characters.

See also 'Introduction to DataControls' for more information.

Hierarchy	Inherits from TDataSource
File Name	\source\SQLiteDataSource.prg

See also TDataSet, DataControls

1.7.3.9.1 TSQLiteDataSource:Properties

■ read Only ■ Assignable ■ Design assignable ■ Run-time assignable

Scope	Name	Type	Initial Value
■	cConnect	Character	" "
■	cPassword	Character	""
■	IDateAsString	Logical	.F.
■	IDoubleQuotes	Logical	.F.
■	IReadToCache	Logical	.T.
■	nAffectedRows	Numeric	0

1.7.3.9.1.1 TSQLiteDataSource:cConnect

String connection to the database. Each engine has its own syntax. Is preferable to use the assistant include it on the Xailer object inspector which will guide to the all process.

Scope	Assignable before the connection to the server
Type	Character
Initial value	""

1.7.3.9.1.2 TSQLiteDataSource:cPassword

Database encryption string. It must be assigned on encrypted databases.

Scope	Assignable
Type	Character
Initial value	""

You must previously encrypt the database with the method Encrypt

1.7.3.9.1.3 TSQLiteDataSource:IDateAsString

Setting this property to true the fields of type 'DATE' and 'DATETIME' will be saved internally on the database as strings.

SQLite does not have a specific field type for dates management, for that reason each programming environment has make its own system. By default Xailer uses only 8 bytes to save the date & time information like most other DBMS do, to save space and to make arithmetical operations with dates a lot easier. Though, other programming environments have choose to save the dates as string with this SQL standard format: 'AAAA-MM-DD HH:MM:SS'

Scope	Assignable
--------------	------------

Type	logical
Initial value	.F.

Select statements change considerably depending on the system used. This search will correctly work with **IDateAsString** set to true:

```
SELECT ... WHERE hiredate > '1990-01-01'
```

However, if **IDateAsString** is set to false, you must use:

```
SELECT ... WHERE hiredate > ctod( '01/01/1980' )
```

Be aware that for this to function we needed to include on SQLITE core functions the following xBase functions:

- local_timestamp()
- dtoc()
- ctod()
- year()
- month()
- day()
- quarter()

For compatibility with MySQL the following functions are included also:

- concat()
- lpad()
- rpad()

If you use the formula of **IDateAsString** to true SQLITE offers some very useful functions:

- date(timestring, modifier, modifier, ...)
- time(timestring, modifier, modifier, ...)
- datetime(timestring, modifier, modifier, ...)
- julianday(timestring, modifier, modifier, ...)
- strftime(format, timestring, modifier, modifier, ...)

If you want to change the date system on a specific database you must do it table by table and in all its date fields. The easiest way to do it is with this single UPDATE command:

To transform from numeric to string format:

```
UPDATE <Table> SET <Field> = year(<Field>) || '-' || printf('%02i',month(<Field>)) || '-' || printf('%02i',day(<Field>))
```

To transform from string to numeric format:

```
UPDATE <Table> SET <Field> = Ctod(<Field>)
```

For further information follow this link: [Funciones de fecha en Sqlite](#)

1.7.3.9.1.4 TSQLiteDatabaseSource:IDoubleQuotes

Setting this property to true the permits to use double quotes on 'Select' statement for compatibility with older versions of SQLite.

Scope	Assignable on design (before connection)
Type	logical
Initial value	.F.

1.7.3.9.1.5 TSQLiteDatabaseSource:IReadToCache

Setting this property to true the complete database is read on the connection process so its available on system cache.

Scope	Assignable on design (before connection)
Type	logical
Initial value	.T.

1.7.3.9.1.6 TSQLiteDatabaseSource:nAffectedRows

Number of rows affected by the last INSERT, UPDATE or DELETE operation done through the Execute() method.

Scope	Read only
Type	Numeric
Initial value	0

1.7.3.9.2 TSQLiteDatabaseSource:Methods

■ Constructor ■ Standard

Type	Name
■	BeginTrans
■	CommitTrans
■	Connect
■	DelTable
■	Disconnect
■	Encrypt
■	Execute
■	File
■	GetTables
■	Query
■	QueryArray

■	QueryArrayHash
■	QueryReport
■	QueryRow
■	QueryRowHash
■	QueryValue
■	RollBackTrans
■	Table

1.7.3.9.2.1 TSQLiteDataSource:BeginTrans

Starts a new transaction.

Type	Standard
Parameters	None
Return value	NIL

1.7.3.9.2.2 TSQLiteDataSource:CommitTrans

Saves any changes and ends the current transaction. It may also start a new transaction.

Type	Standard
Parameters	None
Return value	NIL

1.7.3.9.2.3 TSQLiteDataSource:Connect

Connects to the database.

Type	Standard
Parameters	[<cConnect>] String connection. By default cConnect
Return value	<ISuccess> True if success

1.7.3.9.2.4 TSQLiteDataSource:DelTable

Deletes a table from the database.

Type	Standard
Parameters	<cTableName> Table name
Return value	<ISuccess> True if success

1.7.3.9.2.5 TSQLiteDataSource:Disconnect

Disconnects from the database

Type	Standard
Parameters	None
Return value	<ISuccess> True if success

1.7.3.9.2.6 TSQLiteDataSource:Encrypt

Encrypts, reencrypts or unencrypts a database. If the parameter cPassword is not empty, the database is encrypted or reencrypted with that string, if the parameter is blank the database is unencrypted.

Type	Standard
Parameters	<cPassword> Password string
Return value	<ISuccess> True if success

1.7.3.9.2.7 TSQLiteDataSource:Execute

Executes a command on the database.

Type	Standard
Parameters	<cCommand> Command to execute [<cOperation>] Operation description to be shown in case of a run-time error [<@aData>] Array with all the values returned by the instruction

	in case is a SELECT sentence
Return value	<ISuccess> True if success

After a call to method Execute, the property nAffectedRows returns the number of affected rows.

1.7.3.9.2.8 TSQLiteDataSource:File

Indicates if table name exists in the databse.

Type	Standard
Parameters	<cTableName> Table name
Return value	<ISuccess> True if exists

1.7.3.9.2.9 TSQLiteDataSource:GetTables

Return all the tables and views of a database.

Type	Standard
Parameters	<cMask> Mask to use for searching [<IView>] If true the views will be also included. By default .T.
Return value	<aTables> Array with all the view and table names

1.7.3.9.2.10 TSQLiteDataSource:Query

Creates an TDataset object through a Select instruction in the database.

Type	Standard
Parameters	<cSelect> SELECT instruction [<cProcess>] Process name. Default value: the value assigned by the method.NewProcess
Return value	<oDataset> TDataset object

1.7.3.9.2.11 TSQLiteDatabaseSource:QueryArray

Returns a multidimensional array with all the query information received through a SELECT query in the database.

Type	Standard
Parameters	<cSelect> SELECT instruction [<@aHeaders>] Sets the array with the retrieved column names
Return value	<aData>

1.7.3.9.2.12 TSQLiteDatabaseSource:QueryArrayHash

Returns a hash with all the query information received through a SELECT query in the database.

Type	Standard
Parameters	<cSelect> SELECT instruction [<@aHeaders>] Sets the array with the retrieved column names
Return value	<aData>

1.7.3.9.2.13 TSQLiteDatabaseSource:QueryReport

Creates an TDataset object through a Select instruction in the database but optimized for being used in reports and no suitable for browses. It creates a read only and move forward dataset.

Type	Standard
Parameters	<cSelect> SELECT instruction [<cProcess>] Process name. Default value: the value assigned by the method.NewProcess
Return value	<oDataset> TDataset object

1.7.3.9.2.14 TSQliteDataSource:QueryRow

Returns an array with the information received through a SELECT query in the database.

Type	Standard
Parameters	<cSelect> SELECT instruction [<@aHeaders>] Sets the array with the retrieved column names [aDefault] Default value in case a error is raised
Return value	<aData>

If the select instruction returns more than one row , this method will recover only the first cursor row. The number of elements in the array will be the same as the number of fields in the query.

1.7.3.9.2.15 TSQliteDataSource:QueryRowHash

Returns a hash with the information received through a SELECT query in the database.

Type	Standard
Parameters	<cSelect> SELECT instruction [<@aHeaders>] Sets the array with the retrieved column names [aDefault] Default value in case a error is raised
Return value	<aData>

If the select instruction returns more than one row , this method will recover only the first cursor row.

1.7.3.9.2.16 TSQliteDataSource:QueryValue

Returns the result value from the information received through a SELECT instruction in the database.

Type	Standard
Parameters	<cSelect> SELECT instruction [uDefault] Default value in case a error is raised
Return value	<Value>

If the select instruction returns more than one row or multiple columns, this method will recover only the first column from the first cursor row.

Example:

```
nTotal := oDataSource:QuerValue( "COUNT(*) FROM Customer " )
```

1.7.3.9.2.17 TSQLiteDataSource:RollBackTrans

Cancels any changes made during the current transaction and ends the transaction. It may also start a new transaction..

Type	Standard
Parameters	None
Return value	NIL

1.7.3.9.2.18 TSQLiteDataSource:Table

Creates a TDataset object just indicating ist table name on the database.

Type	Standard
Parameters	<cTableName> Table name [<cProcess>] Process name. Default value: the value assigned by the method.NewProcess
Return value	<oDataset> TDataset object

1.7.3.10 TWebDataSource

Control to manage databases through a Web connection. It lets you access any database remote server using a simple HTTP or HTTPS connection.

Your must include the [x]Harbour library **Tip.lib** on all the projects that you use this control.

The idea is to use Xailer with any database located on a external server using basic HTTP/s connections. The reasons for doing this are mainly two:

1. The ability to create desktop applications with Xailer that access to databases located on Web servers that normally you can not access remotely, but only through a local connection using PHP or whatever.
2. Avoid the use of DLLs, ODBC driver or ADO to connect to any database.

In order to use this control you should take of the following requirements:

- A Web server accesible by your application
- PHP 5.0 or above installed on the Web server
- Database engine accesible through PHP
- Privileges on the database scheme you want to use
- FTP access to the Web server to upload Xailer PHP module

The TWebDataSource control relies on a PHP module that must be uploaded to your Web server. That module is that really connects locally with the database and keeps all traffic with your application. Depending on the database you must use a different module. At present there is only the MySQL module, but the control is prepared to be used by others database engines that will be released in the near future. All the modules are located at the '**Xailer\Source\Enterprise\PHP**' folder.

In order to avoid non authorized use of the PHP module except from our own application and also to avoid that some sensible information like passwords, navigates in Internet, we have included some features:

- All the requests have a validation mechanism that prevents fraudulent use and of course any SQL injection. With any request a validation key is sent that must also be validated by the PHP Module. If the validation key is incorrect the command will not be processed at all. That validation key is based on a encryption string and the command requested. The encryption string must be changed on your application using the cCryptKey property. The value assigned on that property must also be assigned inside the PHP module with exactly the same value. The algorithm used to create the validation key can also be modified through the OnValidate event.
- To prevent the traffic of sensible information through Internet, some of the connection parameters must been set directly on the PHP module, like the database user name and its password. This way is impossible that anyone can access to your sensible information..

Therefore, is necessary to edit the PHP module before uploading it through FTP to your Web server and set the following values:

- Encryption code
- Database user name
- Database password

However this is a extremely easy task since the three variables are defined at the beginning of the module:

```
/*
You must set this values accordly with your application & BD
*/

$cDBUserName = 'neptuno_ro';
$cDBPassword = 'neptuno';
$cCryptKey   = 'change this value';
```

See also 'Introducción a los DataControls' for more information.

Hierarchy	Inherits from TDataSource
File	\source\Enterprise\WebDataSource.prg
See also	DataControls, \Datacontrols\WebDataset sample

1.7.3.10.1 TWebDataSource:Properties

■ Read only
 ■ Assignable
 ■ Design assignable
 ■ Run-time assignable

Scope	Name	Type	Initial Value
■	cConnect	Character	""
■	cCryptKey	Character	"change this value"
■	cDatabase	Character	""
■	cLastError	Character	""
■	cPhpModule	Character	"/xa_wdsMySql.php"
■	ITinyIntAsBoolean	Logical	.T.
■	nDBType	Numeric	dtMYSQL
■	nAffectedRows	Numeric	0
■	nInsertId	Numeric	0
■	nPort	Numeric	80
■	nSendTimeOut	Numeric	30

1.7.3.10.1.1 TWebDataSource:cConnect

Server name or IP.

Scope	Assignable before connecting to Web server
Type	Character
Initial value	""

1.7.3.10.1.2 TWebDataSource:cCryptKey

Encryption chain to be used for communications between your application and the web server. For further information consult the following link.

Scope	Assignable before connecting to Web server
Type	Character
Initial value	"change this value"

1.7.3.10.1.3 TWebDataSource:cDatabase

Database name.

Scope	Assignable before connecting to Web server
Type	Character
Initial value	""

1.7.3.10.1.4 TWebDataSource:cLastError

Description of last error.

Scope	Read only
Type	Character
Initial value	""

1.7.3.10.1.5 TWebDataSource:cPhpModule

Name of the PHP module to be used on the Web server. That module should be modified to establish some parameters: encryption chain and database name and password. That information should be included at the beginning of the module. All the Xailer standard PHP modules can be found under **SourceEnterprisePHP** directory. This property should be set jointly with the nDBType property.

For further information consult the following link.

Scope	Assignable before connecting to Web server
Type	Character
Initial value	"/xa_wdsMySql.php"

1.7.3.10.1.6 TWebDataSource:ITinyIntAsBoolean

If true, TinyInt fields are treated as boolean. In **MySQL** there is no boolean data type and TinyInt is used instead where a value of zero is false and any other value is true. However there are occasions where TinyInt are treated like real numbers.

Scope	Assignable before connecting to Web server
Type	Logical
Initial value	.T.

1.7.3.10.1.7 TWebDataSource:nDBType

Database type to use. This property should be set jointly with the cPhpModule property.

For further information consult the following link.

Scope	Assignable before connecting to Web server
Type	Numeric
Initial value	dtMYSQL
Valores posibles	dtMYSQL, dtSQLITE, dtPOSTGRES, dtORACLE, dtSQLSERVER, dtFIREBIRD

Note: At present it only supports dtMYSQL

1.7.3.10.1.8 TWebDataSource:nAffectedRows

Number of rows affected on last INSERT, UPDATE or DELETE operation.

Scope	Read only
Type	Numeric
Initial value	0

1.7.3.10.1.9 TWebDataSource:nInsertId

Last value introduced on the autoincremental primary key field with latest INSERT operation.

Scope	Read only
Type	Numeric
Initial value	0

1.7.3.10.1.10 TWebDataSource:nPort

Timeout in seconds for email delivery.

Scope	Assignable
Type	Numeric
Initial value	30

1.7.3.10.1.11 TWebDataSource:nSendTimeOut

Server connection port.

Scope	Assignable before connecting to Web server
Type	Numeric
Initial value	80

1.7.3.10.2 TWebDataSource:Methods

■ Constructor ■ Standard

Type	Name
■	BeginTrans
■	BulkExecute
■	CommitTrans
■	Connect
■	CurrentDir
■	DeleteFile
■	Disconnect
■	Execute
■	File
■	ForceDir
■	GetCatalogs
■	GetTables
■	Query
■	IsDir
■	QueryArray
■	QueryArrayHash
■	QueryReport
■	QueryRow
■	QueryRowHash
■	QueryValue
■	MkDir
■	MyIP
■	MyLocation
■	RollBackTrans
■	SendMail
■	Table
■	UpdateFile

1.7.3.10.2.1 TWebDataSource:BeginTrans

Marks the beginning of a new transaction. Unlike other **DataSources**, any **update operations on the database is not done** until a call to the CommitTrans method. Instead of doing the operations, they just simply get achieved and are processed by the BulkExecute method when the transaction ends.

This transaction simulation has a great advantage since it can performs a lot of database operations on a single HTTP request reducing network traffic and response time with absolutely no effort on the programmer.

Important note: Since all the update operations are delayed until the end of the transaction. Any consult operation will not reflect any change made by them.

Type	Standard
Parameters	None
Return value	NIL

1.7.3.10.2.2 TWebDataSource:BulkExecute

Executes one or more concatenated strings separated by semicolons. All the operations are done on a single transaction, so if any of them provokes an error the complete transaction will be rolledback.

Type	Standard
Parameters	<aCommand> Array of SQL strings to execute [<cOperation>] Operation description to be seen in case of an error
Return value	<ISuccess> True if success

After any call to BulkExecute, the property nAffectedRows will return the number of rows affected for all the commands executed.

1.7.3.10.2.3 TWebDataSource:CommitTrans

This method is used to commit a transaction started with the method BeginTrans. However its behaviour is atypical. For further information consult the method BeginTrans..

Type	Standard
Parameters	None
Return value	NIL

1.7.3.10.2.4 TWebDataSource:Connect

Makes the connection with the database

Type	Standard
Parameters	[<cConnect>] Name or IP direction. By default cConnect will be used
Return value	<ISuccess> True if success

1.7.3.10.2.5 TWebDataSource:CurrentDir

Returns the current directory in which the PHP module is installed.

Type	Standard
Parameters	[<IUtf>] If true converts the returned result from UTF to ANSI
Return value	<cDirectory>

1.7.3.10.2.6 TWebDataSource:Disconnect

Disconnects from the database.

Type	Standard
Parameters	None
Return value	<ISuccess> True if success

1.7.3.10.2.7 TWebDataSource>DeleteFile

Deletes a file on the Web server.

Type	Standard
Parameters	<cFilename> File name on the Web server. If you include any path, it must be relative and in accordance with the Web server directory structure.
Return value	<ISuccess> True if success

1.7.3.10.2.8 TWebDataSource:Execute

Executes a command on the database.

Type	Standard
Parameters	<cCommand> Command to execute [<cOperation>] Operation description to be shown in case of a run-time error [<@aData>] Array with all the values returned by the instruction in case is a SELECT sentence
Return value	<ISuccess> True if success

After a call to method Execute, the property nAffectedRows returns the number of affected rows.

1.7.3.10.2.9 TWebDataSource:File

Indicates if table name exists in the database.

Type	Standard
Parameters	<cTableName> Table name
Return value	<ISuccess> True if success

1.7.3.10.2.10 TWebDataSource:ForceDir

This method allows you to create a directory recursively until you create the complete 'path' indicated. The indicated path must be relative to the directory where the PHP module is located.

Type	Standard
Parameters	<cDir> Final directory to create [<IUtf>] If true converts the cDir parameter from ANSI to UTF
Return value	<ICreated> True if created correctly

1.7.3.10.2.11 TWebDataSource:GetCatalogs

Returns all the databases or catalogs on the server.

Type	Standard
Parameters	None
Return value	<aDatabases> Array with all the database names

1.7.3.10.2.12 TWebDataSource:GetTables

Return all the tables and views of a database.

Type	Standard
Parameters	<cMask> Mask to use for searching [<IView>] If true the views will be also included. By default .T.
Return value	<aTables> Array with all the views and table names

1.7.3.10.2.13 TWebDataSource:Query

Creates TDataset object through a Select instruction in the database.

Type	Standard
Parameters	<cSelect> SELECT instruction [<cProcess>] Process name. Default value assigned by the method NewProcess
Return value	<oDataset> TDataset object

1.7.3.10.2.14 TWebDataSource:IsDir

Returns true if the directory passed as a parameter exists. The indicated path must be relative to the directory where the PHP module is located.

Type	Standard
Parameters	<cDir> Directory to check [<IUtf>]

	If true converts the cDir parameter from ANSI to UTF
Return value	<ICreated> True if created correctly

1.7.3.10.2.15 TWebDataSource:QueryArray

Returns a multidimensional array with all the query information received through a SELECT query in the database.

Type	Standard
Parameters	<cSelect> SELECT instruction [<@aHeaders>] Sets the array with the retrieved column names
Return value	<aData>

1.7.3.10.2.16 TWebDataSource:QueryArrayHash

Returns a hash with all the query information received through a SELECT query in the database.

Type	Standard
Parameters	<cSelect> SELECT instruction [<@aHeaders>] Sets the array with the retrieved column names
Return value	<aData>

1.7.3.10.2.17 TWebDataSource:QueryReport

Identical functionality than the Query method.

Type	Standard
Parameters	<cSelect> SELECT instruction [<cProcess>] Process name. Default value assigned by the method NewProcess
Return value	<oDataset> TDataset object

1.7.3.10.2.18 TWebDataSource:QueryRow

Returns an array with the information received through a SELECT query in the database.

Type	Standard
Parameters	<cSelect> SELECT instruction [<aHeaders>] Sets the array with the retrieved column names [aDefault] Default value in case a error is raised
Return value	<aData>

If the select instruction returns more than one row , this method will recover only the first cursor row. The number of elements in the array will be the same as the number of fields in the query.

1.7.3.10.2.19 TWebDataSource:QueryRowHash

Returns a hash with the information received through a SELECT query in the database.

Type	Standard
Parameters	<cSelect> SELECT instruction [<aHeaders>] Sets the array with the retrieved column names [aDefault] Default value in case a error is raised
Return value	<aData>

If the select instruction returns more than one row , this method will recover only the first cursor row.

1.7.3.10.2.20 TWebDataSource:QueryValue

Returns the result value from the information received through a SELECT instruction in the database.

Type	Standard
Parameters	<cSelect> SELECT instruction [uDefault] Default value in case a error is raised
Return value	<Value>

If the select instruction returns more than one row or multiple columns, this method will recover only the first column from the first cursor row.

Example:

```
nTotal := oDataSource:QuerValue( "COUNT(*) FROM Customer " )
```

1.7.3.10.2.21 TWebDataSource:MkDir

This method allows you to create a directory. The indicated path must be relative to the directory where the PHP module is located.

Type	Standard
Parameters	<cDir> Directory to create [<IUtf>] If true converts the cDir parameter from ANSI to UTF
Return value	<ICreated> True if created correctly

1.7.3.10.2.22 TWebDataSource:MyIP

Returns the **IP** which has connect to the data source

Type	Standard
Parameters	None
Return value	<cLocation>

1.7.3.10.2.23 TWebDataSource:MyLocation

Returns the location of the IP which has connect to the data source

Type	Standard
Parameters	None
Return value	<cLocation>

1.7.3.10.2.24 TWebDataSource:RollBackTrans

This method is used to roll back a transaction started with the method `BeginTrans`. However its behaviour is atypical. For further information consult the method `BeginTrans`..

Type	Standard
-------------	----------

Parameters	None
Return value	NIL

1.7.3.10.2.25 TWebDataSource:SendMail

This method permits to send email messages using the web server.

Type	Standard
Parameters	<cTo> : Recipient email account [<cName>] : Recipient email name [<cSubject>] : Message subject [<cBody>] : Message body [<IHtml>] : If true, the format of the message body is in HTML format
Return value	True if success

In case of error, consult the property cLastError.

1.7.3.10.2.26 TWebDataSource:Table

Creates a TDataset object just indicating its table name on the database.

Type	Standard
Parameters	<cTableName> Table name [<cProcess>] Process name. Default value assigned by the method NewProcess
Return value	<oDataset> TDataset object

1.7.3.10.2.27 TWebDataSource:UpdateFile

Creates or updates a file on the Web server.

Type	Standard
Parameters	<cFilename> File name on the Web server. If you include any path, it must be relative and in accordance with the Web server directory structure.

	<cData> Data to include on the file
Return value	<ISuccess> True if success

1.7.3.10.3 TWebDataSource:Events

Name	
	OnValidate

1.7.3.10.3.1 TWebDataSource:OnValidate

Event produced when the validation key is going to be generated based on the encryption string `cCryptKey` and the command to process.

This event permits to modify the algorithm used by Xailer to get the validation key by other done by yourself. Keep in mind that if you modify this algorithm you must also modify the `XA_Validate()` function of the PHP module.

Parameters	<oSender> : Object that triggers the event <cData> : String to process
Return value	<nValidKey>

1.7.3.11 TOdbcDataSource

Class to manage data trough an ODBC connection.

Description:

This class appears in the components palette in the DataSets tab with the "ODBC" name and allows to connect to any database engine trough an ODBC client.

You should install the ODBC client in your development environment and in the final environment where the application will be executed. You should configure in both environments a connection or a DSN (Data Source Name) to the database with the ODBC client and the operating system tool used to configure the ODBC connection through the control panel. For more information, see also the ODBC documentation that includes information about the ODBC client and the operating system.

For more information, see also the 'Introduction to DataControls' chapter.

Hierarchy	Inherits from TDbfDataSource
File Name	\source\OdbcDatasource.prg
See also	TOdbcDataSet, DataControls

1.7.3.11.1 TDbcDataSource:Properties

read Only
 Assignable
 Design assignable
 Run-time assignable

Scope	Name	Type	Initial value
<input type="checkbox"/>	cConnect	Character	."
<input type="checkbox"/>	cDSN	Character	""
<input type="checkbox"/>	cPassword	Character	""
<input type="checkbox"/>	cUser	Character	""
<input type="checkbox"/>	IAutoCommit	Logic	.T.
<input type="checkbox"/>	IDateAsString	Logic	.T.
<input type="checkbox"/>	nAffectedRecords	Logic	.F.

1.7.3.11.1.1 TDbcDataSource:cConnect

String connection to the database.

Scope	Assignable before the connection to the server
Type	Character
Initial value	""

String that is received by the ODBC client to make the connection. If the string is empty when you try to connect, an operating system dialog screen will be shown requesting all the data needed for the connection. In the case that at least the DSN cDSN information has been provided, it is possible that the ODBC client shows other dialog screen requesting the Name, code and some additional information.

Check the first time you connect from the IDE assigning the IConnected property to .T., the cConnect property will be filled automatically.

1.7.3.11.1.2 TDbcDataSource:cDSN

ODBC Data Source Name. Name that is given to the connection in the ODBC configuration dialog. For more information, check the operating system ODBC documentation and the information provided by the database engine which you want to connect to.

Scope	Assignable before the connection to the server
Type	Character
Initial value	""

For more information about the RDD, check the [x]Harbour documentation.

1.7.3.11.1.3 TOdbcDataSource:cPassword

Password to connect to the server. If it is not specified, it is possible that the ODBC control request it in run-time.

Scope	Assignable before the connection to the server
Type	Character
Initial value	""

1.7.3.11.1.4 TOdbcDataSource:cUser

User name to connect to the server. If it is not specified, it is possible that the ODBC control request it in run-time.

Scope	Assignable before the connection to the server
Type	Character
Initial value	""

1.7.3.11.1.5 TOdbcDataSource:lAutoCommit

It is .T. if in every update operation the data are saved in the database.

Scope	Assignable before the connection to the server
Type	Logic
Initial value	.T.

When lAutocommit is .F. all the pending operations are not saved to disk until a CommitAll is performed. RollBackAll revokes all the pending operations.

1.7.3.11.1.6 TOdbcDataSource:lDateAsString

If it is .T., the date fields will be shown as a string. This property is considered for those date fields that includes time information as well.

Scope	Assignable before the connection to the server
Type	Logic
Initial value	.T.

1.7.3.11.1.7 TOdbcDataSource:nAffectedRecords

Number of rows affected by the last INSERT, UPDATE or DELETE operation executed through the Execute() method.

Scope	read Only
Type	Numeric
Initial value	0

1.7.3.11.2 TOdbcDataSource:Methods

■ Constructor ■ Standard

Type	Name
■	CommitAll
■	DelTable
■	Execute
■	File
■	GetTables
■	Query
■	QueryArray
■	QueryRow
■	QueryValue
■	RollBackAll

1.7.3.11.2.1 TOdbcDataSource:CommitAll

Saves all the pending data .

Type	Standard
Parameters	None
Return value	<ISuccess> .T. if the operation is successful

When IAutocommit is .F. all the pending operations are not saved to disk until a CommitAll is performed. RollBackAll revokes all the pending operations.

1.7.3.11.2.2 TOdbcDataSource:DelTable

Deletes the **cTable** table.

Type	Standard
Parameters	<cTable>

	Table name
Return value	<ISuccess> .T. if the operation is successful

1.7.3.11.2.3 TOdbcDataSource:Execute

Execute a SQL command through the ODBC engine.

Type	Standard
Parameters	<cCommand> Command to execute [<cOperation>] Describing name for the operation to be shown in the case that an error is produced.
Return value	<ISuccess> .T. if the operation is successful

1.7.3.11.2.4 TOdbcDataSource:File

Returns .T. if exists the table or view.

Type	Standard
Parameters	<cTable> Table name
Return value	<IExist> .T. if the table exists

1.7.3.11.2.5 TOdbcDataSource:GetTables

Returns an array with all the existing tables.

Type	Standard
Parameters	<cMask> Search table mask [<IView>] .T. to include views. Default value: .T.
Return value	<aTables> Array with existing tables or views

1.7.3.11.2.6 TOdbcDataSource:Query

Creates an TOdbcDataSet object through a Select instruction in the database.

Type	Standard
Parameters	<cSelect> SELECT instruction [<cProcess>] Process name. Default value: the value assigned by the method.NewProcess
Return value	<oOdbcDataSet> TOdbcDataSet object

1.7.3.11.2.7 TOdbcDataSource:QueryArray

Returns a multidimensional array with all the query information received through a SELECT query in the database.

Type	Standard
Parameters	<cSelect> SELECT instruction [<@aHeaders>] Sets the array with the retrieved column names
Return value	<aData>

1.7.3.11.2.8 TOdbcDataSource:QueryRow

Returns an array with the information received through a SELECT query in the database.

Type	Standard
Parameters	<cSelect> SELECT instruction [<@aHeaders>] Sets the array with the retrieved column names
Return value	<aData>

If the select instruction returns more than one row , this method will recover only the first cursor row. The number of elements in the array will be the same as the number of fields in the query.

1.7.3.11.2.9 TDataSource:QueryValue

Returns the result value from the information received through a SELECT instruction in the database.

Type	Standard
Parameters	<cSelect> SELECT instruction
Return value	<Value>

If the select instruction returns more than one row or multiple columns, this method will recover only the first column from the first cursor row.

Example:

```
nTotal := oDataSource:QueryValue( "COUNT(*) FROM Customer " )
```

1.7.3.11.2.10 TDataSource:RollBackAll

Recovers the database status from the last update operation.

Type	Standard
Parameters	None
Return value	<ISuccess> .T. if the operation is successful

When lAutocommit is .F. all the pending operations are not saved to disk until a CommitAll is performed. RollBackAll revokes all the pending operations.

1.7.4 Data Sets

1.7.4.1 TDataField

Class to manage database fields.

Description:

This class is used mainly for the 'DataControls' controls and the TDataSet and descending classes. The TDataField objects will act as containers for all the typical database fields properties and includes the needed functionality to recover and save data in their origin tables.

Hierarchy	Inherits from TComponent
File Name	\source\DataField.prg
See also	TDataSet, TDataSource, DataControls

1.7.4.1.1 TDataFieldProperties

■ read Only ■ Assignable ■ Design assignable ■ Run-time assignable

Scope	Name	Type	Initial value
■	Buffer *	Any	NIL
■	cDisplayName	Character	""
■	cFullName	Character	""
■	cName	Character	""
■	cTable	Character	""
■	cType	Character	""
■	DefValue	Any	NIL
■	InitValue	Any	NIL
■	IAllowsNull	Logic	.F.
■	IEditable	Logic	.F.
■	IModified	Logic	.F.
■	IPrimaryKey	Logic	.F.
■	ISetNullIfEmpty	Logic	.F.
■	nDec	Numeric	0
■	nIndex	Numeric	0
■	nLen	Numeric	0
■	nSQLDec	Numeric	0
■	nSQLLen	Numeric	0
■	nSQLType	Character	""
■	oDataSet	Object	NIL
■	Value	Any	NIL

1.7.4.1.1.1 TDataField:Buffer

Buffer is used to temporary save the information introduced for the user in append and edit operations. It is not needed to access or assign this data directly because the linked Data Control will be responsible to update it.

Scope	Assignable
Type	Character, Date, Numeric o Logic
Initial value	NIL

*** Obsolete property, maintained only for compatibility purposes. You must use the property Value instead.**

1.7.4.1.1.2 TDataField:cDisplayName

Describing filed name to be used for other controls, like TDBBrowse. The default value of this property is the field name value from where it proceeds. However its value is the same as the cName property. This property allows to use a more descriptive text that will be shown as column header in the TDBBrowse controls type.

Scope	Assignable
Type	Character
Initial value	""

1.7.4.1.1.3 TDataField:cFullName

Internal field name prefixed by its table name.

Scope	read Only
Type	Character
Initial value	""

1.7.4.1.1.4 TDataField:cName

Internal field name.

Scope	read Only
Type	Character
Initial value	""

1.7.4.1.1.5 TDataField:cType

Field type: [C]haracter, [N]umeric, [L]ogic, or [D]ate.

Scope	read Only
Type	Character
Initial value	""

1.7.4.1.1.6 TDataField:cTable

Internal field table name.

Scope	read Only
Type	Character
Initial value	""

1.7.4.1.1.7 TDataField:DefValue

Default field value in append operations.

Scope	Assignable
Type	Character, Date, Numeric o Logic
Initial value	NIL

1.7.4.1.1.8 TDataField:InitValue

Initial field value.

Scope	read Only
Type	Character, Date, Numeric o Logic
Initial value	NIL

1.7.4.1.1.9 TDataField:!AllowsNull

It is .T. if the field accepts null values. For DBF type tables, this property is not valid because those tables don't accept null values.

Scope	read Only
Type	Logic
Initial value	.F.

1.7.4.1.1.10 TDataField:IEditable

It is .T. if the field is editable. This property is normally false when the field is not from the main table of the query or is a calculated field.

Scope	read Only
Type	Logic
Initial value	.F.

1.7.4.1.1.11 TDataField:IModified

Returns true if the field value has been modified.

You can force the value of this property assigning its value by hand. This way, you force the value it will return. Be aware that its state gets reset every time the dataset goes into edit mode.

Scope	Assignable
Type	Logic
Initial value	.F.

1.7.4.1.1.12 TDataField:IPrimaryKey

Returns true if the field is part of the primary key.

Scope	Read only
Type	Logic
Initial value	.F.

1.7.4.1.1.13 TDataField:ISetNullIfEmpty

If true and the field value is empty, a NULL value will be saved on the database.

Scope	Assignable
Type	Logic
Initial value	.F.

Note: Useless for DBF tables.

1.7.4.1.1.14 TDataField:nDec

Number of decimal places for the numeric fields.

Scope	read Only
Type	Numeric
Initial value	0

1.7.4.1.1.15 TDataField:nIndex

Field number in the table.

Scope	read Only
Type	Numeric
Initial value	0

1.7.4.1.1.16 TDataField:nLen

Total field length.

Scope	read Only
Type	Numeric
Initial value	0

1.7.4.1.1.17 TDataField:nSQLDec

Number of decimal places for the numeric fields in the SQL engines. See also the file **.sql.ch** from [x]Harbour for more information.

Scope	read Only
Type	Numeric
Initial value	0

1.7.4.1.1.18 TDataField:nSQLLen

Total field length in SQL engines. See also the file **.sql.ch** from [x]Harbour for more information.

Scope	read Only
Type	Numeric
Initial value	0

1.7.4.1.1.19 TDataField:nSQLType

Field type in the origin table in SQL engines. This value represents the field data type using the SQL engine own convention. In some cases this property may be of type string.

Scope	read Only
Type	Numeric
Initial value	0

1.7.4.1.1.20 TDataField:oDataSet

TDataSet proprietary object. It is the object that creates the TDataField.

Scope	Assignable
Type	Object
Initial value	NIL

1.7.4.1.1.21 TDataField:Value

Current field value. The way that this data is recovered or saves is defined by the OnGetValue and OnSetValue events and normally the FieldGet() y FieldPut() methods are responsible of this operation.

Scope	Assignable
Type	Character, Date, Numeric o Logic
Initial value	NIL

1.7.4.1.2 TDataFieldMethods

■ Constructor ■ Standard ■ Only after Create()

Type	Name
■	Blank *
■	FieldGet *
■	FieldPut *
■	HasChanged
■	Load *
■	Valtype
■	VarGet *
■	VarPut *

1.7.4.1.2.1 TDataField:Blank

Establishes an empty value (different than null) in the field. It will assign an empty value to the 'Buffer', based in the field type.

Type	Standard
Parameters	None
Return value	NIL

*** Obsolete method, only maintained for compatibility. Used only internally.**

1.7.4.1.2.2 TDataField:FieldGet

Returns the field value.

Type	Standard
Parameters	None
Return value	<Value> Current field value linked with the TDataField

*** Obsolete method, only maintained for compatibility. You must use the property Value instead.**

1.7.4.1.2.3 TDataField:FieldPut

Assigns a new value to the field.

Type	Standard
Parameters	<Value> New value for the field linked to TDataField
Return value	<ISuccess> .T. if the operation has been successful

*** Obsolete method, only maintained for compatibility. You must use the property Value instead.**

1.7.4.1.2.4 TDataField:HasChanged

Returns true if the value passed as parameter is different than its initial value..

Type	Standard
Parameters	<Value> Value to check
Return value	<IValue> .T. if the value is different

1.7.4.1.2.5 TDataField:Load

Loads the current field value in the 'Buffer' and in InitValue. The IModified is assigned with a .F. value.

Type	Standard
Parameters	None
Return value	NIL

*** Obsolete method, only maintained for compatibility. Used only internally.**

1.7.4.1.2.6 TDataField:Valtype

Returns the field type of the object.

Type	Standard
Parameters	None
Return value	NIL

1.7.4.1.2.7 TDataField:VarGet

Returns the current 'Buffer' value.

Type	Standard
Parameters	None
Return value	<Value>

*** Obsolete method, only maintained for compatibility. You must use the property Value instead.**

1.7.4.1.2.8 TDataField:VarPut

Assigns the 'Value' property value to the to the 'Buffer' property.

Type	Standard
Parameters	<Value> New value to save in the 'Buffer'
Return value	<ISuccess> .T. if the operation has been successful

*** Obsolete method, only maintained for compatibility. You must use the property Value instead.**

1.7.4.1.3 TDataField:Events

Name

OnGetValue

OnSetValue

1.7.4.1.3.1 TDataField:OnGetValue

Event that is produced when the field value is recovered through its value property. Is basically used on calculated fields.

Parameters	<oSender> :
:	Object that triggers the event.
Return value:	Field value

1.7.4.1.3.2 TDataField:OnSetValue

Event that is produced when the field value is assigned. Is basically used on calculated fields.

Parameters	<oSender> :
:	Object that triggers the event
	<Value> :
	New field value
Return value:	NIL

1.7.4.2 TDataSet

Classes to manage tables and database record sets

Description:

This class is the base class for any type of DataSet. It should not be directly used. In other words, they will be descending from their operating classes, for example TDbfDataSet, TSqlQuery or TSqlTable. This class establishes the properties, events and methods that their descending classes will develop. It might be that the user will not need to instantiate directly an object from this class very often.

In some cases you will see how the same property or method is documented in this base class and in the subsequent inherit class level and this is because its definition or behavior are not exactly the same.

If you will develop from scratch any kind of database connection that is not offered by default by Xailer, you will need to inherit the abstract TDataSet class and develop every single property and method to be sure that it will work perfectly with all the DataControls.

Hierarchy Inherits from TComponent
File Name \source\Dataset.prg
See also TDataField, DataControls

1.7.4.2.1 TDataSetProperties

■ Read Only ■ Assignable ■ Design assignable ■ Run-time assignable

Scope	Name	Type	Initial value
■	aFields	Array	{}
■	aLinkedControls	Array	{}
■	aUserFields	Array	{}
■	cLastError	Character	""
■	cName	Character	" "
■	cProcess	Character	""
■	lDisplayErrors	Logic	.T.
■	lOpen	Logic	.F.
■	lReadOnly	Logic	.F.
■	lUpdLinked	Logic	.T.
■	nCursorLocation	Numeric	adUseServer
■	nCursorType	Numeric	adOpenForwardOnly
■	nLastError	Numeric	0
■	nLockType	Numeric	adLockReadOnly
■	nMaxRecords	Numeric	0
■	oDataSource	Object	NIL
■	oRecords	Object	NIL

1.7.4.2.1.1 TDataSet:aFields

TDataField object collection with all the table or dataset fields.

Scope	Assignable
Type	Array
Initial value	{}

The aFields array is created automatically for the TDataSet and takes the info from the TDataField object collection. Much probability you will not need to create manually a TDataField object.

1.7.4.2.1.2 TDataSet:aLinkedControls

Array with all the DataControls linked to TDataSet. All the DataControls will receive and refresh event every time that there is a navigation change in the TDataSet.

Scope	Assignable
Type	Array

Initial value {}

This array will have all the visual controls linked to TDataSet. When a DataControl is linked to a TDataSet through its **oDataSet** property, any TDataSet modification is automatically reflected in all their linked DataControls.

Through the DataControls, it is possible to make add and edit operations without the need to write any line of code. For more information, see also the "Introduction to DataControls" chapter.

1.7.4.2.1.3 TDataSet:aUserFields

TDataField object collections with all the calculated or new virtual fields created by the user.

Scope	Assignable
Type	Array
Initial value	{}

The TDataSet class allows to create calculated fields based in the existing fields, or user fields that can be used as special data containers. For more information, see also the methods AddUserVar () and AddCalcField() and the "Introduction to DataControls" chapter.

1.7.4.2.1.4 TDataSet:cLastError

Last error produced in TDataSet.

Scope	read Only
Type	Character
Initial value	""
See also	nLastError

It is possible to review the errors through its TDataSource, which stores all the errors produced.

1.7.4.2.1.5 TDataSet:cName

Name designed to TDataSet. The name normally indicates the data origin. In a DBF type database, this property indicates the table name. In a SQL database, this property indicates the temporal table used to save rows.

Scope	Assignable
Type	Character
Initial value	""

Note: If you change this property be aware that you must call the **Destroy** method in order

to delete any reference to the previous table.

1.7.4.2.1.6 TDataSet:cProcess

Process name linked to the TDataSet.

Scope	Assignable
Type	Character
Initial value	""

The process name is a literal that identifies all the TDataSets from a specific process. Design a 'cProcess' in every table or open recordset, it is possible to open or close all the TDataSet from the same process using the method 'CloseProcess' or 'OpenProcess' from the TDataSource object.

1.7.4.2.1.7 TDataSet:IDisplayErrors

If it is .T. it will show the possible errors produced by TDataSet during run-time. If its value is .F. it will not show those errors, however, they will be stored in its TDataSource object.

Scope	Assignable
Type	Logic
Initial value	.T.
See also	cLastError, nLastError

1.7.4.2.1.8 TDataSet:IOpen

Opens or closes the TDataSet status. When its value is .T., the TDataSet is open and then it is possible to make all the typical navigation operations with the tables. However, when the TDataSet is closed, its functionality will be limited to determine some parameters that later will be taken into consideration in the open process.

Scope	Assignable
Type	Logic
Initial value	.T.
See also	Open(), Close()

Take into consideration that if the TDataSet has not been created (not only created through **New**, but created with its constructor method Create as well), the open process will not be done until the moment that it will be created using the Create method.

1.7.4.2.1.9 TDataSet:ReadOnly

If it is `.T.`, the table or recordset will be open in read Only mode and then it will not be possible to add, edit or delete any record from the table. This property is useless for SQL Datasets. In that case you should use the property `nLockType`.

Scope	Assignable
Type	Logic
Initial value	<code>.F.</code>
Note	This property only affects to TDataSet if has not been opened with the <code>IOpen</code> property or the <code>Open()</code> method.

1.7.4.2.1.10 TDataSet:UpdLinked

If it is `.T.`, all the `aLinkedControls` controls will receive a refresh notification every time that changes the navigation row from the TDataSet (the record number in the case of DBF type files).

Scope	Run-time assignable
Type	Logic
Initial value	<code>.T.</code>

1.7.4.2.1.11 TDataSet:nCursorLocation

Numerical value with the cursor location to use when opening the DataSet. If the DataSet is already open this property will only be used on next openings.

Scope	Assignable
Type	Numeric
Initial value	<code>adUseClient</code>
Possible values	<code>adUseClient</code> <code>adUseServer</code> <code>adUseNone</code>

- **adUseServer**: Uses data-provider or driver-supplied cursors. These cursors are sometimes very flexible and allow for additional sensitivity to changes others make to the data source. However, some features cannot be simulated with server-side cursors and these features will be unavailable with this setting.
- **adUseClient**: Uses client-side cursors supplied by a local cursor library. Local cursor services often will allow many features that driver-supplied cursors may not, so using this setting may provide an advantage with respect to features that will be enabled.
- **adUseNone**: Does not use cursor services

The possible values are what the ADO engine supports. Not all the DataSources may support all this values, Also ADO, with some specific database provider may not be able to use some of this

values. In that case the Dataset will use the value that best approaches to the given value without generating a run-time error.

For further information in case you use the TADODataSource:

<http://msdn.microsoft.com/library/en-us/ado270/htm/mdprocursorlocation.asp>

1.7.4.2.1.12 TDataSet.nCursorType

Numerical value with the cursor type to use when opening the DataSet. If the DataSet is already open this property will only be used on next openings.

Scope	Assignable
Type	Numeric
Initial value	adOpenStatic
Possible values	adOpenStatic adOpenForwardOnly adOpenDynamic adOpenKeySet

- **adOpenStatic:** Uses a static cursor, which is a static copy of a set of records that you can use to find data or generate reports. Additions, changes, or deletions by other users are not visible.
- **adOpenForwardOnly:** Uses a forward-only cursor. Identical to a static cursor, except that you can only scroll forward through records. This improves performance when you need to make only one pass through a Recordset.
- **adOpenDynamic:** Uses a dynamic cursor. Additions, changes, and deletions by other users are visible, and all types of movement through the Recordset are allowed, except for bookmarks, if the provider doesn't support them..
- **adOpenKeySet:** Uses a keyset cursor. Like a dynamic cursor, except that you can't see records that other users add, although records that other users delete are inaccessible from your Recordset. Data changes by other users are still visible.

The possible values are what the ADO engine supports. Not all the DataSources may support all this values, Also ADO, with some specific database provider may not be able to use some of this values. In that case the Dataset will use the value that best approaches to the given value without generating a run-time error.

For further information in case you use the TADODataSource:

<http://msdn.microsoft.com/library/en-us/ado270/htm/mdprocursorortype.asp>

1.7.4.2.1.13 TDataSet.nLastError

Numeric value from the last error produced by the TDataSet.

Scope	read Only
Type	Numeric

Initial value	0
See also	cLastError

Also it is possible to check the errors through its own TDataSource that stores all the produced errors..

1.7.4.2.1.14 TDataSet:nLockType

Numerical value with the locking scheme to use when opening the DataSet. If the DataSet is already open this property will only be used on next openings.

Scope	Assignable
Type	Numeric
Initial value	adLockUnspecified
Possible values	adLockOptimistic adLockBatchOptimistic adLockPessimistic adLockReadOnly adLockUnspecified

- **adLockOptimistic**: Indicates optimistic locking, record by record. The provider uses optimistic locking, locking records only when you call the Update method.
- **adLockBatchOptimistic**: Indicates optimistic batch updates. Required for batch update mode. (not supported yet).
- **adLockPessimistic**: Indicates pessimistic locking, record by record. The provider does what is necessary to ensure successful editing of the records, usually by locking records at the data source immediately after editing.
- **aLockReadOnly**: Indicates read-only records. You cannot alter the data.
- **aLockUnspecified**: Does not specify a type of lock.

The possible values are what the ADO engine supports. Not all the DataSources may support all this values, Also ADO, with some specific database provider may not be able to use some of this values. In that case the Dataset will use the value that best approaches to the given value without generating a run-time error.

For further information in case you use the TADODataSource:

<http://msdn.microsoft.com/library/en-us/ado270/hm/mdprolocktype.asp>

1.7.4.2.1.15 TDataSet:nMaxRecords

Indicates the maximum number of records to return to the DataSet from a query. If the DataSet is already open this property will only be used on next openings.

Scope	Assignable
Type	Numeric

Initial value 0

1.7.4.2.1.16 TDataSet.oDataSource

TDataSource object from the TDataSet

Scope	Assignable
Type	Object
Initial value	NIL

The oDataSource object specifies the TDataSet origin data. It is like the needed connection with the database. If the oDataSource is modified, it will close automatically the TDataSet if this is open.

1.7.4.2.1.17 TDataSet.oRecords

Internal object responsible of the communication with the database provider. Every DataSource instances its own **oRecords** object when the DataSource is assigned to the DataSet. This property is useless for DBF tables.

Scope	Assignable
Type	Object
Initial value	NIL

Nota: The documentation about this kind of objects will be published with the DataSource creation manual so people can create their own DataSources.

1.7.4.2.2 TDataSet:Methods

■ Constructor ■ Standard □ Only after Create()

Type	Name
■	AddCalcField
■	AddLinkedControl
□	AddNew
■	AddUserVar
□	Assign
□	Blank
□	Bof
□	Cancel
□	CanEdit
□	ClearFilter
■	Clone
□	Close

■	Commit
■	Create
■	DelError
■	Delete
■	DelLinkedControl
■	Edit
■	End
■	Eof
■	FastAddNew
■	FastEdit
■	FastUpdate
■	FieldCount
■	FieldDec
■	FieldDisplayName
■	FieldExist
■	FieldGet
■	FieldGetByName
■	FieldLen
■	FieldName
■	FieldNames
■	FieldPos
■	FieldPut
■	FieldSQLType
■	FieldType
■	Filter
■	GetRecord
■	GetRow
■	GetRowHash
■	GetRows
■	GetRowsHash
■	GoBottom
■	Goto
■	GoTop
■	InitFrom
■	IsReadOnly
■	KeyCount
■	KeyGoto
■	KeyNo
■	Load
■	Locate
■	IOAppend
■	IOEdit
■	Modified
■	nSkip
■	oFieldByName
■	Open

<input type="checkbox"/>	RecCount
<input type="checkbox"/>	Recno
<input type="checkbox"/>	Refresh
<input type="checkbox"/>	RefreshCurrent
<input type="checkbox"/>	ReleaseState
<input type="checkbox"/>	RestoreState
<input type="checkbox"/>	SaveFrom
<input type="checkbox"/>	SaveState
<input type="checkbox"/>	SetFilter
<input type="checkbox"/>	Skip
<input type="checkbox"/>	Sort
<input type="checkbox"/>	Update
<input type="checkbox"/>	UpdateControls
<input type="checkbox"/>	VarGet
<input type="checkbox"/>	VarPut

1.7.4.2.2.1 TDataSet.AddCalcField

Creates a TDataField object to be used by the user as a virtual field, as a result of the codeblock '**bBlock**'. The new object is added automatically to the calculate field array aUserFields.

Type	Standard
Parameters	<p><cField> field Name</p> <p><bBlock> Codeblock to be evaluated when it is needed. This codeblock will receive only one parameter, the TDataField object, and it will return the value of this calculated field.</p> <p>For Example:</p> <pre>{ oField oField:DataSet:Name + ; oField:DataSet:LastName }</pre>
Return value	<oField> Created TDataField value

The calculated fields are very useful to show data that use more than one DataSet field or when it needed to transform an specific field. For example, let's suppose a numeric field "Type" that could have the following values:

1. Normal
2. Exempt
3. Levy

We can do a simple operation to show the labels instead the numbers:

```
Local aValues := { "Normal", "Exempt", "Levy" }
oDataSet.AddCalcField( "cType", { |oField| { aValues[
oField:DataSet:Type ] } } )
```

The calculated fields are prepared only to recover information from the DataSet by default. However, it is possible, using the OnSetValue event from the calculated object, to save data. For example:

```
oField:OnSetValue := { |oField, Value| SaveType( oField:oDataSet, Value ) }
```

```
FUNCTION SaveType( oDataSet, Value )
```

```
  do case
    case Value == "Normal"
      oDataSet:Type := 1
    case Value == "Exempt"
      oDataSet:Type := 2
    case Value == "Levy"
      oDataSet:Type := 3
  end case
```

```
RETURN nil
```

See also "Introduction to DataControls", for more information.

1.7.4.2.2.2 TDataSet:AddLinkedControl

Adds the **oControl** control to the linked controls list **aLinkedControls**

Type	Standard
Parameters	<oControl> Control to add
Return value	<ISuccess> .T. if the operations is successful

This method allows to add an specific DataControl to the TDataSet linked controls. You can get the same result if you assign the TDataSet to the DataControl **oDataSet** property.

1.7.4.2.2.3 TDataSet:AddNew

Enters to edit mode to create a new record.

Type	Only usable with open TDataSet
Parameters	None
Return value	<ISuccess> .T. if it enters to edit mode.

See also 'Introduction to DataControls' for more information.

1.7.4.2.2.4 TDataSet:AddUserVar

Creates a TDataField to be used for the user as an additional data container.

Type	Standard
Parameters	<cVar> Variable name <Value> Value to assign to the buffer's field
Return value	<oField> TDataField object created

The user fields are simple value containers that can be used for the user. The main difference with the calculated fields is that the user fields don't really retrieve information from the DataSet. They only store a value in its edition buffer. They can be very useful when an edition process needs an additional field for any reason.

See also 'Introduction to DataControls' for more information.

1.7.4.2.2.5 TDataSet:Assign

It allows to assign to this dataset all the information of another existing dataset that it receives as a parameter, including its complete cursor and all its data. It is necessary that both datasets are of the same type (instances of the same class) and that their oDataSource property has been assigned.

The great advantage of using this method is that the links of the dataset with its linked visual controls are not lost.

Type	Standard
Parameters	<oDataset> TDataset object
Return value	<ISuccess> True if success

1.7.4.2.2.6 TDataSet:Blank

Initializes the TDataField objects with blanks.

Type	Only usable with open TDataSet
Parameters	[<IUpdLinked>] If TRUE the linked DataControl are updated with blank values. By default .F.

Return value	<Value> Init value assigned
---------------------	---

This method is called internally for the own TDataSet when enters to edit mode through its AddNew method.

See also 'Introduction to DataControls' for more information.

1.7.4.2.2.7 TDataSet:Bof

Indicates if the pointer is at the beginning of the TDataSet (first row or record).

Type	Only usable with open TDataSet
Parameters	None
Return value	<ILogic>

This method is the equivalent of the typical Clipper/[x]Harbour Bof() function.

1.7.4.2.2.8 TDataSet:Cancel

Cancels the edition without update any data.

Type	Only usable with open TDataSet
Parameters	None
Return value	<ISuccess> .T. if the operations is successful

This method cancels the edit mode and disregards any modification done in the TDataSet. See also the AddNew, Edit and Update methods.

1.7.4.2.2.9 TDataSet:CanEdit

Return true if edit is allowed.

Type	Only usable with open TDataSet
Parameters	None
Return	<ICanEdit>

value	
--------------	--

1.7.4.2.2.10 TDataSet:ClearFilter

Cancels the filter in the TDataSet (if any).

Type	Only usable with open TDataSet
Parameters	<p>[<IGoFirst>] If it is .T. the current record will be the first element in the DataSet. Default value: .T.</p> <p>[<IUpd>] If it is .T. all the DataControls linked to the DataSet will be updated. By default its IUpdLinked value</p>
Return value	<p><ISuccess> .T. if the operations is successful</p>

This method clears any exiting filter in the TDataSet and consequently, all its rows will become visible. See also the SetFilter method.

1.7.4.2.2.11 TDataSet:Close

Closes the TdataSets. Is equivalent to IOpen = .F.

Type	Only usable with open TDataSet
Parameters	None
Return value	<p><ISuccess> .T. if the operations is successful</p>

Close the TDataSet means that almost all the data and methods from the TdataSet are not accessible. To re-open the TDataSet, it is possible to use the Open method or assign to the IOpen property the .T. value.

1.7.4.2.2.12 TDataSet:Clone

Creates a cloned TDataSet objet. This method is virtual it is deployed on inherited classes.

Type	Standard
Parameters	None
Return value	<p><oDataSet> New TDataSet object</p>

1.7.4.2.2.13 TDataSet:Commit

Saves all the pending data.

Type	Only usable with open TDataSet
Parameters	None
Return value	<ISuccess> .T. if the operations is successful or NIL

This method is equivalent to the typical Clipper/[x]Harbour DbCommit() function.

1.7.4.2.2.14 TDataSet:Create

Class constructor.

Type	Constructor
Parameters	<oParent> Reference to the TForm proprietary object <oDataSource> Reference to the TDataSource proprietary object <cName> TDataSet name or table name [<cProcess>] Process name. See also cProcess property for more information. By default, its value will be the cProcess property from TDataSource
Return value	<Self> Self reference

Constructor class method. If there is a try to TDataSet, it will not be open until this method is executed.

1.7.4.2.2.15 TDataSet:DelError

Blanks the properties cLastError and nLastError.

Type	Standard
Parameters	None
Return value	None

1.7.4.2.2.16 TDataSet:Delete

Deletes the current row or record.

Type	Only usable with open TDataSet
Parameters	None
Return value	<ISuccess> .T. if the operations is successful

This method is equivalent to the typical Clipper/[x]Harbour DbDelete() function.

1.7.4.2.2.17 TDataSet:DelLinkedControl

Deletes the **oControl** control from the linked controls list **aLinkedControls**

Type	Standard
Parameters	<oControl> Control to remove
Return value	<ISuccess> .T. if the operations is successful

This method allows to delete an specific DataControl from the TDataSet linked controls. You can get the same result if you assign NIL to the DataControl **oDataSet** property.

1.7.4.2.2.18 TDataSet:Edit

Enters to edit mode to edit the current record.

Type	Only usable with open TDataSet
Parameters	None
Return value	<ISuccess> .T. if the operations is successful

This method enters to edit mode. At the same time, unlocks all the DataControls calling to their **Unlock** method and allowing the edition of them. See also the AddNew, Cancel and Update methods.

See also 'Introduction to DataControls' for more information.

1.7.4.2.2.19 TDataSet:End

Destroys the TDataSet, closing it before in the case that it is open. If there is a temporary file created to manage the TDataSet, this file will be destroyed as well.

Type	Standard
Parameters	None
Return value	NIL

1.7.4.2.2.20 TDataSet:Eof

Indicates if the pointer is at the end of the TDataSet (last row or record).

Type	Only usable with open TDataSet
Parameters	None
Return value	<ILogic>

This method is equivalent to the typical Clipper/[x]Harbour Eof() function.

1.7.4.2.2.21 TDataSet:FastAddNew

Enters to **fast** edit mode to create a new record. This method should only be used instead of its homologous AddNew on masive append operations where speed is critical.

Type	Only usable with open TDataSet
Parameters	None
Return value	NIL

The differences of this method with its homologous AddNew are:

- Super fast execution
- No notification to linked Datacontrols
- Impossible to use the Datacontrols for data assignment. The assignment must be done necessarily by code.
- Impossible to cancel the append operation
- The datafield assignment is done directly to the database

Note: This method must be used in conjunction with the method FastUpdate. Never mix the use of this method with the method Update.

See also 'Introduction to DataControls' for more information.

Sample:

```
WITH OBJECT oDataSet
  :FastAddNew()
  :Field1 := 1
  :Field2 := "bla, bla"
  :FastUpdate()
END WITH
```

1.7.4.2.2.22 TDataSet:FastEdit

Enters to **fast** edit mode. This method should only be used instead of its homologous Edit on masive edit operations where spped is critical.

Type	Only usable with open TDataSet
Parameters	None
Return value	NIL

The differences of this method with its homologous Edit are:

- Super fast execution
- No notification to linked Datacontrols
- Impossible to use the Datacontrols for data assignment. The assignment must be done necessarily by code.
- Impossible to cancel the edit operation
- The datafield assignment is done directly to the databae

Note: This method must be used in conjunction with the method FastUpdate. Never mix the use of this method with the method Update.

See also 'Introduction to DataControls' for more information.

Sample:

```
WITH OBJECT oDataSet
  :FastEdit()
  :Field1 := 1
  :Field2 := "bla, bla"
  :FastUpdate()
END WITH
```

1.7.4.2.2.23 TDataSet:FastUpdate

Concludes the operation of append or edit initiated by the methods `FastAddNew` or `FastEdit`. This method should only be used instead of its homologous `Update` on massive edit operations where speed is critical.

Type	Only usable with open TDataSet
Parameters	None
Return value	NIL

The differences of this method with its homologous `Update` are:

- Super fast execution
- No notification to linked Datacontrols
- Impossible to use the Datacontrols for data assignment. The assignment must be done necessarily by code.
- The datafield assignment is done directly to the database

Note: This method must be used in conjunction with the methods `FastAddNew` or `FastEdit`. Never mix the use of this method with the methods `AddNew` or `Edit`.

See also 'Introduction to DataControls' for more information.

Sample:

```
WITH OBJECT oDataSet
  :FastEdit()
  :Field1 := 1
  :Field2 := "bla, bla"
  :FastUpdate()
END WITH
```

1.7.4.2.2.24 TDataSet:FieldCount

Returns the number of fields in the Dataset.

Type	Standard
Parameters	None
Return value	<nLen> Number of fields

1.7.4.2.2.25 TDataSet.FieldDec

Returns the number of decimal places from the **nField** field

Type	Standard
Parameters	<nField> Field name to be checked
Return value	<nLen> Number of decimal places from the nField

1.7.4.2.2.26 TDataSet.FieldDisplayName

Returns the field name to be shown in the **nField** field header.

Type	Standard
Parameters	<nField> Number of Field to check
Return value	<cDisplayName> Field Name

1.7.4.2.2.27 TDataSet.FieldExist

Returns .T. if the '**cField**' field exists in the TDataSet.

Type	Standard
Parameters	<cField> Field name to be checked
Return value	<IExiste> .T. if the field name exists

1.7.4.2.2.28 TDataSet.FieldGet

Returns the field name according to the number **nField**

Type	Only usable with open TDataSet
Parameters	<nField> Field number to be checked

Return value	<Value> Field value
---------------------	-------------------------------------

This method is equivalent to the typical Clipper/[x]Harbour FieldGet() function.

1.7.4.2.2.29 TDataSet:FieldGetByName

Returns the field value according to the **cField** name

Type	Only usable with open TDataSet
Parameters	<cField> Field name to be checked
Return value	<Value> Field value

1.7.4.2.2.30 TDataSet:FieldLen

Returns the **length** from the nField field

Type	Standard
Parameters	<nField> Field number to be checked
Return value	<nLen> Field length

1.7.4.2.2.31 TDataSet:FieldName

Returns the field name according to the **nField** position

Type	Standard
Parameters	<nField> Field number to be checked
Return value	<cField> Field Name

1.7.4.2.2.32 TDataSet.FieldNames

Returns an array with all the field names.

Type	Standard
Parameters	None
Return value	<aFields> Array with file names

1.7.4.2.2.33 TDataSet.FieldPos

Returns the **cField** field position in the TDataSet.

Type	Standard
Parameters	<cField> Field name to be checked
Return value	<nIndex> Field position. 0 if not found()

1.7.4.2.2.34 TDataSet.FieldPut

Assigns the '**Value**' value to the field according to the '**nField**' number.

Type	Only usable with open TDataSet
Parameters	<nField> Field number to be checked <Value> Field value
Return value	NIL

This method is equivalent to the typical Clipper/[x]Harbour FieldPut() function.

1.7.4.2.2.35 TDataSet.FieldSQLType

Returns the SQL data type, from the TDataSource **nField** field, according to the Windows API.

Type	Standard
Parameters	<nField> Field number to be checked

Return value	<nType> SQL type
---------------------	----------------------------------

The ODBC TDataSource type can get one of the following values:

```
#define SQL_CHAR          1
#define SQL_NUMERIC       2
#define SQL_DECIMAL       3
#define SQL_INTEGER       4
#define SQL_SMALLINT      5
#define SQL_FLOAT         6
#define SQL_REAL          7
#define SQL_DOUBLE        8
#define SQL_DATE          9
#define SQL_TIME          10
#define SQL_TIMESTAMP     11
#define SQL_VARCHAR       12
#define SQL_LONGVARCHAR   -1
#define SQL_BINARY        -2
#define SQL_VARBINARY     -3
#define SQL_LONGVARBINARY -4
#define SQL_BIGINT        -5
#define SQL_TINYINT       -6
#define SQL_BIT           -7
```

1.7.4.2.2.36 TDataSet:FieldType

'nField' data type, according to the Xbase standard.

Type	Standard
Parameters	<nField> Field number to be checked
Return value	<cType> C Character N Numeric D Date L Logic M Memo

1.7.4.2.2.37 TDataSet:Filter

Establish a filter in the TDataSet.

Type	Only usable with open TDataSet
Parameters	<cExpresion>

	Filter expression [<I GoFirst >] If it is .T. the current record will be the first element in the DataSet. Default value: .T. [<I Upd >] If it is .T. all the DataControls linked to the DataSet will be updated. By default its IUpdLinked value
Return value	<c OldExpression > Old filter expression

This method allows to define any filter in the TDataSet. See also the ClearFilter method.

The filter expression is compatible with the typical Clipper/[x]Harbour DbSetFilter() function.

1.7.4.2.2.38 TDataSet:GetRecord

Returns the TExStruct object with all the current record information from the table. An TExStruct object is like an array with the advantage that its members can be accessible by its name instead by number. In this case the names correspond with the table field names.

Type	Usable only when the TDataSet is open
Parameters	None
Return value	<o Struct > TExStruct object with he current record information

1.7.4.2.2.39 TDataset:GetRow

Returns an array with the field values of a specific record on the dataset.

Type	Usable only when the TDataSet is open
Parameters	<[n Row]> Row to retrieve. By default current row <[I Move]> If true the record pointer is changed to the value indicated on first parameter. By default .T., otherwise the old record pointer is restored on exit
Return value	<a Data > Array data

See also the method GetRecord.

1.7.4.2.2.40 TDataSet.GetRowHash

Returns a Hash with the field values of a specific record on the dataset.

Type	Usable only when the TDataSet is open
Parameters	<[nRow]> Row to retrieve. By default current row
Return value	<hHash> Hash data
See also	GetRecord, GetRow

See also the method

1.7.4.2.2.41 TDataSet.GetRows

Returns a multi-dimensional array with all the Dataset records.

Type	Usable only when the TDataSet is open
Parameters	None
Return value	<aData> Array data

Important note:

The array returned by this method is normally a copy of the data, so any modification done to the array will not affect the Dataset itself. Though, is possible, that some database engine, and for speed reasons, returns an array which is the same that the one used internally by the the Dataset. For that reason you should always consider the array returned as a **read-only** array. If you change any array element, and this array is also used by the dataset you may get undesired errors, specially if you do edition operations.

1.7.4.2.2.42 TDataSet.GetRowsHash

Returns a Hash with all the Dataset records.

Type	Usable only when the TDataSet is open
Parameters	None
Return value	<hHash> Hash data

1.7.4.2.2.43 TDataSet:GoBottom

Moves the navigation pointer to the last TDataSet row or record.

Type	Only usable with open TDataSet
Parameters	None
Return value	NIL

This method is equivalent to the typical Clipper/[x]Harbour DbGoBottom() function.

1.7.4.2.2.44 TDataSet:Goto

Moves the navigation pointer to the '**nPos**' TDataset row or record.

Type	Only usable with open TDataSet
Parameters	<nPos> New position
Return value	NIL
See also	KeyGoto

This method is equivalent to the typical Clipper/[x]Harbour DbGoTo() function.

1.7.4.2.2.45 TDataSet:GoTop

Moves the navigation pointer to the first TDataSet row or record.

Type	Only usable with open TDataSet
Parameters	None
Return value	NIL

This method is equivalent to the typical Clipper/[x]Harbour DbGoTop() function.

1.7.4.2.2.46 TDataSet:InitFrom

Initializes the buffer for all the TDataSet TDataField fields with the information from other DataSet.

Type	Usable only when the TDataSet is open
Parameters	<oFrom> Origin TDataSet data
Return value	NIL

This operation will not make any modification to the table, because only will initialize the buffer to edit in the TDataSet rather than edit directly the data. The Datasets don't need to have the same structure. Only those TDataField buffers with the same name in both TDataSets will be initialized.

1.7.4.2.2.47 TDataSet:IsReadOnly

Indicates if the DataSet is read only.

Type	Only usable with open TDataSet
Parameters	None
Return value	<ILogic>

The DataSet has two properties involved with its read only state. The property IReadOnly which is used for DBF tables and the property nLockType for other DataSet types. This method gives you its read only state independently of the DataSet type.

1.7.4.2.2.48 TDataSet:KeyCount

Returns the total number of record from the TDataSet.

Type	Only usable with open TDataSet
Parameters	<IExact> If .T., the return value will be exact
Return value	<nTotal> Total number of records
See also	RecCount

This method is equivalent to the typical Clipper/[x]Harbour OrdKeyCount() function.

This method, like the Clipper function, does not take into consideration the deleted records and the current existing filters. However, it considers the filters specified through '**Scopes**'. See SetScope method in the TDbfDataSet class).

You can use the set the **IExact** property from the TDataSource to .T. or you can pass a true value as first parameter, to get the exact value from the existing records. However, use this property very carefully and use it only in very small tables, because this method is very slow.

The TDbfDataSet class from TDataSet has a '**IExact**' property that initializes its TDataSource with the value of this property and indicates the **KeyCount** and **KeyNo** methods behavior.

1.7.4.2.2.49 TDataSet:KeyGoto

Moves the navigation pointer to the relative '**nPos**' position in the TDataSet according to the current active index.

Type	Only usable with open TDataSet
Parameters	<nPos> New relative position
Return value	NIL
See also	Goto

This method is equivalent to the typical Clipper/[x]Harbour OrdKeyGoto() function.

1.7.4.2.2.50 TDataSet:KeyNo

Returns the current row or record relative position from the TDataSet based in the current index.

Type	Only usable with open TDataSet
Parameters	<IExact> If .T., the return value will be exact
Return value	<nPos> Relative position based in the current active index.

This method is equivalent to the typical Clipper/[x]Harbour OrdKeyNo() function.

This method, like the Clipper function, does not take into consideration the deleted records and the current existing filters. However, it considers the filters specified through '**Scopes**'. See SetScope method in the TDbfDataSet class).

You can use the set the **IExact** property from the TDataSource to .T. or you can pass a true value as first parameter, to get the exact value from the existing records. However, use this property very carefully and use it only in very small tables, because this method is very slow.

The TDbfDataSet class from TDataSet has a '**IExact**' property that initializes its TDataSource with the value of this property and indicates the **KeyCount** and **KeyNo** methods behavior.

1.7.4.2.2.51 TDataSet:Load

Initializes the TDataField objects with the current TDataSet value.

Type	Only usable with open TDataSet
Parameters	None
Return value	NIL

This method calls to the Load method from every TDataField object found in the aFields and aUserFields arrays

This method is internally called by the TDataSet every time that enters to edit mode, through the AddNew or Edit methods.

1.7.4.2.2.52 TDataSet:Locate

Locates a record in the TDataSet.

Type	Only usable with open TDataSet
Parameters	<cExpression> Expression to search <IContinue> If true it will search from the actual position [<IUpdLinked>] If it is .T. all the DataControls linked to the DataSet will be updated. By default its IUpdLinked value
Return value	<ISuccess> True if success

1.7.4.2.2.53 TDataSet:IOnAppend

Indicates if the TDataSet has called its AddNew method and it is in an append process.

Type	Only usable with open TDataSet
Parameters	None
Return value	<ILogic>

1.7.4.2.2.54 TDataSet:IOneEdit

Indicates if the TDataSet has called its Edit or AddNew methods and it is in an edit process.

Type	Only usable with open TDataSet
Parameters	None
Return value	<ILogic>

1.7.4.2.2.55 TDataSet:Modified

Indicates if any TDataSet field has been modified in an edit process.

Type	Only usable with open TDataSet
Parameters	None
Return value	<ILogic>

1.7.4.2.2.56 TDataSet:nSkip

Moves the navigation pointer '**nPos**' positions in the TDataSet based in the current active index.

Type	Only usable with open TDataSet
Parameters	<nPos> Number of records to skip. It can be a negative value.
Return value	<nSkipped> Number of real records skipped.
See also	Goto, Skip

Take into consideration that this method does not establish Eof()=.T. when it tries to skip further to the end of the TDataSet. This method should be used instead of Skip when a TDBBrowse control is attached to the dataset, otherwise the browse data may get corrupted.

1.7.4.2.2.57 TDataSet:oFieldByName

Returns the TDataField object from the '**cField**' field in the TDataSet.

Type	Standard
Parameters	<cField> Field name to be checked

Return value	<oField> TDataField object or o NIL if it is not found
---------------------	---

Is also possible to access to the TDatafield object using its name as a member of the class with the parameter **dsOBJECT**. For example:

```
Msginfo( oDataSet:Code( dsOBJECT ):Type )
```

1.7.4.2.2.58 TDataSet:Open

Opens the TDataSet. It is the equivalent to make IOpen = .T.

Type	Standard
Parameters	None
Return value	<ISuccess> .T. if the operations is successful

When you open a TDataSet means that almost all the TDataSet data and methods start to be accessible. To close the TDataSet you can use the Close method or set the IOpen property to .F..

Take into consideration that if the TDataSet has not been created yet the real open process will not be done until the moment that is create calling the Create method. The instantiate process must be completed not only creating the TDataSet using **New**, but also creating it through the **Create** constructor.

1.7.4.2.2.59 TDataSet:RecCount

Returns the total number of records from the TDataSet.

Type	Only usable with open TDataSet
Parameters	None
Return value	<nTotal> Total number of records
See also	KeyCount

This method is equivalent to the typical Clipper/[x]Harbour RecCount() function.

This method, like the Clipper function, does not take into consideration the deleted records, the current existing filters and the filters specified through '**Scopes**'. In other words, it returns the number of physical records existing in the table.

1.7.4.2.2.60 TDataSet:RecNo

Returns the current row or record physical position from the TDataSet.

Type	Only usable with open TDataSet
Parameters	None
Return value	<nPos> Physic position

This method is equivalent and it is based in the typical Clipper/[x]Harbour RecNo() function.

This method, like the Clipper function, does not take into consideration the deleted records, the current existing filters and the filters specified through '**Scopes**'. In other words, it returns the current physical record position in the table.

1.7.4.2.2.61 TDataSet:Refresh

Refreshes all the TDataSet information causing a new reading data operation.

Type	Only usable with open TDataSet
Parameters	None
Return value	NIL

1.7.4.2.2.62 TDataSet:RefreshCurrent

Refreshes the information from the current TDataSet record.

Type	Only usable with open TDataSet
Parameters	None
Return value	NIL

1.7.4.2.2.63 TDataSet:ReleaseState

Releases the last state saved through the SaveState method.

Type	Usable only when the TDbfDataSet is open
Parameters	[<IRestoreUpdateLinked>] If is not NIL, the IUpdLinked property is updated with this value
Return value	<ISuccess> .T. if the operations is successful

1.7.4.2.2.64 TDataSet:RestoreState

Restores the last state saved through the `SaveState` method.

Type	Usable only when the <code>TDbfDataSet</code> is open
Parameters	<code><IRestoreUpdateLinked></code> If it is not NIL, the <code>IUpdLinked</code> property is updated with this value
Return value	<code><ISuccess></code> .T. if the operations is successful

1.7.4.2.2.65 TDataSet:SaveFrom

Saves the current record with information from other `DataSet`.

Type	Usable only when the <code>TDataSet</code> is open
Parameters	<code><oFrom></code> Origin <code>TDataSet</code> data or <code>TExStruct</code> object retrieved by method <code>GetRecord</code>
Return value	NIL

Both `Dataset` might not have the same structure. Only those fields with the same name in both `TDataSets` will be saved.

1.7.4.2.2.66 TDataSet:SaveState

Saves the current table state (position, filters, etc) in an internal stack that can be recovered with the `RestoreState` method. It is possible also to disregard the last `SaveState` process with the `ReleaseState` method.

Type	Usable only when the <code>TDataSet</code> is open
Parameters	<code><IStopUpdateLinked></code> If it is not NIL, the <code>IUpdLinked</code> property is update with the opposite of its value.
Return value	<code><ISuccess></code> .T. if the operations is successful

The only parameter of this property is designed to save the table state and at the same time avoid that any subsequent operation affects the linked controls. You can pass a parameter in the `RestoreState` method to allow that the linked controls start to receive events from its `DataSet`.
Example:

```

With object oDataSet
  // Save state and update the DataControls
  :SaveState( .t. )
  :GoTop()
  // All this moves will not affect to the DataControls
  Do While !:Eof()
    .....
    :Skip()
  Enddo
  // It will recover the table state and the DataControls start to
  receive events.
  :RestoreState( .t. )
End with

```

1.7.4.2.2.67 TDataSet:SetFilter

Establish a filter in the TDataSet.

Type	Only usable with open TDataSet
Parameters	<cExpression> Filter expression [<IgoFirst>] If it is .T. the current record will be the first element in the DataSet. Default value: .T. [<IUpd>] If it is .T. all the DataControls linked to the DataSet will be updated. By default its IUpdLinked value
Return value	<cOldExpression> Old filter expression

This method allows to filter the TDataSet. See also the ClearFilter method.

This method is compatible with the typical Clipper/[x]Harbour DbSetFilter() function.

1.7.4.2.2.68 TDataSet:Skip

Moves the navigation pointer '**nPos**' positions in the TDataSet based in the current active index.

Type	Only usable with open TDataSet
Parameters	<nPos> Number of records to skip. It can be a negative value.
Return value	NIL
See also	Goto, nSkip

This method is equivalent and it is based in the typical Clipper/[x]Harbour DbSkip() function. This method should not be used when a TDBBrowse control is attached to the dataset, since the browse data may get corrupted, you should use the method nSkip instead.

1.7.4.2.2.69 TDataSet:Sort

Sorts the TDataSet.

Type	Only usable with open TDataSet
Parameters	<cExpression> Sort expression. Consult the documentation of the linked DataSource for further information [<IGoFirst>] If true a GoTop operations is performed. By default .T. [<IUpdLinked>] If it is .T. all the DataControls linked to the DataSet will be updated. By default its IUpdLinked value
Return value	<ISuccess>

1.7.4.2.2.70 TDataSet:Update

Ends the edition updating the input data or adding a new record if AddNew was used to enter to the edit mode, or modifying the data from the current record if the Edit method was used.

Type	Only usable with open TDataSet
Parameters	None
Return value	<ISuccess> .T. if the operations is successful

See also the AddNew, Edit and Cancel methods.

1.7.4.2.2.71 TDataSet:UpdateControls

Refreshes all the Datacontrols linked to the TDataSet through its aLinkedControls array

Type	Only usable with open TDataSet
Parameters	None
Return value	NIL

1.7.4.2.2.72 TDataSet:VarGet

Returns the edit buffer value according to the '**nField**' number.

Type	Only usable with open TDataSet
Parameters	<nField> Field number to be checked
Return value	<Value> Buffer value

This method retrieves the **buffer** value from the TDataField object assigned to the '**nField**' field.

*** Obsolete method maintained only for compatibility. Use method FieldGet instead.**

1.7.4.2.2.73 TDataSet:VarPut

Assigns the '**Value**' value to the TDataField buffer assigned to the '**nField**' field.

Type	Only usable with open TDataSet
Parameters	<nField> Field number to be checked <Value> New value
Return value	NIL

*** Obsolete method maintained only for compatibility. Use method FieldPut instead.**

1.7.4.2.3 TDataSet:Events

Name
OnAddNew
OnCancel
OnClose
OnCreate
OnEdit
OnOpen
OnPostChange
OnPostClose

OnPostDelete
OnPostOpen
OnPostSave
OnPreChange
OnPreDelete
OnPreSave
OnUpdate

1.7.4.2.3.1 TDataSet:OnAddNew

Event that is produced when the edit mode is activated with the AddNew method.

Parameters	<oSender> : Object that triggers the event.
Return value:	NIL

1.7.4.2.3.2 TDataSet:OnCancel

Event that is produced when the edit mode is canceled with the Cancel method.

Parameters	<oSender> : Object that triggers the event.
Return value:	NIL

1.7.4.2.3.3 TDataSet:OnClose

Event that is produced when the TDataSet is closed. If the event has been assigned and it returns a .F. value, the closing process is aborted.

Parameters	<oSender> : Object that triggers the event.
Return value:	<Logic>

1.7.4.2.3.4 TDataSet:OnCreate

Event that is produced when the TDataSet is created (when the Create constructor method is executed).

Parameters	<oSender> : Object that triggers the event.
Return value:	NIL

1.7.4.2.3.5 TDataSet:OnEdit

Event that is produced when the edit mode is activated with the Edit method.

Parameters	<oSender> : Object that triggers the event.
Return value:	NIL

1.7.4.2.3.6 TDataSet:OnOpen

Event that is produced when the TDataSet is open. If the event has been assigned and it returns a .F. value, the open process is aborted.

Parameters	<oSender> : Object that triggers the event.
Return value:	<ICancel>

1.7.4.2.3.7 TDataSet:OnPostChange

Event that is produced after the navigation pointer changes position in the TDataSet.

Parameters	<oSender> : Object that triggers the event. <nOperation> Operation type <nParam1> First parameter according to the operation <nParam2> Second parameter according to the operation
Return value:	NIL

The **nOperation** can be: (dataset.ch)

```
#define XA_DSNAV_GOTOP      0
#define XA_DSNAV_GOBOTTOM  1
#define XA_DSNAV_GOTO      2
```

```
#define XA_DSNAV_KEYGOTO 3
#define XA_DSNAV_SKIP 4
#define XA_DSNAV_SEEK 5
#define XA_DSNAV_LOCATE 6
#define XA_DSNAV_CONTINUE 7
#define XA_DSNAV_ADDNEW 8
#define XA_DSNAV_DELETE 9
#define XA_DSNAV_EDIT 10
#define XA_DSNAV_OTHER 11
```

nParam1 and **nParam2** depend from the operation type and the event:

XA_DSNAV_GOTOP: Not used

XA_DSNAV_GOBOTTOM: Not used

XA_DSNAV_GOTO: Current record number

XA_DSNAV_KEYGOTO: Current record number

XA_DSNAV_SKIP: Number of skipped records, records to be skipped.

XA_DSNAV_SEEK: Current record number, ISuccess

XA_DSNAV_LOCATE: Search expression

XA_DSNAV_CONTINUE: Not used

XA_DSNAV_ADDNEW: Current record number, Previous record number

XA_DSNAV_EDIT: Current record number

XA_DSNAV_DELETE: Current record number, Previous record number

XA_DSNAV_OTHER: Not used

1.7.4.2.3.8 TDataSet:OnPostClose

Event that is produced after the TDataSet is closed.

Parameters	<oSender>: Object that triggers the event.
Return value:	NIL

1.7.4.2.3.9 TDataSet:OnPostDelete

Event that is produced after a record is deleted in the TDataSet.

Parameters	<oSender> : Object that triggers the event.
Return value:	NIL

1.7.4.2.3.10 TDataSet:OnPostOpen

Event that is produced after the TDataSet is opened.

Parameters	<oSender> : Object that triggers the event.
Return value:	NIL

1.7.4.2.3.11 TDataSet:OnPostSave

Event that is produced after a record is saved in the TDataSet.

Parameters	<oSender> : Object that triggers the event.
Return value:	NIL

1.7.4.2.3.12 TDataSet:OnPreChange

Event that is produced before the navigation pointer changes position in the TDataSet.

Parameters	<oSender> : Object that triggers the event. <nOperation> Operation type <nParam1> First parameter according to the operation <nParam2> Second parameter according to the operation
Return value:	NIL

The **nOperation** can be: (dataset.ch)

```

#define XA_DSNAV_GOTOP      0
#define XA_DSNAV_GOBOTTOM  1
#define XA_DSNAV_GOTO      2
#define XA_DSNAV_KEYGOTO   3
#define XA_DSNAV_SKIP      4
#define XA_DSNAV_SEEK      5
#define XA_DSNAV_LOCATE    6
#define XA_DSNAV_CONTINUE  7
#define XA_DSNAV_ADDNEW    8
#define XA_DSNAV_DELETE    9
#define XA_DSNAV_EDIT      10
#define XA_DSNAV_OTHER     11

```

Param1 and **Param2** depend from the operation type and the event:

XA_DSNAV_GOTOP: Not used

XA_DSNAV_GOBOTTOM: Not used

XA_DSNAV_GOTO: Current record number

XA_DSNAV_KEYGOTO: Current record number

XA_DSNAV_SKIP: Number of records to skip

XA_DSNAV_SEEK: Current record number

XA_DSNAV_LOCATE: Search expression

XA_DSNAV_CONTINUE: Not used

XA_DSNAV_ADDNEW: Current record number

XA_DSNAV_DELETE : Current record number

XA_DSNAV_EDIT: Current record number

XA_DSNAV_OTHER: Not used

1.7.4.2.3.13 TDataSet:OnPreDelete

Event that is produced before to delete a record in the TDataSet. If the event has been assigned and it returns a .F. value, the delete process is aborted.

Parameters	<oSender>: Object that triggers the event.
Return value:	<ICancel>

1.7.4.2.3.14 TDataSet.OnPreSave

Event that is produced before to save a record in the TDataSet. If the event has been assigned and this returns a .F. value, the save process is canceled.

Parameters	<oSender>: Object that triggers the event.
Return value:	<ICancel>

1.7.4.2.3.15 TDataSet.OnUpdate

Event that is produced when exits the edition with the Update method.

Parameters	<oSender>: Object that triggers the event.
Return value:	NIL

1.7.4.3 TDbfDataSet

Class to manage DBF type tables through its data type origins TDataSource: NTX, CDX or ADS.

Description:

This class allows to manage DBF type tables with the native [x]Harbour format. Several properties and methods are encapsulated calls to the xHarbour native functions to manage tables.

There are several advantages of using TDbfDataSet instead to access the directly the tables. For more information, see also the "Introduction to DataControls" chapter.

Hierarchy	Inherits from TDataSet
File Name	\source\DbfDataset.prg
See also	TDataField, DataControls

1.7.4.3.1 TDbfDataSetProperties

■ read Only ■ Assignable ■ Design assignable ■ Run-time assignable

Scope	Name	Type	Initial value
■	aldxFiles	Array	{}
■	cAlias	Character	""
■	cName	Character	""
■	cPassword	Character	""
■	lKeyExact	Logic	.F.

□	INew	Logic	.T.
■	IReuseDel	Logic	.T.
□	IShared	Logic	.T.

1.7.4.3.1.1 TDbfDataSet:aldxFiles

Array with all the index files that must be open at the same time that the TDbfDataSet.

Scope	Assignable
Type	Array
Initial value	{}

This array will be fill before the TDbfDataSet is open. Any further modification in this array will not have any effect until the TDbfDataSet it is closed and opened again.

You can use the AddIdxFile method, that works even if the DataSet is open or not.

1.7.4.3.1.2 TDbfDataSet:cAlias

Alias name.

Scope	Read only
Type	Character
Initial value	""

This property is equivalent to the typical Clipper/[x]Harbour Alias() function.

1.7.4.3.1.3 TDbfDataSet:cName

Table name.

Scope	Assignable
Type	Character
Initial value	""

The TDataSource needed extension is assigned in the case that is not indicated (normally DBF).

It is not needed to indicate the full file name path, due this information is specified normally by the TDataSource '**cConnect**' property.

1.7.4.3.1.4 TDbfDataSet:cPassword

Password string to use for table encryption.

Scope	Assignable
Type	Character
Initial value	""

For more RDD information, please check the [x]Harbour documentation.

1.7.4.3.1.5 TDbfDataSet:IKeyExact

If it is .T. the KeyCount and KeyNo will return exact values.

Scope	Standard
Type	Logic
Initial value	.F.

If the TDbfDataSet has assigned a TDataSource, the initial value of this property will be defined by another property with the same name in the TDataSource.

The KeyCount and KeyNo Clipper basic methods, do not take into consideration the deleted records and the current existing filters. However, it considers the filters specified through '**Scopes**'. (See SetScope method in the TDbfDataSet class.)

If you need an exact value of the current records, you can use the **IKeyExact** property. However, use this property very carefully and use it only in very small tables, because the performance will be very slow with the KeyNo and KeyCount methods.

1.7.4.3.1.6 TDbfDataSet:INew

If it is .T., a table is open in a new work area. This property has not effect if it is modified after the table has been opened with its Open method or its IOpen property.

Scope	Assignable before the DataSet is open
Type	Logic
Initial value	.T.

This property is equivalent with the NEW clause in the typical Clipper/[x]Harbour USE command. The NEW clause corresponds with the INew = . T.

1.7.4.3.1.7 TDbfDataSet:IReUseDel

If it is .T., it will use automatically the deleted records (if any), in further append operations.

Scope	Standard
Type	Logic
Initial value	.T.

The technique to recover deleted records is simple: basically all the fields are blanked and deleted later. Logically, if you have at least on index, the deleted record will be at the beginning of the table when the index is active.

The record recover is done in the append process. Before to do an APPEND BLANK, you try to recover the deleted record changing the 'SET DELETED' to ON and moving the navigation pointer to the first table record with the active index. If there is any deleted record, this will be at the beginning, because all its fields are blanked. In the case that there is not any deleted record, a normal append process is done. If the table is not indexed, there is no problem, simply it will not recover deleted records.

This record recovery system has a small problem: due the change of all its fields to blank, the record changes its relative position in the index and however, the navigation pointer en the table change if there is an active index. To solve this problem, the TDbfDataSet Delete method moves the navigation pointer to the next record according to the active index, or it will move the pointer to Eof() en the case that the next record does not exists.

You need to take this small detail into consideration when you use a loop for multiple deletions, for example:

```
do while oDs:Code := "0001"
    oDs>Delete()
    oDs:Skip()
Enddo
```

This code will not work, because the oDs>Delete() will **skip** and will not delete some records. If you need multiple deletions, we recommend to use the following loop:

```
do while oDs:Seek( "0001" )
    oDs>Delete()
Enddo
```

1.7.4.3.1.8 TDbfDataSet:IShared

If it is .T., it will open the table in shared mode. This property has not effect if it is modified after the table has been opened with its Open method or its IOpen property.

Scope	Assignable before the DataSet is open
Type	Logic
Initial value	.T.

This property is equivalent with the EXCLUSIVE clause in the typical Clipper/[x]Harbour USE

command. The EXCLUSIVE clause corresponds with the IShared = . F.

1.7.4.3.2 TDbfDataSet:Methods

■ Constructor ■ Standard ■ Only after Create()

Typ	Name
■	AddIdxFile
■	AddTag
■	AddTempTag
■	Alias
■	Append
■	Area
■	ClearRelations
■	ClearScope
■	Clone
■	CompleteSeek
■	Continue
■	Delete
■	Deleted
■	DeleteTag DelTag
■	DelTempTag
■	FileLock
■	FileUnlock
■	ForcedSeek
■	Found
■	GetScope
■	GetTags
■	IsRecordLocked
■	Locate
■	OrdBagName
■	OrdCount
■	OrdFor
■	OrdKey
■	OrdKeyVal
■	OrdName
■	OrdNumber
■	OrdSetFocus
■	Pack
■	Recall
■	RecLock
■	RecUnlock
■	Reindex
■	ReleaseState
■	RestoreState

Save
SaveState
SaveToArray
Seek
Select
SetRelation
SetScope
Sort
Struct
Zap

1.7.4.3.2.1 TDbfDataSet:AddIdxFile

Add a new index file to the aldxFiles array to be open automatically when the table is open. If the table is already open, the index file is added to the array and the index is open. It is not needed to indicate the full file name path, due this information is specified normally by the TDataSource 'cConnect' property.

Type	Standard
Parameters	<cFile> File name
Return value	NIL

This method is equivalent to the typical Clipper/[x]Harbour OrdListAdd() function.

1.7.4.3.2.2 TDbfDataSet:AddTag

Adds a new 'Tag' or sort code to an existing index file.

Type	Usable only when the TDbfDataSet is open
Parameters	<cTag> Tag name <cExp> Index expression <cFor> FOR expression <cFile> Bag file name [<oProgressBar>] ProgressBar optional control to show the index process [<ITemporal>] If true the tag will be temporary and will only exist until the file is close. TAdsDataSource does not support this feature but does not force any error

	[<IUnique>] If true the tag will be created with the Unique clause. By default false.
Return value	NIL

This method is equivalent to the typical Clipper/[x]Harbour OrdConSet() and OrdCreate() functions.

1.7.4.3.2.3 TDbfDataSet:AddTempTag

Adds a new temporary 'Tag'. The tag will only exist until the file is close or DelTempTag is called. In case of TAdsDataSource a call to DelTempTag is needed to delete the temporal tag.

Type	Usable only when the TDbfDataSet is open
Parameters	<cTag> Tag name <cExp> Index expression <cFor> FOR expression [<oProgressBar>] ProgressBar optional control to show the index process
Return value	NIL

1.7.4.3.2.4 TDbfDataSet:Alias

Returns the '**Alias**' table name.

Type	Usable only when the TDbfDataSet is open
Parameters	None
Return value	<cAlias> Alias Name

This method is equivalent to the typical Clipper/[x]Harbour Alias() function.

One of the main DataSets objects is not to worry about the alias or area numbers of all the open tables. However this method exists to provide to the user an open door to use the DBF tables in the traditional way, just in case that is needed.

Example:

```
( oDataSet:Alias() )->( DbGoTop() )
```

1.7.4.3.2.5 TDbfDataSet:Append

Adds a new record to the table.

Type	Usable only when the TDbfDataSet is open
Parameters	<p>[<nTimeOut>] Retry time. the default value is: TDataSource:nTimeOut. zero value means undefined time.</p> <p>[<IReleaseOldLocks>] If it is .T. it will release previous locks. Default value: .T.</p> <p>[<IEvents>] If it is .T. the events will be triggered. Default: .T.</p>
Return value	<p><ISuccess> .T. if the operations is successful</p>

This method is equivalent to the typical Clipper/[x]Harbour DBAppend() function.

1.7.4.3.2.6 TDbfDataSet:Area

Returns the area number of the table.

Type	Usable only when the TDbfDataSet is open
Parameters	None
Return value	<nArea>

This method is equivalent to the typical Clipper/[x]Harbour Select() function.

1.7.4.3.2.7 TDbfDataSet:ClearRelations

Removes any existing relation established with SetRelation.

Type	Usable only when the TDbfDataSet is open
Parameters	None
Return value	NIL

This method is equivalent to the typical Clipper/[x]Harbour DbClearRel() function.

1.7.4.3.2.8 TDbfDataSet:ClearScope

Removes any existing 'Scope' established with SetScope.

Type	Usable only when the TDbfDataSet is open
Parameters	[<IRefresh>] If .T. all the Data Controls linked with the TDataSet will be refreshed. By default its IUpdLinked value
Return value	<aScope> Old scope (Top, Bottom)

This method is equivalent to the typical Clipper/[x]Harbour OrdScore(nType, NIL) function.

1.7.4.3.2.9 TDbfDataSet:Clone

Clones actual DataSet.

Type	Standard
Parameters	None
Return value	<oDataSet> New cloned DataSet

This method duplicates almost all the properties of the actual DataSet to a new one and creates it. The new cloned DataSet will use the same table but on a different work area. The cloned DataSet state will be identical from its original except on the following:

- Different active area and Alias
- No DataControls linked
- No events inherited from the original DataSet
- No Stack state saved by SaveState and RestoreState
- It will not be in Edit mode even if the original was
- No temporal sort done with the Sort() method

Note: It can not be used with tables opened in exclusive mode. On that it return a NIL value.

1.7.4.3.2.10 TDbfDataSet:CompleteSeek

Searches for a record in the complete TDataSet based in the current active index, independently of any filter or scope.

Be aware that filters and scopes are restored after the search and is possible that the record pointer may not be on the founded record.

Type	Usable only when the TDbfDataSet is open
Parameters	<Value> Search value [<ISoftSeek>]

	<p>If it is .T., it will be a soft seek. Default value: .F. [<ILast>] If it is .T., it will find the last Value occurrence. Default value: .F. [<IError>] If it is .T. and the seek operations fails, it will produce a run-time error. Default value: .F.. This parameter can be used when you are sure that the Seek operation will find a record. See also ForcedSeek method.</p>
Return value	<ISuccess> .T. if the operations is successful
See also	Seek, ForcedSeek

Important note: When the record is not found, it will keep the navigation pointer in its original position, and not in Eof() + 1, as is made by the DBSeek() function.

This method is equivalent and it is based in the Clipper/[x]Harbour DbSeek() function.

1.7.4.3.2.11 TDbfDataSet:Continue

Continues the search in the table based in the last call to the Locate method from the current position.

Type	Usable only when the TDbfDataSet is open
Parameters	None
Return value	<ISuccess> True if the search has been successful

This method is equivalent to the typical Clipper/[x]Harbour __DbContinue() function and CONTINUE command.

1.7.4.3.2.12 TDbfDataSet>Delete

Deletes the current record.

Type	Usable only when the TDbfDataSet is open
Parameters	[<nTimeOut>] Retry time. the default value is: TDataSource:nTimeOut. zero value means undefined time. [<IEvents>] If it is .T. the events will be triggered. Default: .T.
Return value	<ISuccess> .T. if the operations is successful

The main difference with the **DbDelete()** function is that his method changes recno() table to the next valid record when the application is configured to ignore the record deleted with the 'SET DELETE ON' command. This detail is very important specially when there are massive delete operations. For example:

```
do while oDs:Codigo := "0001"
    oDs>Delete()
    oDs:Skip()
Enddo
```

This code will not work, because the oDs>Delete() will **skip** and will not delete some records. If you need multiple deletions, we recommend to use the following loop:

```
do while oDs:Seek( "0001" )
    oDs>Delete()
Enddo
```

The TDbfDataSet has a mechanism to recover the deleted records and it is activated when the IReUseDel is set o .T.. **Is very important that you review the documentation about that property to know more about how this mechanism works y how it can affect your programs.**

This method is equivalent to the typical Clipper/[x]Harbour DbDelete() function.

1.7.4.3.2.13 TDbfDataSet:Deleted

Returns the delete status from the current record.

Type	Usable only when the TDbfDataSet is open
Parameters	None
Return value	<IDeleted> .T. if deleted

This method is equivalent to the typical Clipper/[x]Harbour Deleted() function.

1.7.4.3.2.14 TDbfDataSet:DelTag

Deletes a tag from the index file (Bag) indicated

Type	Usable only when the TDbfDataSet is open
Parameters	<cTag> Tag to delete [<cFile>] Tag Index file or Bag. Default: current Bag
Return value	NIL

This method is equivalent to the typical Clipper/[x]Harbour OrdDestroy() function.

1.7.4.3.2.15 TDbfDataSet:DelTempTag

Deletes a 'Tag' created by the method AddTempTag.

Type	Usable only when the TDbfDataSet is open
Parameters	<cTag> Tag to delete
Return value	NIL

1.7.4.3.2.16 TDbfDataSet:FilLock

Blocks the file avoiding its modification for any user.

Type	Usable only when the TDbfDataSet is open
Parameters	[<nTimeOut>] Retry time. the default value is: TDataSource:nTimeOut. Zero value means undefined time.
Return value	<ISuccess> .T. if the operations is successful

This method is equivalent to the typical Clipper/[x]Harbour FLock() function.

1.7.4.3.2.17 TDbfDataSet:FilUnLock

Unblocks the file previously blocked with the FilLock method.

Type	Usable only when the TDbfDataSet is open
Parameters	None
Return value	NIL

This method is equivalent to the typical Clipper/[x]Harbour DbUnlock() function.

1.7.4.3.2.18 TDbfDataSet:ForcedSeek

Seek operation on the table and if this is not successful, an execution error is produced. This method can be used in search operations that will never return .F. values, for example, the client information in the invoices table. In those cases, it is normal that the index is corrupted or that the client information has been deleted by accident. Then, it makes sense to produce an error.

Type	Usable only when the TDbfDataSet is open
Parameters	<xValue> Search expression
Return value	NIL
See also	Seek, CompleteSeek

This method is based in the typical Clipper/[x]Harbour DbSeek() function.

1.7.4.3.2.19 TDbfDataSet:Found

Indicates if the last search operation with the Locate, Continue or Seek methods was successful

Type	Usable only when the TDbfDataSet is open
Parameters	None
Return value	<ISuccess> .T. if the search operation was successful

This method is equivalent to the typical Clipper/[x]Harbour Found() function.

1.7.4.3.2.20 TDbfDataSet:GetScope

Returns an array with the existing scope range in an specific Tag with the format { SCOPE_TOP, SCOPE_BOTTOM}.

Type	Usable only when the TDbfDataSet is open
Parameters	[<cTag>] Tag to be checked. Default: current tag.
Return value	<aInfo> Array with the Scope information

It is important that the Scope has been established with the SetScope method, because it will not work if is done directly with the OrdScope function.

1.7.4.3.2.21 TDbfDataSet.GetTags

Returns an array with all the existing table tags.

Type	Usable only when the TDbfDataSet is open
Parameters	<IUserTags> If .T., it will not include all the tags that start with the "_" character. This avoids to include possible internal tags that we don't need or want to see. Default value: .T. <ICapFirst> If .T. it returns the tags in lowercase with the first character in uppercase. Default value: .T.
Return value	<ainfo> Array with the tag information

1.7.4.3.2.22 TDbfDataSet.IsRecordLocked

Returns the locked status from the current record.

Type	Usable only when the TDbfDataSet is open
Parameters	None
Return value	<ILocked> .T. if it is blocked

This method is equivalent to the typical Clipper/[x]Harbour DbrLockList() function.

1.7.4.3.2.23 TDbfDataSet.Locate

Makes a sequential search in the table.

Type	Usable only when the TDbfDataSet is open
Parameters	<cExp> Search expression <IContinue> If it is .T. the search will continue from the current position <IUpdate> If it is .T. the linked controls will be updated. Default value: .T.
Return value	<ISuccess> .T. if the search operation has been successful

This method is equivalent to the typical Clipper/[x]Harbour __DBLocate() function or LOCATE command.

1.7.4.3.2.24 TDbfDataSet:OrdBagName

Returns the Bag file name of a specific **Tag** based on its order or name.

Type	Usable only when the TDbfDataSet is open
Parameters	[<xTag>]: Tag number or name. By default active Tag
Return value	<cFile>: Bag file name where resides the Tag

This method is equivalent to the Clipper/[x]Harbour OrdBagName() function.

1.7.4.3.2.25 TDbfDataSet:OrdCount

Retrieves the total number of keys included in an index.

Type	Usable only when the TDbfDataSet is open
Parameters	[<xTag>]: Tag name or number. By default active Tag [<cFile>]: Bag file where resides the Tag. Default value: the active Bag.
Return value	<nTags>

This method is equivalent to the [x]Harbour OrdCount() function.

1.7.4.3.2.26 TDbfDataSet:OrdFor

Retrieves the FOR expression of an index.

Type	Usable only when the TDbfDataSet is open
Parameters	[<xTag>]: Tag name or number. By default active Tag [<cFile>]: Bag file where resides the Tag. Default value: the active Bag
Return value	<cExpression>

This method is equivalent to the [x]Harbour OrdFor() function.

1.7.4.3.2.27 TDbfDataSet:OrdKey

Retrieves the key expression of an index.

Type	Usable only when the TDbfDataSet is open
Parameters	[<xTag>]: Tag name or number. By default active Tag [<cFile>]: Bag file where resides the Tag. Default value: the active Bag
Return value	<cKey>

This method is equivalent to the [x]Harbour OrdKey() function.

1.7.4.3.2.28 TDbfDataSet:OrdKeyVal

Retrieves the value of the key expression from the active index.

Type	Usable only when the TDbfDataSet is open
Parameters	None
Return value	<xValue>

This method is equivalent to the [x]Harbour OrdKeyVal() function.

1.7.4.3.2.29 TDbfDataSet:OrdName

Returns the name of a specific **Tag** based on its order.

Type	Usable only when the TDbfDataSet is open
Parameters	[<nTag>]: Tag number. By default active Tag [<cFile>]: Bag file where resides the Tag. Default value: the active Bag.
Return value	<cTag>: Tagname

This method is equivalent to the Clipper/[x]Harbour OrdName() function.

1.7.4.3.2.30 TDbfDataSet:OrdNumber

Returns the number of a specific **Tag** based on its name.

Type	Usable only when the TDbfDataSet is open
Parameters	<p>[<cTag>]: Tag name. By default active Tag</p> <p>[<cFile>]: Bag file where resides the Tag. Default value: the active Bag.</p>
Return value	<cTag>: Tagname

This method is equivalent to the Clipper/[x]Harbour OrdNumber() function.

1.7.4.3.2.31 TDbfDataSet:OrdSetFocus

changes the active tag.

Type	Usable only when the TDbfDataSet is open
Parameters	<p><xTag> Tag to be selected. It can have a character or numeric value. To establish the natural sort, use the 0 value.</p> <p>[<cFile>] Bag file where resides the Tag. Default value: the active Bag.</p> <p>[<IError>] If it is .T. and the tag is not found, it produces a run-time error. Default value: .T.</p> <p>[<IRefresh>] If .T. all the Data Controls linked with the TDataSet will be refreshed. By default its IUpdLinked value</p>
Return value	NIL

This method is equivalent to the typical Clipper/[x]Harbour OrdSetFocus() function.

1.7.4.3.2.32 TDbfDataSet:Pack

Packs the table deleting physically all the deleted record.

Type	Usable only when the TDbfDataSet is open
Parameters	<p>[<ITry>] If it is .T., the pack process will be done, but if for any reason it is not possible to proceed, it will not return any error. Default value: .T.</p>

Return value	<ISuccess> .T. if the operations is successful
---------------------	---

To pack a table you need to open the table in exclusive mode. If the table was open in exclusive mode, the operation will be successful. However, when the table is open in shared mode, this method will try to open it in exclusive mode to proceed with the pack operation. If **ITry** = .T. and it was not possible to open the table in exclusive mode, it will not produce any error.

This method saves the complete table status before the process, and restore it after the pack operation.

This method is equivalent and is based in the Clipper/[x]Harbour __dbPAck() function and PACK command.

1.7.4.3.2.33 TDbfDataSet:Recall

Restores a deleted record.

Type	Usable only when the TDbfDataSet is open
Parameters	None
Return value	<ISuccess> .T. if the operations is successful

This method is bases in the typical Clipper/[x]Harbour DbRecall() function. However, it does not work in the same way, because the **Clipper/[x]Harbour function** tries to recover the current record (where the current navigation pointer is pointing) and this **Xailer method** tries to recover any deleted record to recycle it in an append process.

See also the IReUseDel property to have more information about the TDbfDataSet deleted record recovery mechanisms.

1.7.4.3.2.34 TDbfDataSet:RecLock

Lock a record in a table. When the record is locked, it is not possible to make any modification to it.

Type	Usable only when the TDbfDataSet is open
Parameters	[<nRecno> Record number. Default value: the current record [<nTimeOut> Retry time. the default value is: TDataSource:nTimeOut. Zero value means undefined time.
Return value	<ISuccess> .T. if the operations is successful

This method is equivalent to the Clipper/[x]Harbour DbrLock() function.

1.7.4.3.2.35 TDbfDataSet:RecUnLock

Unlock a record in the table. When the record is unlocked, it is possible to modify it.

Type	Usable only when the TDbfDataSet is open
Parameters	[<nRecno> Record number. Default value: the current record
Return value	NIL

This method is equivalent to the Clipper/[x]Harbour DbrUnLock() function.

1.7.4.3.2.36 TDbfDataSet:Reindex

Reindex the table generating all the active indexes.

Type	Usable only when the TDbfDataSet is open
Parameters	[<ITry> If it is .T., it will try to make the Reindex process. If for any reason, it is not possible to do it, it will not return any error. Default value: .T. [<oProgressBar> ProgressBar control type to show the Reindex progress. [<IRefresh> If .T. all the Data Controls linked with the TDataSet will be refreshed. By default its IUpdLinked value
Return value	<ISuccess> .T. if the operations is successful

You need to open a table in exclusive mode to Reindex it. If the table is open in exclusive mode, the index operation will be successful. However, when the table is open in shared mode, this method will try to open it in exclusive mode to proceed with the Reindex process. If **ITry** = .T. and it was not possible to open the table in exclusive mode, no error will be produced.

This method saves the complete table status before the process, and restore it after the index operation.

This method is equivalent and it is based in the Clipper/[x]Harbour OrdListRebuild() function and REINDEX command.

1.7.4.3.2.37 TDbfDataSet:ReleaseState

Releases the last state saved through the `SaveState` method.

Type	Usable only when the TDbfDataSet is open
Parameters	[<IRestoreUpdateLinked>] If is not NIL, the IUpdLinked property is updated with this value
Return value	<ISuccess> .T. if the operations is successful

1.7.4.3.2.38 TDbfDataSet:RestoreState

Restores the last state saved through the `SaveState` method.

Type	Usable only when the TDbfDataSet is open
Parameters	[<IRestoreUpdateLinked>] f is not NIL, the IUpdLinked property is updated with this value
Return value	<ISuccess> .T. if the operations is successful

1.7.4.3.2.39 TDbfDataSet:Save

Saves the current record with the TDataField buffer information from each field.

Type	Usable only when the TDbfDataSet is open
Parameters	[<IForced>] If it is .T., all the field will be saved even those that have not changed. Default value: .F. [<IUnlock>] If it is .T. the record will be unlocked after updated. Default value: .T. [<IEvents>] If it is .T., it will trigger the events. Default value: .T.
Return value	<ISuccess> .T. if the operations is successful

1.7.4.3.2.40 TDbfDataSet:SaveState

Saves the current table state (position, tag, scopes, filters, etc) in an internal stack that can be recovered with the `RestoreState` method. It is possible also to disregard the last `SaveState` process with the `ReleaseState` method.

Type	Usable only when the TDbfDataSet is open
Parameters	[<IStopUpdateLinked> If it is not NIL, the IUpdLinked property is update with the opposite of its value.
Return value	<ISuccess> .T. if the operations is successful

The only parameter of this property is designed to save the table state and at the same time avoid that any subsequent operation affects the linked controls. You can pass a parameter in the RestoreState method to allow that the linked controls start to receive events from its DataSet. Example:

```
With object oDataSet
  // Save state and update the DataControls
  :SaveState( .t. )
  :GoTop()
  // All this moves will not affect to the DataControls
  Do While !:Eof()
    .....
    :Skip()
  Enddo
  // It will recover the table state and the DataControls start to
  receive events.
  :RestoreState( .t. )
End with
```

1.7.4.3.2.41 TDbfDataSet:SaveToArray

This method allows to copy data from one table to one or more dimension array. aFields can be an array with field names or a codeblock that receives as the first parameter a DataSet. Example:

```
aItems := oDS:SaveToArray( { |Self, nArrayPos| ::Code + ": " + Trim(
::Name ) } )
aItems := oDs:SaveToArray( {"Code", "Name"} )
```

Type	Usable only when the TDbfDataSet is open
Parameters	<aFields> Array with the field names to be moved [<bFor> For expression in codeblock form. [<bWhile> While expression in codeblock form
Return value	<aData> Array with the data copied

1.7.4.3.2.42 TDbfDataSet:Seek

Searches for a record in the TDataSet based in the current active index.

Type	Usable only when the TDbfDataSet is open
Parameters	<p><Value> Search value</p> <p>[<ISoftSeek>] If it is .T., it will be a soft seek. Default value: .F.</p> <p>[<ILast>] If it is .T., it will find the last Value occurrence. Default value: .F.</p> <p>[<IError>] If it is .T. and the seek operations fails, it will produce a run-time error. Default value: .F.. This parameter can be used when you are sure that the Seek operation will find a record. See also ForcedSeek method.</p> <p>[<IAI!>] If it is .T. the seek operation will be done on the complete table, independently of any filter or scope. Default value: .F..Be aware that filters and scopes are restored after the search and is possible that the record pointer may not be on the founded record.</p>
Return value	<p><ISuccess> .T. if the operations is successful</p>
See also	ForcedSeek, CompleteSeek

Important note: When the record is not found, it will keep the navigation pointer in its original position, and not in Eof() + 1, as is made by the DBSeek() function.

This method is equivalent and it is based in the Clipper/[x]Harbour DbSeek() function.

1.7.4.3.2.43 TDbfDataSet:Select

Select a table as active table.

Type	Usable only when the TDbfDataSet is open
Parameters	None
Return value	NIL

This method is equivalent and it is based in the Clipper/[x]Harbour DbSelect() function.

One of the DataSet main objectives is forget about the aliases or the area numbers of every open table. However, this method exists to provide to the programmer an open door to use the DBF tables in the traditional way, and can be used if it is needed.

1.7.4.3.2.44 TDbfDataSet:SetRelation

Establishes a relation between a current TDbfDataSet and other DataSet.

Type	Usable only when the TDbfDataSet is open
Parameters	<cField> Field name when the relation is established. <oDataSet> TDbfDataSet object to select [<IAdd>] If it is .T., it will keep the existing relations. Default value: .T. [<IScoped>] If it is .T., a scope will be done on the related dataset. Default value: .F.
Return value	NIL

This method is equivalent and it is based in the Clipper/[x]Harbour DbSetRelation() function.

1.7.4.3.2.45 TDbfDataSet:SetScope

Establish a new scope for the current tag.

Type	Usable only when the TDbfDataSet is open
Parameters	[<xTop>] If it is not NIL, it will establish a 'Top Scope' with this value. [<xBottom>] If is not NIL, it will establish a 'Bottom Scope' with this value. Otherwise, it will use the same xTop value. [<IGoFirst>] If it is .T. it will do a GoTop after the scope has been established. [<IRefresh>] If .T. all the Data Controls linked with the TDataSet will be refreshed. By default its IUpdLinked value
Return value	NIL

This method is equivalent and it is based in the Clipper/[x]Harbour OrdScope() function.

1.7.4.3.2.46 TDbfDataSet:Sort

Sorts a table based in an expression.

Type	Usable only when the TDbfDataSet is open
Parameters	<cExp> Sort expression [<IGoFirst>] If it is .T. it will do a GoTop after the sort has been. Default value: .T. [<IDescend>] If it is .T., it will be a descending sort. Default value: .F. [<cFor>] FOR expression as literal [<oProgressBar>] Optional ProgressBar control type to show the sort process.
Return value	<ISuccess> .T. if success

The table is not really sorted, as is done by the dbase SORT command. It simply creates a temporal Tag over a temporary Bag file that the DataSet will destroy when the table is closed. This method allows in a very easy way to sort any table bases in any field. Logically this process might take some time, depending of the number of records.

1.7.4.3.2.47 TDbfDataSet:Struct

Returns an array with the table structure.

Type	Usable only when the TDbfDataSet is open
Parameters	None
Return value	<aInfo> Array with the table structure.

This method is equivalent and it is based in the Clipper/[x]Harbour DbStruct() function.

1.7.4.3.2.48 TDbfDataSet:Zap

Deletes the complete table.

Type	Usable only when the TDbfDataSet is open
Parameters	[<ITry>] If it is .T., it will try to make the zap operation, but if for any reason it is not possible to do it, it will not produce an error. Default value: .T.

	<IRefresh> If .T. all the Data Controls linked with the TDataSet will be refreshed. By default its IUpdLinked value
Return value	<ISuccess> .T. if the operations is successful

To delete completely a table, it is needed to open it in exclusive mode. If the table is open in exclusive mode, the operation will be executed successfully. However, when the table is open in share mode, this method will try to reopen it in exclusive mode to proceed to delete it. if **ITry** = .T. and the has not been possible to open the table in exclusive mode, it will not produce an error.

This method is equivalent and it is based in the Clipper/[x]Harbour __DbZap() function and ZAP command.

1.7.4.3.3 TDbfDataSet:Events

Name	
	OnPostAppend
	OnPreAppend

1.7.4.3.3.1 TDbfDataSet:OnPostAppend

Event that is produced after adding a record in the TDataSet.

Parameters	<oSender> : Object that triggers the event.
Return value:	NIL

1.7.4.3.3.2 TDbfDataSet:OnPreAppend

Event that is produced before to add a record in the TDbfDataSet. if the event has been assigned and it returns a .F. value, the append process is aborted.

Parameters	<oSender> : Object that triggers the event.
Return value:	<ICancel>

1.7.4.4 TMemDataSet

Class to manage tables in memory

Description:

This class allows to manage tables in memory using only arrays and offering the same functionality than the DataSet based in DBF files. This class does not requires any TDataSource associated to work.

There are some advantages to use the TMemDataSet control instead the DataSets. For more information, see also the 'Introduction to DataControls'.

Important note:

The use of this control with TDbBrowse objects, permits to create complex filters with its filter bar:

- Use of the sequence '--' to indicate initial-end values. For example: 10-15
- Any combination of boolean operator '&&' for AND and '||' para OR. For example 1 || 3

Hierarchy	Inherits from TDataSet
File name	\source\MemDataSet.prg
See also	TDataField, DataControls

1.7.4.4.1 TMemDataSet:Properties

■ read Only
 ■ Assignable
 ■ Design assignable
 ■ Run-time assignable

Scope	Name	Type	Initial value
■	aData	Array	{}
■	aWork	Character	""
■	!AbortOnErrors	Logical	.F.

1.7.4.4.1.1 TMemDataSet:aData

Array with the original DataSet information. This array can be assigned through the Open method.

Scope	readOnly
Type	Array
Initial value	{}

1.7.4.4.1.2 TMemDataSet:aWork

Array with the DataSet work information. This array is a reduced and/or sorted version from the aData original array with the filters and sort order set.

Scope	readOnly
--------------	----------

Type	Array
Initial value	{}

Note: Take into consideration that aWork is no an duplicated array from aData. It is only duplicated the first array dimension.

1.7.4.4.1.3 TMemDataSet:AbortOnErrors

If it is .T., it will produce an run-time error when there is access data error.

Scope	Assignable
Type	Logic
Initial value	.F.

1.7.4.4.2 TMemDataSet:Methods

■ Constructor ■ Standard ■ Only after Create()

Type	Name
■	AddField
■	Clone
■	Filter SetFilter
■	GetRow
■	GetStructFrom
■	Open
■	OrgRecno
■	ReleaseState
■	RestoreState
■	SaveState
■	Search
■	Sort
■	SqlInsert
■	SqlUpdate
■	Zap

1.7.4.4.2.1 TMemDataSet:AddField

Adds a new field to the DataSet. This method should **only** be used when no array is given as a parameter to the Open() method because the array is empty but you need to establish the fields in advance. See also the method GetStructFrom.

Type	Standard
Parameters	<cFieldName> Field name

Return value	<oDataField> TDataField object
---------------------	-----------------------------------

Important note: This method can only be used before opening the DataSet

Sample:

```

METHOD MemDataSet1Create( oSender ) CLASS TForm1

    WITH OBJECT oSender
        WITH OBJECT :AddField( "Nombre" )
            :cType := "C"
            :nLen := 10
        END WITH
        WITH OBJECT :AddField( "Tipo" )
            :cType := "C"
            :nLen := 1
        END WITH
        WITH OBJECT :AddField( "Long" )
            :cType := "N"
            :nLen := 15
        END WITH
        WITH OBJECT :AddField( "Dec" )
            :cType := "N"
            :nLen := 10
        END WITH
        :Open()
    END WITH

RETURN Nil

```

1.7.4.4.2.2 TMemDataSet:Clone

Clones the current DataSet.

Type	Standard
Parameters	None
Return value	<oDataSet> New cloned DataSet

This method duplicates most of the properties of the current DataSet in a new one and it creates it. The cloned DataSet will use the same array, however any change in the original DataSet will affect to the cloned one.

1.7.4.4.2.3 TMemDataSet:Filter

Filters the TDataSet records.

Type	Only usable with TDataSet open
Parameters	<cExpression> Expression to search. It can be any expression that includes the field names used in the Open method [<IGoFirst>] If it is .T. the current record will be the first element in the DataSet. Default value: .T. [<IUpd>] If it is .T. all the DataControls linked to the DataSet will be updated. By default its IUpdLinked value.
Return value	<cFilter> Old filter expression
See also	Filter

Example:

```
oMemDataSet:Open( Direcorry(), {"Name", "Length", "Date", "Time",
"Type" } )
oMemDataSet:SetFilter( "Upper(Name) = 'Readme' " )
```

1.7.4.4.2.4 TMemDataSet:GetRow

Returns as an array the current row data.

Type	Standard
Parameters	None
Return value	<aLine>

1.7.4.4.2.5 TMemDataSet:GetStructFrom

Sets the Dataset structure based on a DBF table or a TDataSet object.

Type	Standard
Parameters	<aDbfStruct oDataSet > Array with the DBF structure with the same format that the DbCreate() function or a TDataSet object
Return value	<ISuccess> True if success

1.7.4.4.2.6 TMemDataSet:Open

Opens the DataSet.

Type	Standard
Parameters	<aData> / <hHash>: Multidimensional array with the values to be used for the DataSet with the format: <pre>{ {Col1-1, Col1-2, ..., Col1-N} , ; {Col2-1, Col2-2, ..., Col2-N} , ; {ColN-1, ColN-2, ..., ColN-N} }</pre> / Hash or array of hashes with all the information [<aNames>]: Array with the column names. Default value: 'FIELD #1', 'FIELD #2', ... [<IEvalEvents>]: If it is .T. it will evaluate the DataSet open events. Default: .T.
Return value	<ISuccess> .T. if the operation is successful

This method is the only responsible to facilitate to the DataSet all the information to be shown.

1.7.4.4.2.7 TMemDataset:OrgRecno

Return the original record pointer even if the dataset is filtered or sorted.

Type	Standard
Parameters	None
Return value	<nRecno> Original record

1.7.4.4.2.8 TMemDataSet:ReleaseState

Disregards the last state saved through the SaveState method.

Type	Only usable with TDataSet open
Parameters	[<IRestoreUpdateLinked>] If it is different that NIL the IUpdLinked property is updated with this value
Return value	<ISuccess> .T. if the operation is successful

1.7.4.4.2.9 TMemDataSet:RestoreState

Restores the last state saved through the SaveState method.

Type	Only usable with TDataSet open
Parameters	[<IRestoreUpdateLinked>] If it is different that NIL the IUpdLinked property is updated with this value
Return value	<ISuccess> .T. if the operation is successful

1.7.4.4.2.10 TMemDataSet:SaveState

Saves the current table status (position, tags, scopes, filters, etc) in an internal pile and it can be recovered with the RestoreState method. It is possible also to disregard the last save operation with the ReleaseState method.

Type	Only usable with TDataSet open
Parameters	[<IStopUpdateLinked>] If it is different that NIL the IUpdLinked property is updated with this value
Return value	<ISuccess> .T. if the operation is successful

The only parameter of this property is designed to save the table status and avoid that any following operation affects the linked controls. In the RestoreState method you can pass a parameter to allow that the linked controls receive the events of its DataSet.

Example:

```
With object oDataSet
  // Saves the current state and the DataControl will not be updated
  DataControls
  :SaveState( .t. )
  :GoTop()
  // All these moves will not affect the DataControls
  Do While !:Eof()
    .....
    :Skip()
  Enddo
  // Restores the status of the table and the DataControls start
  receiving events
  :RestoreState( .t. )
End with
```

1.7.4.4.2.11 TMemDataSet:Search

Searches for a record in the TDataSet.

Type	Only usable with TDataSet open
Parameters	<cExpression> Expression to search. It can be any expression that includes the field names used in the Open method [<INext>] If it is .T. the search operation will be done from the current record
Return value	<ISuccess> .T. if the operation is successful
See also	Filter

Example:

```
oMemDataSet:Open( Direcorry(), { "Name", "Length", "Date", "Time",
  "Type" } )
oMemDataSet:Search( "Upper(Name) = 'Readme' " )
```

1.7.4.4.2.12 TMemDataSet:Sort

Sorts the DataSet in ascending or descending form for any of its columns.

Type	Only usable with TMemDataSet open
Parameters	<SortInfo> It can be one of the following: <ul style="list-style-type: none"> • String like the ORDER BY SQL sentence. The list of fields separated by commas and optionally the use of the 'DESC' clause on any field to indicate descending sort for that column • Array or a string with the column numbers to be used in the sort process. A negative value indicates a descending sort for that column. • String with the column numbers to be used in the sort process. A negative value indicates a descending sort for that column. [<IGoFirst>] If .T. the current record will be located as the first element in the DataSet. Default value: .T. [<IUpd>] If .T. it will update all the DataControls linked to the DataSet. Default value: .T.
Return value	NIL

Examples:

```
oMemDataSet:Sort( "first, last, salary DESC" )  
oMemDataSet:Sort( "-1, 2, -3" )  
oMemDataSet:Sort( {-1, 2, -3} )
```

1.7.4.4.2.13 TMemDataset:SqlInsert

Returns a string with the **insert** SQL sentence of all the columns that have changed their value.

Type	Only usable with TDataSet open
Parameters	None
Return value	<cSql> SQL sentence
See also	SqlUpdate

1.7.4.4.2.14 TMemDataSet:SqlUpdate

Returns a string with the **update** SQL sentence of all the columns that have changed their value.

Type	Only usable with TDataSet open
Parameters	<aWhereFields cField> Array with all the columns that are going to be used in the WHERE expression. <aWhereValues xValue> Array with all the columns values that are going to be used in the WHERE expression.
Return value	<cSql> SQL sentence
See also	SqlUpdate

1.7.4.4.2.15 TMemDataSet:Zap

Deletes all rows of the recordset.

Type	Only usable with TMemDataSet open
Parameters	None
Return value	NIL

1.7.4.5 TObdcDataSet

Class to manage RecordSets proceeding from TObdcDataSource.

Description:

This class allows to manage RecordSets proceeding from ODBC connections through the TObdcDataSource object.

As you can see, this class inherits from the TDbfDataSet class and the reason is that it uses a temporary DBF table like a buffer, to store data. If you open a table in exclusive mode without any index active, the process speed is practically the same than using an array to save the information.

The advantages of inherit from the TDbfDataSet is that you can use all the methods and data from that class, making possible, for example, use filters, create indexes, search information, etc.

For more information see also 'Introduction to DataControls'.

Hierarchy	Inherits from TDbfDataSet
File Name	\source\OdbcDataset.prg
See also	TDataSet, TDataField, DataControls

1.7.4.5.1 TObdcDataSetProperties

■ read Only ■ Assignable ■ Design assignable ■ Run-time assignable

Scope	Name	Type	Initial value
■	cSelect	Character	" "
■	AsynFetch	Logic	.F.
■	INoCompare	Logic	.F.

1.7.4.5.1.1 TObdcDataSet:cSelect

SELECT command used. See also the SQL engine documentation to check the complete SELECT instruction.

Scope	Assignable before the DataSet is open
Type	Character
Initial value	" "

When the DataSet is open and you want to execute another query in the object, you can close the object with the IOpen property and reassign a new value to **cSelect** and then, reopen again the DataSet.

1.7.4.5.1.2 Todbcdataset:AsynFetch

If it is .T., the data recovery will be done asynchrony.

Scope	Assignable before the DataSet is open
Type	Logic
Initial value	.F.

This property must be used when the number of records to be recovered in a dataset is very high or when the database server connection is very slow. Check that after the first record is recovered, the control backs to the application and the remaining records are recovered in the background.

If there is any linked TDBBrowse control to the DataSet, you will see how the cursor changes from an arrow to an hourglass icon when the cursor is moved over the browse and there are data to be recovered.

When the recovered process has finalized, the OnComplete event is triggered.

1.7.4.5.1.3 Todbcdataset:InoCompare

If it is .T., will not compare the edited record with the real record before to save, avoiding an optimist lock and saving the record.

Scope	Assignable
Type	Logic
Initial value	.F.

The Todbcdataset will make an optimist block by default. It checks if the record has been changed after the system enters to edit mode with the Edit method. If there is at least a field that has changed, the process is stopped without any update operation to the table unless that the **INoCompare** property is set to .T..

1.7.4.5.2 Todbcdataset:Methods

■ Constructor ■ Standard

Typ	Name
■	CanEdit
■	Complete
■	FieldsIsNull
■	GetPrimaryKeys
■	IsComplete
■	RecordHasChanged
■	SetPrimaryKeys

1.7.4.5.2.1 TObdcDataSet:CanEdit

.T. if it is possible to add or edit new records in the DataSet. To edit the object is needed the complete primary key in the RecordSet: the table must have a primary key and all the primary key fields must be included in the SELECT instruction. Some drivers (for example, access) can manage primary tables but they don't offer this information to their ODBC controller. In those cases you can establish the fields that belong to the primary key with the SetPrimaryKeys method.

Type	Only usable when the DataSet is open
Parameters	None
Return value	<ICanEdit> .T. if it is possible to add or edit new record to the DataSet

See also 'Introduction to DataControls' for more information.

1.7.4.5.2.2 TObdcDataSet:Complete

Recovers all the cursor data.

Type	Only usable when the DataSet is open
Parameters	None
Return value	<ISuccess> .T. is the operation is successful

It only makes sense to use this method when the IAsynFetch is set to .T.: in this case the data recovery is made asynchrony.

See also 'Introduction to DataControls' for more information.

1.7.4.5.2.3 TObdcDataSet:FieldsIsNull

Returns .T. if the '**nField**' is null.

Type	Only usable when the DataSet is open
Parameters	<nField> Field to check
Return value	<IIsNull> .T. if it is null

A field with a null value, is a field that has never had a value assigned. The DBF tables type does not consider null values, but most of the SQL database engines do..

For more information, check 'Introduction to DataControls'.

1.7.4.5.2.4 TObdcDataSet:GetPrimaryKeys

Returns an array with all the fields that build the primary key.

Type	Only usable when the DataSet is open
Parameters	None
Return value	<aFields> Field name fields

For more information, check 'Introduction to DataControls'.

1.7.4.5.2.5 TObdcDataSet:IsComplete

.T. if the data recovering operation has been completed.

Type	Only usable when the DataSet is open
Parameters	None
Return value	<IComplete> .T. if the operation has been completed.

See also the IAsynFetch and Complete properties. For more information, check 'Introduction to DataControls'.

1.7.4.5.2.6 TObdcDataSet:RecordHasChanged

.T. if the current record has changed in the ODBC database after it enters to edit mode.

Type	Only usable when the DataSet is open
Parameters	None
Return value	<IChange> .T. if it has been changed

See also the INoCompare property. for more information, check 'Introduction to DataControls'.

1.7.4.5.2.7 TObdcDataSet:SetPrimaryKeys

Allows to establish the fields that build the table primary key. The results can be catastrophic if the values for **aFields** are not coherent.

Type	Only usable when the DataSet is open
-------------	--------------------------------------

Parameters	<aFields> Field name array
Return value	NIL

For more information see also 'Introduction to DataControls'.

1.7.4.5.3 TOracleDataSet:Events

Name	
OnComplete	

1.7.4.5.3.1 TOracleDataSet:OnComplete

Event that is produced when the data recover operation has been completed.

Parameters	<oSender> Object that triggers the event
Return value	NIL

1.7.4.6 Stored procedures

Xailer incorporates a complete stored procedures manager. Most SQL database management systems that are truly client-server incorporate the ability to create stored procedures and functions within the server itself. Its use has great advantages over conventional SQL statements, such as:

- It is code that is executed directly in the database engine, which has a fast and direct access to the data that needs to be manipulated, so in many occasions the information traffic between the client and the server is drastically reduced.
- It allows to move all the logic of the business from our application to the databases. In this way, all our applications can use this procedure without having to repeat the code in each application that we create.
- Facilitates tracking model-view-controller pattern.
- You can receive additional information from the cursor that with SQL statements is not possible.

For further information follow this [link](#).

In the management of stored procedures in Xailer, three classes are involved:

1. Base class: TSqlProc which is the main class and is the one that provides practically all of the necessary functionality. An instance of this class represents a reference to a stored procedure hosted on the database server.
2. The TSqlParam class which represents each of the parameters that a stored procedure can receive. These parameters can be input, output or both. They offer great functionality that far exceeds the capabilities of simple SQL statements.

3. The `TSqlProcs` class which represents a collection of `TSqlProc` objects. It usually includes references to all existing stored procedures in the database itself since Xailer IDE automatically can construct a module that includes a class that inherits from `TSqlProcs` and has all the stored procedures of the database. In order to build module automatically, you only have to use the context menu of the `TDataSource` and choose the option 'Create stored procedures.

At the moment only operational for `TMariaDbDataSource` and `TMySQLDataSource`.

1.7.4.6.1 TSqlProc

Class for handling stored procedures.

Hierarchy TComponent descendant
File \source\Enterprise\SqlProc.prg

1.7.4.6.1.1 TSqlProc:Properties

■ Read only ■ Assignable □ Design assignable ■ Run-time assignable

Scope	Name	Type	Initial value
□	aSqlParams	Array	{}
■	aResults	Array	{}
□	cName	Character	" "
■	lDisplayErrors	Logical	.T.
■	lOpen	Logical	.F.
■	oDataSource	Object	NIL

`TSqlParam` objects collection with all the stored procedure parameters.

Scope	Assignable
Type	<code>TSqlParam</code> array
Initial value	{}

It will rarely manipulate this Array directly. Its elements creation is do through the method `AddParam` and to retrieve an concrete element yo may use the method `ParamByName` or access directly by its ordinal position.

`TMemDataset` objects collection with all the cursors returned by the stored procedure when is executed.

Scope	Read only
Type	<code>TMemDataset</code> array
Initial value	{}

A stored procedure can return none or multiple cursors, depending on the number of SELECT statements that exist in the procedure. This Array receives Object TMemDataset for each of the cursors received.

It is indicated that they are read only because their creation can only be done by the object itself. However, it is always possible to edit those datasets in memory and then perform the necessary persistence with other stored procedures or SQL statements.

Stored procedure name on the database.

Scope	Design assignable
Type	Character
Initial value	""

If true run-time errors are shown on the screen.

Scope	Assignable
Type	Logical
Initial value	.T.

TDataSource object owner of this TSqlProc.

Scope	Assignable
Type	Object
Initial value	NIL

The Object oDataSource marks the data source of the TSqlProc, that is, it is like the necessary connection with the database. Logically modifying the oDataSource causes an automatic closing of the Object if it were open.

Open or close the state of the Object. When its value becomes true, the execution of the stored procedure occurs, which is equivalent to calling the Open method.

Scope	Assignable
Type	Logical
Initial value	.T.

Ver también Open(), Close()

It is important that the object has been perfectly instantiated by indicating its name, its parameters and its oDataSource object before setting this property to true. Otherwise, it is most likely that an execution error occurs.

1.7.4.6.1.2 TSqlProc:Methods

■ Constructor ■ Standard ■ Only after Create()

Type	Name
■	AddParam
■	Close
■	Create
■	Open
■	OutputParams
■	ParamByName
■	ReturnValue
■	Run

Adds a parameter to the TSqlProc object, creating a new TSqlParam object which is included to its aSqlParams property.

Type	Standard
Parameters	<p><cName> Parameter name</p> <p>[<nDirection>] Direction. It corresponds with the property nDirection. By default adParamInput</p> <p>[<nType>] Type. It corresponds with the property nType. By default adEmpty</p> <p>[<nSize>] Size. It corresponds with the property nSize. By default 0</p> <p>[<Initial>] Initial value. By default NIL</p>
Return value	<p><oSqlParam> TSqlParam object</p>

Close the TSqlProc object. Is the equivalent to set IOpen to false.

Type	Only after TSqlProc is open
Parameters	None
Return value	<p><ISuccess> True if success</p>

Closing a TSqlProc object implies that most of its data and methods become inaccessible.

Class constructor.

Type	Constructor
Parameters	<oParent> Reference to its TForm owner <oDataSource> Reference to its TDataSource owner <cName> Stored procedure name
Return value	<Self> Reference to itself

Opens or runs the stored procedure. Is equivalent to setting the property IOpen to true.

Type	Standard
Parameters	None
Return value	<ISuccess> True if success

Opening a TSqlProc implies that most of the data and methods of the TSqlProc start to be accessible. To close the TSqlProc you can use the Close method or put your IOpen property to false.

Returns an array of TSqlParam objects of type output or input-output.

Type	Standard
Parameters	None
Return value	<aParams> TSqlParam array

Return a TSqlParam object with a specific parameter name from its aSqlParams array property.

Type	Standard
Parameters	<cName>

	Parameter name
Return value	<oSqlParam> NIL TSqlParm object

Returns the value returned by the stored procedure. For this method to return any value other than NIL it is necessary that there is a parameter with the nDirection property to adParamReturnValue.

Type	Standard
Parameters	None
Return value	Any

1.7.4.6.1.3 TSqlProc:Events

Name	
	OnCreate
	OnClose
	OnOpen
	OnPostClose
	OnPostOpen

Event triggered when the TSqlProc is created.

Parameters	<oSender>: Object that triggers the event.
Return value	NIL

Event triggered when the TSqlProc closes executing its method Close or setting his property IOpen to false.

Parameters	<oSender>: Object that triggers the event.
Return value	NIL

Event triggered before the TSqlProc opens due a execution of its method Open or setting its property IOpen to true.

Parameters	<oSender> Object that triggers the event.
Return value	NIL

Event triggered after the TSqlProc is close due a execution of its method Close or setting its property IOpen to false.

Parameters	<oSender> Object that triggers the event.
Return value	NIL

Event triggered after the TSqlProc opens due a execution of its method Open or setting its property IOpen to true.

Parameters	<oSender> Object that triggers the event.
Return value	NIL

1.7.4.6.2 TSqlParam

Class for handling the Parameters of stored procedures created with TSqlProc.

1.7.4.6.2.1 TSqlParam:Properties

■ Read only ■ Assignable ■ Design assignable ■ Run-time assignable

Scope	Name	Type	Initial value
■	cName	Character	" "
■	Initial	Any	""
■	nDirection	Numeric	adParamUnknown
■	nSize	Numeric	0
■	nType	Numeric	adEmpty
■	Value	Any	Initial
■	oSqlProc	Object	NIL

Parameter name. It corresponds to a classic variable name.

Scope	Design assignable
Type	Character

Initial value	""
----------------------	----

Parameter initial value.

Scope	Design assignable
Type	Any
Initial value	""

Parameter direction. The parameters of a stored procedure can receive or return values or both at the same time. The output address parameters can be very useful for returning additional information, such as: the new value of a self-incremental field.

Scope	Design assignable
Type	Numeric
Initial value	adParamUnknown
Possible values	adParamUnknown adParamInput adParamOutput adParamInputOutput adParamReturnValue

Parameter size in bytes. This property can be automatically assigned on the constructor with his type property.

Scope	Design assignable
Type	Numeric
Initial value	0

Parameter type on the database server.

Scope	Design assignable
Type	Numeric
Initial value	adEmpty
Possible values	adBSTR adBigInt adBinary adBoolean adChar

```

adCurrency
adDBDate
adDBTime
adDBTimeStamp
adDate
adDecimal
adDouble
adEmpty
adInteger
adLongVarBinary
adLongVarChar
adLongVarChar
adLongVarChar
adNumeric
adSingle
adSmallInt
adTinyInt
adUnsignedBigInt
adUnsignedInt
adUnsignedSmallInt
adUnsignedTinyInt
adVarBinary
adVarChar
adVarNumeric
adVarChar
adWChar

```

Consult the database documentation for further information.

Parameter value. If it is a output parameter, when you execute the stored procedure, this property will have the value returned by the stored procedure.

Scope	Design assignable
Type	Any
Initial value	""

Reference to its TSqlProc object. This property is assigned on the object creation.

Scope	Design assignable
Type	Object
Initial value	NIL

1.7.4.6.2.2 TSqParams:Methods

■ Constructor ■ Standard ■ Only after Create()

Type	Name
■	Create
■	IsInputParam
■	IsOutputParam

Class constructor.

Type	Constructor
Parameters	<p><oSqlProc> Reference to its TSqProc owner</p> <p><cName> Parameter name</p> <p>[<nDirection>] Direction. It corresponds with the property nDirection. By default adParamInput</p> <p>[<nType>] Type. It corresponds with the property nType. By default adEmpty</p> <p>[<nSize>] Size. It corresponds with the property nSize. By default 0</p> <p>[<Initial>] Initial value. By default NIL</p>
Return value	<p><Self> Reference to itself</p>

Class constructor. Normally this method is not executed directly, instead is called through the AddParam method of the class TSqProc.

Returns true if the parameter is of type input or input-output.

Type	Standard
Parameters	None
Return value	<IInput>

Returns true if the parameter is of type output or input-output.

Type	Standard
Parameters	None
Return value	<IOutput>

1.7.4.7 TSqIProcs

Class for handling collections of stored procedures. This class is used by the IDE when it builds all the stored procedures of a certain database. Instanced objects of this class have a collection of stored procedures that can be accessed by their own name and even send any parameter to them. For example:

```
oMyProc := oProcs:GetSiguieteUsuario( nEmpresa )
```

oProcs is the object that contains all the methods of the class, which if desired can be built by the IDE itself using the context menu of the DataSource. **GetNextUser** would correspond to the name of a stored procedure present in the database. **oMyProc** is an instance of the TSqIProc class which offers all the necessary functionality to interact with the database.

Example of construction by the IDE:

```
#include "Xailer.ch"

CLASS TxaContaProcs FROM TSqIProcs
    METHOD CopiarEmpresa( vOld, BYREF vNew ) // --> TSqIProc
    METHOD CrearEmpresa( BYREF vEmpresa, vDbVersion )// --> TSqIProc
    METHOD DestruirEmpresa( vEmpresa ) // --> TSqIProc
    METHOD EstablecerDefecto( vEmpresa, vTabla ) // --> TSqIProc
    METHOD GetSiguieteUsuario( vEmpresa ) // --> TSqIProc
    METHOD IsSuperAdmin( vName, vClave ) // --> TSqIProc
    METHOD RecalcularSaldos( vEmp ) // --> TSqIProc
    METHOD RenombrarEmpresa( vOld, BYREF vNew ) // --> TSqIProc
    METHOD RestaurarDefecto( vEmpresa, vTabla ) // --> TSqIProc
END CLASS

//-----

METHOD CopiarEmpresa( vOld, BYREF vNew ) CLASS TxaContaProcs

    LOCAL oProc AS CLASS TSqIProc
    LOCAL lSuccess

    oProc := ::ProcByName( "CopiarEmpresa" )

    IF oProc == NIL
        oProc := TSqIProc():Create( NIL, ::oDataSource, "CopiarEmpresa" )
        ::AddProc( oProc )
        oProc:AddParam( "vOld", adParamInput, adInteger, 0, vOld )
        oProc:AddParam( "vNew", adParamInputOutput, adInteger, 0, vNew )
    ELSE
        oProc:Close()
        oProc:SetParamValues( { vOld, vNew } )
    END IF

    IF ( lSuccess := oProc:Run() )
        vNew := oProc:vNew
    END IF

RETURN IIF( lSuccess, oProc, NIL )

//-----

METHOD CrearEmpresa( BYREF vEmpresa, vDbVersion ) CLASS TxaContaProcs
....
```

To use it anywhere in your programs you will only have to instantiate an object of that class with the following instruction:

```
oProcs := TxaContaProcs():Create( Self, ::oMariaDBDataSource1 )
```

It is recommended that oProcs be a public variable and instanced only once when loading your program

And then to execute any procedure would:

```
LOCAL oProc AS CLASS TSqIProc
oProc := oProcs:GetSiguienteUsuario( nEmpresa )
Msginfo( oProc:ReturnValue )
oProc:Close()
```

And that's all!

The **TSqIProcs** base class has few members and all the important are already mentioned. These are all its members:

- Property **aProcs**: Array with all TSqIProc objects defined
- Class constructor **Create**(<oParent>, <oDataSource>)
- Method **AddProc**(<oProc>) that basically its used by the IDE to generate the code
- Method **DelProc**(<oProc>) to delete a TSqIProc object
- Method **ProcByName**(<cName>) which return a TSqIProc with a specific <cName>
- Event **OnCreate**(<oSender>) to perform any operation when the object is instanced

Hierarchy	TComponent descendant
File	\source\Enterprise\SqlProc.prg

1.7.4.8 TSQLQuery

Class to manage SQL type tables through its data **SQL** type origins, for example TADODataSource.

By the way, there is an exception, since this type of DataSet is not compatible with TDBCDataSource. However the use of TDBCDataSource is not recommended since is better to use ADO through ODBC wich gives better performance and flexibility.

Description:

This class allows to manage SQL type tables across a 'SELECT' sentence.

Hierarchy	Inherits from TDataSet
File Name	\source\SQLQuery.prg
See also	TSQTable, TDataField, DataControls

1.7.4.8.1 TSQLQueryProperties

read Only
 Assignable
 Design assignable
 Run-time assignable

Scope	Name	Type	Initial value
<input type="checkbox"/>	aSQLParams	Array	{}
<input type="checkbox"/>	cSelect	Character	" "

1.7.4.8.1.1 TSQLQuery:aSQLParams

Parameters array to be used in conjunction with cSelect property.

The '?' signs found on the original SQL sentence are replaced by the elements of the array. The first '?' found on the SQL sentence is replaced by the first array element, and so on, until all the '?' signs are processed.

Scope	Assignable (before open)
Type	Array
Initial value	{}

Sample:

```

WITH OBJECT oDataset
  :cSelect      := "SELECT * FROM CLIENTS WHEEW Name = ? AND Age = ?
AND HIREDATE >= ?"
  :aSQLParams := { cSelect, "John", 25, Ctod( "01/01/2008" ) }
END WITH
  
```

1.7.4.8.1.2 TSQLQuery:cSelect

SELECT sentence to execute.

Scope	Assignable
Type	Character
Initial value	" "

1.7.4.9 TSQLTable

Class to manage SQL type tables through its data **SQL** type origins, for example TADODataSource.

By the way, there is an exception, since this type of DataSet is not compatible with TODBDataSource. However the use of TODBDataSource is not recommended since is better to use ADO through ODBC wich gives better performance and flexibility.

Description:

This class allows to manage SQL type tables just indicating its name.

Hierarchy Inherits from TDataSet
File Name \source\SQLTable.prg
See also TSQLQuery, TDataField, DataControls

1.7.4.9.1 TSQLTableProperties

read Only
 Assignable
 Design assignable
 Run-time assignable

Scope	Name	Type	Initial value
<input type="checkbox"/>	cTableName	Character	""

1.7.4.9.1.1 TSQLTable:cTableName

Table name to open.

Scope	Assignable
Type	Character
Initial value	""

1.7.5 Controls

1.7.5.1 TDBBrowse

This class is used to show and edit TDataSet objects in a Browse. The use of this class is linked to the DataControls. For more information, see also the introductiontoDataControls section.

	FIRST	LAST	STREET	CITY	NPAGO
▶	CLIFFORD	ABELSON	31831 WALNUT STREET	RIVER EDGE	2
	BRUCE	ABELSON	22236 W WALNUT	LEES SUMMI	5
	DOMINIC	ACKER	11414 GALLERIA BLVD	WILLIAMSTO'	3
	DANIEL	ADIOS	2457 GREENWAY DR	VACAVILLE	1
	KANDASAMY	AGUAYO	26220 CREEK CROSSING	BETHLEHEM	1
	KANDASAMY	AGUAYO	13645 FOCHT AVENUE	MONTVILLE	1
	LEN	AMARAL	1230 REYGER ST	KOWLOON B,	1
	BARRY	AMMANN	5440 S.W. BLUE INN COU	SOLANA BEA	1
	DAVID	BAILEY	9059 MAXFIELD CT	SCOTTSDALE	1

Hierarchy Inherits from TBrowse
See also TDbfBrowse, TArrayBrowse
File name \source\DBBrowse.prg

1.7.5.1.1 TDBBrowse:Properties

■ read Only
 ■ Assignable
 ■ Design assignable
 ■ Run-time assignable

Scope	Name	Type	Initial value
■	cMsgNoFilter	Character	""
■	IAutoOrder	Logic	.F.
■	IAutoSeek	Logic	.F.
■	oDataSet	Object	NIL

1.7.5.1.1.1 TDBBrowse:cMsgNoFilter

Text message to be shown when there is not possible to filter an specific column. It is used together with the Browse IFilterBar property and the column object FilterEval property.

Scope:	Assignable
Type:	Character
Initial value:	""

The class allows to filter any column by default, however there are some cases where it makes no sense to do it, for example in a column that shows an image. In those cases you should assign the NIL value to the column object FilterEval property and indicate with this property the message that you want to show then the users tries to filter by a column that does not allow to do it. If you leave this property blank, the Browse simply will beep when the is not possible to filter by that specific column.

1.7.5.1.1.2 TDBBrowse:IAutoOrder

If TRUE the browse could be sorted by any column by just clicking on its header. A second click will change the sort from ascending to descending or viceversa.

Scope:	Assignable
Type:	Logic
Initial value:	.F.

1.7.5.1.1.3 TDBBrowse:IAutoSeek

If true the incremental search will be activated on the browse. By default the search is done over the first visible column, unless it has been established another column using the IAutoOrder property and clicking on any column header; on that case the seek will be done on the ordered column.

Its operations is as follows: The first time you press an alphanumeric key within the browse, it performs a search of the key pressed on the sorted column. If it finds it, the browse is redirected to that row, otherwise beeps. If the search was successful and you press another key, the browse now search again for the sum of two keystrokes the same way. Internally the browse is

storing the search value, if you want you can display it in a TLabel control type using the property oSeek. If you change the current Browse record with the navigation keys or the mouse, the internal search value is initialized to empty string.

This property is incompatible with the IFastEdit property and the overload of OnSeek event.

Scope:	Assignable
Type:	Logic
Initial value:	.F.

Note: Be aware that except when the Browse uses as data source a TMemDataset object, on the rest of the cases the work is done directly by the database engine. On slow connections and heavy duty servers we recommend to use memory datasets.

1.7.5.1.1.4 TDBBrowse.oDataSet

TDataSet object type linked to the control.

Scope:	Assignable
Type:	Object
Initial value:	NIL

Nota: The TDataSet object must be linked only with one DBBrowse control type. If you try to assign the same TDataSet to other TDBBrowse it will cause that all the navigation events from the TDataSet will be received only in the last assigned DBBrowse.

1.7.5.2 TDBBrowseMod

This class is used to display and edit TDataSet objects in a TBrowseMod control . The use of this class is related to DataControls. See the DataControls Introduction section for more information.

The screenshot shows a window titled 'Form1' with a 'Filter' and 'Edit' button. A warning message states: 'The edition is done in a memory array. There is no persistence on the underlying database table'. Below the message is a table with columns: First, Last, Street, Npago, and an empty column. The table contains the following data:

First	Last	Street	Npago	
ROBIN	LAFORTU...	23974 NORTH REDGUM	1	
CLYDE	WILKS	15337 MARTINA STREET	3	
LOREN	HOMNICK	21459 MESA RIDGE ROAD	4	
STEWART	WOO	25782 MCDONALD'S PLAZA	5	
TINA	JOST	21132 RAFFAELE PLACE	1	
MARK	LOUMAN	23772 WILLOUGHBY STREET	1	
DAVE	YEAGER	14394 YALE AVENUE	1	

Hierarchy TBrowseMod descendant
File \source\DBBrowseMod.prg

1.7.5.2.1 TDBBrowseMod:Propiedades

■ Read only ■ Assignable ■ Design assignable ■ Run-time assignable

Scope	Name	Type	Initial value
■	oDataSet	Objeto	NIL

1.7.5.2.1.1 TDBBrowseMod:oDataSet

TDataSet object type linked to the control.

Ámbito:	Asignable
Tipo:	Objeto
Valor inicial:	NIL

Note: The TDataSet should only be linked to a single DBBrowse type control. Any attempt to re-assign the same TDataSet to another TDBBrowseMod object will cause all navigation events of the TDataSet to be received only in the last assigned DBBrowse.

1.7.5.2.2 TDBBrowseMod:Methods

■ Constructor ■ Standard ■ Only after Create()

Type	Name
■	DataCol

1.7.5.2.2.1 TDbBrowseMod:DataCol

Allows to retrieve the TDbBrwColMod column object when the field name is known.

Type	Standard
Parameters	<cDataField>
Return value	TDbBrwColMod object

1.7.5.3 TDBBrwColMod

Class inherited from TBrwColMod adapted to be used as Data control.

Hierarchy TBrwColMod descendant
File \source\DBBrwColMod.prg

1.7.5.3.1 TDBBrwColMod:Properties

■ Read-only ■ Assignable ■ Design assignable ■ Run-time assignable

Scope	Name	Type	Initial value
■	cDataField	Character	""
■	oDataField	Object	NIL

1.7.5.3.1.1 TDBBrwColMod:oDataField

TDataField object type linked to the control.

Ámbito:	Asignable
Tipo:	Objeto
Valor inicial:	NIL

This property establishes all the data view and edit mechanisms linked to the control. If this property is not assigned, the control will not show any information from the database and it will not be possible to edit it neither.

This property can be assigned directly through a TDataField object or can be assigned a literal with the field name. En this case if its oDataSet property has been already assigned, the control will search for that field in the oDataSet and if it found it will replace the literal with the own TDataField object found.

If assigns this property directly with a TDataField object, the control will automatically update the oDataSet property with the proprietary TDataSet object from the TDataField passed.

1.7.5.3.1.2 TDBBrwColMod:cDataField

Read-only property that receives the name of the assigned field in oDataField.

Ámbito:	Sólo lectura
Tipo:	Carácter
Valor inicial:	""

1.7.5.4 TDBCardBox

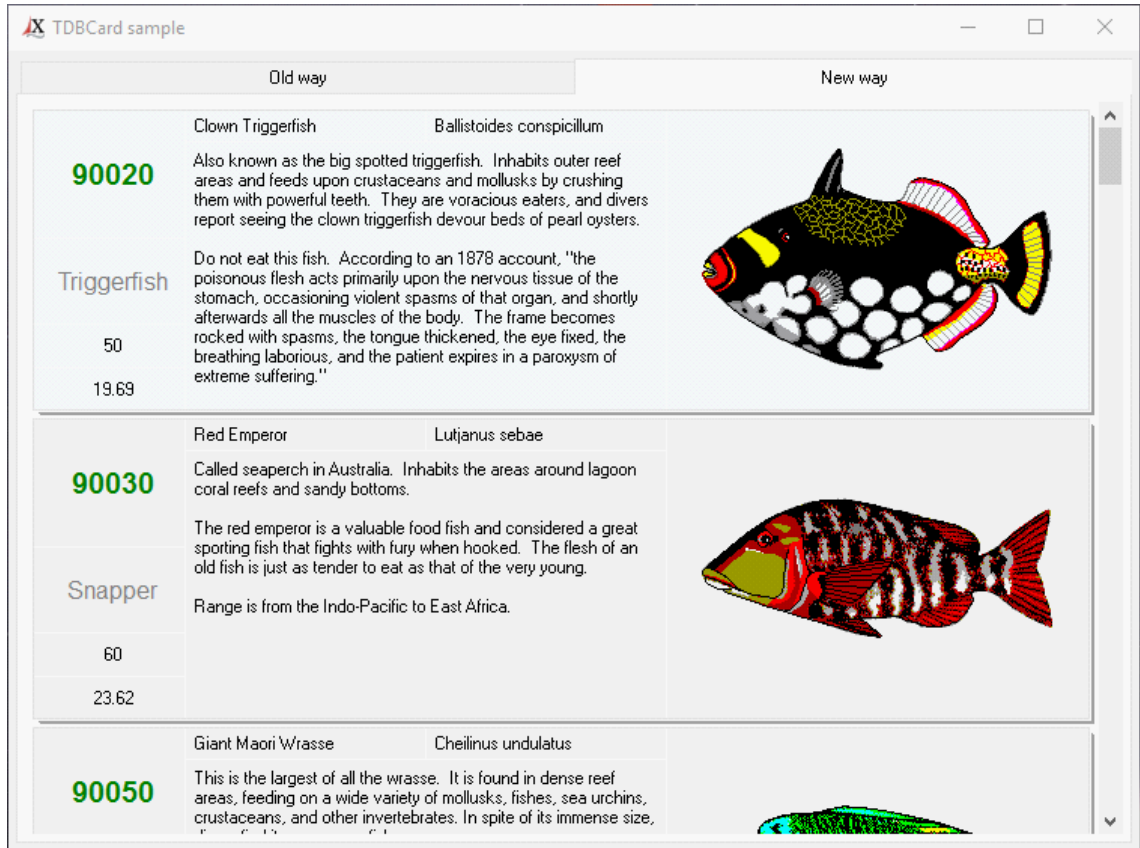
This class represents a TCardBox specialized on data management through DataControls.

Each element of this class is a **TDBCardItem** object which is TCardItem descendant. Its only difference is that includes two new properties:

- **oDataField** to establish the field name of its dataset container

- **cNullValue**, value to show when the dataset value is NULL

And its **nColumn** property is no longer available.



Hierarchy File

TCardBody descendant
 \source\DBCCardBody.prg

1.7.5.4.1 TDBCardBody:Properties

■ read Only ■ Assignable ■ Design assignable ■ Run-time assignable

Scope	Name	Type	Initial value
■	oDataSet	Object	NIL

1.7.5.4.1.1 TDBCardBody:oDataSet

TDataSet object type linked to the control.

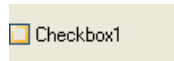
Scope:	Assignable
Type:	Object
Initial value:	NIL

Nota:

The TDataSet object must be linked only with one **TDBCardBox** control type. If you try to assign the same TDataSet to other TDBCardbox it will cause that all the navigation events from the TDataSet will be received only in the last assigned **TDBCardBox**.

1.7.5.5 TDBCheckBox

This class represents a Windows CheckBox control specialized to manage data through the DataControls.



Hierarchy Inherits from TCheckBox
File name \source\DBCheckBox.prg

1.7.5.5.1 TDBCheckBox:Properties

■ read Only ■ Assignable ■ Design assignable ■ Run-time assignable

Scope	Name	Type	Initial value
■	IAutosave	Logic	.T.
■	IEditable	Logic	.T.
■	oDataField	Object	NIL
■	oDataSet	Object	NIL

1.7.5.5.1.1 TDBCheckBox:IAutoSave

If it is .F., its value will not be updated on the underlying database.

Scope:	Assignable
Type:	Logic
Initial value:	.T.

This property is normally used when you want to show a value of a specific field but you do not want that the value is updated in any manner, even if its value has been modified. Its common use is when you want to show the value of non editable fields like, for example, auto incremental fields.

1.7.5.5.1.2 TDBCheckBox:IEditable

If it is .F., it is not possible to edit the control, and the CheckBox will be read only.

Scope:	Assignable
Type:	Logic
Initial value:	.T.

1.7.5.5.1.3 TDBCheckBox:oDataField

TDataField object type linked to the control.

Scope:	Assignable
Type:	Object
Initial value:	NIL

This property is responsible to establish all the mechanisms to edit or view the data linked to the control. If this property is not assigned, the control will not show any information from the database and it is not possible to edit it neither.

This property can be assigned directly through a TDataField object or it can be indicated with a field name literal. In this case if its oDataSet property has already been assigned, the control will search the field in the oDataSet and if it finds it will replace the literal with the TDataField object found.

If this property is assigned directly with a TDataField object, the control will automatically update the oDataSet property with the proprietary TDataSet object form the passed TDataField.

1.7.5.5.1.4 TDBCheckBox:oDataSet

TDataSet object type linked to the control.

Scope:	Assignable
Type:	Object
Initial value:	NIL

This property indicates that the TDataSet object (data collection) will be linked to the control.

It must indicate its oDataField object to complete the link.

1.7.5.5.2 TDBCheckBox:Methods

Constructor
 Standard
 Only after Create()

Type	Name
<input checked="" type="checkbox"/>	Value

1.7.5.5.2.1 TDBCheckBox:Value

Returns the current control value. This method is common to all the DataControls.

Type	Only after Create()
Parameters	None
Return value	CheckBox state

If the linked oDataSet is in edit mode, this property returns the 'buffer' value from the linked oDataField object. Otherwise, this property will return the real field value from the database.

1.7.5.6 TDBCheckBoxMod

This class represents a windows TCheckBoxMod control specialized in handling data using DataControls.

Hierarchy	TCheckBoxMod descendant
File	\source\DBCheckBoxMod.prg

1.7.5.6.1 TDBCheckBoxMod:Properties

Read only
 Assignable
 Design assignable
 Run-time assignable

Scope	Name	Type	Initial value
<input type="checkbox"/>	IAutoSave	Logical	.T.
<input type="checkbox"/>	IEditable	Logical	.T.
<input type="checkbox"/>	oDataField	Object	NIL
<input type="checkbox"/>	oDataSet	Object	NIL

1.7.5.6.1.1 TDBCheckBoxMod:IAutoSave

If False, the control will not save its value in the database.

Scope:	Assignable
Type:	Logical
Initial value:	.T.

This property is used when you want to display the value of a particular field but you do not want to save it, even if you have changed its value. Its most frequent use is when you want to display values of non-editable fields such as auto incremental fields.

1.7.5.6.1.2 TDBCheckBoxMod:IEditable

If False, the control cannot be edited and always remains read-only.

Scope:	Assignable
Type:	Logical
Initial value:	.T.

1.7.5.6.1.3 TDBCheckBoxMod:oDataField

Object of type TDataField that linked to the control..

Scope:	Assignable
Type:	Object
Initial value:	NIL

This property establishes all the data view and edit mechanisms linked to the control. If this property is not assigned, the control will not show any information from the database and it will not be possible to edit it neither.

This property can be assigned directly through a TDataField object or can be assigned a literal with the field name. En this case if its oDataSet property has been already assigned, the control will search for that field in the oDataSet and if it found it will replace the literal with the own TDataField object found.

If assigns this property directly with a TDataField object, the control will automatically update the oDataSet property with the proprietary TDataSet object from the TDataField passed.

1.7.5.6.1.4 TDBCheckBoxMod:oDataSet

TDataSet object type linked to the control.

Scope:	Assignable
Type:	Object
Initial value:	NIL

This property indicates that the TDataSet object (record set) will be linked to the control.

You should indicate its oDataField objet as well to complete the link process.

1.7.5.6.2 TDBCheckBoxMod:Methods

■ Constructor ■ Standard ■ Only after Create()

Type	Name
■	Value

1.7.5.6.2.1 TDBCheckBoxMod:Value

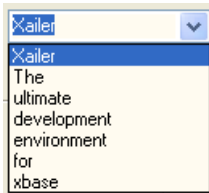
Returns the current value of the control. This method is common to all DataControls.

Type	Only after Create()
Parameters	None
Return value	Checkbox state

If the oDataSet bound to a control that is in edit mode, this property returns the buffer value of the bound oDataField object, otherwise, this property returns the actual value of the field in the database.

1.7.5.7 TDBComboBox

This class represents a Windows ComboBox control type specialized in managing data through DataControls.



Hierarchy Inherits from TComboBox
File name \source\DBComboBox.prg

1.7.5.7.1 TDBComboBox:Properties

■ read Only ■ Assignable ■ Design assignable ■ Run-time assignable

Scope	Name	Type	Initial value
■	altemsBound	Array	{}
■	IAutoSave	Logic	.T.
■	IEditable	Logic	.T.
■	nDataType	Numeric	dtDEFAULT
■	oDataField	Object	NIL
■	oDataSet	Object	NIL

1.7.5.7.1.1 TDBComboBox:altemsBound

Array with all the values the control will use to retrieve and assign its information to the dataset.

This property must be used in conjunction with the property `nDataType` assigning it a value of **dtBOUND** and the property `nStyle` with the value `csDROPDOWNLIST`. This way the `altems` array will only be used to show the elements in the combobox and the **altemsBound** array will be used to retrieve and save its value on the dataset. Logically the dimension of `altems` and `altemsBound` should be equal.

Scope:	Assignable
Type:	Array
Initial value:	{}

1.7.5.7.1.2 TDBComboBox:IAutoSave

If it is `.F.`, its value will not be updated on the underlying database.

Scope:	Assignable
Type:	Logic
Initial value:	<code>.T.</code>

This property is normally used when you want to show a value of a specific field but you do not want that the value is updated in any manner, even if its value has been modified. Its common use is when you want to show the value of non-editable fields like, for example, auto-incremental fields.

1.7.5.7.1.3 TDBComboBox:IEditable

If it is `.F.`, it is not possible to edit the control, and the ComboBox will be read-only.

Scope:	Assignable
Type:	Logic
Initial value:	<code>.T.</code>

1.7.5.7.1.4 TDBComboBox:nDataType

Indicates how the search will be done in the ComboBox.

Scope:	Assignable
Type:	Numeric

Initial value:	dtDEFAULT
Possible values:	dtDEFAULT, dtINDEX, dtSTRING, dtBOUND

- **dtDEFAULT:** for oDataField objects that return numeric values, it will relate this numeric value with the nIndex ComboBox property. For character type values, it will search the string in the ComboBox list of elements.
- **dtINDEX:** Regardless the oDataField data type, the search will be done over the nIndex property. In the case of character type values, they will be transformed to numeric format adding zeros to the left according to the field length. This value is typically used when you save numeric values in character type fields, with zeros to the left. For example: "0001".
- **dtSTRING:** Regardless the oDataField data type, the search will be done over the ComboBox list of elements.
- **dtBOUND:** The search and assign will not be done into altems array but on altemsBound array.

Note: The **dtBOUND** style is **incompatible** with the property ISort set to true because the array altemsBound does not reflect the sort modifications. This style is also **incompatible** with the property nStyle different than csDROPDOWNLIST.

1.7.5.7.1.5 TDBComboBox:oDataField

TDataField object type linked to the control.

Scope:	Assignable
Type:	Object
Initial value:	NIL

This property is responsible to establish all the mechanisms to edit or view the data linked to the control. If this property is not assigned, the control will not show any information from the database and it is not possible to edit it neither.

This property can be assigned directly through a TDataField object or it can be indicated with a field name literal. In this case if its oDataSet property has already been assigned, the control will search the field in the oDataSet and if it finds it will replace the literal with the TDataField object found.

If this property is assigned directly with a TDataField object, the control will automatically update the oDataSet property with the proprietary TDataSet object form the passed TDataField.

1.7.5.7.1.6 TDBCombokBox:oDataSet

TDataSet object type linked to the control.

Scope:	Assignable
Type:	Object

Initial value: NIL

This property indicates that the TDataSet object (data collection) will be linked with the control. It should indicate as well its oDataField object to complete the link.

1.7.5.7.2 TDBComboBox:Methods

■ Constructor ■ Standard ■ Only after Create()

Type	Name
■	Value

1.7.5.7.2.1 TDBCombokBox:Value

Returns the current control value. This method is common to all the DataControls.

Type	Only after Create()
Parameters	None
Return value	ComboBox state

If the linked oDataSet is in edit mode, this property returns the 'buffer' value from the linked oDataField, otherwise, this property returns the real value from the database field.

1.7.5.8 TDBComboBoxMod

This class represents a windows TComboBoxMod control specialized in handling data using DataControls.

Hierarchy TComboBoxMod descendant
File \source\DBComboBox.prg

1.7.5.8.1 TDBComboBox:Propiedades

■ Read only ■ Assignable ■ Design assignable ■ Run-time assignable

Scope	Name	Type	Initial value
■	altemsBound	Array	{}
■	cHint	Character	""
■	IAutoSave	Logical	.T.
■	IEditable	Logical	.T.
■	nDataType	Numeric	dtDEFAULT
■	oDataField	Objeto	NIL

■	oDataSet	Objeto	NIL
---	----------	--------	-----

1.7.5.8.1.1 TDBComboBox:altemsBound

Array with the values to be used by the control to retrieve and save the information in the Dataset.

This property must be used in association with the nDataType property by setting it to **dtBOUND** and the nStyle property to csDROPDOWNLIST. This way, the altems array will only be used to display the items in the combo-box and the altemsBound array will be used to retrieve and save its value on the dataset. Logically, the dimension of altems and altemsBound must be the same.

Scope:	Assignable
Type:	Array
Initial value:	{}

1.7.5.8.1.2 TDBComboBox:cHint

Text to be displayed when no element is selected.

Scope:	Assignable
Type:	Character
Initial value:	""

1.7.5.8.1.3 TDBComboBox:IAutoSave

If False, the control will not save its value in the database.

Scope:	Assignable
Type:	Logical
Initial value:	.T.

This property is used when you want to display the value of a particular field but you do not want to save it, even if you have changed its value. Its most frequent use is when you want to display values of non-editable fields such as auto incremental fields.

1.7.5.8.1.4 TDBComboBox:IEditable

If False, the control cannot be edited and always remains read-only.

Scope:	Assignable
Type:	Logical
Initial value:	.T.

1.7.5.8.1.5 TDBComboBox:nDataType

Indicates the way in which the combo box list will be searched.

Scope:	Assignable
Type:	Numeric
Initial value:	dtDEFAULT
Possible values:	dtDEFAULT, dtINDEX, dtSTRING, dtBOUND

- **dtDEFAULT:** For oDataField objects that return numeric values, the numeric value will be matched with the nIndex property of the combo-box. For character values, the string will be searched for in the list of combo-box items
- **dtINDEX:** Regardless of the type of the oDataField, the search will be done on the nIndex property. In the case of character type values it will be converted to numeric format by putting as many leading zeros as the length of the field. This value is the typical search when storing numeric data in character type fields. For example: "0001"
- **dtSTRING:** Regardless of the type of the oDataField, the search will be done on the comboBox items elements
- **dtBOUND:** The search and the assignment will be done over the altemsBound array

Note: The **dtBOUND** style is **incompatible** with the property ISort to true, because the array altemsBound do not reflect the sort changes.

1.7.5.8.1.6 TDBComboBox:oDataField

Object del Type TDataField vinculado con el control.

Scope:	Assignable
Type:	Object
Initial value:	NIL

Esta propiedad es la que establece todo el mecanismo de visualización y edición del dato vinculado al control. Hasta que no se asigne esta propiedad el control no muestra ninguna información sobre la base de datos ni tampoco es posible la edición del mismo.

Esta propiedad puede ser asignada directamente a través de un Object TDataField o se le puede indicar un literal con el Name del campo. En este último caso si su propiedad oDataSet ya ha sido asignada el control buscará dicho campo en el propio oDataSet, y si lo encuentra sustituirá el literal por el propio Object TDataField encontrado.

Si se asigna esta propiedad directamente con un Object TDataField, el control automáticamente actualizará la propiedad oDataSet con el Object TDataSet propietario del propio TDataField pasado.

1.7.5.8.1.7 TDBComboBox:oDataSet

TDataSet object type linked to the control.

Scope:	Assignable
Type:	Objeto
Initial value:	NIL

This property indicates that the TDataSet object (record set) will be linked to the control.

You should indicate its oDataField objet as well to complete the link process.

1.7.5.8.2 TDBComboBox:Methods

■ Constructor ■ Estándar ■ Only after Create()

Type	Name
■	Value

1.7.5.8.2.1 TDBComboBox:Value

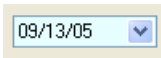
Devuelve el valor actual del control. Este método es común a todos los DataControls

Type	Only after Create()
Parameters	None
Return value	ComboBoxstate

If the oDataSet bound to a control that is in edit mode, this property returns the buffer value of the bound oDataField object, otherwise, this property returns the actual value of the field in the database.

1.7.5.9 TDBDateEdit

This class represents a TDateEdit Windows control, specialized to manage data through DataControls.



Hierarchy Inherits from TEditBtn
File name \source\DBEditBtn.prg

1.7.5.9.1 TDBDateEdit:Properties

read Only
 Assignable
 Design assignable
 Run-time assignable

Scope	Name	Type	Initial value
<input type="checkbox"/>	IAutosave	Logic	.T.
<input type="checkbox"/>	IEditable	Logic	.T.
<input type="checkbox"/>	oDataField	Object	NIL
<input type="checkbox"/>	oDataSet	Object	NIL

1.7.5.9.1.1 TDBDateEdit:IAutoSave

If it is .F., its value will not be updated on the underlying database.

Scope:	Assignable
Type:	Logic
Initial value:	.T.

This property is normally used when you want to show a value of a specific field but you do not want that the value is updated in any manner, even if its value has been modified. Its common use is when you want to show the value of non editable fields like, for example, auto incremental fields.

1.7.5.9.1.2 TDBDateEdit:IEditable

If it is .F. the control can not be edited and will be treated as read only.

Scope:	Assignable
Type:	Lógico
Initial value:	.T.

1.7.5.9.1.3 TDBDateEdit:oDataField

TDataField object type linked to the control.

Scope:	Assignable
Type:	Object
Initial value:	NIL

This property establishes all the display and edit mechanisms from the linked data to the control. If this property is not assigned, the control will not show any information from the database and it is not possible to edit it neither.

This property can be assigned directly through a TDataField object or it is possible to indicate a literal with the field name. In this case if its oDataSetproperty has been already assigned, the control will search for that control in the oDataSet and if it is found it will replace the literal with the TDataField object found.

If this property is assigned automatically with a TDataField object, the control will automatically update the oDataSet property with the proprietary TDataSet object from the TDataField passed.

1.7.5.9.1.4 TDBDateEdit:oDataSet

TDataSet object type linked to the control.

Scope:	Assignable
Type:	Object
Initial value:	NIL

This property indicates that the TDataSet object (record set) will be linked to the control.

It must be indicated its oDataField object to have the link complete.

1.7.5.10 TDBDateEditMod

This class represents a windows TDateEditMod control specialized in handling data using DataControls.

Hierarchy	TDateEditMod descendant
File	\source\DBDateEditMod.prg

1.7.5.10.1 TDBDateEditMod:Properties

■ Read only ■ Assignable ■ Design assignable ■ Run-time assignable

Scope	Name	Type	Initial value
■	IAutoSave	Logical	.T.
■	IEditable	Logical	.T.
■	oDataField	Object	NIL
■	oDataSet	Object	NIL

1.7.5.10.1.1 TDBDateEditMod:IAutoSave

If False, the control will not save its value in the database.

Scope:	Assignable
Type:	Logical
Initial value:	.T.

This property is used when you want to display the value of a particular field but you do not want to save it, even if you have changed its value. Its most frequent use is when you want to display values of non-editable fields such as auto incremental fields.

1.7.5.10.1.2 TDBDateEditMod:IEditable

If False, the control cannot be edited and always remains read-only.

Scope:	Assignable
Type:	Logical
Initial value:	.T.

1.7.5.10.1.3 TDBDateEditMod:oDataField

Object del Type TDataField vinculado con el control.

Scope:	Assignable
Type:	Object
Initial value:	NIL

Esta propiedad es la que establece todo el mecanismo de visualización y edición del dato vinculado al control. Hasta que no se asigne esta propiedad el control no muestra ninguna información sobre la base de datos ni tampoco es posible la edición del mismo.

Esta propiedad puede ser asignada directamente a través de un Object TDataField o se le puede indicar un literal con el Name del campo. En este último caso si su propiedad oDataSet ya ha sido asignada el control buscará dicho campo en el propio oDataSet, y si lo encuentra sustituirá

el literal por el propio Object TDataField encontrado.

Si se asigna esta propiedad directamente con un Object TDataField, el control automáticamente actualizará la propiedad oDataSet con el Object TDataSet propietario del propio TDataField pasado.

1.7.5.10.1.4 TDBDateEditMod:oDataSet

TDataSet object type linked to the control.

Scope:	Assignable
Type:	Objeto
Initial value:	NIL

This property indicates that the TDataSet object (record set) will be linked to the control.

You should indicate its oDataField objet as well to complete the link process.

1.7.5.11 TDBDatePicker

This class represents an standard windows date edit control, specialized to manage data through DataControls.



Hierarchy	Inherits from TDatePicker
See also	TDateEdit
File	\source\DBDatePicker.prg

1.7.5.11.1 TDBDatePicker:Properties

■ read Only ■ Assignable ■ Design assignable ■ Run-time assignable

Scop e	Name	Type	Initial value
■	IAutosave	Logic	.T.
■	IEditable	Logic	.T.
■	oDataField	Object	NIL

■	oDataSet	Object	NIL
---	----------	--------	-----

1.7.5.11.1.1 TDBDatePicker:IAutoSave

If it is .F., its value will not be updated on the underlying database.

Scope:	Assignable
Type:	Logic
Initial value:	.T.

This property is normally used when you want to show a value of a specific field but you do not want that the value is updated in any manner, even if its value has been modified. Its common use is when you want to show the value of non editable fields like, for example, auto incremental fields.

1.7.5.11.1.2 TDBDatePicker:Editable

If it is .F. the control can not be edited and will be treated as read only.

Scope:	Assignable
Type:	Lógico
Initial value:	.T.

1.7.5.11.1.3 TDBDatePicker:oDataField

TDataField object type linked to the control.

Scope:	Assignable
Type:	Object
Initial value:	NIL

This property establishes all the display and edit mechanisms from the linked data to the control. If this property is not assigned, the control will not show any information from the database and it is not possible to edit it neither.

This property can be assigned directly through a TDataField object or it is possible to indicate a literal with the field name. In this case if its oDataSet property has been already assigned, the control will search for that control in the oDataSet and if it is found it will replace the literal with the TDataField object found.

If this property is assigned automatically with a TDataField object, the control will automatically update the oDataSet property with the proprietary TDataSet object from the TDataField passed.

1.7.5.11.1.4 TDBDatePicker:oDataSet

TDataSet object type linked to the control.

Scope:	Assignable
Type:	Object
Initial value:	NIL

This property indicates that the TDataSet object (record set) will be linked to the control.

It must be indicated its oDataField object to have the link complete.

1.7.5.11.2 TDBDatePicker:Methods

■ Constructor ■ Standard ■ Only after Create()

Typ Name
■ Value

1.7.5.11.2.1 TDBDataPicker:Value

Returns the current control value. This method is common to all the DataControls.

Type	Only after Create()
Parameters	None
Return value	CheckBox state

If the linked oDataSet is in edit mode, this property returns the 'buffer' value from the linked oDataField object. Otherwise, this property will return the real field value from the database.

1.7.5.12 TDBEdit

This class represents an specialized TEdit windows control to manage data through the DataControls.



Hierarchy	Inherits from TEdit
File name	\source\DBEdit.prg

1.7.5.12.1 TDBEdit:Properties

read Only
 Assignable
 Design assignable
 Run-time assignable

Scope	Name	Type	Initial value
<input type="checkbox"/>	IAutosave	Logic	.T.
<input type="checkbox"/>	IEditable	Logic	.T.
<input type="checkbox"/>	oDataField	Object	NIL
<input type="checkbox"/>	oDataSet	Object	NIL

1.7.5.12.1.1 TDBEdit:IAutoSave

If it is .F., its value will not be updated on the underlying database.

Scope:	Assignable
Type:	Logic
Initial value:	.T.

This property is normally used when you want to show a value of a specific field but you do not want that the value is updated in any manner, even if its value has been modified. Its common use is when you want to show the value of non editable fields like, for example, auto incremental fields.

1.7.5.12.1.2 TDBEdit:IEditable

If it is .F., the control can not be edited and it will be read only.

Scope:	Assignable
Type:	Logic
Initial value:	.T.

1.7.5.12.1.3 TDBEdit:oDataField

TDataField object type linked to the control.

Scope:	Assignable
Type:	Object
Initial value:	NIL

This property establishes all the data view and edit mechanisms linked to the control. If this property is not assigned, the control will not show any information from the database and it will not be possible to edit it neither.

This property can be assigned directly through a TDataField object or can be assigned a literal with the field name. In this case if its oDataSet property has been already assigned, the control will search for that field in the oDataSet and if it found it will replace the literal with the own

TDataField object found.

If assigns this property directly with a TDataField object, the control will automatically update the oDataSet property with the proprietary TDataSet object from the TDataField passed.

1.7.5.12.1.4 TDBEdit:oDataSet

TDataSet object type linked to the control.

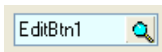
Scope:	Assignable
Type:	Object
Initial value:	NIL

This property indicates that the TDataSet object (record set) will be linked to the control.

You should indicate its oDataField object as well to complete the link process.

1.7.5.13 TDBEditBtn

This class represents an specialized Windows TEditBtn control to manage data through the DataControls.



Hierarchy	Inherits from TEditBtn
Filename	\source\DBEditBtn.prg

1.7.5.13.1 TDBEditBtn:Properties

■ read Only ■ Assignable ■ Design assignable ■ Run-time assignable

Scope	Name	Type	Initial value
■	IAutosave	Logic	.T.
■	IEditable	Logic	.T.
■	oDataField	Object	NIL
■	oDataSet	Object	NIL

1.7.5.13.1.1 TDBEditBtn:IAutoSave

If it is .F., its value will not be updated on the underlying database.

Scope:	Assignable
Type:	Logic

Initial value: .T.

This property is normally used when you want to show a value of a specific field but you do not want that the value is updated in any manner, even if its value has been modified. Its common use is when you want to show the value of non-editable fields like, for example, auto-incremental fields.

1.7.5.13.1.2 TDBEditBtn:IEditable

If it is .F., the control can not be edited and will be read only.

Scope:	Assignable
Type:	Logic
Initial value:	.T.

1.7.5.13.1.3 TDBEditBtn:oDataField

TDataField object type linked to the control.

Scope:	Assignable
Type:	Object
Initial value:	NIL

This property establishes all the data view and edit mechanisms linked to the control. If this property is not assigned, the control will not show any information from the database and it will not be possible to edit it neither.

This property can be assigned directly through a TDataField object or can be assigned a literal with the field name. In this case if its oDataSet property has been already assigned, the control will search for that field in the oDataSet and if it found it will replace the literal with the own TDataField object found.

If assigns this property directly with a TDataField object, the control will automatically update the oDataSet property with the proprietary TDataSet object from the TDataField passed.

1.7.5.13.1.4 TDBEditBtn:oDataSet

TDataSet object type linked to the control.

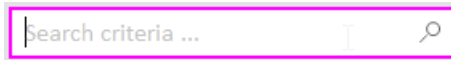
Scope:	Assignable
Type:	Object
Initial value:	NIL

This property indicates that the TDataSet object (record set) will be linked to the control.

You should indicate its oDataField objet as well to complete the link process.

1.7.5.14 TDBEditMod

This class represent a TEditMod control specialized on the management of DataControls.



Hierarchy TEditMod descendant
File \source\DBEditMod.prg

1.7.5.14.1 TDBEditMod:Properties

read Only
 Assignable
 Design assignable
 Run-time assignable

Scope	Name	Type	Initial value
<input checked="" type="checkbox"/>	IAutosave	Logic	.T.
<input checked="" type="checkbox"/>	IEditable	Logic	.T.
<input checked="" type="checkbox"/>	oDataField	Object	NIL
<input checked="" type="checkbox"/>	oDataSet	Object	NIL

1.7.5.14.1.1 TDBEditMod:IAutoSave

If it is .F. the control can not be edited and will be treated as read only.

Scope:	Assignable
Type:	Logic
Initial value:	.T.

1.7.5.14.1.2 TDBEditMod:IEditable

If it is .F. the control can not be edited and will be treated as read only.

Scope:	Assignable
Type:	Logic
Initial value:	.T.

1.7.5.14.1.3 TDBEditMod:oDataField

TDataField object type linked to the control.

Scope:	Assignable
Type:	Object
Initial value:	NIL

This property establishes all the display and edit mechanisms from the linked data to the control. If this property is not assigned, the control will not show any information from the database and it is not possible to edit it neither.

This property can be assigned directly through a TDataField object or it is possible to indicate a literal with the field name. In this case if its oDataSetproperty has been already assigned, the control will search for that control in the oDataSet and if it is found it will replace the literal with the TDataField object found.

If this property is assigned automatically with a TDataField object, the control will automatically update the oDataSet property with the proprietary TDataSet object from the TDataField passed.

1.7.5.14.1.4 TDBEditMod:oDataSet

TDataSet object type linked to the control.

Scope:	Assignable
Type:	Object
Initial value:	NIL

This property indicates that the TDataSet object (record set) will be linked to the control.

It must be indicated its oDataField object to have the link complete.

1.7.5.15 TDBHyperLink

This class represents an special THyperLink control to be used together with the DataControls.

Description:

The TDBHyperLink class represents a specialized THyperLink to manage data through the DataControl.s The difference with THyperLink is basically 2 properties: oDataSet to indicate the data origin and oDataField to indicate the oDataSet field to be used in the link. The typical cText property from the THyperLink object is hidden in the IDE object inspector because it does not make sense because now the control information will depend of the field value that we have indicated with the oDataField property.

When the control has been linked with any TDataset field, any TDataset navigation operation will

be reflected in the TDBHyperLink control value automatically.

Hierarchy Inherits from THyperLink
File name \source\DBHyperLink1.prg

1.7.5.15.1 TDBHyperLink:Properties

■ read Only ■ Assignable ■ Design assignable ■ Run-time assignable

Scope	Name	Type	Initial value
■	oDataField	Object or Character	NIL
■	oDataSet	Object	NIL

1.7.5.15.1.1 TDBHyperLink:oDataField

Establishes the oDataSet objet field to be linked with the control.

Scope	Assignable
Type	Object or Character
Initial value	NIL

The value can be a literal or an TDataField object proceeding from the oDataSet.

In the case that a literal is introduced in the property, the control will search this literal as field name in the oDataSet object. If is found the value of this property is updated with its corresponding TDataField object. In the case that this field is not found or the oDataSet has not been assigned yet, the oDataField property will keep the literal value and it will try to reassign it like TDataField object every time that the oDataSet property value is modified.

1.7.5.15.1.2 TDBHyperLink:oDataSet

Establishes the TDataSet object to be linked with the control.

Scope	Assignable
Type	Object
Initial value	NIL

1.7.5.15.2 TDBHyperLink:Methods

■ Constructor ■ Standard

Type	Name
■	Refresh

1.7.5.15.2.1 TDBHyperLink:Refresh

Updates the control value with its oDataset object information.

Type	Standard
Parameters	None
Return value	NIL

The difference with the classic Refresh method inherited from the TControl class, this method re-read its value from its oDataset object.

1.7.5.16 TDBLabel

This class represents an special text label control to be used together with the DataControls.

Description:

The TDBLabel class represents a specialized text label to manage data through the DataControl.s The difference with TLabel is basically 2 properties: oDataSet to indicate the data origin and oDataField to indicate the oDataSet field to be used in the link. The typical cText property from the TLabel object is hidden in the IDE object inspector because it does not make sense because now the control information will depend of the field value that we have indicated with the oDataField property.

When the control has been linked with any TDataset field, any TDataset navigation operation will be reflected in the TDBLabel control value automatically.

Hierarchy	Inherits from TLabel
File name	\source\DBLabel.prg

1.7.5.16.1 TDBLabel:Properties

■ read Only ■ Assignable ■ Design assignable ■ Run-time assignable

Scope	Name	Type	Initial value
■	IAutosize	Logic	.F.
■	oDataField	Object or Character	NIL
■	oDataSet	Object	NIL

1.7.5.16.1.1 TDBLabel:IAutosize

Automatically adjust the area where the control is drawn.

Scope	Assignable
Type	Logic
Initial value	.F.

This property takes .F. as default value, instead of .T. like its TLabel parent class.

1.7.5.16.1.2 TDBLabel:oDataField

Establishes the oDataSet objet field to be linked with the control.

Scope	Assignable
Type	Object or Character
Initial value	NIL

The value can be a literal or an TDataField object proceeding from the oDataSet.

In the case that a literal is introduced in the property, the control will search this literal as field name in the oDataSet object. If is found the value of this property is updated with its corresponding TDataField object. In the case that this field is not found or the oDataSet has not been assigned yet, the oDataField property will keep the literal value and it will try to reassign it like TDataField object every time that the oDataSet property value is modified.

1.7.5.16.1.3 TDBLabel:oDataSet

Establishes the TDataSet object to be linked with the control.

Scope	Assignable
Type	Object
Initial value	NIL

1.7.5.16.2 TDBLabel:Methods

■ Constructor ■ Standard

Type	Name
■	Refresh

1.7.5.16.2.1 TDBLabel:Refresh

Updates the control value with its oDataset object information.

Type	Standard
Parameters	None
Return value	NIL

The difference with the classic Refresh method inherited from the TControl class, this method re-read its value from its oDataset object.

1.7.5.17 TDBListBox

This class represents a specialized Windows ListBox control to manage data through DataControls.



Hierarchy	Inherits from TListBox
File name	\source\DBListBox.prg

1.7.5.17.1 TDBListoBox:Properties

■ read Only ■ Assignable ■ Design assignable ■ Run-time assignable

Scope	Name	Type	Initial value
■	altemsBound	Array	{}
■	IAutoSave	Logic	.T.
■	IEditable	Logic	.T.
■	nDataType	Numeric	dtDEFAULT
■	oDataField	Object	NIL
■	oDataSet	Object	NIL

1.7.5.17.1.1 TDBListBox:altemsBound

Array with all the values the contol will use to retrieve and assign its information to the dataset.

This property must be used in conjunction with the property nDataType assigning it a value of **dtBOUND**. This way the altems array will be used to show the elements in the listbox and the **altemsBound** array will be used to retrieve and save its value on the dataset. Logically the dimension of altems and altemsBound should be equal.

Scope:	Assignable
Type:	Array
Initial value:	{}

1.7.5.17.1.2 TDBListBox:IAutoSave

If it is .F., its value will not be updated on the underlying database.

Scope:	Assignable
Type:	Logic
Initial value:	.T.

This property is normally used when you want to show a value of a specific field but you do not want that the value is updated in any manner, even if its value has been modified. Its common use is when you want to show the value of non-editable fields like, for example, auto-incremental fields.

1.7.5.17.1.3 TDBListBox:IEditable

If it is .F. the control can not be edited and it will be treated as read only.

Scope:	Assignable
Type:	Logic
Initial value:	.T.

1.7.5.17.1.4 TDBListBox:nDataType

Indicates the search type in the control.

Scope:	Assignable
Type:	Numeric
Initial value:	.T.
Possible values:	dtDEFAULT, dtINDEX, dtSTRING, dtBOUND

- **dtDEFAULT:** for oDataField objects that return numeric values, it will correspond its value with the ListBox nIndex property. For character type values, it will search the string in the ListBox list of elements.
- **dtINDEX:** Regardless the oDataField type, the search will be done over the nIndex property. In the case of character type it will pass to numeric format including zeros to the left. This is the typical search when you have numeric values in character type fields. For example, "0001".
- **dtSTRING:** Regardless the oDataField type, the search will be done on the ListBox list of

elements.

- **dtBOUND:** The search and assign will not be done into `altems` array but on `altemsBound` array

Note: The **dtBOUND** style is **incompatible** with the property `ISort` set to true because the array `altemsBound` does not reflect the sort modifications.

1.7.5.17.1.5 TDBListBox:oDataField

TDataField object type linked to the control

Scope:	Assignable
Type:	Object
Initial value:	NIL

This property establishes all the display and edit mechanisms from the linked data to the control. If this property is not assigned, the control will not show any information from the database and it is not possible to edit it neither.

This property can be assigned directly through a TDataField object or it is possible to indicate a literal with the field name. In this case if its `oDataSet` property has been already assigned, the control will search for that control in the `oDataSet` and if it is found it will replace the literal with the TDataField object found.

If this property is assigned automatically with a TDataField object, the control will automatically update the `oDataSet` property with the proprietary TDataSet object from the TDataField passed.

1.7.5.17.1.6 TDBListBox:oDataSet

TDataSet object type linked to the control.

Scope:	Assignable
Type:	Object
Initial value:	NIL

This property indicates that the TDataSet object (record set) will be linked to the control.

It must be indicated its `oDataField` object to have the link complete.

1.7.5.17.2 TDBListBox:Methods

■ Constructor ■ Standard ■ Only after Create()

Typ	Name
■	Value

1.7.5.17.2.1 TDBListBox:Value

Returns the current control value. This method is common to all the DataControls.

Type	Only after Create()
Parameters	None
Return value	ListBox status

If the linked oDataSet control is in edit mode, this property returns the 'buffer' value from the oDataField linked object, otherwise this property returns the real value from the field in the database.

1.7.5.18 TDBListBoxMod

This class represents a windows TListBoxMod control specialized in handling data using DataControls.

Hierarchy	TListBoxMod descendant
File	\source\DBListBoxMod.prg

1.7.5.18.1 TDBListBoxMod:Properties

■ Read only ■ Assignable ■ Design assignable ■ Run-time assignable

Scope	Name	Type	Initial value
■	altemsBound	Array	{}
■	IAutoSave	Logical	.T.
■	IEditable	Logical	.T.
■	nDataType	Numeric	dtDEFAULT
■	oDataField	Object	NIL
■	oDataSet	Object	NIL

1.7.5.18.1.1 TDBListBoxMod:altemsBound

Array with the values to be used by the control to retrieve and save the information in the Dataset.

This property must be used in association with the nDataType property by setting it to **dtBOUND** and the nStyle property to **csDROPDOWNLIST**. This way, the altems array will only be used to display the items in the combo-box and the altemsBound array will be used to retrieve and save its value on the dataset. Logically, the dimension of altems and altemsBound must be the same.

Scope:	Assignable
---------------	------------

Type:	Array
Initial value:	{}

1.7.5.18.1.2 TDBListBoxMod:IAutoSave

If False, the control will not save its value in the database.

Scope:	Assignable
Type:	Logical
Initial value:	.T.

This property is used when you want to display the value of a particular field but you do not want to save it, even if you have changed its value. Its most frequent use is when you want to display values of non-editable fields such as auto incremental fields.

1.7.5.18.1.3 TDBListBoxMod:IEditable

If False, the control cannot be edited and always remains read-only.

Scope:	Assignable
Type:	Logical
Initial value:	.T.

1.7.5.18.1.4 TDBListBoxMod:nDataType

Indicates the way in which the combo box list will be searched.

Scope:	Assignable
Type:	Numeric
Initial value:	dtDEFAULT
Possible values:	dtDEFAULT, dtINDEX, dtSTRING, dtBOUND

- **dtDEFAULT:** For oDataField objects that return numeric values, the numeric value will be matched with the nIndex property of the combo-box. For character values, the string will be searched for in the list of combo-box items
- **dtINDEX:** Regardless of the type of the oDataField, the search will be done on the nIndex property. In the case of character type values it will be converted to numeric format by putting as many leading zeros as the length of the field. This value is the typical search when storing numeric data in character type fields. For example: "0001"
- **dtSTRING:** Regardless of the type of the oDataField, the search will be done on the comobox

items elements

- **dtBOUND:** The search and the assignment will be done over the `altemsBound` array

Note: The **dtBOUND** style is **incompatible** with the property `ISort` to true, because the array `altemsBound` does not reflect the sort changes.

1.7.5.18.1.5 TDBListBoxMod:oDataField

Object of the type `TDataField` linked to the control.

Scope:	Assignable
Type:	Object
Initial value:	NIL

This property is the one that establishes the entire mechanism of visualization and editing of the data linked to the control. Until it is not assigned, the control does not show any information about the database nor is it possible to edit it.

This property can be assigned directly through an `Object TDataField` or it can be indicated with a literal with the name of the field. In the latter case, if the property `oDataSet` has already been assigned, the control will search for that field in the `oDataSet`, and if it finds it, it will replace the literal with the `Object TDataField` found.

If this property is assigned directly with an `Object TDataField`, the control will automatically update the `oDataSet` property with the `Object TDataSet` owner of the `TDataField`.

1.7.5.18.1.6 TDBListBoxMod:oDataSet

`TDataSet` object type linked to the control.

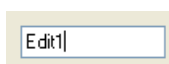
Scope:	Assignable
Type:	Objeto
Initial value:	NIL

This property indicates that the `TDataSet` object (record set) will be linked to the control.

You should indicate its `oDataField` object as well to complete the link process.

1.7.5.19 TDBMaskEdit

This class represents a `TMaskEdit` windows control, specialized in managing data through `DataControls`.



Hierarchy Inherits from TMaskEdit
File name \source\DBMaskEdit.prg

1.7.5.19.1 TDBMaskEdit:Properties

read Only Assignable Design assignable Run-time assignable

Scope	Name	Type	Initial value
<input type="checkbox"/>	IAutosave	Logic	.T.
<input type="checkbox"/>	IEditable	Logic	.T.
<input type="checkbox"/>	oDataField	Object	NIL
<input type="checkbox"/>	oDataSet	Object	NIL

1.7.5.19.1.1 TDBMaskEdit:IAutoSave

If it is .F., its value will not be updated on the underlying database.

Scope:	Assignable
Type:	Logic
Initial value:	.T.

This property is normally used when you want to show a value of a specific field but you do not want that the value is updated in any manner, even if its value has been modified. Its common use is when you want to show the value of non editable fields like, for example, auto incremental fields.

1.7.5.19.1.2 TDBMaskEdit:IEditable

If it is .F. the control can not be edited and will be treated as read only.

Scope:	Assignable
Type:	Logic
Initial value:	.T.

1.7.5.19.1.3 TDBMaskEdit:oDataField

TDataField object type linked to the control.

Scope:	Assignable
Type:	Object
Initial value:	NIL

This property establishes all the display and edit mechanisms from the linked data to the control. If this property is not assigned, the control will not show any information from the database and it is not possible to edit it neither.

This property can be assigned directly through a TDataField object or it is possible to indicate a literal with the field name. In this case if its oDataSet property has been already assigned, the control will search for that control in the oDataSet and if it is found it will replace the literal with the TDataField object found.

If this property is assigned automatically with a TDataField object, the control will automatically update the oDataSet property with the proprietary TDataSet object from the TDataField passed.

1.7.5.19.1.4 TDBMaskEdit:oDataSet

TDataSet object type linked to the control.

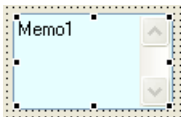
Scope:	Assignable
Type:	Object
Initial value:	NIL

This property indicates that the TDataSet object (record set) will be linked to the control.

It must be indicated its oDataField object to have the link complete.

1.7.5.20 TDBMemo

This class represents a TMemo Windows control, specialized on managing data through the DataControls.



Hierarchy	Inherits from TMaskEdit
File name	\source\DBMaskEdit.prg

1.7.5.20.1 TDBMemo:Properties

■ read Only ■ Assignable ■ Design assignable ■ Run-time assignable

Scope	Name	Type	Initial value
■	IAutosave	Logic	.T.
■	IEditable	Logic	.T.
■	oDataField	Object	NIL
■	oDataSet	Object	NIL

1.7.5.20.1.1 TDBMemo:IAutoSave

If it is .F., its value will not be updated on the underlying database.

Scope:	Assignable
Type:	Logic
Initial value:	.T.

This property is normally used when you want to show a value of a specific field but you do not want that the value is updated in any manner, even if its value has been modified. Its common use is when you want to show the value of non editable fields like, for example, auto incremental fields.

1.7.5.20.1.2 TDBMemo:IEditable

If it is .F. the control can not be edited and it will be treated as read only.

Scope:	Assignable
Type:	Logic
Initial value:	.T.

1.7.5.20.1.3 TDBMemo:oDataField

TDataField object type linked to the control.

Scope:	Assignable
Type:	Object
Initial value:	NIL

This property establishes all the display and edit mechanisms from the linked data to the control. If this property is not assigned, the control will not show any information from the database and it is not possible to edit it neither.

This property can be assigned directly through a TDataField object or it is possible to indicate a literal with the field name. In this case if its oDataSet property has been already assigned, the control will search for that control in the oDataSet and if it is found it will replace the literal with the TDataField object found.

If this property is assigned automatically with a TDataField object, the control will automatically update the oDataSet property with the proprietary TDataSet object from the TDataField passed.

1.7.5.20.1.4 TDBMemo:oDataSet

TDataSet object type linked to the control.

Scope:	Assignable
Type:	Object
Initial value:	NIL

This property indicates that the TDataSet object (record set) will be linked to the control.

It must be indicated its oDataField object to have the link complete.

1.7.5.21 TDBMemoMod

This class represents a TMemoMod control, specialized on managing data through the DataControls.



Hierarchy TMemoMod descendant
File \source\DBMemoMod.prg

1.7.5.21.1 TDBMemoMod:Properties

■ read Only ■ Assignable ■ Design assignable ■ Run-time assignable

Scope	Name	Type	Initial value
■	IAutosave	Logic	.T.
■	IEditable	Logic	.T.
■	oDataField	Object	NIL
■	oDataSet	Object	NIL

1.7.5.21.1.1 TDBMemoMod:IAutoSave

Si falso el control no se producirá el salvado de su valor en la base de datos.

Ámbito:	Asignable
Tipo:	Lógico
Valor inicial:	.T.

Esta propiedad se utiliza cuando se desea mostrar el valor de un determinado campo pero en ningún caso se desea su grabación, incluso aunque se haya cambiado el valor del mismo. Su uso más frecuente es cuando se desean mostrar valores de campos no editables como por ejemplo, campos del tipo autoincremental.

1.7.5.21.1.2 TDBMemoMod:IEditable

Si falso el control no se puede editar permaneciendo siempre como de sólo lectura.

Ámbito:	Asignable
Tipo:	Lógico
Valor inicial:	.T.

1.7.5.21.1.3 TDBMemoMod:oDataField

Objeto del tipo TDataField vinculado con el control.

Ámbito:	Asignable
Tipo:	Objeto
Valor inicial:	NIL

Esta propiedad es la que establece todo el mecanismo de visualización y edición del dato vinculado al control. Hasta que no se asigne esta propiedad el control no muestra ninguna información sobre la base de datos ni tampoco es posible la edición del mismo.

Esta propiedad puede ser asignada directamente a través de un objeto TDataField o se le puede indicar un literal con el nombre del campo. En este último caso si su propiedad oDataSet ya ha sido asignada el control buscará dicho campo en el propio oDataSet, y si lo encuentra sustituirá el literal por el propio objeto TDataField encontrado.

Si se asigna esta propiedad directamente con un objeto TDataField, el control automáticamente actualizará la propiedad oDataSet con el objeto TDataSet propietario del propio TDataField pasado.

1.7.5.21.1.4 TDBMemoMod:oDataSet

Objeto del tipo TDataSet vinculado con el control.

Ámbito:	Asignable
Tipo:	Objeto
Valor inicial:	NIL

Esta propiedad indica que objeto TDataSet (colección de registros) estará vinculado con el control.

Deberá indicar igualmente su objeto oDataField para realizar el enlace completo.

1.7.5.22 TDBNavigator

This class represents a ToolBar type control specialized on Datasets basic operations, like Dataset navigation, creation, edition and record suppression.



Hierarchy	TControl descendant
See also	TToolButton
File	\source\ToolBar.prg

1.7.5.22.1 TDBNavigator:Properties

■ Read only ■ Assignable ■ Design assignable ■ Run-time assignable

Scope	Name	Type	Initial value
■	aToolTips	Array	{}
■	IBtnCancel	Logical	.T.
■	IBtnDelete	Logical	.T.
■	IBtnEdit	Logical	.T.
■	IBtnFirst	Logical	.T.
■	IBtnInsert	Logical	.T.
■	IBtnLast	Logical	.T.
■	IBtnNext	Logical	.T.
■	IBtnPrevious	Logical	.T.
■	IBtnRefresh	Logical	0
■	IBtnUpdate	Logical	0
■	nAlign	Numeric	alNONE
■	oDataSet	Object	NIL

1.7.5.22.1.1 TDBNavigator:aTooltips

Array with all the buttons tooltips of the Toolbar. Be aware that there are 10 possible buttons and always on this order:

1. First
2. Previous
3. Next
4. Last
5. Add
6. Delete
7. Edit
8. Ok
9. Cancel
10. Refresh

You should use a 10 elements array and each array element should correspond with one of the previous buttons, even if some of them are not visible.

Scope	Assignable
Type	Matriz
Initial value	{}

1.7.5.22.1.2 TDBNavigator:IBtnCancel

If true the Toolbar will show the Cancel button.

Scope	Assignable
Type	Logical
Initial value	.T.

1.7.5.22.1.3 TDBNavigator:IBtnDelete

If true the Toolbar will show the Delete button.

Scope	Assignable
Type	Logical
Initial value	.T.

1.7.5.22.1.4 TDBNavigator:IBtnEdit

If true the Toolbar will show the Edit button.

Scope	Assignable
--------------	------------

Type	Logical
Initial value	.T.

1.7.5.22.1.5 TDBNavigator:IBtnFirst

If true the Toolbar will show the 'Go First' button.

Scope	Assignable
Type	Logical
Initial value	.T.

1.7.5.22.1.6 TDBNavigator:IBtnInsert

If true the Toolbar will show the Insert button.

Scope	Assignable
Type	Logical
Initial value	.T.

1.7.5.22.1.7 TDBNavigator:IBtnLast

If true the Toolbar will show the 'Go last' button.

Scope	Assignable
Type	Logical
Initial value	.T.

1.7.5.22.1.8 TDBNavigator:IBtnNext

If true the Toolbar will show the 'Go next' button.

Scope	Assignable
Type	Logical
Initial value	.T.

1.7.5.22.1.9 TDBNavigator:IBtnPrevious

If true the Toolbar will show the 'Go previous' button.

Scope	Assignable
Type	Logical
Initial value	.T.

1.7.5.22.1.10 TDBNavigator:IBtnRefresh

If true the Toolbar will show the Refresh button.

Scope	Assignable
Type	Logical
Initial value	.T.

1.7.5.22.1.11 TDBNavigator:IBtnUpdate

If true the Toolbar will show the Update button.

Scope	Assignable
Type	Logical
Initial value	.T.

1.7.5.22.1.12 TDBNavigator:nAlign

Control alignment in its oParent object container.

Scope:	Assignable
Type:	Numeric
Initial value:	aINONE
Possible values:	aINONE, alLEFT, alTOP, alRIGHT, alBOTTOM, alCLIENT

Description:

This property allows to the controls to adjust its dimensions and position to its oParent object container. The alignment can be:

- **None:** Default value
- **Left:** The control is aligned to the left of its oParent control and it takes the height of its client container.
- **Upper:** The control is aligned to the top of its oParent control and it takes the width of its client

- container.
- **Right:** The control is aligned to the right of its oParent control and it takes the height of its client container.
 - **Bottom:** The control is aligned to the bottom of its oParent control and it takes the width of its client container.
 - **Client:** The control is aligned the client area of its oParent control adjusts itself its size.

1.7.5.22.1.13 TDBNavigator:oDataSet

TDataSet object type linked to the control.

Scope	Assignable
Type	Object
Initial value	NIL

1.7.5.22.2 TDBNavigator:Events

Name
OnClickCancel
OnClickDelete
OnClickEdit
OnClickFirst
OnClickInsert
OnClickLast
OnClickNext
OnClickPrevious
OnClickRefresh
OnClickUpdate

1.7.5.22.2.1 TDBNavigator:OnClickCancel

Event that is produced when the Delete button is pressed.

Parameters	<oSender>:
:	Reference to the object that triggers the event
Return value:	If it returns false the default operation of the button is canceled

1.7.5.22.2.2 TDBNavigator:OnClickDelete

Event that is produced when the Delete button is pressed.

Parameters	<oSender> :
:	Reference to the object that triggers the event
Return value:	If it returns false the default operation of the button is canceled

1.7.5.22.2.3 TDBNavigator:OnClickEdit

Event that is produced when the Edit button is pressed.

Parameters	<oSender> :
:	Reference to the object that triggers the event
Return value:	If it returns false the default operation of the button is canceled

1.7.5.22.2.4 TDBNavigator:OnClickFirst

Event that is produced when the 'Go first' button is pressed..

Parameters	<oSender> :
:	Reference to the object that triggers the event
Return value:	If it returns false the default operation of the button is canceled

1.7.5.22.2.5 TDBNavigator:OnClickInsert

Event that is produced when the Insert button is pressed.

Parameters	<oSender> :
:	Reference to the object that triggers the event
Return value:	If it returns false the default operation of the button is canceled

1.7.5.22.2.6 TDBNavigator:OnClickLast

Event that is produced when the 'Go Last' button is pressed.

Parameters	<oSender> :
:	Reference to the object that triggers the event
Return value:	If it returns false the default operation of the button is canceled

1.7.5.22.2.7 TDBNavigator:OnClickNext

Event that is produced when the 'Go next' button is pressed..

Parameters	<oSender> :
:	Reference to the object that triggers the event
Return value:	If it returns false the default operation of the button is canceled

1.7.5.22.2.8 TDBNavigator:OnClickPrevious

Event that is produced when the 'Go previous' button is pressed.

Parameters	<oSender> :
:	Reference to the object that triggers the event
Return value:	If it returns false the default operation of the button is canceled

1.7.5.22.2.9 TDBNavigator:OnClickRefresh

Event that is produced when the Refresh button is pressed.

Parameters	<oSender> :
:	Reference to the object that triggers the event
Return value:	If it returns false the default operation of the button is canceled

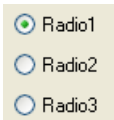
1.7.5.22.2.10 TDBNavigator:OnClickUpdate

Event that is produced when the Update button is pressed..

Parameters	<oSender>:
:	Reference to the object that triggers the event
Return value:	If it returns false the default operation of the button is canceled

1.7.5.23 TDBRadioMenu

This class represents a RadioMenu button specialized to manage data through the DataControls.



Hierarchy	Inherits from TRadioMenu
File name	\source\DBRadioMenu.prg

1.7.5.23.1 TDBRadioMenu:Properties

■ read Only ■ Assignable ■ Design assignable ■ Run-time assignable

Scope	Name	Type	Initial value
■	IAutosave	Logic	.T.
■	IEditable	Logic	.T.
■	oDataField	Object	NIL
■	oDataSet	Object	NIL

1.7.5.23.1.1 TDBRadioMenu:IAutoSave

If it is .F., its value will not be updated on the underlying database.

Scope:	Assignable
Type:	Logic
Initial value:	.T.

This property is normally used when you want to show a value of a specific field but you do not want that the value is updated in any manner, even if its value has been modified. Its common use is when you want to show the value of non editable fields like, for example, auto incremental fields.

1.7.5.23.1.2 TDBRadioMenu:IEditable

If it is .F. the control can not be edited and it will be treated as read only.

Scope:	Assignable
Type:	Logic
Initial value:	.T.

1.7.5.23.1.3 TDBRadioMenu:oDataField

TDataField object type linked to the control.

Scope:	Assignable
Type:	Object
Initial value:	NIL

This property establishes all the display and edit mechanisms from the linked data to the control. If this property is not assigned, the control will not show any information from the database and it is not possible to edit it neither.

This property can be assigned directly through a TDataField object or it is possible to indicate a literal with the field name. In this case if its oDataSetproperty has been already assigned, the control will search for that control in the oDataSet and if it is found it will replace the literal with the TDataField object found.

If this property is assigned automatically with a TDataField object, the control will automatically update the oDataSet property with the proprietary TDataSet object from the TDataField passed.

1.7.5.23.1.4 TDBRadioMenu:oDataSet

TDataSet object type linked to the control.

Scope:	Assignable
Type:	Object
Initial value:	NIL

This property indicates that the TDataSet object (record set) will be linked to the control.

It must be indicated its oDataField object to have the link complete.

1.7.5.23.2 TDBRadioMenu:Methods

Constructor Standard Only after Create()

Typ	Name
<input type="checkbox"/>	Value

1.7.5.23.2.1 TDBRadioMenu:Value

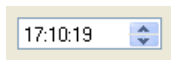
Returns the current control value. This method is common to all the DataControls.

Type	Only after Create()
Parameters	None
Return value	Checkbox status

If the linked oDataSet control is in edit mode, this property returns the 'buffer' value from the oDataField linked object, otherwise this property returns the real value from the field in the database.

1.7.5.24 TDBTimePicker

This class represents a TTimePicker control specialized to manage data through the DataControls.



Hierarchy Inherits from TTimePicker
File name \source\DBTimePicker.prg

1.7.5.24.1 TDBTimePicker:Properties

read Only Assignable Design assignable Run-time assignable

Scope	Name	Type	Initial value
<input type="checkbox"/>	IAutosave	Logic	.T.
<input type="checkbox"/>	IEditable	Logic	.T.
<input type="checkbox"/>	oDataField	Object	NIL
<input type="checkbox"/>	oDataSet	Object	NIL

1.7.5.24.1.1 TDBTimePicker:IAutoSave

If it is .F., its value will not be updated on the underlying database.

Scope:	Assignable
---------------	------------

Type:	Logic
Initial value:	.T.

This property is normally used when you want to show a value of a specific field but you do not want that the value is updated in any manner, even if its value has been modified. Its common use is when you want to show the value of non-editable fields like, for example, auto-incremental fields.

1.7.5.24.1.2 TDBTimePickerEditable

If it is .F. the control can not be edited and will be kept as read-only.

Scope:	Assignable
Type:	Logic
Initial value:	.T.

1.7.5.24.1.3 TDBTimePicker:oDataField

TDataField object type linked to the control.

Scope:	Assignable
Type:	Object
Initial value:	NIL

This property establishes all the display and edit mechanisms for the data linked to the control. If this property is not assigned the control will not show any information from the database and it is not possible to edit it neither.

This property can be assigned directly through a TDataField object or can be indicated with a text with the field name. In this case if its oDataSet has already assigned the control will look for that field in the oDataSet and if it is found it will replace the text for the TDataField object found.

If this property is assigned directly with a TDataField object, the control automatically updates the oDataSet property with the proprietary TDataSet object from the TDataField passed.

1.7.5.24.1.4 TDBTimePicker:oDataSet

TDataSet object type linked to the control.

Scope:	Assignable
Type:	Object
Initial value:	NIL

This property indicates that the TDataSet object (DataSet) will be linked to the control.

You must indicate as well its oDataField object to complete the link.

1.7.5.24.2 TDBTimePicker:Methods

■ Constructor ■ Standard ■ Only after Create()

Type	Name
■	Value

1.7.5.24.2.1 TDBTimePicker:Value

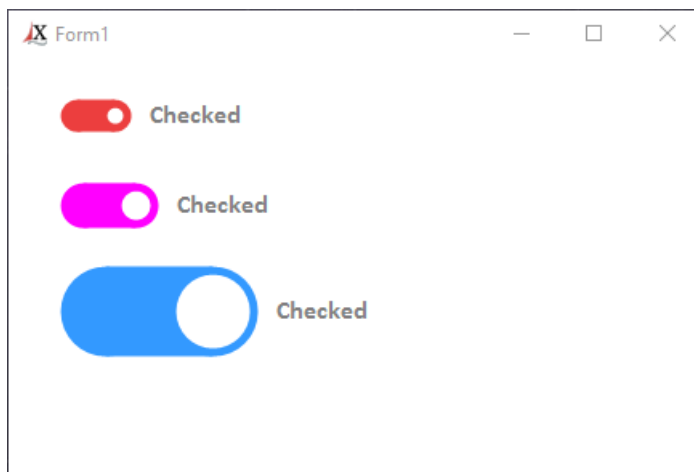
Returns the current control value. This method is common to all the DataControls.

Type	Only after Create()
Parameters	None
Return value	TimePickervalue

If the oDataSet linked to the control is in edit mode, this property returns the 'buffer' value from the oDataField linked object. Otherwise, this property will return the real DataField value from the database.

1.7.5.25 TDBSwitch

This class represents TSwitch control specialized on data management through DataControls.



Hierarchy
File

TSwitch descendant
\source\DBSwitch.prg

1.7.5.25.1 TDBSwitch:Properties

■ read Only ■ Assignable ■ Design assignable ■ Run-time assignable

Scope	Name	Type	Initial value
■	IAutosave	Logic	.T.
■	IEditable	Logic	.T.
■	oDataField	Object	NIL
■	oDataSet	Object	NIL

1.7.5.25.1.1 TDBSwitch:IAutoSave

If it is .F., its value will not be updated on the underlying database.

Scope:	Assignable
Type:	Logic
Initial value:	.T.

This property is normally used when you want to show a value of a specific field but you do not want that the value is updated in any manner, even if its value has been modified. Its common use is when you want to show the value of non editable fields like, for example, auto incremental fields.

1.7.5.25.1.2 TDBSwitch:IEditable

If it is .F. the control can not be edited and it will be treated as read only.

Scope:	Assignable
Type:	Logic
Initial value:	.T.

1.7.5.25.1.3 TDBSwitch:oDataField

TDataField object type linked to the control.

Scope:	Assignable
Type:	Object
Initial value:	NIL

This property establishes all the display and edit mechanisms from the linked data to the control. If this property is not assigned, the control will not show any information from the database and it is not possible to edit it neither.

This property can be assigned directly through a TDataField object or it is possible to indicate a literal with the field name. In this case if its oDataSet property has been already assigned, the control will search for that control in the oDataSet and if it is found it will replace the literal with the

TDataField object found.

If this property is assigned automatically with a TDataField object, the control will automatically update the oDataSet property with the proprietary TDataSet object from the TDataField passed.

1.7.5.25.1.4 TDBSwitch:oDataSet

TDataSet object type linked to the control.

Scope:	Assignable
Type:	Object
Initial value:	NIL

This property indicates that the TDataSet object (record set) will be linked to the control.

It must be indicated its oDataField object to have the link complete.

1.7.5.25.2 TDBSwitch:Methods

■ Constructor ■ Standard ■ Only after Create()

Type	Name
■	Value

1.7.5.25.2.1 TDBSwitch:Value

Returns the value of the control. This method is common to all DataControls

Type	Only after Create()
Parámetros	None
Return value	State value

If the linked oDataSet is in edit mode, this property returns the 'buffer' value from the linked oDataField object. Otherwise, this property will return the real field value from the database.

1.8 Other Classes and Objects

1.8.1 TBarcode

This class allows the printing of bar codes of very different formats. It is based on the **hbZebra** library included in Harbour.

Hierarchy TComponent descendant

File Internal

1.8.1.1 TBarcode:Properties

■ Read only
 ■ Assignable
 ■ Design assignable
 ■ Run-time assignable

Scope	Name	Type	Initial value
■	cText	Character	""
■	nLastError	Numeric	0
■	nType	Numeric	zbNONE

1.8.1.1.1 TBarcode:cText

Bar code descriptor.

Scope	Assignable
Type	Character
Initial value	""

1.8.1.1.2 TBarcode:nLastError

Last error code.

Scope	Read only
Type	Numeric
Initial value	0

1.8.1.1.3 TBarcode:nType

Bar code type.

Scope	Assignable
Type	Numeric
Initial value	zbNONE
Possible values	zbNONE zbEAN2 (Numeric, 2 positions) zbEAN5 (Numeric, 5 positions) zbEAN13 (Numeric, 13 positions) zbEAN8 (Numeric, 13 positions) zbUPCA (Numeric, 12 positions) zbUPCE (Numeric, 6 positions) zbCODE128 (alphanumeric, variable) zbCODE93 (A-Z, 0-9, '-.\$/+%', variable)

[zbCODE39](#) (A-Z, 0-9, '-.\$/+%', variable)
[zbCODE11](#) (Numeric, variable)
[zbCODABAR](#) (A-Z, 0-9, '-.\$/+%', variable)
[zbITF](#) (Numeric, variable)
[zbMSI](#) (Numeric, variable)
[zbPDF417](#) (ASCII limited, variable)
[zbDATAMATRIX](#) (ASCII complete, variable)
[zbQRCODE](#) (ASCII limited, variable)

For further information about all the code bar types, consult the following links:

- **EAN2** - https://en.wikipedia.org/wiki/EAN_2
- **EAN5** - https://en.wikipedia.org/wiki/EAN_5
- **EAN13** - https://en.wikipedia.org/wiki/European_Article_Number
- **EAN8** - https://en.wikipedia.org/wiki/Global_Trade_Item_Number
- **UPCA** - https://en.wikipedia.org/wiki/Universal_Product_Code
- **UPCE** - https://en.wikipedia.org/wiki/Universal_Product_Code#UPC-E
- **CODE128** - https://en.wikipedia.org/wiki/Code_128
- **CODE39** - https://en.wikipedia.org/wiki/Code_39
- **CODE93** - https://en.wikipedia.org/wiki/Code_93
- **CODE11** - https://en.wikipedia.org/wiki/Code_11
- **CODABAR** - <https://en.wikipedia.org/wiki/Codabar>
- **ITF** - https://en.wikipedia.org/wiki/Interleaved_2_of_5
- **MSI** - https://en.wikipedia.org/wiki/MSI_Barcode
- **PDF417** - <https://en.wikipedia.org/wiki/PDF417>
- **DATAMATRIX** - https://en.wikipedia.org/wiki/Data_Matrix
- **QRCODE** - https://en.wikipedia.org/wiki/QR_code

1.8.1.2 TBarcode:Methods

■ Constructor ■ Standard

Type	Name
■	AsString
■	GetErrorDescription
■	GetPicture
■	Print

1.8.1.2.1 TBarcode:AsString

Returns the bard code in bit mask format.

Type	Standard
Parameters	None
Return value	<cBarcode>

1.8.1.2.2 TBarcode:GetErrorDescription

Last error description.

Type	Standard
Parameters	None
Return value	<cError>

1.8.1.2.3 TBarcode:GetPicture

Retrieves the bar code as a TPicture object. Is important to establish the nBkGndMode property to [blSTRETCH](#) on its object container.

Type	Standard
Parameters	<nFlags> Flag modifier. Review hbzebra.ch
Return value	<TPicture>

1.8.1.2.4 TBarcode:Print

Prints the bar code an a specific TCanvas object.

Type	Standard
Parameters	<oCanvas> TCanvas object <aRect> Rectangle of coordinates according to the current unit of measurement of <i>oCanvas</i> <nFlags> Flag modifier. Review hbzebra.ch
Return value	<NIL>

1.8.2 TExStruct

The purpose of this class is to create the data structure that can be used through its member names instead of its numbers, as it happen with the arrays. A similar functionality can be found with Hash types or associative arrays that are only available in [x]Harbour. But this class some important's advantages over Hash types like:

- Strong typed members. You can force that any member become of a specific type and the casting will be done automatically to that type when ever possible
- Array save and restore methods are generated and used in the same order as members creation, and not sorted by key like in Hash types, that make them much useful used in

conjunction with huge arrays

- They admit a default initial value that can be easily retrieved calling the Reset method

Description:

This class can be used basically through the Xailer commands offered by Xailer in the ExStruct.ch file. You need to include manually this file in the PRG's header if you want to use this class.

The command involved in the structure creation are:

```
STRUC <oSt>
```

```
ENDSTRUC
```

```
MEMBER <xData, ...> ;
[ AS <LOGICAL, NUMERIC, NUMBER, CHARACTER, DATE, BLOCK, ARRAY, OBJECT,
STRUC> ] ;
[ <DEFAULT, INIT> <default> ] ;
[ SIZE <nSize> ]
```

The following can be a typical example:

```
STRUC oSt
  MEMBER cCode, cName AS CHARACTER
  MEMBER nSalary AS NUMERIC
  MEMBER dBirth AS DATE
ENDSTRUC
```

You can use the 'oSt' structure in the this way:

```
oSt:cCode := "0001"
```

Hierarchy Inherits from nobody
File Name \source\ExStruct.prg

1.8.2.1 TExStruct:Methods

■ Constructor ■ Standard

Type	Name
■	AddMember
■	Clone
■	GetDefaults
■	GetKeys
■	GetMember
■	GetValue
■	GetValues
■	GetName
■	Len
■	MemberList
■	Modified

■	New
■	Reset
■	SetValue
■	SetValues
■	ToJson

1.8.2.1.1 TExStruct:AddMember

Adds a new member to the structure

Type	Standard
Parameters	<p><xName>: Name or array names with the new structure members.</p> <p>[<cType>]: It can be any basic [x/Harbour type. If it is not specified, it will take any type.</p> <p>[<xDefault>]: Default value. NIL If is not specified.</p> <p>[<nSize>]: Data size. This parameter is only applicable to the character type. Default value: any.</p>
Return value	NIL

Description:

This method allows to create new members in the structure. This method is called internally by the MEMBER command.

1.8.2.1.2 TExStruct:Clone

Creates a new TExStruct object exactly the same as the actual.

Type	Standard
Parameters	None
Return value	oStruct

1.8.2.1.3 TExStruct:GetDefaults

Returns an array with all the members default values of the object.

Type	Standard
Parameters	None
Return Value	<aList>

1.8.2.1.4 TExStruct:GetKeys

Returns an array with all the members name of the object.

Type	Standard
Parameters	None
Return Value	<aList>

1.8.2.1.5 TExStruct:GetValue

Returns the value of any member of the object. Its use is equivalent as using its member name directly. For example this two lines of code return identical values:

```
oStruc:cName  
oStruc:GetValue( "cName" )
```

Type	Standard
Parameters	<cName>: Membername
Return Value	<xValue>: Membervalue

1.8.2.1.6 TExStruct:GetValues

Returns an array with all the members values of the object.

Type	Standard
Parameters	None
Return Value	<aList>

1.8.2.1.7 TExStruct:GetName

Returns a member name giving its ordinal creation order.

Type	Standard
Parameters	<nOrder>
Return Value	<cName>

1.8.2.1.8 TExStruct:GetMember

Returns the creation order of any member.

Type	Standard
Parameters	<cName>: Member name
Return Value	<nOrder>: Creation order

1.8.2.1.9 TExStruct:Len

Returns the number of members in the object.

Type	Standard
Parameters	None
Return Value	<nLen>: Members total

1.8.2.1.10 TExStruct:MemberList

Returns an array with all the structure members names.

Type	Standard
Parameters	None
Return value	<aList>

1.8.2.1.11 TExStruct:Modified

Returns true if the value of any member of the object is modified from its original value.

Type	Standard
Parameters	<cName> Membername
Return Value	<IValue>

1.8.2.1.12 TExStruct:New

Class constructor.

Type	Constructor
Parameters	None
Return value	Self reference

1.8.2.1.13 TExStruct:Reset

Initializes all the members with their default values.

Type	Standard
Parameters	None
Return Value	NIL

1.8.2.1.14 TExStruct:SetValue

Sets the value of any member of the object. Its use is equivalent as using directly its member name. For example this two lines of code are equivalent:

```
oStruc:cName := "Mike"  
oStruc:SetValue( "cName", "Mike" )
```

Type	Standard
Parameters	<cName> : Membername <xValue> : Newvalue
Return Value	NIL

1.8.2.1.15 TExStruct:SetValues

Assign values to all the members of the object. The assignment is done in the same order as the members creation.

Type	Standard
Parameters	<aList> : Data array
Return Value	NIL

1.8.2.1.16 TExStruct.ToJson

Returns the object as a JSON string.

Type	Standard
Parameters	<IType> : If true the object members will include their type with this construction: <NAME>_<TYPE> By default false
Return Value	JSON string

1.8.3 TFileName

Class to manipulate filenames.

Description:

The TFileName class allows to manipulate Windows files.

It is also possible to use directly the following functions, to make operations with filenames:

- [FileDrive](#)(**<cFileName>**) equivalent to the Drive method.
- [FileExtension](#)(**<cFileName>**) equivalent to the Extension method.
- [FileFullName](#)(**<cFileName>**) equivalent to the FullName method.
- [FileHasDrive](#)(**<cFileName>**) equivalent to the HasDrive method.
- [FileHasExtension](#)(**<cFileName>**) equivalent to the HasExtension method.
- [FileHasPath](#)(**<cFileName>**) equivalent to the HasPath method.
- [FilePath](#)(**<cFileName>**) equivalent to the Path method.
- [FileSetExtension](#)(**<cFileName>**, **cExtension**) equivalent to the SetExtension method.
- [FileSetName](#)(**<cFileName>**, **cName**) equivalent to the SetName method.
- [FileSetPath](#)(**<cFileName>**, **cPath**) equivalent to the SetPath method.
- [FileShortName](#)(**<cFileName>**) equivalent to the ShortName method.
- [FileUnique](#)(**<cPath>**, **cExtension**, **cRootName**) equivalent to the Unique method.

Hierarchy Inherits from TComponent
File name \source\FileName.prg

1.8.3.1 TFileName.Properties

■ read Only ■ Assignable ■ Design assignable ■ Run-time assignable

Scope	Name	Type	Initial value
■	cFileName	Character	""

1.8.3.1.1 TFileName.cFileName

Indicates the filename.

Scope	Assignable
Type	Character
Initial value	""

1.8.3.2 TFileName.Methods

■ Constructor ■ Standard

Type	Name
■	Create
■	Drive
■	Extension
■	FullName
■	HasDrive
■	HasExtension
■	HasPath
■	New
■	Path
■	SetExtension
■	SetName
■	SetPath
■	ShortName
■	Unique
■	WithExtension
■	WithName
■	WithPath

1.8.3.2.1 TFileName:Drive

Indicates the file drive.

Type	Standard
Parameters	None
Return value	<cDrive> String with the file drive identification.

1.8.3.2.2 TFileName:Extension

Provides the filename extension.

Type	Standard
Parameters	None
Return value	<cExtension> String with the filename extension

1.8.3.2.3 TFileName:FullName

Provides the full filename, without the path.

Type	Standard
Parameters	None
Return value	<cName> String with the filename (without the path)

1.8.3.2.4 TFileName:HasDrive

Indicates if the filename includes the drive identifier.

Type	Standard
Parameters	None
Return value	<IDrive> .T. if the filename contains the drive identifier. .F. if it does not include it.

1.8.3.2.5 TFileName:HasExtension

Indicates if the filename has an extension.

Type	Standard
Parameters	None
Return value	<IExtensión> .T. if the filename has an extension

1.8.3.2.6 TFileName:HasPath

Indicates if the filename includes the path.

Type	Standard
Parameters	None
Return value	<IPath> .T. if the filename includes the path

1.8.3.2.7 TFileName:New

Class constructor.

Type	Constructor
Parameters	<cFilename> Filename
Return value	Self reference

1.8.3.2.8 TFileName:Path

Provides the filename path.

Type	Standard
Parameters	None
Return value	<cPath> String with the filename path

1.8.3.2.9 TFileName:SetExtension

Replaces the filename extension with a new one.

Type	Standard
Parameters	<cExtensión> Indicates the new filename extension <IUpdate> if it is .T., ::cFileName will be updated.; Default value: .T.
Return value	<cFileName> Returns the new filename

1.8.3.2.10 TFileName:SetName

Replaces the filename with a new one.

Type	Standard
Parameters	<cName> Indicates the new filename <IUpdate> If it is .T., ::cFileName will be updated; Default value: .T.
Return value	<cFileName> Returns the new filename

1.8.3.2.11 TFileName:SetPath

Replaces the filename path with a new one..

Type	Standard
Parameters	<cPath> Indicates the new filename path <IUpdate> If it is .T., ::cFileName will be updated; Default value: .T.
Return value	<cFileName> Returns the filename with the new path

1.8.3.2.12 TFileName:ShortName

Provides the filename, without the path and extension.

Type	Standard
Parameters	None
Return value	<cName> String with the filename, without path and extension

1.8.3.2.13 TFileName:Unique

Provides the filename, the path and the extension of a unique temporary file.

Type	Standard
Parameters	<cExtension> File Extension. Default value: "tmp" <cRootName>

	Basic filename. Default: "XA_"
Return value	<cFileName> Returns the temporary filename.

Important Note: This method always tries to erase the file so it gives you the chance to reuse old and orphan temporal files due a run time error or a power down. Logically if the temporal file is in use the function will not be able to delete it and it will return another valid name. If you do not want that functionality you may use the function `GetTempFileName()`

1.8.3.2.14 TFileName:WithExtension

Replaces the file extension for a new one.

Type	Standard
Parameters	<cExtension> Indicates the new file extension
Return value	<cFileName> Returns the filename, with a new extension

1.8.3.2.15 TFileName:WithName

Replaces the filename with a new one.

Type	Standard
Parameters	<cName> Indicates the new file name
Return value	<cFileName> Returns the new filename

1.8.3.2.16 TFileName:WithPath

Replaces the filename path for a new one.

Type	Standard
Parameters	<cPath> New path
Return value	<cFileName> Returns the filename, with the new path.

1.8.4 TRecentList

Manage of most recently used lists (MRU)

Description:

This class is used basically to manage the most recently used lists. The class that manages these lists is the TRecentListMenu class that inherits from this class.

Hierarchy Inherits from TComponent
File Name \source\RecentList.prg

1.8.4.1 TRecentList:Properties

■ read Only ■ Assignable ■ Design assignable ■ Run-time assignable

Scope	Name	Type	Initial value
■	altems	Array	{}
■	clniFile	Character	""
■	clniSection	Character	""
■	IAutoRemove	Logic	.F.
■	IAutoSave	Logic	.F.
■	nMaxItems	Numeric	5

1.8.4.1.1 TRecentList:altems

Recent list of elements that manages the control.

Scope	Assignable
Type	Array
Initial value	{}

1.8.4.1.2 TRecentList:clniFile

INI filename that stores the list of elements.

Scope	Assignable
Type	Character
Initial value	""

1.8.4.1.3 TRecentList:cIniSection

INI file section name that stores the list of elements.

Scope	Assignable
Type	Character
Initial value	""

1.8.4.1.4 TRecentList:IAutoRemove

Removes all the elements that don't exist. This property only makes sense when the list manages file names.

Scope	Assignable
Type	Logic
Initial value	.F.

1.8.4.1.5 TRecentList:IAutoSave

Indicates if the list must be saved every time that a element is added, modified or deleted.

Scope	Assignable
Type	Logic
Initial value	.F.

1.8.4.1.6 TRecentList:nMaxItems

Indicates the maximum number of elements that manages the list.

Scope	Assignable
Type	Numeric
Initial value	5

1.8.4.2 TRecentList:Methods

■ Constructor ■ Standard

Typ	Name
■	Add
■	Delete

■	DeleteItems
■	Read
■	Refresh
■	SetIniFile
■	SetItems
■	Write

1.8.4.2.1 TRecentList:Add

Adds a new element to the list.

Type	Standard
Parameters	<cltem> Elements description
Return value	NIL

1.8.4.2.2 TRecentList:Delete

Deletes one elements from the list.

Type	Standard
Parameters	<nItem> Index from the element in the list.
Return value	NIL

1.8.4.2.3 TRecentList:DeleteItems

Deletes all the elements from the list.

Type	Standard
Parameters	None
Return value	NIL

1.8.4.2.4 TRecentList:Read

Reads a list of elements from the selected file.

Type	Standard
Parameters	<cFile> INI file name. The default value is "ApplicationName.ini" <cSection> INI file section name. Default: "RecentList"
Return value	NIL

1.8.4.2.5 TRecentList:Refresh

Refresh the elements in the list. Checks that the number of elements is not higher than nMaxItems and saves the changes if the IAutoSave property is active.

Type	Standard
Parameters	None
Return value	NIL

1.8.4.2.6 TRecentList:SetIniFile

Assigns the file name and the section that stores the recent list of elements.

Type	Standard
Parameters	<cFile> INI file name. The default value is "ApplicationName.ini" <cSection> INI file section name. Default: "RecentList"
Return value	NIL

1.8.4.2.7 TRecentList:SetItems

Assigns a new list of elements. It is equivalent to assign the altems property.

Type	Standard
Parameters	<altems> List of elements

Return value	NIL
---------------------	-----

1.8.4.2.8 TRecentList:Write

Writes the list of elements in the selected file.

Type	Standard
Parameters	<cFile> INI file name. The default value is "ApplicationName.ini" <cSection> INI file section name. Default: "RecentList"
Return value	NIL

1.8.5 TRecentListMenu

This class manage the Recent list menus (MRU).

Description:

Recent list menus management (MRU).

Hierarchy	Inherits from TRecentList
File name	\source\RecentMenu.prg

1.8.5.1 TRecentListMenu:Properties

■ read Only ■ Assignable ■ Design assignable ■ Run-time assignable

Scope	Name	Type	Initial value
■	cMessage	Character	"Open the selected file"
■	cTextEmpty	Character	"<None>"
■	IFullPath	Logic	.F.
■	INumerate	Logic	.F.
■	IPrefix	Logic	.F.
■	ISeparator	Logic	.T.
■	ISubMenu	Logic	.F.
■	IText	Logic	"Recent files"
■	nMaxWidth	Numeric	0
■	nMenu	Numeric	1
■	nMenuPos	Numeric	0
■	oMenu	Object	NIL

1.8.5.1.1 TRecentListMenu:cMessage

Message that is shown in the status bar.

Scope	Design assignable
Type	Character
Initial value	"Open the selected file"

1.8.5.1.2 TRecentListMenu:IFullPath

Indicates if needs to show the complete file path or only the filename and extension.

Scope	Assignable
Type	Logic
Initial value	.F.

1.8.5.1.3 TRecentListMenu:INumerate

Numbers the elements, indicating its position in the list.

Scope	Assignable
Type	Logic
Initial value	.F.

1.8.5.1.4 TRecentListMenu:IPrefix

It is used with INumerate and underlines the number that indicates the element position in the list. In this way it is possible to use the number like a menu shortcut.

Scope	Assignable
Type	Logic
Initial value	.F.

1.8.5.1.5 TRecentListMenu:ISeparator

Includes a separator before the first element.

Scope	Design assignable
Type	Logic

Initial value	.T.
----------------------	-----

1.8.5.1.6 TRecentListMenu:lSubMenu

Indicates if it is needed to add entries below nMenuPos or in a submenu.

Scope	Design assignable
Type	Logic
Initial value	.F.

1.8.5.1.7 TRecentListMenu:lText

Submenu's title

Scope	Design assignable
Type	Logic
Initial value	.F.

1.8.5.1.8 TRecentListMenu:cTextEmpty

Description shown by the menu when it does not have any element.

Scope	Design assignable
Type	Character
Initial value	"<None>"

1.8.5.1.9 TRecentListMenu:nMaxWidth

Maximum number of characters to be shown.

Scope	Assignable
Type	Numeric
Initial value	0

1.8.5.1.10 TRecentListMenu:nMenu

Menu index, if the list depends from the main menu.

Scope	Design assignable
Type	Numeric
Initial value	1

1.8.5.1.11 TRecentListMenu:nMenuPos

First list element position.

Scope	Design assignable
Type	Numeric
Initial value	0

1.8.5.1.12 TRecentListMenu:oMenu

Menu object where the list is assigned.

Scope	Assignable
Type	Object
Initial value	NIL

1.8.5.2 TRecentListMenu:Methods

■ Constructor ■ Standard

Typ	Name
■	Create
■	Refresh

1.8.5.2.1 TRecentListMenu:Create

Creates the menu items list and inserts it in its belonging menu.

Type	Constructor
Parameters	<oMenu> Menu object where the list is assigned <nMenu> Index menu, if the list depends from the main menu

	<nMenuPos> First list element position.
Return value	Self reference (Self)

1.8.5.2.2 TRecentListMenu:Refresh

Refresh the elements from the list.

Type	Standard
Parameters	None
Return value	NIL

1.8.5.3 TRecentListMenu:Events

Name	
	OnClick

1.8.5.3.1 TRecentListMenu:OnClick

Event that is triggered when an element from the menu is selected.

Parameters	<oSender> Reference to the object that triggers the event <cFile> Selected file name
Return value	NIL

1.8.6 TUnzipFile

Class to decompress zip files.

Description:

The TUnzipFile class allows to decompress windows zip files.

This class descends from the **TZipArchive** class that establishes the common TZipFile and TUnzipFile functionality. The **TZipArchive** class is not documented due is its an internal function and has no utility for the programmer.

Note for xHarbour:

To use this class you must include the **HBZIP.LIB** and **ZLIB.LIB** libraries into your project.

Note for Harbour:

To use this class you must include the **HBMZIP.LIB**, **MINIZIP.LIB** and **ZLIB.LIB** libraries into your project.

Hierarchy Inherits from TZipArchive
File name \source\ZipArchive.prg

1.8.6.1 TUnzipFile:Properties

■ read Only ■ Assignable ■ Design assignable ■ Run-time assignable

Scope	Name	Type	Initial value
■	aFileMask	Array	*.*
■	aFiles *	Array	{}
■	cDirectory	Character	""
■	cFileName	Character	""
■	cLastError ^	Character	""
■	cPassword	Character	""
■	lIncludePath	Logic	.T.
■	nBufferSize ^	Numeric	256
■	nCount *	Numeric	0
■	nExtractBufferSize *	Numeric	32
■	nLastError ^	Numeric	0
■	nReadBufferSize *	Numeric	16
■	nWriteBufferSize *	Numeric	64

* Only available for xHarbour

^ Only available for Harbour

1.8.6.1.1 TUnzipFile:aFileMask

Mask from the files to be extracted. It can be defined a string like "*.cpp" or a list and path files { "*.cpp", "\include\test.h" }. Both, relative and absolute paths are supported.

Scope	Assignable
Type	Array
Initial value	*.*

1.8.6.1.2 TUnzipFile:aFiles

File list from the Zip file.

Scope	readOnly
Type	Array
Initial value	{}

Only available for xHarbour

1.8.6.1.3 TUnzipFile:cDirectory

Folder or directory where the file will be decompressed.

Scope	Assignable
Type	Character
Initial value	""

1.8.6.1.4 TUnzipFile:cFileName

Zip filename

Scope	Assignable
Type	Character
Initial value	""

1.8.6.1.5 TUnzipFile:cLastError

Description of last error produced.

Scope	Read only
Type	Character
Initial value	""

1.8.6.1.6 TUnzipFile:cPassword

Zip file password.

Scope	Assignable
--------------	------------

Type	Character
Initial value	""

1.8.6.1.7 TUnzipFile:IncludePath

Indicates if the path for the files will be restored.

Scope	Assignable
Type	Logic
Initial value	.T.

1.8.6.1.8 TUnzipFile:nBufferSize

Decompress buffer size in Kb.

Scope	Assignable
Type	Numeric
Initial value	256

Only available for Harbour

1.8.6.1.9 TUnzipFile:nCount

Indicates the number of files included in the Zip file.

Scope	read Only
Type	Numeric
Initial value	0

Only available for xHarbour

1.8.6.1.10 TUnzipFile:nExtractBufferSize

Extract buffer size in Kb.

Scope	Assignable
Type	Numeric
Initial value	32

Only available for xHarbour

1.8.6.1.11 TUnzipFile:nLastError

Last error number.

Scope	readOnly
Type	Numeric
Initial value	0

Only available for Harbour

1.8.6.1.12 TUnzipFile:nReadBufferSize

Read buffer size in Kb.

Scope	Assignable
Type	Numeric
Initial value	16

Only available for xHarbour

1.8.6.1.13 TUnzipFile:nWriteBufferSize

Write buffer size in Kb.

Scope	Assignable
Type	Numeric
Initial value	64

Only available for xHarbour

1.8.6.2 TUnzipFile:Methods

■ Constructor ■ Standard

Typ	Name
■	DeleteFiles ^
■	FilesCount ^
■	FilesInfo ^
■	FileSize ^
■	HasPassword ^

■ Run | Execute

* Only available for xHarbour

^ Only available for Harbour

1.8.6.2.1 TUnZipFile:FilesCount

Returns the number of file in the ZIP file.

Type	Standard
Parameters	None
Return value	<Count> Number of files

Only available for Harbour

1.8.6.2.2 TUnZipFile>DeleteFiles

Deletes inner files from the ZIP file.

Type	Standard
Parameters	<aFiles> Filename array list
Return value	<ISuccess> True if success

Note: The process stops returning false if any of the files can not be deleted. Previous deleted files are not restored.

Only available for Harbour

1.8.6.2.3 TUnZipFile:FilesInfo

Returns an array with information of all the files in the ZIP file.

Type	Standard
Parameters	None
Return value	<aFiles> A multi-dimensional array with the following items on its subelements: - File name - File size - Compression method - Compressed size - Compressing ratio - Date - Time

- CRC
- Attributes
- Encrypted (Yes/No)
- Comment

Only available for Harbour

1.8.6.2.4 TUnZipFile:FileSize

Returns the total bytes size of all the files in the ZIP file.

Type	Standard
Parameters	None
Return value	<nTotal> Bytes total

Only available for Harbour

1.8.6.2.5 TUnZipFile:HasPassword

Returns true if the ZIP file is encrypted.

Type	Standard
Parameters	None
Return value	<IValue> True if encrypted

Only available for Harbour

1.8.6.2.6 TUnzipFile:Run

Unzip the files from the zip file.

Type	Standard
Parameters	None
Return value	<ISuccess> .T. if the files were unzipped successfully

1.8.6.3 TUnzipFile:Events

Name
OnChangeDisk *
OnChangeFile ^
OnProgress

* Only available for xHarbour

^ Only available for Harbour

1.8.6.3.1 TUnzipFile:OnChangeDisk

Unzip the zip file in multiple disks.

Parameters	<oSender> Reference to the object that triggers the event <nDisk> Disk number where the files are being read
Return value	NIL

Only available for xHarbour

1.8.6.3.2 TUnZipFile:OnChangeFile

Event that is triggered when a new file is going to be decompressed.

Parameters	<oSender> Reference to the object that triggers the event <cFile> File name
Return value	NIL

Only available for Harbour

1.8.6.3.3 TUnZipFile:OnProgress

Depending on the xBase compiler its behaviour is different:

For xHarbour:

Indicates the partial progress according with the number of files that are being compressing.

Parameters	<oSender> Reference to the object that triggers the event <cFile> File name that is being compressing
-------------------	--

	<nPos> File order number
Return value	NIL

For Harbour:

Indicates the partial progress according with the bytes decompressed.

Parameters	<oSender> Reference to the object that triggers the event <nBytes> Bytes compressed so far <nTotal> Bytes to compress
Return value	NIL

1.8.7 TWiaScan

Class to manage escaners using Windows WIA (Windows image acquisition) API that is available on any Microsoft operating system since Windows XP.

Hierarchy	TComponent descendant
File	Internal

1.8.7.1 TWiaScan:Properties

■ Read only ■ Assignable ■ Design assignable ■ Run-time assignable

Scope	Name	Type	Initial value
■	cDeviceId	String	Nil
■	nImageMode	Numeric	imColor
■	nResolution	Numeric	150

1.8.7.1.1 TWiaScan:cDeviceId

Device identifier to use for the scan. It should be one of the values returned by the method Devices.

Scope	Assignable
Type	String
Initial value	NIL

1.8.7.1.2 TWiaScan:nImageMode

Image type to scan.

Scope	Assignable
Type	Numeric
Initial value	imColor
Possible values	imColor, imGray, imBW

1.8.7.1.3 TWiaScan:nResolution

Pixel resolution to use in the scan.

Scope	Assignable
Type	Numeric
Initial value	150

1.8.7.2 TWiaScan:Methods

■ Constructor ■ Standard

Type	Name
■	DeviceInfo
■	Devices
■	ImageSize
■	Scan

1.8.7.2.1 TWiaScan:DeviceInfo

Returns information about a specific device and mode.

Type	Standard
Parameters	[<cDeviceId>]: Device id. Optional [<nMode>]: Image mode: imColor, imGray, imBW. Optional
Return value	<aInfo> Array with supported resolutions. For example: { 150, 300, 600, 1200, 2400, 4800 }

1.8.7.2.2 TWiaScan:Devices

Returns an array with all the available devices.

Type	Standard
Parameters	Ninguno
Return value	<aDevices>

1.8.7.2.3 TWiaScan:ImageSize

Returns the image size for a specific scanning options.

Type	Standard
Parameters	[<cDeviceId>]: Device id. Optional [<nMode>]: Image mode: imColor, imGray, imBW. Optional. [<nResolution>]: Resolution. Optional [<aRect>]: Coordinates array {xPos, yPos, xExtent, yExtent}. Optional.
Return value	<nImageSize> Image size

1.8.7.2.4 TWiaScan:Scan

Starts scan process.

Type	Standard
Parameters	[<cDeviceId>]: Device id. Optional [<IPreview>]: Preview scan. Optional. By default false. [<nMode>]: Image mode: imColor, imGray, imBW. Optional. [<nResolution>]: Resolution. Optional [<aRect>]: Coordinates array {xPos, yPos, xExtent, yExtent}. Optional.
Return value	<cImageData> Scanned data

1.8.7.3 TWiaScan:Events

Name
OnScan

1.8.7.3.1 TWiaScan:OnScan

Event produce during the scan.

Parameters:	<oSender>: A reference to the object that launches de event <nPercent>: Percentage done
Return value	Nil

1.8.8 TZipFile

This class encapsulates the manage of zip files.

Description:

The TFileName class allows to manipulate Zip files.

The TZipFile descends from the **TZipArchive** class that establishes the common functionality for the TZipFile and TUnZipFile classes. The **TZipArchive** class is not documented due all this functionality is internal and it has not utility for the programmer.

Note for xHarbour:

To use this class you must include the **HBZIP.LIB** and **ZLIB.LIB** libraries into your project.

Note for Harbour:

To use this class you must include the **HBMZIP.LIB**, **MINIZIP.LIB** and **ZLIB.LIB** libraries into your project.

Hierarchy	Inherits from nobody
File name	\source\ZipArchive.prg

1.8.8.1 TZipFile:Properties

■ read Only
 ■ Assignable
 ■ Design assignable
 ■ Run-time assignable

Scope	Name	Type	Initial value
■	aFileMask	Array	"*.*"
■	aFiles	Array	
■	cComment	Character	""
■	cFileName	Character	""
■	clastError ^	Character	""
■	cPassword	Character	""

■	nBufferSize ^	Numeric	256
■	nCompressionLevel	Numeric	pkDEFLATE
■	nCount *	Numeric	
■	nExtractBufferSize *	Numeric	32
■	nLastError ^	Numeric	0
■	nReadBufferSize *	Numeric	16
■	nSpandMode *	Numeric	pkNOSPAN
■	nWriteBufferSize *	Numeric	64
■	lIncludePath	Logic	.T.
■	lMasksComPath	Logic	.F.
■	lOverwrite	Logic	.T.
■	lRecurse	Logic	.F.
■	lUnicode	Logic	.F.

* Only available for xHarbour

^ Only available for Harbour

1.8.8.1.1 TZipFile:aFileMask

Mask from the files to be included. It can be defined a string like "*.cpp" or a mask list and path files {"*.cpp", "include*.h"}. Both, relative and absolute paths are supported. This property is incompatible with the method AddFile.

Scope	Assignable
Type	Array
Initial value	"*.*"

1.8.8.1.2 TZipFile:aFiles

File list to compress. This list is clear out on any change of aFileMask.

Scope	readOnly
Type	Array
Initial value	{}

1.8.8.1.3 TZipFile:cComment

ZIP comment.

Scope	Assignable
Type	Character
Initial value	""

1.8.8.1.4 TZipFile:cFileName

ZIP filename to create.

Scope	Assignable
Type	Character
Initial value	""

1.8.8.1.5 TZipFile:cLastError

Description of last error produced.

Scope	Read only
Type	Character
Initial value	""

1.8.8.1.6 TZipFile:cPassword

Zip file password.

Scope	Assignable
Type	Character
Initial value	""

1.8.8.1.7 TZipFile:nBufferSize

Compress buffer size in Kb.

Scope	Assignable
Type	Numeric
Initial value	256

Only available for Harbour

1.8.8.1.8 TZipFile:nCompressionLevel

Data compression level in the zip file.

Scope	Assignable
--------------	------------

Type	Numeric
Initial value	pkDEFLATE
Possible values	pkSTORE, pk1, pk2, pk3, pk4, pkDEFLATE, pk6, pk7, pk8, pkBEST

1.8.8.1.9 TZipFile:nCount

Number of files in the zip file.

Scope	readOnly
Type	Numeric
Initial value	0

Only available for xHarbour

In Harbour you can use the expression `Len(oZipFile:aFiles)` to retrieve the number of files to compress.

1.8.8.1.10 TZipFile:nExtractBuferSize

Extract buffer size in Kb.

Scope	Assignable
Type	Numeric
Initial value	32

Only available for xHarbour

See the property `nBufferSize` on Harbour.

1.8.8.1.11 TZipFile:nLastError

Last error number.

Scope	readOnly
Type	Numeric
Initial value	0

Only available for Harbour

1.8.8.1.12 TZipFile:nReadBufferSize

Read buffer size in Kb.

Scope	Assignable
Type	Numeric
Initial value	16

Only available for xHarbour

See the property nBufferSize on Harbour.

1.8.8.1.13 TZipFile:nSpanMode

Indicates if the zip file will be created in a rigid support media (pkNOSPAN) or not (pkSPAN).

Scope	Assignable
Type	Numeric
Initial value	pkNOSPAN
Possible values	pkNOSPAN, pkSPAN

Only available for xHarbour

1.8.8.1.14 TZipFile:nWriteBufferSize

Write buffer size in Kb.

Scope	Assignable
Type	Numeric
Initial value	64

Only available for xHarbour

See the property nBufferSize on Harbour.

1.8.8.1.15 TZipFile:lIncludePath

Includes the file path.

Scope	Assignable
Type	Logic

Initial value	.T.
----------------------	-----

1.8.8.1.16 TZipFile:IMasksComPath

If true relatives paths are created with the masks common path.

Scope	Assignable
Type	Logic
Initial value	.F.

1.8.8.1.17 TZipFile:IOverwrite

Overwrites the file, if exists.

Scope	Assignable
Type	Logic
Initial value	.T.

1.8.8.1.18 TZipFile:IRrecurse

Includes the files from the lower folders.

Scope	Assignable
Type	Logic
Initial value	.F.

1.8.8.1.19 TZipFile:IUnicode

If true Unicode codification will be used for file names.

Scope	Assignable
Type	Logic
Initial value	.F.

1.8.8.2 TZipFile:Methods

■ Constructor ■ Standard

Type	Name
■	AddFile
■	FileSize ^
■	Run Execute

* Only available for xHarbour

^ Only available for Harbour

1.8.8.2.1 TZipFile:AddFile

Adds a file to be compressed. This method is incompatible with the property aFileMask.

Type	Standard
Parameters	<cFile> File to add
Return value	NIL

1.8.8.2.2 TZipFile:FileSize

Returns the number of bytes of all the files to compress.

Type	Standard
Parameters	None
Return value	<nTotal> Total bytes

1.8.8.2.3 TZipFile:Run

Creates the zip file.

Type	Standard
Parameters	None
Return value	<ISuccess> .T. if the file has been created successfully

1.8.8.3 TZipFile:Events

Name
OnChangeDisk *
OnChangeFile ^

OnProgress

*** Only available for xHarbour****^ Only available for Harbour**

1.8.8.3.1 TZipFile:OnChangeDisk

Allows to save the zip file in multiple disks.

Parameters	<oSender> Reference to the object that triggers the event <nDisk> Disk number order
Return value	NIL

Only available for xHarbour

1.8.8.3.2 TZipFile:OnChangeFile

Event that is triggered when a new file is going to be compressed.

Parameters	<oSender> Reference to the object that triggers the event <cFile> File name
Return value	NIL

Only available for Harbour

1.8.8.3.3 TZipFile:OnProgress

Depending on the xBase compiler its behaviour is different:

For xHarbour:

Indicates the partial progress according with the number of files that are being compressing.

Parameters	<oSender> Reference to the object that triggers the event <cFile> File name that is being compressing <nPos> File order number
Return value	NIL

For Harbour:

Indicates the partial progress according with the bytes compressed.

Parameters	<oSender> Reference to the object that triggers the event <nBytes> Bytes compressed so far <nTotal> Bytes to compress
Return value	NIL

1.9 Printing

1.9.1 Report Generator TReport

Xailer includes a simple report generator **TReport** that allows to make any kind of column reports. This report system does not include a visual designer because has been designed to create quickly reports using commands and some lines of code directly. The advantages of this system are: very easy to use, fast report creation and flexibility to adapt to possible future changes.

The screenshot shows a window titled "Xailer Report" with a toolbar containing navigation and printing icons. Below the toolbar is a zoom control set to "Ancho de página". The main content area displays a table titled "My first Xailer report" with the following data:

Rango	First	Last	Salary
157	CLIFFORD	ABELSON	134.400
250	BRUCE	ABELSON	73.800
257	DOMINIC	ACKER	21.900
269	DANIEL	ADKINS	111.500
21	KANDASAMY	AGUAYO	150.800
251	KANDASAMY	AGUAYO	85.600
329	LEN	AMARAL	125.300
16	BARRY	AMMANN	70.500
106	DAVID	BAILEY	2.300
197	TED	BAILEY	48.900
399	JAMES	BANSICO	75.100
363	DEAN	BANTA	15.700
422	DEAN	BANTA	107.800
288	CHUCK	BARKALOW	128.900
425	LILLIAN	BARKALOW	18.900
226	BRYAN	BARNES	28.800
301	HERB	BASSETT	56.700
463	VINCENT	BECKERMAN	2.500
487	JACKIE	BECKERMAN	133.300
396	LOREN	BEILHARZ	125.300
263	BARBARA	BLANCHI	8.000
693	GRAHAM	BLOCKEY	150.400

The report format that can be created with the TReport is the classic report with multiple columns and the option to add multiple group levels to get subtotals in the desired column. It is based in the basic dBase REPORT FORM command with a lot of new features and adapted to be used in a graphic environment. All the reports designed with the TReport have the following zones:

- Header
- Title

- Columns
- Groups
- Footer

The header, the title and the footer are the horizontal elements in the reports and they have one or multiple lines from the left to the right in the report width. Every element corresponds to an object from the TRptLine class and they are at the same time properties from the TReport class: the header is the oHeader property, the title is the oTitle property and the footer is the oFooter property.

The columns are objects from the TRptColumn class that are stored in the aColumns properties from the TReport object. The column objects store the title information and the information to be displayed in the column. In both cases it is possible to display several lines. It allows to display, for example, one record from the database in the same column, but in more than one line.

The groups are objects from the TRptGroup class and are stored in the aGroups property from the TReport object. This is the class responsible to make all the group and total process.

Even when they are several classes involved, the report creation is very simple, through the commands included by Xailer. In order to use them, you must include the **REPORT.CH** header file in all the modules where you will create reports.

By default, the report generator is prepared to manage the active DBF table, with any FOR active clause, and to the end of the table (!Eof()). However this default behavior can be easily changed using the bSkip, bWhile and bFor properties.

The TReport includes an event system that allows to adapt the reports to any possibility, because allows to the programmer to take total report control and modify or adapt it in the way he needs. For example, to add a logo in the first page is as simple as capture the event OnStartPage, check that is the first page with the TReport:nPage property and in the case of the first page, paint the logo using directly the TCanvas object from the current printout: TReport:oDevice:oCanvas.

1.9.1.1 Commands

To create a report:

```
REPORT <oReport> [ OF <oParent> ] ;
[ TITLE <bTitle, ...> [ ALIGN <nTitleAlign> ] ] ;
[ HEADER <bHeader, ...> [ ALIGN <nHeaderAlign> ] ] ;
[ FOOTER <bFooter, ...> [ ALIGN <nFooterAlign> ] ] ;
[ FONT <oFont, ...> ] ;
[ PEN <oPen> ] ;
[ SUMMARY ] ;
[ PREVIEW [ MODAL ] ] ;
[ TOPRINTER <oPrinter> ] ;
[ DATASET <oDataset> ] ;
[ JOBNAME <cJobName> ] ;
[ TO PDF <cPdf> ] ;
[ KEEP ]
```

OF	Allows to establish the parent form from the report. Optional.
TITLE	Report title. You may indicate more than one line separating every expression by commas. Check

	that it is processed as a 'code-block' and allows to write any expression that can be evaluated as a string. Optional.
HEADER	Report header. You may indicate more than one line separating every expression by commas. Check that it is processed as a 'code-block' and allows to write any expression that can be evaluated as a string. Optional.
FOOTER	Report footer. You may indicate more than one line separating every expression by commas. Check that it is processed as a 'code-block' and allows to write any expression that can be evaluated as a string. Optional.
FONT	Establishes the TFont object types to be used in the report. You may indicate more than one object separating every of them by commas. It is the programmer responsibility to destroy the TFont objects when the report has been finished. Optional.
PEN	Establishes the TPen object to be used to paint any type of line in the report. It is the programmer responsibility to destroy the TPen object when the report has been finished. Optional.
SUMMARY	This clause establishes that only prints the information in groups. Optional.
PREVIEW MODAL	This clause establishes that the report will be previewed instead to be printed. If you include the additional MODAL clause, the preview windows will be modal and the application will stop until the user closes the preview windows. Optional.
TO PRINTER	Allows to indicate exactly the TPrinter object where the printout will be done. Optional.
DATASET	Allows to indicate the TDatasetr object to use for printing. For further information consult the property oDataset. Optional.
JOBNAME	Printer job descriptive name. Optional.
ALIGN	Allows to indicate the alignment that may be taLEFT, taRIGHT or taCENTER. Optional.
TO PDF	Allows direct printing to PDF indicating the file name in <cPdf>. This clause is incompatible with the KEEP clause.
KEEP	This clause allows reports to be accumulated and not printed directly. When the report has the KEEP clause, the report is saved in memory and is not printed. You can nest as many reports as you like using the KEEP clause. As soon as you generate a report that does not have such a clause, all saved reports will be printed, along with this last one. This clause is incompatible with the TO PDF clause.

To create a column:

```

COLUMN [ <oRptCol> ] OF <oReport>;
[ TITLE <bTitle, ...> ] ;
[ DATA <bData, ...> ] ;
[ PICTURE <cPicture, ...> ] ;
[ FONT <nFont> ] ;
[ CHARSIZE <nSize> ] ;
[ TOTAL ] ;
[ ALIGN <nAlign> ] ;
[ COLOR TEXT <nClrText> ] ;
[ COLOR PANE <nClrPane> ] ;
[ COLOR TITLE TEXT <nClrText> ] ;
[ COLOR TITLE PANE <nClrPane> ]

```

OF	Column's TReport object.
TITLE	Column title. You may indicate more than one line separating every expression by commas. Check that it is processed as a 'code-block' and allows to write any expression that can be evaluated as a string. Optional.
DATA	Column data. You may indicate more than one line separating every expression by commas. It is processed as a 'code-block' so it is allowed to type any valid expression. If you indicate the PICTURE clause it will show the value using that mask. Optional.
PICTURE	Mask to be used in the column's data. You may indicate more than one mask, separating them by commas. Every mask will be used in every line of the DATA clause. Optional.
FONT	Font number to be used according to the FONT clause definition in the REPORT command or in the oReport:aFonts property. Optional.
CHARSIZE	Allows to establish the column width based in the standard width of the uppercase "B" character. Optional.
TOTAL	Indicates if the column will be totalized. In the case that this has more than one line, all the lines involved will be totalized. To avoid that one specific line will be totalized, transform it to a character type. Optional.
ALIGN	Allows to indicate the alignment that may be taLEFT, taRIGHT or taCENTER. Optional.
COLOR TEXT	Color for the column's text. Optional.
COLOR PANE	Column's background color. Optional.

To create groups:

```

GROUP [ <oRptGrp> ] [ OF <oReport> ] ;
[ ON <bGroup> ] ;

```

```
[ HEADER <bHeader> ] ;
[ FOOTER <bFooter> ] ;
[ FONT <nFont> ] ;
[ EJECT ]
```

OF	TReport object for the group.
HEADER	Text to show in the group header. Check that it is processed as a 'code-block' and allows to write any expression that can be evaluated as a string. Optional.
FOOTER	Text to show in the group footer. Check that it is processed as a 'code-block' and allows to write any expression that can be evaluated as a string. Optional.
FONT	Font number to be used according to the FONT definition for the REPORT command or the oReport:aFonts property. Optional.
EJECT	Indicates if there is a page break every time that the group is broken. Optional.

Note: If more than one group is created, the group break expression needs to be from general to specific. For example:

```
GROUP oGroup1 OF oReport ON Cli->Country
GROUP oGroup2 OF oReport ON Cli->Country + Cli->PostalCode
```

To activate the report:

```
RUN REPORT <oReport> ;
[ FOR <bFor> ] ;
[ WHILE <bWhile> ] ;
[ ONINIT <OnInit> ] ;
[ ONEND <OnEnd> ] ;
[ ONPOSTEND <OnPostEnd> ] ;
[ ONSTARTPAGE <OnStartPage> ] ;
[ ONENDPAGE <OnEndPage> ] ;
[ ONPOSTPAGE <OnPostPage> ] ;
[ ONSTARTGROUP <OnStartGroup> ] ;
[ ONENDGROUP <OnEndGroup> ] ;
[ ONPOSTGROUP <OnPostGroup> ] ;
[ ONSTARTLINE <OnStartLine> ] ;
[ ONENDLINE <OnEndLine> ] ;
[ ONCHANGE <OnChange> ] ;
[ ONCHANGED <OnChanged>
```

FOR	FOR expression to evaluate the report. It is { .T. } by default. This expression is evaluated in every report iteration. If it returns FALSE the current iteration will not be included in the report and a new iteration is processed. For more information see the TReport:bFor property. Optional.
------------	---

WHILE	WHILE expression to evaluate the report. It is <code>{ !Eof() }</code> by default. This expression is evaluated in every report iteration. If it returns FALSE the report is stopped immediately. For more information see the <code>TReport:bWhile</code> property. Optional.
ON INIT	Event that is produced when starts the printout. For more information see the <code>TReport:OnInit</code> event. Optional.
ON END	Event that is produced when finishes the printout, but before to print the global totals. For more information see the <code>TReport:OnEnd</code> event. Optional.
ON POST END	Event that is produced when finishes the printout, but after the global total printout. For more information see the <code>TReport:OnPostEnd</code> event. Optional.
ON STARTPAGE	Event that is produced when starts the page printout. For more information see the <code>TReport:OnStartPage</code> event. Optional.
ON ENDPAGE	Event that is produced when the page printout is finished, but before the page totals are printed. For more information see the <code>TReport:OnEndPage</code> event. Optional.
ON POSTPAGE	Event that is produced when the page printout is finished, but after the page totals are printed. For more information see the <code>TReport:OnPostPage</code> event. Optional.
ON STARTGROUP	Event that is produced when the group printout starts. Besides the <code>oSender</code> parameter it receives a second <code>oGroup</code> parameter. For more information see the <code>TReport:OnStartGroup</code> event. Optional.
ON ENDGROUP	Event that is produced when the group printout is finished, before to print the totals. Besides the <code>oSender</code> parameter, it receives a second parameter <code>oGroup</code> . For more information see the <code>TReport:OnEndGroup</code> event. Optional.
ON POSTGROUP	Event that is produced when the group printout is finished, but after the totals are printed. Besides the <code>oSender</code> parameter, it receives a second parameter <code>oGroup</code> . For more information see the <code>TReport:OnPostGroup</code> event. Optional.
ON STARTLINE	Event that is produced every time that the printout will start a new line. Besides the <code>oSender</code> parameter it receives a second <code>nHeight</code> parameter with the height specified in pixels to move forward. For more information see the <code>TReport:OnStartLine</code> event. Optional.
ON ENDLINE	Event that is produced every time that a line printout is finished. Besides the <code>oSender</code> parameter, it receives a second <code>nHeight</code> parameter

	with the height specified in pixels to move forward. For more information see the TReport:OnEndLine event. Optional.
ON CHANGE	Event that is produced before to start a new iteration in the report loop. Optional.
ON CHANGED	Event that is produced after is produced the interaction the in report loop. Optional.

Example:

```
#include "XAILER.CH"
#include "REPORT.CH"

REPORT oReport ;
  TITLE "My first Xailer report" ;
  HEADER DtoC( Date() ) + " " + Time() ;
  FOOTER "Page: " + lTrim( Str( oReport:nPage ) ) ALIGN taCENTER ;
  PREVIEW

COLUMN OF oReport ;
  TITLE "Recno" ;
  DATA Recno() ;
  ALIGN taRIGHT

COLUMN OF oReport ;
  TITLE "First" ;
  DATA Customer->First

COLUMN OF oReport ;
  TITLE "Last" ;
  DATA Customer->Last

COLUMN OF oReport ;
  TITLE "Street" ;
  DATA Customer->Street

COLUMN OF oReport ;
  TITLE "City" ;
  DATA Customer->City

COLUMN OF oReport ;
  TITLE "Salary" ;
  DATA Customer->Salary ;
  ALIGN taRIGHT ;
  PICTURE "@E 99,999,999" ;
  TOTAL

GROUP oGroup OF oReport ;
  ON Upper( Left( Customer->First, 1 ) ) ;
  FOOTER "Letter " + oGroup:cValue + " :"
```

RUN REPORT oReport

1.9.1.2 Classes

The report generator is formed by the following classes:

TReport	Main class responsible for the printout and it encapsulates to the other existing classes
TRptColumn	Specific class to manage the report columns
TRptLine	Specific class to manage the horizontal report information, like the title, the header o the footer
TRptGroup	Specific class to manage the report groups

For more information, see also the Get started chapter.

1.9.1.2.1 TReport

Main class from the simple report generator. It is the container for the rest of classes involved in the report generation.

Hierarchy	Inherits from TComponent
File name	\source\Report.prg

1.9.1.2.1.1 TReport:Properties

■ read Only ■ Assignable ■ Design assignable ■ Assignable en run-time

Scope	Name	Type	Initial value
■	aColumns	Array	{}
■	aFonts	Array	{}
■	aGroups	Array	{}
■	bFor	code-block	{ .T. }
■	bSkip	code-block	NIL
■	bWhile	code-block	{ !EOF() }
■	cGrandTotal	Character	""
■	cJobName	Character	"Xailer Report"
■	cPageTotal	Character	""
■	cPdfFilename	Character	""
■	cXLSDData	Character	""
■	IAutoLand	Logic	.T.
■	ICancel	Logic	.F.
■	IPageTotal	Logic	.T.
■	IPreview	Logic	.F.
■	IPreviewModal	Logic	.F.
■	IKeepWithNext	Logic	.F.

■	lSummary	Logic	.F.
■	lTotal	Logic	.F.
■	lXLSNumeric	Logic	.T.
■	nBottomMargin	Numeric	100
■	nClrLined	Numeric	clWhite
■	nCounter	Numeric	0
■	nGridStyle	Numeric	rgNONE
■	nGroupLine	Numeric	rISINGLE
■	nLeftMargin	Numeric	100
■	nLineHeight	Numeric	0
■	nPage	Numeric	0
■	nRow	Numeric	0
■	nSeparator	Numeric	0
■	nTitleDnLine	Numeric	rIDOUBLE
■	nTitleUpLine	Numeric	rIDOUBLE
■	nTopMargin	Numeric	100
■	nTotalLine	Numeric	rIDOUBLE
■	oDataset	Object	NIL
■	oDevice	Object	NIL
■	oFooter	Object	NIL
■	oHeader	Object	NIL
■	oParent	Object	NIL
■	oPen	Object	NIL
■	oTitle	Object	NIL

Array of TRptColumn objects with all the report columns.

Scope:	Assignable
Type:	Array
Initial value:	{}

Array of TFont objects used in the report.

Scope:	Assignable before Create()
Type:	Array
Initial value:	{}

It is important to assign this property before to create the TReport object due the provided fonts are cloned for the printer device in the Create constructor.

It is the programmer responsibility to destroy the TFont objects when the printout has been finalized.

Array of TRptGroup objects with all the report groups.

Scope:	Assignable
Type:	Array
Initial value:	{}

FOR expression to filter the report. It can be any expression in a code block form that returns a logic value. This expression will be evaluated in every loop interaction and in the case that the indicated expression returns false, that iteration will be disregarded and will not be printed.

Scope:	Assignable
Type:	Code-block
Initial value:	{ .T. }

Expression to go through all the report iterations. It can be any expression in a code block form. If this property is not defined, the default process will be to do a SKIP on the active DBF table.

Scope:	Assignable
Type:	Code-block
Initial value:	NIL

WHILE expression to filter the report. It will print when an specific situation is produced. It can be any expression in a code block form that returns a logic value. This expression will be evaluated in every loop interaction and in the case that the indicated expression returns false, that interaction will be disregarded and the report will stop immediately. This property is initialized by default with a simple !EOF() on the active DBF table.

Scope:	Assignable
Type:	Code-block
Initial value:	{ !EOF() }

Text to be shown in the report's final totals line, in the first column.

Scope:	Assignable
Type:	Character

Initial value:	""
-----------------------	----

Text to be shown in the Printer Job Management and in the title in the preview report window.

Scope:	Assignable
Type:	Character
Initial value:	"Xailer report"

Text to show in the total line of every page, in the first column.

Scope:	Assignable
Type:	Character
Initial value:	""

Name of the PDF file where the report will be generated.

Scope:	Assignable
Type:	Character
Initial value:	""

Buffer with the information to send to to Excel when using the method ToExcel.

Scope:	Assignable
Type:	Character
Initial value:	""

When sending the report to Excel, this property holds the data string with the information that is sent to it. Its format has this particularities:

- Use a carriage return and line feed (CRLF) to change the row on the spreadsheet
- Use the Tab character (Chr(9)) to change the cell on the spreadsheet
- The first spreadsheet column must not be used for personal purposes because its used by group headers and footer that are in boldface

Through the TReport object events, except OnStartLine, OnEndLine, OnStartPage, OnEndPage y OnPostPage you can manage this property adding more information, being aware of its format

already described.

If it is .T. the report will automatically adjust to Landscape in the case that it does not fit in Portrait mode.

Scope:	Assignable
Type:	Logic
Initial value:	.T.

If it is .T. indicates that the report has been stopped pressing the Cancel button.

Scope:	readOnly
Type:	Logic
Initial value:	.F.

If it is .T. the report will show a subtotal line in every page. Otherwise, it will show only the total line at the end of the report.

Scope:	Assignable
Type:	Logic
Initial value:	.T.

If it is .T. the report will be shown in a preview window.

Scope:	Assignable
Type:	Logic
Initial value:	.F.

If it is .T. and the IPreview is .T., the report will be shown in the preview windows in modal type.

Scope:	Assignable
Type:	Logic
Initial value:	.F.

If .T. the present document will be accumulated with next document and will not be shown when completed. In order to preview the full document with all the generated reports, you must set this property to .F. on the last report or call the method PreviewNow.

Scope:	Assignable
Type:	Logic
Initial value:	.F.

If it is .T. the report will show only the groups subtotals.

Scope:	Assignable
Type:	Logic
Initial value:	.F.

If it is .T. any report column has a total clause.

Scope:	read Only
Type:	Logic
Initial value:	.F.

If true, when sending the report to Excel, all the numeric columns will be ported as numbers and not as strings.

Ámbito:	Asignable
Tipo:	Lógico
Valor inicial:	.T.

This property is only present for compatibility since previous versions of Xailer made the export to Excel as character strings.

Indicates the Bottom margin specified in millimeters * 10 (100 == 1 centimeter).

Scope:	Assignable
Type:	Numeric
Initial value:	100

Background color to simulate the alternate or 'accounting' printout.

Scope:	Assignable
Type:	Numeric
Initial value:	clWhite

See the Appendix to check the colors available

Indicates the counter for the number of iterations produced by the report.

Scope:	read Only
Type:	Numeric
Initial value:	0

Grid style to use in the report.

Scope:	Assignable
Type:	Numeric
Possible values:	rgNONE, rgVERTICAL, rgHORIZONTAL, rgBOTH
Initial value:	rgNONE

Line style to be shown in the report in every group subtotal.

Scope:	Assignable
Type:	Numeric
Possible values:	rINONE, rISINGLE, rIDOUBLE
Initial value:	rISINGLE

Indicates the Left margin specified in millimeters * 10 (100 == 1 centimeter).

Scope:	Assignable
---------------	------------

Type:	Numeric
Initial value:	100

Indicates the standard height in pixels. This value is automatically calculated by the report based in the first defined Font, but it can be changed.

Scope:	Assignable
Type:	Numeric
Initial value:	0

Returns the current report page.

Scope:	Assignable
Type:	Numeric
Initial value:	0

Indicates the current report row, in pixels. If you need to modify the current row position is better to use the NewLine and BackLine methods.

Scope:	Assignable
Type:	Numeric
Initial value:	0

Indicates the standard width based in the 'B' character from the first defined font. This value is used to separate the different report columns. It is calculated automatically but it can be changed.

Scope:	Assignable
Type:	Numeric
Initial value:	0

Line style to be shown below the title line.

Scope:	Assignable
Type:	Numeric

Possible values:	rINONE, rISINGLE, rIDOUBLE
Initial value:	rIDOUBLE

Line style to be shown above the tile line.

Scope:	Assignable
Type:	Numeric
Possible values:	rINONE, rISINGLE, rIDOUBLE
Initial value:	rIDOUBLE

Indicates the Top margin specified in millimeters * 10 (100 == 1 centimeter).

Scope:	Assignable
Type:	Numeric
Initial value:	100

Indicates the line style to be shown by the report in the totals line.

Scope:	Assignable
Type:	Numeric
Possible values:	rINONE, rISINGLE, rIDOUBLE
Initial value:	rIDOUBLE

TDataset object to use for the printing process.

Scope:	Assignable
Type:	Object
Initial value:	NIL

By default the TReport object uses the active DBF alias to perform the report using the properties bFor and bSkip. When this property is set, the report uses it instead of the active Alias.

The report object uses automatically the dataset, positioning the dataset at its beginning and restoring its state when the report is completely finished.

TPrinter object when the printout will be done.

Scope:	Assignable
Type:	Object
Initial value:	NIL

TRptLine object with all the report footer information.

Scope:	Assignable
Type:	Object
Initial value:	NIL

TRptLine object with all the report header information.

Scope:	Assignable
Type:	Object
Initial value:	NIL

Proprietary TForm object type from the report. Optional.

Scope:	Assignable
Type:	Object
Initial value:	NIL

TPen object type used to print the lines in the report.

Scope:	Assignable
Type:	Object
Initial value:	NIL

The oPen object is automatically destroyed by the printer TCanvas object from in the moment that the printout is finished.

TRptLine object with all the report title information.

Scope:	Assignable
Type:	Object
Initial value:	NIL

1.9.1.2.1.2 TReport:Methods

■ Constructor ■ Standard ■ Only after Create()

Type	Name
■	AddColumn
■	AddGroup
■	Adjust
■	AreGroups
■	BackLine
■	DelColumn
■	DelGroup
■	EndLine
■	Finished
■	LinesLeft
■	NeedNewPage
■	NewLine
■	NewPage
■	PrintAtCol
■	PreviewNow
■	Run
■	StartLine
■	ToExcel

Adds a new column to the report.

Type	Only after Create
Parameters	<oCol> Object TRptColumn
Return value	NIL

Adds a new group to the report.

Type	Only after Create
Parameters	<oCol> Object TRptGroup
Return value	NIL

Method to adjust automatically the report coordinates and dimensions. It is called automatically before to start printing.

Type	Only after Create
Parameters	None
Return value	NIL

Returns .T. if there is a group defined in the report.

Type	Only after Create
Parameters	None
Return value	<IExist> .T. if there are groups in the report

Go backward a number of lines the current report printing row.

Type	Only after Create
Parameters	<nLines> Number of lines to go back. Default: 1.
Return value	NIL

Deletes an specific report column.

Type	Only after Create
Parameters	<nCol> Column position number to eliminate.
Return	NIL

value	
--------------	--

Deletes an specific report group.

Type	Only after Create
Parameters	<nGroup> Group position number to eliminate.
Return value	NIL

Method that is called by the report generator every time that it finishes to print a line.

If you will print lines manually in any event, it is important that you execute the StartLine method before to print it. When you finish to print the whole line, call the EndLine method.

Type	Only after Create
Parameters	<nHeight> Line height specified in pixels. Default: the nLineHeight property value.
Return value	NIL

Method that indicates if the report has finished to print

Type	Only after Create
Parameters	None
Return value	<IFinished> .T. if the report has finished.

Method that indicates the remaining number of lines to print in the current page.

Type	Only after Create
Parameters	None
Return value	<nLines> Number of lines

Returns .T. if the report needs a new page to print the next line.

Type	Only after Create
Parameters	None
Return value	<IYes> .T. if the report needs a new page to print the next line

Move the printer cursor one line forward.

Type	Only after Create
Parameters	<nHeight> Line height in pixels. Default: the nLineHeight property value
Return value	NIL

Move the printer cursor one page forward.

Type	Only after Create
Parameters	None
Return value	NIL

Prints text on the same position of a specific column. Namely, uses the column Y coordinate and the actual X coordinate (nRow) of the report.

Type	Only after Create
Parameters	<nCol>: Column number to use as reference <cText>: Text to print [<nAlign>]: Text alignment. By default the property nAlignment of the column [<nFont>]: Font number to use. By default the property nDataFont of the column [<nColor>]: Text color. By default nClrText of the column
Return	NIL

value

Forces the preview of all the generated reports. It must be used in conjunction with IPrintWithNext.

Type	Only after Create
Parameters	None
Return value	NIL

Executes the report.

Type	Only after Create
Parameters	None
Return value	<ISuccess> .T. if the process has been processed correctly

Method that is called by the report every time that it prints a line.

If you will print lines manually in any event, it is important that you execute the StartLine method before to print it. When you finish to print the whole line, call the EndLine method.

Type	Only after Create
Parameters	<nHeight> Line height in pixels. Default: the nLineHeight property value
Return value	NIL

Sends the report to the Excel. The Excel application should be installed on the PC.

This method generates the report in the same way than method Run(). If you have already run the report, normally it should be necessary to reset the database navigation pointer with oDataset:Gotop() or equivalent.

This method does not trigger any event.

Type	Only after Create
Parameters	<[cFile]> Filename of the XLS file to create. If empty the

	report will be send to Excel and then it will be shown.
Return value	<ISuccess> True if success

1.9.1.2.1.3 TReport:Events

Name
OnChange
OnChanged
OnEnd
OnEndGroup
OnEndLine
OnEndPage
OnInit
OnPostEnd
OnPostGroup
OnPostPage
OnPreChange
OnStartGroup
OnStartLine
OnStartPage

Event that is triggered before a new iteration.

Parameters	<oSender>: Object that triggers the event.
Return value:	NIL

Event that is triggered after the report finishes to print an iteration.

Parameters	<oSender>: Object that triggers the event.
Return value:	NIL

Event that is triggered when the report finishes the printout, but before to print the general totals.

Parameters	<oSender>:
-------------------	------------

:	Object that triggers the event.
Return value:	NIL

Event that is triggered at the end of the group, but before to print the group subtotals.

Parameters	<oSender>:
:	Object that triggers the event.
	<oGroup>
	Object TRptGroup
Return value:	NIL

Event that is triggered when the report finishes to print a line.

Parameters	<oSender>:
:	Object that triggers the event.
	<nHeight>
	Line height in pixels
Return value:	NIL

Event that is triggered when the printout starts.

Parameters	<oSender>:
:	Object that triggers the event.
Return value:	NIL

Event that is triggered when the report finishes the printout of one page, but before to print its totals.

Parameters	<oSender>:
:	Object that triggers the event.
Return value:	NIL

Event that is triggered when the report finishes to print, but after the general totals are printed.

Parameters	<oSender> :
:	Object that triggers the event.
Return value:	NIL

Event that is triggered at the end of a group, but after their subtotals are printed.

Parameters	<oSender> :
:	Object that triggers the event.
	<oGroup> Object TRptGroup
Return value:	NIL

Event that is triggered when the report finishes the printout, but after their totals are printed.

Parameters	<oSender> :
:	Object that triggers the event.
Return value:	NIL

Event that is triggered before a new iteration and before the columns data is printed.

Parameters	<oSender> :
:	Object that triggers the event.
Return value:	NIL

Event that is triggered when the report starts a new group.

Parameters	<oSender> :
:	Object that triggers the event.
	<oGroup> Object TRptGroup
Return value:	NIL

Event that is triggered when the report starts printing a line.

Parameters	<oSender> : Object that triggers the event.
:	<nHeight> Line height in pixels
Return value:	NIL

Event that is triggered when starts a page printout.

Parameters	<oSender> : Object that triggers the event.
Return value:	NIL

1.9.1.2.2 TRptColumn

This class represents all the column objects included in the report. All the report column collection is saved also with the aColumns property from it container report.

Hierarchy	Inherits from TComponent
File name	\source\Report.prg

1.9.1.2.2.1 TRptColumn:Properties

■ read Only ■ Assignable ■ Design assignable ■ Assignable en run-time

Scope	Name	Type	Initial value
■	aData	Array	{}
■	aPicture	Array	{}
■	aTitle	Array	{}
■	cTotalPict	Character	""
■	ITotal	Logic	.F.
■	nAlignment	Numeric	taLEFT
■	nCharSize	Numeric	0
■	nClrPane	Numeric	clWhite
■	nClrText	Numeric	clBlack
■	nDataFont	Numeric	1
■	nTitleAlignment	Numeric	-1
■	nTitleClrPane	Numeric	clWhite
■	nTitleClrText	Numeric	clBlack
■	nTitleFont	Numeric	1
■	nTitleHeight	Numeric	0

■	nTotal	Numeric	0
■	nTotalFont	Numeric	1
■	nWidth	Numeric	0
■	oReport	Object	NIL

Codeblock array with all the print information.

Scope:	Design assignable
Type:	Array
Initial value:	{}

Every element has the information to be print per line in every interaction. One iteration can produce more than one line in the report, however the following expressions are valid:

```
oCol:aData := { {|| Cliente->Name }, {|| Cliente->LastName } }
```

In this example every report iteration will print two lines: the client name in the first line and the last name in the second line.

The codeblock expressions can be any basic type: Character, Number, Logic or Date. If you specify the aPicture you can establish a mask or template for a better presentation.

Masks array to be used in every line to be printed.

Scope:	Design assignable
Type:	Array
Initial value:	{}

Every array element corresponds with the mask to be used in every aData array element. For example:

```
oCol:aPicture := { "9999999999", "@!" }
```

For more information, see also the aData property.

Codeblocks array with all the column headers or titles.

Scope:	Design assignable
Type:	Array
Initial value:	{}

Every array element corresponds with one title line. The column title can use more than one line. For example:

```
oCol:aTitle:= { {|| "Name", {|| "Last Name" } }
```

In this example the title has two lines: "Name" in the first line and "Last Name" in the second line.

The codeblocks expressions must be character type.

Mask to use on the columns totals. If it is not specified, it will use the fist picture indicated in the aPicture array.

Scope:	Design assignable
Type:	Character
Initial value:	""

.T. if the column will have totals.

Scope:	Design assignable
Type:	Logic
Initial value:	.F.

In the case that the column has more than one line, it will totalize all the numeric type expressions in aData. If you need that any of the aData expressions will not be part of the figure, convert them to character type in the same codeblock expression.

Indicates the column alignment.

Scope:	Design assignable
Type:	Numeric
Possible values:	taLEFT, taCENTER, taRIGHT
Initial value:	taLEFT

Indicates the column width in characters. The 'B' character is used as base in the calculation.

Scope:	Design assignable
Type:	Numeric
Initial value:	0

Indicates the column background color.

Scope:	Design assignable
Type:	Numeric
Initial value:	clWhite

See the Appendix to check the color available

Indicates the column text color.

Scope:	Design assignable
Type:	Numeric
Initial value:	clBlack

See the Appendix to check the color available

Indicates the font number to use for the data printout. This number corresponds to the aFonts array from its oReport container. By default it will use the first font defined in its TReport container object.

Scope:	Design assignable
Type:	Numeric
Initial value:	1

If you want to change the font for specific column values you can use the OnPrintData event.

Indicates the column title alignment.

Scope:	Design assignable
Type:	Numeric
Possible values:	-1, taLEFT, taCENTER, taRIGHT
Initial value:	-1

A -1 value uses the alignment of its property nAlignment

Indicates the column title background color.

Scope:	Design assignable
Type:	Numeric
Initial value:	clWhite

See the Appendix to check the color available

Indicates the column title text color.

Scope:	Design assignable
Type:	Numeric
Initial value:	clBlack

See the Appendix to check the color available

Font number to be used to print the column titles. This number corresponds to the aFonts array from its oReport container. By default it will use the first font defined in its TReport container object.

Scope:	Design assignable
Type:	Numeric
Initial value:	1

If you want to change the font for specific column values you can use the OnPrintTitle event.

Indicates the title height in pixels.

Scope:	read Only
Type:	Numeric
Initial value:	0

Indicates the column total value.

Scope:	read Only
Type:	Numeric
Initial value:	0

Font number to be used to print the column totals. This number corresponds to the `aFonts` array from its `oReport` container. By default it will use the first font defined in its `TReport` container object.

Scope:	Design assignable
Type:	Numeric
Initial value:	1

If you want to change the font for an specific column you can use the `OnPrintTotal` event.

Indicates the column width specified in pixels.

Scope:	read Only
Type:	Numeric
Initial value:	0

Indicates the column `TReport` proprietary object.

Scope:	Design assignable
Type:	Object
Initial value:	NIL

1.9.1.2.2.2 TRptColumn:Events

Name
OnAddToTotal
OnPrintData
OnPrintTitle
OnPrintTotal

Event that is triggered when the column is totalized. With this event you can cancel or even modify the column totalization on specific circumstances.

Parameters	<oSender> :
:	Object that triggers the event.
	<nLine> :

	Line to print. It corresponds to the aData array element <Value>: Current column and line expression
Return value:	If it returns a value distinct of NIL, that value will be used for column totalization

Event that is triggered when the column information is printed.

Parameters	<p><oSender>: Object that triggers the event.</p> <p><nLine>: Line to print. It corresponds to the aData array element</p> <p><@cText>: Text to be printed. It is passed by reference and however it allows to modify its value</p> <p><@nFont>: Font number to use from the oReport:aFonts array. It is passed by reference and however it allows to modify its value.</p> <p><@nColor>: Color to be used for the text. It is passed by reference and however it allows to modify its value.</p> <p><Value>: Current column expression. In contrast with cText this is the original data without any picture transformation and can be any basic type.</p>
Return value:	NIL

Event that is triggered when the report prints the column title.

Parameters	<p><oSender>: Object that triggers the event.</p> <p><nLine>: Line to print. It corresponds to the aTitle array element.</p> <p><@cText>: Text to be printed. It is passed by reference and however it allows to modify its value</p> <p><@nFont>: Font number to use from the oReport:aFonts array. It is passed by reference and however it allows to modify its value</p> <p><@nColor>: Color to be used for the text. It is passed by reference and however it allows to modify its value</p>
-------------------	--

Return value: NIL

Event that is triggered when the column totals are printed.

Parameters

- <oSender>:**
Object that triggers the event.
- <@cText>:**
Text to print. It is passed by reference and however it allows to modify its value
- <@nFont>:**
Font number to be used from the oReport:aFonts array. It is passed by reference and however it allows to modify its value
- <@nColor>:**
Color to use for the text. It is passed by reference and however it allows to modify its value

Return value: NIL

1.9.1.2.3 TRptLine

This class represents all the horizontal objects from the report. It corresponds to the oTitle, oHeader and oFooter properties from its container report.

Hierarchy Inherits from TComponent
File name \source\Report.prg

1.9.1.2.3.1 TRptLine:Properties

■ read Only ■ Assignable ■ Design assignable ■ Assignable en run-time

Scope	Name	Type	Initial value
■	aAlignment	Array	{}
■	aFonts	Array	{}
■	aLine	Array	{}
■	nColor	Character	clBlack
■	nHeight	Numeric	0
■	nWidth	Numeric	0
■	oReport	Object	NIL

Numeric array with all the line alignments to be printed.

Scope: Design assignable

Type:	Array
Initial value:	{}

Every array element corresponds to the alignment of every aLine array element. However the following expressions are valid:

```
oLine:aAlignment := { taLEFT, taRIGHT }
```

The possible values for every array element are: taLEFT, taRIGHT and taCENTER.

Important note:

You may have a double alignment on the text, left and right on the same time using the string '|' (double pipe) as a separator for both alignments on the text to print. On that case, the value of this property is useless.

Numeric array with the font number to be used to print.

Scope:	Design assignable
Type:	Array
Initial value:	{}

Every array element corresponds to the font number in the oReport:aFonts to be used on every aLine array element.

Codeblock array with all the lines to print.

Scope:	Design assignable
Type:	Array
Initial value:	{}

Every array element corresponds to one print line. For example:

```
oLine:aLines := { {|| "Costumers list", {|| "Year: " + Str( Year( Date() ), 4 ) } }
```

The codeblock expression must be character type. To change the alignment of every line use the aAlignment property.

Indicates the text color.

Scope:	Design assignable
Type:	Numeric
Initial value:	clBlack

To change the text color for an specific line, use the OnPrintLine event.

See the appendix to check the colors available

Indicates the height in pixels.

Scope:	read Only
Type:	Numeric
Initial value:	0

Indicates the width in pixels.

Scope:	read Only
Type:	Numeric
Initial value:	0

Proprietary TReport object.

Scope:	Design assignable
Type:	Object
Initial value:	NIL

1.9.1.2.3.2 TRptLine:Events

Name
OnPrintLine

Event that is triggered when it is printed every line.

Parameters	<oSender>:
-------------------	------------

:	Object that triggers the event. <nLine>: Line to print. It corresponds to the aLine array. <@cText>: Text to print. It is passed by reference and however it allows to modify its value <@nFont>: Font number to be used from the oReport:aFonts array. It is passed by reference and however it allows to modify its value <@nColor>: Color to use for the text. It is passed by reference and however it allows to modify its value
Return value:	NIL

1.9.1.2.4 TRptGroup

This class represents all the report groups. They are contained in the oReport:aGroups property from its container report.

Hierarchy	Inherits from TComponent
File name	\\source\\Report.prg

1.9.1.2.4.1 TRptGroup:Properties

■ read Only
 ■ Assignable
 ■ Design assignable
 ■ Assignable en run-time

Scope	Name	Type	Initial value
■	aTotal	Array	{}
■	bFooter	Code-block	{ "Total..." }
■	bGroup	Code-block	{ "" }
■	bHeader	Code-block	{ "" }
■	cOldValue	Character	""
■	cValue	Character	""
■	lEject	Logic	.F.
■	nClrText	Numeric	clBlack
■	nCounter	Numeric	0
■	nFont	Numeric	1
■	nFooterHeight	Numeric	0
■	nHeaderHeight	Numeric	0
■	nOrder	Numeric	0
■	oReport	Object	NIL

Numeric array with the totals of all the report columns. This array is initialized to zero every time that there is a new group.

Scope:	read Only
Type:	Array
Initial value:	{}

Codeblock with the text to be shown as group footer..

Scope:	Design assignable
Type:	Code-block
Initial value:	{ "Total..." }

The codeblock expression must be character type.

Codeblock with the expression that will identify the group condition. When the value of this expression changes, it will create a group break.

Scope:	Design assignable
Type:	Code-block
Initial value:	{ "" }

The codeblock expression must be character type. For example:

```
oGroup:bGroup := { || Customers->Zipcode }
```

Codeblock with the text to be shown as group header.

Scope:	Design assignable
Type:	Code-block
Initial value:	{ "" }

The codeblock expression must be character type.

Indicates the value for the last but one evaluated expression for the bGroup property.

Scope:	read Only
---------------	-----------

Type:	Character
Initial value:	""

Indicates the value for the last evaluated expression for the bGroup property.

Scope:	read Only
Type:	Character
Initial value:	""

If it is .T. it will eject the page if there is a group break.

Scope:	Design assignable
Type:	Logic
Initial value:	.F.

Text color.

Scope:	Design assignable
Type:	Numeric
Initial value:	clBlack

To change the colors you can also use the OnPrintHeader, OnPrintFooter or OnPrintTotal events.

See the Appendix to check the colors available

Counter with the number of elements totalized in the group. This value is initialized to zero every time that there is a group break.

Scope:	read Only
Type:	Array
Initial value:	{}

Numeric value with the font number to be used based in the oReport:aFonts array.

Scope:	Design assignable
Type:	Numeric
Initial value:	1

To change the font you can use also the OnPrintHeader, OnPrintFooter or OnPrintTotal events.

Indicates the footer height specified in pixels.

Scope:	read Only
Type:	Numeric
Initial value:	0

Indicates the header height specified in pixels.

Scope:	read Only
Type:	Numeric
Initial value:	0

Group creation order in its container report.

Scope:	read Only
Type:	Numeric
Initial value:	0

Proprietary TReport object.

Scope:	Design assignable
Type:	Object
Initial value:	NIL

1.9.1.2.4.2 TRptGroup:Events

Name
OnPrintFooter
OnPrintHeader
OnPrintTotal

Event that is triggered when the print footer is printed.

Parameters	<oSender> : Object that triggers the event. <@cText> : Text to print. It is passed by reference and however it allows to modify its value <@nFont> : Font number to use from the oReport:aFonts array. It is passed by reference and however it allows to modify its value <@nColor> : Color to use for the text. It is passed by reference and however it allows to modify its value
Return value:	NIL

Event that is triggered when the header group is printed.

Parameters	<oSender> : Object that triggers the event. <@cText> : Text to print. It is passed by reference and however it allows to modify its value <@nFont> : Font number to use from the oReport:aFonts array. It is passed by reference and however it allows to modify its value <@nColor> : Color to use for the text. It is passed by reference and however it allows to modify its value
Return value:	NIL

Event that is triggered when the group total is printed.

Parameters	<oSender> : Object that triggers the event.
-------------------	---

	<@cText> : Text to print. It is passed by reference and however it allows to modify its value
	<@nFont> : Font number to use from the oReport:aFonts array. It is passed by reference and however it allows to modify its value
	<@nColor> : Color to use for the text. It is passed by reference and however it allows to modify its value
Return value:	NIL

1.9.2 FastReport for Xailer

Introduction

FastReport (FR) is a visual report generator that provides to your applications the ability to generate multiple types of reports quickly and efficiently.

This report generator provides all the tools necessary to develop complex and showy reports.

FastReport for Xailer is based on FastReport from Fast Report Inc. <http://www.fast-report.com/> that also has versions for Delphi (VCL and FireMonkey), .NET and Mono

The first adaptation to [x]Harbour compiler was done by Sergey Spirin from ParitetSoft. Unfortunately the product was completely discontinued. Xailer, knowing the great dependency of many Xailer users to this great product decided to create its own adaptation taking advantage of Xailer's visual interface, but trying to maintain the compatibility with the old ParitetSoft product, in order to minimize as much as possible the adaptations of old reports.

The visual interface of FastReport for Xailer is really easy to use, and there are no complicated installations processes. You just need to add a single DLL (frx.dll) to the application folder.

FastReport for Xailer integrates natively with the programming language, being able to use variables, methods or functions defined in our programs and supports any data format that our application may use. You can use from DBF tables with NTX or CDX indexes, Advantage Database Server (ADT tables), data from database engines via ODBC or ADO or MySql, MariaDB or SQLite datasets or even simple data arrays.

FastReport can do all kind of reports, from columnar reports, labels, graphics, bar codes, calendars, crosstabs reports, invoices, large tickets, photo and rich text printing.

Any kind of printing on any printer can be done with FastReport.

FastReport features

- No complicated installation, only a royalty-free DLL is needed on your application directory.
- Powerful visual designer, multilingual and easy to use.

- Completely WYSIWYG
- Band design oriented.
- Supports any type of components: lines, boxes, texts, images, OLE, graphics, bar codes, rich text, etcetera.
- Export functionality with multiple formats available (PDF, HTML, RTF, CSV, TXT, XLS, XML, JPG, BMP, TIFF, GIF, ODS, ODT, eMail), from the preview window or directly on your source code without showing anything on the screen.
- Support for any kind of printer, including dot matrix printing.
- Internal script editor with syntax highlighting.
- Support of various scripting languages: PascalScript, C++Script, Javascript and BasicScript.
- Scripting debugger.
- Runtime access of any Xailer variable, function or work area.
- Great speed on report generation and export.
- Third party libraries to expand its possibilities. For example QR bidimensional codes.
- Report preview with search and edit capabilities.
- Text rotation.
- Paragraphs justification.
- Support of simple HTML tags on text objects (Font,color,b,i,u,sub,sup).
- Multiple types of fonts, sizes, styles and colors.
 - **Note:** You may use any font on the developing process, but that font must exist on the end user PC, in case not, a font substitute will be used.
- Word wrapping.
- Use of functional URLs (in preview mode).
- Zoom on the visual designer.
- Guides and rules.
- Configurable grid in mm, inches or pixels.
- Wizard for most common reports.
- Object copy to Windows clipboard.
- Unlimited Undo/Redo.
- The reports can contain their own forms, for example, to ask for some parameters. You may have all the forms you need. FastReport supports the common objects used on any form designer: buttons, text edit, check boxes, etcetera.
- Los informes pueden contener sus propios formularios, por ejemplo para pedir parámetros antes de preparar un informe. Se pueden tener tantos diálogos como sea necesario. FastReport usa el mismo diseñador para crear y mantener los diálogos y soporta los controles de Windows más comúnmente usados como botones, gets, check boxes, etc.

- Storage of all the reports in XML format. Compressed (GZip compatible) also supported..
- Bidirectional text output for specific languages.
- Support of third party components through OLE, like Excel files for example.
- Any kind of page dimensions and even with unlimited length.
- Includes a full functional demo with the only limitation that only prints the first five pages.

FastReport components

FastReport has four main components:

- A dynamic link library FRX.DLL that includes the FastReport engine, including all the languages that may use. The library can be copied to any directory pointed out by your PATH environment variable. But normally the best place is to copy it to the same directory of your application.
- TFastReport main class to manage the reports. This module is included on Xailer's libraries so you will not need to worry about it.
- TFrDataset and its descendents: TFrXailerDataset, TFrArrayDataset and TFrDbfDataset are classes which its purpose it to connect the the different Xailer source datasets with FastReport itself.
- TFrOptions and its descendants: TFrEngineOptions, TFrPreviewOptions, TFrPrintOptions and TFrReportOptions are classes that maybe used to get or set all the properties of any report.

Using the Report Generator

FastReport is not intended to be used as normal application, is not an EXE file, is a report generator that is embedded into your program.

To use it you just need to include the following code in your application:

```
oFr:=TFastReport():New()  
oFr:DesignReport()  
oFr:End()
```

When running your application and executing this code, the report designer will be shown with all its features so you will be able to design your own report and then save it with any name, for example "Sales report.Fr3" which is a XML file that we can open with any text editor in case necessary.

In order to run a already created report to a preview window or directly to the printer, a few lines more are necessary:

```
oFr:=TFastReport():New()  
oFr:LoadFromFile("Sales report.fr3")  
oFr:ShowReport()  
oFr:End()
```

Really simple.

Using the Report Designer

The use of FastReport is really simple and there is a lot of documentation available. For example:

http://www.fast-report.com/en/download/public_files/142/

http://www.fast-report.com/en/download/public_files/146/

http://www.fast-report.com/en/download/public_files/148/

For further information consult the following link:

<http://www.fast-report.com>

Acknowledgements

We would like to thanks Bingen Ugaldebere and René Flores for its great job on the Spanish documentation of FastReport and for allow us to use part of their work on the Spanish version of this manual.

Likewise, be grateful with Sergey Spirin (RIP) for its work on making FastReport accesible to a lot of Harbour users. We collect its witness and expect to fulfill the expectations of all the [x]Harbour community.

1.9.2.1 Sergey Spirin compatibility

FastReport for Xailer has made a great effort in order to be the maximum compatible with the Sergey Spirin product, but still there are minor differences in subjects we believe are not crucial, like:

- Specific BLOB methods since our version does not need any special treatment for them
- SetEventHandler method since we use a better approach with specific events
- Resync method since is absolutely not necessary on our system
- SetTxtDataset method since we believe is useless
- Multiple reports mode
- Manual build reports

Our base class is TFastReport but you can also instantiate your objects from **frReportManager()** which adds to **TFastReport** some compatibility methods with Sergey product. Which are:

- Constructor **New**(cFilename)
- Constructor **Init**()
- Method **DestroyFR**()
- Method **SetWorkArea**(cAlias, nArea, IOem, aRangeParams)
- Method **SetFieldAliases**(cAlias, cFieldAliases)
- Method **SetUserDataset**(cAlias, cFields, bGotop, bSkipPlus1, bSkipMinus1, bCheckEof, bGetValue)

Important note: You must call the **Create** method() after calling the **New**() constructor.

If in the future and due user request more compatibility is requested surely it will be added only to the **frReportManager()** class.

1.9.2.2 Register

In order to stop seeing the annoying watermark and the five pages limitation you must register you version of FastReport for [x]Harbour & Xailer on every application you may create. This can be done by setting the properties TFastReport:cFRLicense and TFasReport:cXALicense every time you create a TFastReport object. This can be a heavy task that can be resolved with just a few lines of code on any of your source code modules. Just include the following code:

```
#include "Xailer.ch"

#pragma TEXTHIDDEN(1)
CLASS TFastReport FROM XFastReport

    VAR cFRLicense INIT "1234-5678-9012-3456" // your FR License goes
here
    VAR cXALicense INIT "XAFR-00000000-01" // your XA License goes
here
END CLASS
#pragma TEXTHIDDEN(0)
```

The TEXTHIDDEN commands permit to obfuscate your license on the executable. Without those lines your license can be easily accessed. We highly appreciate your cooperation on this issue.

1.9.2.3 TFastReport

Xailer main report class.

The reports created by FastReport are saved in XML files with the extension FR3 and include not only the report information but important context information on where the report should be printed or shown. For this reason, you will see that from Xailer's IDE many of the report properties are not available and that is because the persistence of those properties are directly handled by FastReport and there is no point to do the same job by Xailer's IDE. In any case any report property can be set on your own source code.

Certain operations can only be made after the report is loaded. For example: since is possible to create a new variable category, if you have not set a specific report to load, when you load it the new category will be lost and superseded with the categories of the own report. For that reason is importante to set the report filename before modifying any of its properties.

Hierarchy TComponent descendant
File \source\FastReport.prg

1.9.2.3.1 TFastReport:Properties

■ Read only ■ Assignable ■ Design assignable ■ Runtime assignable

Scope	Name	Type	Initial Value
■	aDatasets	Array	{}

■	cDllPath	String	""
■	cFileName	String	""
■	cFRLicense	String	""
■	cLanguage	Cáacter	""
■	cLastError	String	""
■	cReslco	String	""
■	cResName	String	""
■	cTitle	String	""
■	cXALicense	String	""
■	lAbortOnErrors	Logical	.T.
■	lDelFRVars	Logical	.T.
■	lSynchroData	Logical	.F.
■	nLastError	Numeric	0
■	oEngineOptions	Object	TFrEngineOptions
■	oPreviewOptions	Object	TFrPreviewOptions
■	oPrintOptions	Object	TFrPrintOptions
■	oReportOptions	Object	TFrReportOptions

1.9.2.3.1.1 TFastReport:aDatasets

Array of TFrDataset objects connected to the report.

Scope	Read only
Type	Array
Initial Value	{}

1.9.2.3.1.2 TFastReport:cDllPath

Frx.dll path. It must be set before the call to method **Create()** or on the OnBeforeCreate event.

Scope	Assignable
Type	String
Initial Value	""

1.9.2.3.1.3 TFastReport:cFileName

Name of the FR3 or FP3 file that contains the report.

Scope	Assignable
Type	String
Initial Value	""

The set of this property provokes the deletion of the cResName property.

1.9.2.3.1.4 TFastReport:cFRLicense

FastReport license provided by FastReport Inc.

Scope	Assignable
Type	String
Initial Value	""
See also	cXALicense

This property must be assigned on every report you create in order to deactivate the evaluation mode. To avoid the work of introducing the **cFRLicense** and the **cXALicense** license, as convenient is to overload its **T** class introducing its values. For example:

```
CLASS TFastReport FROM XFastReport

    PROPERTY cFRLicense INIT "1234-5678-9012-3456"
    PROPERTY cXALicense INIT "XAFR-00000000-01"

END CLASS
```

1.9.2.3.1.5 TFastReport:cLanguage

Language to be used on the report designer and preview window.

Scope	Assignable
Type	String
Initial Value	""
Possible values	English, Brazil, French, German, Italian, Portuguese, Russian, Spanish, Greek

All languages literals are contained in FastReport for Xailer DLL. If your language is not available on this list, you can use the LoadLangFile method to load from an external file your own language.

1.9.2.3.1.6 TFastReport:cLastError

Last error produced.

Scope	Read only
Type	String

Initial Value	""
----------------------	----

1.9.2.3.1.7 TFastReport:cReslco

Resource name that holds the report image (ICO).

Scope	Assignable
Type	String
Initial Value	""

1.9.2.3.1.8 TFastReport:cResName

Resource name for FR3 or FP3 report information.

Scope	Assignable
Type	String
Initial Value	""

The set of this property provokes the deletion of the cFileName property.

1.9.2.3.1.9 TFastReport:cTitle

Report title.

Scope	Assignable
Type	String
Initial Value	""

1.9.2.3.1.10 TFastReport:cXALicense

FastReport for Xailer license provided by Xailer.

Scope	Assignable
Type	String
Initial Value	""
See also	cFRLicense

This property must be assigned on every report you create in order to deactivate the evaluation mode. To avoid the work of introducing the **cXALicense** and the cFRLicense license, as convenient is to overload its **T** class introducing its values. For example:

```
CLASS TFastReport FROM XFastReport

    PROPERTY cFRLicense INIT "1234-5678-9012-3456"
    PROPERTY cXALicense INIT "XAFR-00000000-01"

END CLASS
```

1.9.2.3.1.11 TFastReport:IAbortOnErrors

If true, any error will provoke a runtime error that can be trapped with a `TRY . . CATCH`.

Scope	Assignable
Type	Logical
Initial Value	.T.

1.9.2.3.1.12 TFastReport:IDelFRVars

If true, reports variables included on the FR3 file will be deleted on load. When this property is set to true, the variables created by the user before the loading are first saved and restored after the load process, so they maintain intact.

Scope	Assignable
Type	Logical
Initial Value	.T.

1.9.2.3.1.13 TFastReport:ISynchroData

If true, the report generation also synchronizes the underlying datasets. This consumes more time to generate the report but it gives you more information if you are trapping any report event.

Scope	Assignable
Type	Logical
Initial Value	.T.

1.9.2.3.1.14 TFastReport:nLastError

Last ordinal error number produced.

Scope	Read only
--------------	-----------

Type	Logical
Initial Value	""
See also	cLastError

1.9.2.3.1.15 TFastReport:oEngineOptions

TFrEngineOptions object that holds all the properties of the report relating to the print engine.

Scope	Read only
Type	Object
Initial Value	TFrEngineOptions

1.9.2.3.1.16 TFastReport:oPreviewOptions

TFrPreviewOptions object that holds all the properties of the report relating to the preview engine.

Scope	Read only
Type	Object
Initial Value	TFrPreviewOptions

1.9.2.3.1.17 TFastReport:oPrintOptions

TFrPrintOptions object that holds all the properties of the report relating to the printing process.

Scope	Read only
Type	Object
Initial Value	TFrPreviewOptions

1.9.2.3.1.18 TFastReport:oReportOptions

TFrReportOptions object that holds all the properties of the report relating to the report itself.

Scope	Read only
Type	Object
Initial Value	TFrReportOptions

1.9.2.3.2 TFastReport:Methods

■ Constructor ■ Standard ■ Only after Create()

Type	Name
■	AddArray
■	AddCategory
■	AddDataSet
■	AddDbf
■	AddFunction
■	AddVariable
■	Calc
■	CategoriesList
■	Clear
■	ClearData
■	ClearMasterDetail
■	ClearVariables
■	DelDataset
■	DelError
■	DeleteCategory
■	DeleteVariable
■	DesignReport
■	DoExport
■	FieldValue
■	GetErrors
■	GetProperty
■	GetVariable
■	LoadFPFile
■	LoadFromFile
■	LoadFromResource
■	LoadFromString
■	LoadLangFile
■	LoadStyleSheetFromFile
■	LoadStyleSheetFromString
■	PrepareReport
■	PreviewClose
■	Print
■	RestoreVariables
■	SaveToFile
■	SaveToFPFile
■	SaveToString
■	SaveVariables
■	SetADOCConnectStr
■	SetFormatSettings
■	SetIcon
■	SetLicense
■	SetProperty
■	SetTitle

■	SetVariable
■	ShowPreparedReport
■	ShowReport
■	StopReport
■	VariablesList

1.9.2.3.2.1 TFastReport:AddArray

Adds a new dataset to the report using as data source an array.

Type	Standard
Parameters	<cName> Name that will be given to dataset on design time <aData> Datga array [<aFields>] Field names for every column on the array.
Return Value	TFrArrayDataset object

1.9.2.3.2.2 TFastReport:AddCategory

Adds a new variable category to the report.

Type	After loading the report
Parameters	<cName> Category name
Return Value	True if success

Note: Is important to create the category after the call to PrepareReport or in the OnAfterLoad event.

1.9.2.3.2.3 TFastReport:AddDataset

Adds a new dataset to the report using as data source a Xailer dataset.

Type	Standard
Parameters	<cName> Name that will be given to dataset on design time <oDataset> Xailer dataset
Return Value	TFrXailerDataset object

1.9.2.3.2.4 TFastReport:AddDbf

Adds a new dataset to the report using as data source one or more DBF files.

Type	Standard
Parameters	<cName> Name that will be given to dataset on design time [<aFields>] Field names for every column on the report
Return Value	TFrDbfDataset object

The **aFields** parameter is an array with all the field names you want to include. There are some important features:

- You can indicate the table source prefixing the field with its alias. For example: `Clients->Code`. If you do not indicate the alias, the active area when the report is loaded is used.
- Is possible to indicate all the fields of a table using the '*' character. For example: `Clients->*`.
- Is possible to modify the name of a field for design time using the clause 'AS'. For example: `Clients->id AS Code`.

In order to use more than one work area is necessary to set the relations between them using the xBase command `SET RELATION`.

1.9.2.3.2.5 TFastReport:AddFunction

Adds a new function to the report. Allows that any function application is accesible to the report engine.

Type	After loading the report
Parameters	<cPrototype> Function prototype in PascalScript <cCategory> Category where the function should appear on the designer <cDescription> Description to be shown on the report designer
Return Value	True if success

FastReport for Xailer includes in the [x]Harbour category and for compatibility reasons with the discontinued product *FastReport for [x]Harbour* some of its functions. However this is not really necessary since any function of your application can be accessed from the report generator with only this method. You only need to know how to perform the function prototyping in PascalScript.

For example:

```
oFR.AddFunction( "function NtoL(n: extended):String", "My category", ;
                "Converts a number to a string" )
```

With this simple line you will gain access to the `NtoL()` function on your report. It's that easy. To

use it in your reports can use even the wizard that provides the designer. This would be an example of its use:

```
[NtoL( <CUSTOMERS . "CustNo" > ) ]
```

Do not worry if a runtime error occurs within your own application when you call a function defined in the report. FastReport for Xailer is able to recover itself from it and even show the error description on the report.

1.9.2.3.2.6 TFastReport:AddVariable

Adds a new variable to the report.

Type	After loading the report
Parameters	<cName> Variable name on the report designer <Value> Variable value
Return Value	True if success

The variable will be created on the variables root tree unless you have create a category first with the method AddCategory.

1.9.2.3.2.7 TFastReport:Calc

Allows to calculate any PascalScript expression on the report environment. Returns the value of the expression.

Type	After loading the report
Parameters	<cExpression> Expression to evaluate
Return Value	Value

1.9.2.3.2.8 TFastReport:CategoriesList

Returns an array with all the available categories.

Type	After loading the report
Parameters	None
Return Value	altems

1.9.2.3.2.9 TFastReport:Clear

Clears the contents of the report and leave it in the same state as if he had not given any report.

Type	After loading the report
Parameters	None
Return Value	True if success

1.9.2.3.2.10 TFastReport:ClearData

Clears the contents of the report.

Type	After loading the report
Parameters	None
Return Value	NIL

1.9.2.3.2.11 TFastReport:ClearMasterDetail

Deletes a master-detail relation.

Type	Standard
Parameters	<cMasterName> Name of master dataset <cDetailName> Name of detail dataset
Return Value	True if success

1.9.2.3.2.12 TFastReport:ClearVariables

Clears all the variables of the report.

Type	After loading the report
Parameters	None
Return Value	True if success

1.9.2.3.2.13 TFastReport:DelDataset

Deletes a dataset from the report.

Type	Standard
Parameters	<oDataset> TFrDataset object to delete
Return Value	True if success

1.9.2.3.2.14 TFastReport:DelError

Deletes last error produced.

Type	Standard
Parameters	None
Return Value	True if success

1.9.2.3.2.15 TFastReport>DeleteCategory

Deletes a report category.

Type	After loading the report
Parameters	<cName> Category name
Return Value	True if success

1.9.2.3.2.16 TFastReport>DeleteVariable

Deletes a report variable.

Type	After loading the report
Parameters	<cName> Variable name
Return Value	True if success

1.9.2.3.2.17 TFastReport:DesignReport

Runs the report designer.

Type	After loading the report
Parameters	None
Return Value	True if success

Important Note: If the report has internal master-detail operations its advisable to call the PrepareReport method first in order to achieve consistent results on preview.

1.9.2.3.2.18 TFastReport:DoExport

Exports the report with the selected filter.

Type	After loading the report
Parameters	<cFilter> Filter to use. Possible values PDFEXPORT HTMLEXPORT RTFEXPORT CSVEXPORT XLSEXPORT DOTMATRIXEXPORT BMPEXPORT JPGEXPORT TIFFEXPORT GIFEXPORT SIMPLETEXTEEXPORT MAILEXPORT ODSEXPORT ODTEXPORT XMLEXPORT EMFEXPORT BIFEXPORT
Return Value	True if success

For further information consult FastReport documentation.

1.9.2.3.2.19 TFastReport:FieldValue

Any dataset field value.

Type	After loading the report and during report generation. It should be used inside of the any report generation events.
-------------	--

Parameters	<cTable> Table name <cField> Field name
Return Value	Value

1.9.2.3.2.20 TFastReport:GetErrors

Retrieves an array of strings with all the errors produced.

Type	After loading the report
Parameters	None
Return Value	<aErrors> Array of errors

1.9.2.3.2.21 TFastReport:GetProperty

Retrieves any report property. Note that you can specify a complex expression as '`REPORT.ReportOptions.cAuthor`'.

Type	After loading the report
Parameters	<cObjectName> Object name. If blank the <code>REPORT</code> object will be used. Possible values BMPEXport CSVEXPORT DESIGNER DotMatrixExport GIFEXPORT HTMLEXPORT HTMLExport JPEGEXPORT MailExport ODSEXport ODTEXport PDFEXPORT REPORT RTFEXPORT RTFEXPORT SimpleTextExport TIFFEXPORT XLSEXport XMLExport <cName> Variable name or expression
Return Value	Property value

1.9.2.3.2.22 TFastReport:GetVariable

Retrieves a report variable value.

Type	After loading the report
Parameters	<cName> Variable name
Return Value	Variable value

1.9.2.3.2.23 TFastReport:LoadFPFile

Loads a prepared report form file.

Type	Standard
Parameters	<cFilename> File name
Return Value	True if success

1.9.2.3.2.24 TFastReport:LoadFromFile

Loads a report from file.

Type	Standard
Parameters	<cFilename> File name
Return Value	True if success

1.9.2.3.2.25 TFastReport:LoadFromResource

Loads a report from resources.

Type	Standard
Parameters	<cResname> Resource name
Return Value	True if success

1.9.2.3.2.26 TFastReport:LoadFromString

Loads a report from a string.

Type	Standard
Parameters	<cString> String with all the report information
Return Value	True if success

1.9.2.3.2.27 TFastReport:LoadLangFile

Loads a XML file with the language to use on the report designer and preview window.

Type	Standard
Parameters	<cFilename> Filename
Return Value	True if success
See also	cLanguage

1.9.2.3.2.28 TFastReport:LoadStyleSheetFromFile

Loads a style sheet from file.

Type	After loading the report
Parameters	<cFilename> File name
Return Value	True if success

For further information consult FastReport documentation.

1.9.2.3.2.29 TFastReport:LoadStyleSheetFromString

Loads a style sheet from a string.

Type	After loading the report
Parameters	<cString> String with the information
Return Value	True if success

For further information consult FastReport documentation.

1.9.2.3.2.30 TFastReport:PrepareReport

Prepares report.

Type	After loading the report
Parameters	[<IKeepLastReport>] If true previous pages form old reports will not be deleted. By default .F.
Return Value	True if success

1.9.2.3.2.31 TFastReport:PreviewClose

Closes the original FastReport preview window.

Type	After loading the report
Parameters	None
Return Value	True if success

1.9.2.3.2.32 TFastReport:Print

Prints the report.

Type	After loading the report
Parameters	None
Return Value	True if success

1.9.2.3.2.33 TFastReport:RestoreVariables

Restores report variables saved with a previous call to SaveVariables.

Type	Standard
Parameters	None
Return Value	True if success

1.9.2.3.2.34 TFastReport:SaveToFile

Saves the report to a file.

Type	Standard
-------------	----------

Parameters	<cFilename> File name
Return Value	True if success

1.9.2.3.2.35 TFastReport:SaveToFPFile

Saves a report with all its data to a file.

Type	Standard
Parameters	<cFilename> File name
Return Value	True if success

1.9.2.3.2.36 TFastReport:SaveToString

Returns the report definition as a string.

Type	Standard
Parameters	<None>
Return Value	<cReportDefinition>

1.9.2.3.2.37 TFastReport:SaveVariables

Saves internally the actual report variables.

Type	Standard
Parameters	None
Return Value	True if success

See also: RestoreVariables.

1.9.2.3.2.38 TFastReport:SetADOCConnectStr

Sets the ADO connection string for the report.

Type	Standard
Parameters	<cADOCConnect> Connection string
Return Value	True if success

1.9.2.3.2.39 TFastReport:SetFormatSettings

Sets the default value format for most locale global variables .

Type	Standard
Parameters	<nSetting> Type of setting. <xValue> Newvalue
Return Value	True if success

nSetting is described in Frh.ch:

```
#define frxCurrencyString 1
#define frxCurrencyFormat 2
#define frxNegCurrFormat 3
#define frxThousandSeparator 4
#define frxDecimalSeparator 5
#define frxCurrencyDecimals 6
#define frxDateSeparator 7
#define frxShortDateFormat 8
#define frxLongDateFormat 9
#define frxTimeSeparator 10
#define frxTimeAMString 11
#define frxTimePMString 12
#define frxShortTimeFormat 13
#define frxLongTimeFormat 14
#define frxShortMonthNames 15
#define frxLongMonthNames 16
#define frxShortDayNames 17
#define frxLongDayNames 18
#define frxTwoDigitYearCenturyWindow 20
#define frxListSeparator 21
```

For **xValue** read follow description:

The initial values of these variables are fetched from the system registry using the GetLocaleInfo function in the Win32 API. The description of each variable specifies the LOCALE_XXXX constant used to fetch the initial value.

CurrencyString - Defines the currency symbol used in floating-point to decimal conversions. The initial value is fetched from LOCALE_SCURRENCY.

CurrencyFormat - Defines the currency symbol placement and separation used in floating-point to decimal conversions. Possible values are:

```
0 = '$1'
1 = '1$'
2 = '$ 1'
3 = '1 $'
```

The initial value is fetched from LOCALE_ICURRENCY.

NegCurrFormat - Defines the currency format for used in floating-point to decimal conversions of negative numbers. Possible values are:

0 = '(\$1)' 4 = '(1\$)' 8 = '-1 \$' 12 = '\$ -1'
1 = '-\$1' 5 = '-1\$' 9 = '-\$ 1' 13 = '1- \$'
2 = '\$-1' 6 = '1-\$' 10 = '1 \$-' 14 = '(\$ 1)'
3 = '\$1-' 7 = '1\$-' 11 = '\$ 1-' 15 = '(1 \$)'

The initial value is fetched from `LOCALE_INEGCURRE`.

ThousandSeparator - The character used to separate thousands in numbers with more than three digits to the left of the decimal separator. The initial value is fetched from `LOCALE_STHOUSAND`. A value of #0 indicates no thousand separator character should be output even if the format string specifies thousand separators.

DecimalSeparator - The character used to separate the integer part from the fractional part of a number. The initial value is fetched from `LOCALE_SDECIMAL`. `DecimalSeparator` must be a non-zero value.

CurrencyDecimals - The number of digits to the right of the decimal point in a currency amount. The initial value is fetched from `LOCALE_ICURRDIGITS`.

DateSeparator - The character used to separate the year, month, and day parts of a date value. The initial value is fetched from `LOCALE_SDATE`.

ShortDateFormat - The format string used to convert a date value to a short string suitable for editing. For a complete description of date and time format strings, refer to the documentation for the `FormatDate` function. The short date format should only use the date separator character and the m, mm, d, dd, yy, and yyyy format specifiers. The initial value is fetched from `LOCALE_SSHORTDATE`.

LongDateFormat - The format string used to convert a date value to a long string suitable for display but not for editing. For a complete description of date and time format strings, refer to the documentation for the `FormatDate` function. The initial value is fetched from `LOCALE_SLONGDATE`.

TimeSeparator - The character used to separate the hour, minute, and second parts of a time value. The initial value is fetched from `LOCALE_STIME`.

TimeAMString - The suffix string used for time values between 00:00 and 11:59 in 12-hour clock format. The initial value is fetched from `LOCALE_S1159`.

TimePMString - The suffix string used for time values between 12:00 and 23:59 in 12-hour clock format. The initial value is fetched from `LOCALE_S2359`.

ShortTimeFormat - The format string used to convert a time value to a short string with only hours and minutes. The default value is computed from `LOCALE_ITIME` and `LOCALE_ITLZERO`.

LongTimeFormat - The format string used to convert a time value to a long string with hours, minutes, and seconds. The default value is computed from `LOCALE_ITIME` and `LOCALE_ITLZERO`.

ShortMonthNames - Array of strings containing short month names. The mmm format specifier in a format string passed to `FormatDate` causes a short month name to be substituted. The default values are fetched from the `LOCALE_SABBREVMONTHNAME` system locale entries.

LongMonthNames - Array of strings containing long month names. The mmmm format specifier in a format string passed to `FormatDate` causes a long month name to be substituted. The default

values are fetched from the LOCALE_SMONTHNAME system locale entries.

ShortDayNames - Array of strings containing short day names. The ddd format specifier in a format string passed to FormatDate causes a short day name to be substituted. The default values are fetched from the LOCALE_SABBREVDAYNAME system locale entries.

LongDayNames - Array of strings containing long day names. The dddd format specifier in a format string passed to FormatDate causes a long day name to be substituted. The default values are fetched from the LOCALE_SDAYNAME system locale entries.

ListSeparator - The character used to separate items in a list. The initial value is fetched from LOCALE_SLIST.

TwoDigitYearCenturyWindow - Determines what century is added to two digit years when converting string dates to numeric dates. This value is subtracted from the current year before extracting the century. This can be used to extend the lifetime of existing applications that are inextricably tied to 2 digit year data entry. The best solution to Year 2000 (Y2k) issues is not to accept 2 digit years at all - require 4 digit years in data entry to eliminate century ambiguities.

1.9.2.3.2.40 TFastReport:SetIcon

Sets the resource icon from resources.

Type	Standard
Parameters	<cResname> Resource name. Must be BMP type
Return Value	True if success

1.9.2.3.2.41 TFastReport:SetLicense

Sets the license to deactivate the evaluation mode.

Type	Standard
Parameters	<cFRLicense> FastReport license <cXALicense> FastReport for Xailer license
Return Value	True if success
See also	cFRLicense, cXALicense

1.9.2.3.2.42 TFastReport:SetProperty

Sets a report property. Note that you can use complex expression like '`REPORT.ReportOptions.cAuthor = 'Peter'`'.

Type	After loading the report
Parameters	<p><cObjectName> Object name. If blank the <code>REPORT</code> object will be used. Possible values</p> <p>BMPExport CSVExport DESIGNER DotMatrixExport GIFExport HTMLEXPORT HTMLExport JPEGExport MailExport ODSExport ODTExport PDFExport REPORT RTFExport RTFEXPORT SimpleTextExport TIFExport XLSExport XMLExport</p> <p><cName> Variable name or expression</p> <p><xValue> Newvalue</p>
Return Value	NIL

1.9.2.3.2.43 TFastReport:SetTitle

Sets report title.

Type	Standard
Parameters	<p><cTitle> Report title</p>
Return Value	True if success

1.9.2.3.2.44 TFastReport:SetVariable

Sets the value of a report variable.

Type	After loading the report
Parameters	<cName> Variable name <xValor> Newvalue
Return Value	True if success

1.9.2.3.2.45 TFastReport:ShowPreparedReport

Shows report preview of already prepared report with the method PrepareReport.

Type	After loading the report
Parameters	[<IKeepLastReport>] If true previous preview pages form old reports will not be deleted. By default .F. [<oControl>] TFRPreview control where to show the report
Return Value	True if success

1.9.2.3.2.46 TFastReport:ShowReport

Shows report preview.

Type	After loading the report
Parameters	[<IKeepLastReport>] If true previous preview pages form old reports will not be deleted. By default .F. [<oControl>] TFRPreview control where to show the report
Return Value	True if success

1.9.2.3.2.47 TFastReport:StopReport

End the report immediately.

Type	After loading the report
Parameters	None
Return Value	NIL

1.9.2.3.2.48 TFastReport:VariablesList

Returns an array with all the variables of a defined category.

Type	After loading the report
Parameters	<cCategory> Category name
Return Value	altems

1.9.2.3.3 TFastReport:Events

Name
OnAfterLoad
OnAfterPrint
OnAfterPrintReport
OnBeforeCreate
OnBeforeConnect
OnBeforeLoad
OnBeforePrint
OnBeginDoc
OnBtnGenPdf
OnBtnPrint
OnBtnSendMail
OnClickObject
OnDbClickObject
OnDesignLoadReport
OnDesignSaveReport
OnEndDoc
OnEndPreview
OnGetValue
OnMouseOverObject
OnPageChanged
OnPreview
OnPrintPage
OnPrintReport
OnProgress
OnProgressStart
OnProgressStop

1.9.2.3.3.1 TFastRport:OnAfterLoad

Event that fires when the report is loaded. This is a good place to change the value of any property of the report.

Parameters	<oSender> : Object that triggers the event.
Return Value	NIL

1.9.2.3.3.2 TFastReport:OnAfterPrint

Event that fires every time you print any report object.

Parameters	<oSender> : TFastReport object that triggers the event <cObject> Object name
Return Value	NIL

1.9.2.3.3.3 TFastReport:OnAfterPrintReport

Event that is fired after the report is complete.

Parameters	<oSender> : Object that triggers the event.
Return Value	NIL

1.9.2.3.3.4 TFastReport:OnBeforeCreate

Event that fires when the control is created and before the DLL is loaded.

Parameters	<oSender> : TFastReport object that triggers the event
Return Value	NIL

1.9.2.3.3.5 TFastReport:OnBeforeConnect

Event that fires whenever a connection is made with a ADO connector.

Parameters	<oSender> :
-------------------	--------------------------

	TFastReport object that triggers the event <cObject> ADO object name
Return Value	NIL

1.9.2.3.3.6 TFastReport:OnBeforeLoad

Event that fires before the report is loaded but after all data and its relations are loaded.

Parameters	<oSender>: Object that triggers the event.
Return Value	NIL

1.9.2.3.3.7 TFastReport:OnBeforePrint

Event that fires before any object gets printed.

Parameters	<oSender>: Object that triggers the event. <cObject> Object name
Return Value	NIL

1.9.2.3.3.8 TFastReport:OnBeginDoc

Event that triggers when you start creating the report document.

Parameters	<oSender>: Object that triggers the event.
Return Value	NIL

1.9.2.3.3.9 TFastReport:OnBtnGenPdf

Event that triggers when you push the PDF button of the toolbar preview window.

Parameters	<oSender>: Object that triggers the event.
Return Value	If a value different from NIL is returned the default button operation will be canceled.

1.9.2.3.3.10 TFastReport:OnBtnPrint

Event that triggers when you push the 'Print' button of the toolbar preview window.

Parameters	<oSender> : Object that triggers the event.
Return Value	If a value different from NIL is returned the default button operation will be canceled.

1.9.2.3.3.11 TFastReport:OnBtnSendMail

Event that triggers when you push the 'Send mail' button of the toolbar preview window.

Parameters	<oSender> : Object that triggers the event.
Return Value	If a value different from NIL is returned the default button operation will be canceled.

1.9.2.3.3.12 TFastReport:OnClickObject

Event that is fired whenever you click on any object in the report.

Parameters	<oSender> : TFastReport object that triggers the event <cObject> Object name
Return Value	NIL

1.9.2.3.3.13 TFastReport:OnDbIclickObject

Event that is fired whenever you double-click on any object in the report.

Parameters	<oSender> : TFastReport object that triggers the event <cObject> Object name
Return Value	NIL

1.9.2.3.3.14 TFastReport:OnDesignLoadReport

Event that fires when loading the report from the designer.

Parameters	<oSender> : Object that triggers the event.
Return Value	true if success on the load operation

1.9.2.3.3.15 TFastReport:OnDesignSaveReport

Event that fires when saving the report from the designer.

Parameters	<oSender> : Object that triggers the event. <ISaveAs> True if option 'Save as...' is used
Return Value	true if success on the save operation

1.9.2.3.3.16 TFastReport:OnEndDoc

Event that fires when you finish creating the report document.

Parameters	<oSender> : Object that triggers the event.
Return Value	NIL

1.9.2.3.3.17 TFastReport:OnEndPreview

Event that fires when you close the report preview.

Parameters	<oSender> : Object that triggers the event.
Return Value	NIL

1.9.2.3.3.18 TFastReport:OnGetValue

Event that is fired before evaluating a variable. If this event is overloaded you can change the value of it with the return value.

Parameters	<oSender> :
-------------------	--------------------------

	TFastReport object that triggers the event <cVar> Variable name
Return Value	<xValue> new value

1.9.2.3.3.19 TFastReport:OnMouseOverObject

Event that is fired whenever the mouse is positioned over any object of the report.

Parameters	<oSender>: TFastReport that triggers the event <cObject> Object name
Return Value	NIL

1.9.2.3.3.20 TFastReport:OnPageChanged

Event that is fired whenever the preview window page changes.

Parameters	<oSender>: TFastReport that triggers the event <nPage> Page number
Return Value	NIL

1.9.2.3.3.21 TFastReport:OnPreview

Event that triggers at the beginning of the preview of the report.

Parameters	<oSender>: Object that triggers the event.
Return Value	NIL

1.9.2.3.3.22 TFastReport:OnPrintPage

Event that fires before each page of the report.

Parameters	<oSender>: TFastReport object that triggers the event <nCopyPage>
-------------------	---

	Page copy number
Return Value	NIL

1.9.2.3.3.23 TFastReport:OnPrintReport

Event that fires when the report begins printing.

Parameters	<oSender> : Object that triggers the event.
Return Value	NIL

1.9.2.3.3.24 TFastReport:OnProgress

Event that is fired to mark the progress of the report.

Parameters	<oSender> : TFastReport object that triggers the event <nProgressType> 0 Building 1 Exporting 2 Printing <nProgress> Page number in process
Return Value	NIL

1.9.2.3.3.25 TFastReport:OnProgressStart

Event that fires when the report starts a process.

Parameters	<oSender> : TFastReport object that triggers the event <nProgressType> 0 Building 1 Exporting 2 Printing <nProgress> Page number in process
Return Value	NIL

1.9.2.3.3.26 TFastReport:OnProgressStop

Event that fires when interrupting the process of creating the report.

Parameters	<oSender> : TFastReport object that triggers the event <nProgressType> 0 Building 1 Exporting 2 Printing <nProgress> Page number in process
Return Value	NIL

1.9.2.4 TFrDataset

Class to manage all the TFastReport data connectors.

This class is responsible for making the connection between FastReport reports and the data handled by our application. It supports three types of data sources and for each of them there is a specialized class that inherits from **TFrDataset**, which are:

- TFrXailerDataset
- TFrDbfDataset
- TFrArrayDataset

In no case should directly instantiate objects of this class.

Hierarchy TComponent descendant
File \source\FrDataset.prg

1.9.2.4.1 TFrDataset:Properties

■ Read only ■ Assignable ■ Design assignable ■ Runtime assignable

Scope	Name	Type	Initial Value
■	aRelationFields	Array	{}
■	cName	String	""
■	lLoadOnDemand	Logical	.F.
■	nLoaded	Numeric	0
■	nMaxRecsOnDesign	Numeric	100
■	oDsMaster	Object	NIL
■	oReport	Object	NIL

1.9.2.4.1.1 TFrDataset:aRelationFields

String array with all the master-detail relations to be used in conjuntio with the oDsMaster property.

Sets the fields to use for the master-detail relation. Each field must ocypy a single array element. In case the field names do not match in both datasets you will have to use the following expression to indicate it:

```
'cDetailField=cMasterField'
```

In order to achieve a special ordering on the detail dataset you can include any extra field (last array element) on the detail field side separated by ';'. For example:

```
'cDetailField1;cDetailField2=cMasterField1'
```

Scope	Standard
Type	Array
Initial Value	{}

1.9.2.4.1.2 TFrDataset:cName

Dataset name on the report designer. This property must be set with any value In case is not assigned a default value will be given.

This property must be sent before the report is loaded.

Scope	Design assignable
Type	String
Initial Value	""

1.9.2.4.1.3 TFrDataset:lLoadOnDemand

If true the report data will be loaded on demand. This specially useful on very large tables because Fast-Report normally can begin printing without having all the report data. In most cases should not be necessary its use.

Important note: You must not change the DBF table state during the printing process. Avoid to use tables that are also displayed within a browse at the same time.

This property is overloaded as read only for TFrArrayDataset class so its not usable.

Scope	Design assignable
Type	Logical
Initial Value	.F.

1.9.2.4.1.4 TFrDataset:nLoaded

Indicates the data load state of the dataset.

Scope	Read only
Type	Numeric
Initial Value	0
Possible values	0 Not loaded 1 Loaded for design 2 Complete load

1.9.2.4.1.5 TFrDataset:nMaxRecsOnDesign

Sets the maximum number of records that will be downloaded when using the report designer. This property allows you to limit the number of records to be loaded for design processes to not delay carrying large amount of data. A value of zero means to download the complete set of records.

Scope	Design assignable
Type	Numeric
Initial Value	100

1.9.2.4.1.6 TFrDataset:oDsMaster

TFrDataset inherited object that sets the 'Master' dataset on a master-detail relation.

Scope	Standard
Type	TFrDataset object
Initial Value	NIL
See also	aRelationFields

1.9.2.4.1.7 TFrDataset:oReport

TFastReport owner of the dataset. Must be assigned before calling its **Create** method.

Scope	Design assignable
Type	TFastReport object
Initial Value	NIL

1.9.2.4.2 TFrDataset:Methods

■ Constructor ■ Standard ■ Only after Create()

Type	Name
■	IsActive
■	IsLoaded
■	Refresh
■	SetMaster

1.9.2.4.2.1 TFrDataset:IsActive

Returns true if the dataset is active and assigned to its oReport object.

Type	Standard
Parameters	None
Return Value	True if assigned

1.9.2.4.2.2 TFrDataset:IsLoaded

Returns true if the dataset is already loaded.

Type	Standard
Parameters	None
Return Value	True if loaded

1.9.2.4.2.3 TFrDataset:Refresh

It causes a reload of the data.

Type	Standard
Parameters	None
Return Value	True if success

1.9.2.4.2.4 TFrDataset:SetMaster

Sets a master-detail relation.

Type	Standard
Parameters	<oMasteDataset> Master TFrdataset object <aFields>

Fields on both datasets that will be used to set the relation. Each field must occupy an element in the array. When both fields name are different you can use the following expression to indicate it:

```
'cDetailField=cMasterField'
```

Return Value NIL

1.9.2.4.3 TFrDataset:Events

Name
OnAfterLoad
OnClose
OnFirst
OnNext
OnOpen
OnPrior

1.9.2.4.3.1 TFrDataset:OnAfterLoad

Event that fires when the report is loaded. This is a good place to change the value of any property of the report.

Parameters	<oSender>: Object that triggers the event.
Return Value	NIL

1.9.2.4.3.2 TFrDataset:OnClose

Event that fires when you close the dataset.

Parameters	<oSender>: Object that triggers the event.
Return Value	NIL

1.9.2.4.3.3 TFrDataset:OnFirst

Event that triggers when the dataset is positioned at its first record.

Parameters	<oSender>: Object that triggers the event.
Return Value	NIL

1.9.2.4.3.4 TFrDataset:OnNext

Event that triggers when positioning the dataset on next record.

Parameters	<oSender>: Object that triggers the event.
Return Value	NIL

1.9.2.4.3.5 TFrDataset:OnOpen

Event that triggers when you open the dataset.

Parameters	<oSender>: Object that triggers the event.
Return Value	NIL

1.9.2.4.3.6 TFrDataset:OnPrior

Event that triggers when positioning the dataset on the previous record.

Parameters	<oSender>: Object that triggers the event.
Return Value	NIL

1.9.2.5 TFrXailerDataset

Class specialized on connecting Xailer **TDataset** objects with FastReport.

Hierarchy TFrDataset descendant
File \source\FrDataset.prg

1.9.2.5.1 TFrXailerDataset:Properties

■ Read only ■ Assignable ■ Design assignable ■ Runtime assignable

Scope	Name	Type	Initial Value
■	aFields	Array	{"*"}
■	oDataset	Object	NIL

1.9.2.5.1.1 TFrXailerDataset:aFields

The **aFields** is an array with all the field names you want to include. There are some important features:

- Is possible to indicate all the fields of a dataset using the '*' character. Default value.
- Is possible to modify the name of a field for design time using the clause 'AS'. For example: [id AS Code](#).

Scope	Design assignable
Type	Array
Initial Value	{}

Note: Is important to set this property before loading the report in order to avoid a reload.

1.9.2.5.1.2 TFrXailerDataset:oDataset

TDataset inherited object that holds the data.

Scope	Standard
Type	TDataset object
Initial Value	NIL

1.9.2.6 TFrDbfDataset

Class specialized on connecting DBF files with FastReport.

Hierarchy	TFrDataset descendant
File	\source\FrDataset.prg

1.9.2.6.1 TFrDbfDataset:Properties

■ Read only ■ Assignable ■ Design assignable ■ Runtime assignable

Scope	Name	Type	Initial Value
■	aFields	Array	{}

1.9.2.6.1.1 TFrDbfDataset:aFields

The **aFields** is an array with all the field names you want to include. There are some important features:

- You can indicate the table source prefixing the field with its alias. For example:

`Clients->Code`. If you do not indicate the alias, the active area when the report is loaded is used.

- Is possible to indicate all the fields of a table using the '*' character. For example:
`Clients->*`.
- Is possible to modify the name of a field for design time using the clause 'AS'. For example:
`Clients->id AS Code`.

In order to use more than one work area is necessary to set the relations between them using the xBase command `SET RELATION`.

Scope	Design assignable
Type	Array
Initial Value	{}

1.9.2.7 TFrArrayDataset

Class specialized on connecting arrays with FastReport..

Hierarchy	TFrDataset descendant
File	\source\FrDataset.prg

1.9.2.7.1 TFrArrayDataset:Properties

Read only
 Assignable
 Design assignable
 Runtime assignable

Scope	Name	Type	Initial Value
<input type="checkbox"/>	aData	Array	{}
<input type="checkbox"/>	aFields	Array	{}

1.9.2.7.1.1 TFrArrayDataset:aData

Array containing the data to be used by report.

Scope	Design assignable
Type	Array
Initial Value	{}

1.9.2.7.1.2 TFrArrayDataset:aFields

Field structure for every array column. The structure is the same used by the `DbStruct()` function:

```
{ { cName, cType, nLength, nDecimals}, ... }
```

You can also use a literal string using the comma as separator:

```
{ "Name,Type, Length, Decimals", ... }
```

Scope	Design assignable
Type	Array
Initial Value	{}

1.9.2.8 TFrEngineOptions

Class for handling the FastReport print engine.

All the **TFrEngine???** classes indeed they inherit from a base class with the name **TFrOptions**, but is not documented since none of its members can be used directly.

Hierarchy File TComponent descendant
 \source\FrOptions.prg

1.9.2.8.1 TFrEngineOptions:Properties

■ Read only ■ Assignable ■ Design assignable ■ Runtime assignable

Scope	Name	Type	Initial Value
■	cTempDir	String	""
■	IConvertNulls	Logical	.T.
■	IDoublePass	Logical	.F.
■	IgnoreDevByZero	Logical	.F.
■	IPrintIfEmpty	Logical	.T.
■	ISilentMode	Logical	.F.
■	IUseFileCache	Logical	.F.
■	nMaxMemSize	Numeric	10
■	nNewSilentMode	Numeric	simMessageBoxes (0)

1.9.2.8.1.1 TFrEngineOptions:cTempDir

Specifies a path to the directory for storing temporary files.

Scope	Assignable
Type	String
Initial Value	""

1.9.2.8.1.2 TFrEngineOptions:ICovertNulls

Converts the “Null” value of the DB field into “0,” “False,” or empty string, depending on the field type.

Scope	Assignable
Type	Logical
Initial Value	.T.

1.9.2.8.1.3 TFrEngineOptions:IDoublePass

Makes a report a two-pass one.

Scope	Assignable
Type	Logical
Initial Value	.F.

1.9.2.8.1.4 TFrEngineOptions:IgnoreDevByZero

Ignores division by zero errors.

Scope	Assignable
Type	Logical
Initial Value	.F.

1.9.2.8.1.5 TFrEngineOptions:IPrintIfEmpty

Defines, whether it is necessary to print a blank report (one which containing no data lines).

Scope	Assignable
Type	Logical
Initial Value	.T.

1.9.2.8.1.6 TFrEngineOptions:ISilentMode

“Silent” mode. Thus all messages about errors can only be retrieved with the GetErrors method.

Scope	Assignable
Type	Logical

Initial Value	.F.
----------------------	-----

1.9.2.8.1.7 TFrEngineOptions:IUseFileCache

Defines, whether it is necessary to use report pages caching into the file (see the “nMaxMemSize” property).

Scope	Assignable
Type	Logical
Initial Value	.F.

1.9.2.8.1.8 TFrEngineOptions:nMaxMemSize

The maximum size of memory in Mbytes, allocated to the report pages’ cache. It becomes useful in cases when the “IUseFileCache” property is equal to .T.. If a report begins to occupy more memory during construction, caching of the constructed report pages into a temporary file is performed. This property is inexact and allows only approximate determination of the memory limit.

Scope	Assignable
Type	Numeric
Initial Value	10

1.9.2.8.1.9 TFrEngineOptions:nNewSilentMode

Property to set ISilentMode with a better precision.

Scope	Assignable
Type	Numeric
Initial Value	simMessageBoxes
Possible values	simMessageBoxes (0) Shows a dialog for every error occurred simSilent (1): Same behaviour that ISilentMode set to true simReTrhow (2): Forces a runtime error

1.9.2.9 TFrPreviewOptions

Class for handling the FastReport preview engine.

All the **TFrEngine???** classes indeed they inherit from a base class with the name **TFrOptions**, but is not documented since none of its members can be used directly.

Hierarchy TComponent descendant
File \source\FrOptions.prg

1.9.2.9.1 TFrPreviewOptions:Properties

■ Read only ■ Assignable ■ Design assignable ■ Runtime assignable

Scope	Name	Type	Initial Value
■	IAllowEdit	Logical	.F.
■	IDoubleBuffered	Logical	.T.
■	IMaximized	Logical	.T.
■	IModal	Logical	.T.
■	IOutlineExpand	Logical	.T.
■	IOutlineVisible	Logical	.F.
■	IPictureCacheInFile	Logical	.F.
■	IShowCaptions	Logical	.F.
■	nButtons	Numeric	4095
■	nOutLineWidth	Numeric	180
■	nZoom	Numeric	1.00

1.9.2.9.1.1 TFrPreviewOptions:IAllowEdit

Enables or disables a finished report editing. Always false in DEMO versions.

Scope	Assignable
Type	Logical
Initial Value	.F.

1.9.2.9.1.2 TFrPreviewOptions:IDoubleBuffered

A double-buffer mode for the preview window. If enabled (by default), the preview window will not flicker during repainting, but the process speed would be reduced.

Scope	Assignable
Type	Logical
Initial Value	.T.

1.9.2.9.1.3 TFrPreviewOptions:IMaximized

Defines whether the preview window is maximized.

Scope	Assignable
Type	Logical
Initial Value	.T.

1.9.2.9.1.4 TFrPreviewOptions:IModal

Defines whether the preview window is modal.

Scope	Assignable
Type	Logical
Initial Value	.T.

1.9.2.9.1.5 TFrPreviewOptions:IOutlineExpand

Defines whether the report outline is expanded or not.

Scope	Assignable
Type	Logical
Initial Value	.T.

1.9.2.9.1.6 TFrPreviewOptions:IOutlineVisible

Defines whether the panel with the report outline is visible.

Scope	Assignable
Type	Logical
Initial Value	.F.

1.9.2.9.1.7 TFrPreviewOptions:IPictureCacheInFile

Defines whether images should be saved on cache.

Scope	Assignable
--------------	------------

Type	Logical
Initial Value	.F.

1.9.2.9.1.8 TFrPreviewOptions:IShowCaptions

Defines whether it is necessary to display button captions. When enabling this property, you should limit the number of the displayed buttons in the `nButtons` property, since all the buttons would not find room on the screen

Scope	Assignable
Type	Logical
Initial Value	.F.

1.9.2.9.1.9 TFrPreviewOptions:nButtons

A set of buttons, which will be available in the preview window.

Scope	Assignable
Type	Numeric
Initial Value	4095
Possible values	Any <code>nOr()</code> combination of the following values:
	<code>pbPrint</code> 1
	<code>pbLoad</code> 2
	<code>pbSave</code> 4
	<code>pbExport</code> 8
	<code>pbZoom</code> 16
	<code>pbFind</code> 32
	<code>pbOutline</code> 64
	<code>pbPageSetup</code> 128
	<code>pbTools</code> 256
	<code>pbEdit</code> 512
	<code>pbNavigator</code> 1024
	<code>pbExportQuick</code> 2048
	<code>pbNoClose</code> 4096
	<code>pbNoFullScreen</code> 8192
	<code>pbNoEmail</code> 16384

1.9.2.9.1.10 TFrPreviewOptions:nOutlineWidth

Defines width of the panel with the report outline.

Scope	Assignable
--------------	------------

Type	Numeric
Initial Value	180

1.9.2.9.1.11 TFrPreviewOptions:nZoom

The default zooming value.

Scope	Assignable
Type	Numeric
Initial Value	1.00

1.9.2.9.2 TFrPreviewOptions:Methods

■ Constructor ■ Standard ■ Only after Create()

Type	Name
■	SetBounds

1.9.2.9.2.1 TFrPreviewOptions:SetBounds

Sets the preview window coordinates.

Type	Standard
Parameters	<nLeft> Left coordinate <nTop> Top coordinate <nWidth> Window width in pixels <nHeight> Window height in pixels
Return Value	NIL

1.9.2.10 TFrPrintOptions

Class for handling the FastReport printing.

All the **TFrEngine???** classes indeed they inherit from a base class with the name **TFrOptions**, but is not documented since none of its members can be used directly.

Hierarchy	TComponent descendant
File	\source\FrOptions.prg

1.9.2.10.1 TFrPrintOptions:Properties

■ Read only ■ Assignable ■ Design assignable ■ Runtime assignable

Scope	Name	Type	Initial Value
■	cPageNumbers	String	""
■	cPrinter	String	"Default"
■	ICollate	Logical	.T.
■	IRreverse	Logical	.F.
■	IShowDialog	Logical	.T.
■	nCopies	Numeric	1
■	nPrintPages	Numeric	ppAll

1.9.2.10.1.1 TFrPrintOptions:cPageNumbers

Page numbers, which are to be printed. For example, "1,3,5-12,17-".

Scope	Assignable
Type	String
Initial Value	""

1.9.2.10.1.2 TFrPrintOptions:cPrinter

Printer name.

Scope	Assignable
Type	String
Initial Value	"Default"

1.9.2.10.1.3 TFrPrintOptions:ICollate

Whether to collate the copies.

Scope	Assignable
Type	Logical
Initial Value	.T.

1.9.2.10.1.4 TFrPrintOptions:IRreverse

Whether to reverse page printing.

Scope	Assignable
Type	Logical
Initial Value	.F.

1.9.2.10.1.5 TFrPrintOptions:IShowDialog

Whether to display a print dialogue.

Scope	Assignable
Type	Logical
Initial Value	.T.

1.9.2.10.1.6 TFrPrintOptions:nCopies

Number of copies.

Scope	Assignable
Type	Numeric
Initial Value	1

1.9.2.10.1.7 TFrPrintOptions:nPrintPages

Defines the pages to be printed.

Scope	Assignable
Type	Numeric
Initial Value	ppAll
Possible values	ppAll 0
	ppOdd 1
	ppEven 2

1.9.2.10.2 TFrPrintOptions:Methods

■ Constructor ■ Standard ■ Only after Create()

Type	Name
■	ClearOptions

1.9.2.10.2.1 TFrPrintOptions:ClearOptions

Resets all the properties to their default value.

Type	Standard
Parameters	None
Return Value	NIL

1.9.2.11 TFrReportOptions

Class for handling the report itself.

All the **TFrEngine???** classes indeed they inherit from a base class with the name **TFrOptions**, but is not documented since none of its members can be used directly.

Hierarchy TComponent descendant
File \source\FrOptions.prg

1.9.2.11.1 TFrReportOptions:Properties

■ Read only ■ Assignable ■ Design assignable ■ Runtime assignable

Scope	Name	Type	Initial Value
■	aDescription	Array	{}
■	cAuthor	String	""
■	cInitString	String	""
■	cPassword	String	""
■	lCompressed	Logical	.F.

1.9.2.11.1.1 TFrReportOptions:aDescription

String array with report description.

Scope	Assignable
Type	Array
Initial Value	{}

1.9.2.11.1.2 TFrReportOptions:cAuthor

Report author.

Scope	Assignable
Type	Cadena
Initial Value	""

1.9.2.11.1.3 TFrReportOptions:clnitString

Escape sequence for dot matrix printers.

Scope	Assignable
Type	Cadena
Initial Value	""

1.9.2.11.1.4 TFrReportOptions:cPassword

Password for encrypted reports.

Scope	Assignable
Type	Cadena
Initial Value	""

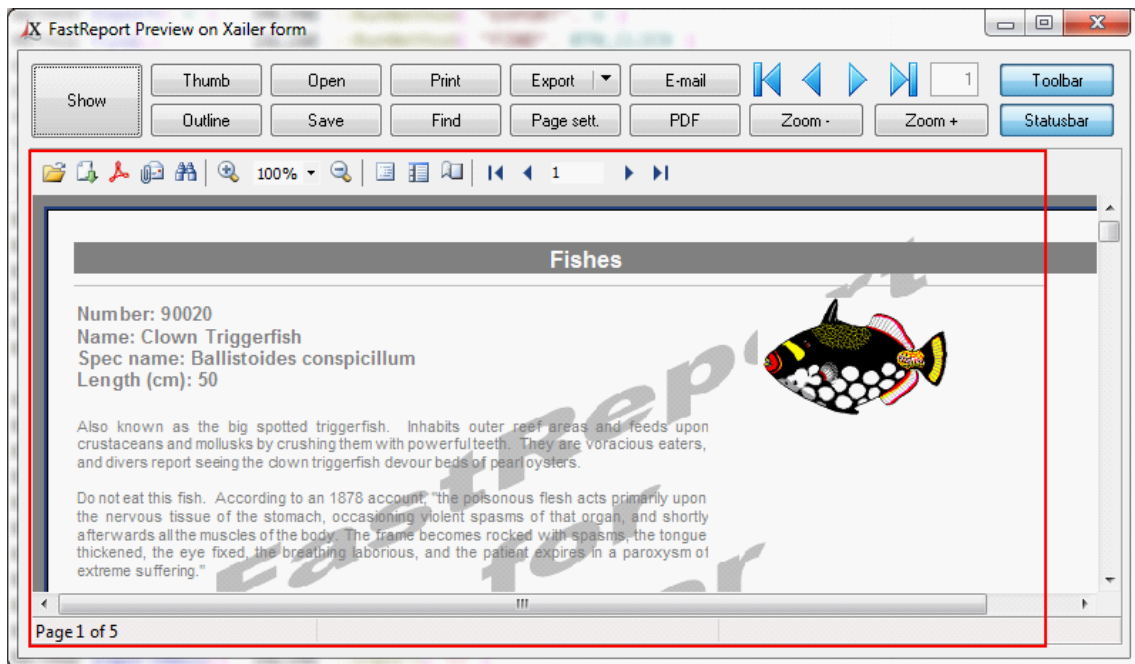
1.9.2.11.1.5 TFrReportOptions:lCompressed

Whether the report is compressed.

Scope	Assignable
Type	Logical
Initial Value	.F.

1.9.3 TFrPreview

This class represents a FastReport preview control. It lets you preview the reports generated by FastReport inside a Xailer form.



Hierarchy
File

Descendent of TStdControl
 \source\FrPreview.prg

1.9.3.1 TFrPreview:Properties

■ Read only ■ Assignable ■ Design assignable ■ Run-time assignable

Scope	Name	Type	Initial value
■	IBtnClose	Logical	.F.
■	IBtnEdit	Logical	.F.
■	IBtnEmail	Logical	.F.
■	IBtnExport	Logical	.T.
■	IBtnFind	Logical	.F.
■	IBtnNav	Logical	.T.
■	IBtnOpen	Logical	.F.
■	IBtnOutline	Logical	.F.
■	IBtnPage	Logical	.T.
■	IBtnPDF	Logical	.T.
■	IBtnPrint	Logical	.T.
■	IBtnSave	Logical	.F.
■	IBtnThumb	Logical	.F.
■	IBtnZoom	Logical	.T.
■	IStatusBar	Logical	.T.

■	lToolbar	Logical	.T.
■	nBorderStyle	Numeric	bvETCHED
■	nHeight	Numeric	100
■	nPage	Numeric	0
■	nPageCount	Numeric	0
■	nWidth	Numeric	100
■	nZoomMode	Numeric	4
■	oReport	Objeto	NIL

1.9.3.1.1 TFrPreview:lBtnEdit

if true the design button will be show on the toolbar if present.

Scope:	Assignable
Type:	Logical
Initial value:	.F.

Note: Is only functional on the registered version of FastReport for Xailer.

1.9.3.1.2 TFrPreview:lBtnClose

if true the Close button will be show on the toolbar if present.

Scope:	Assignable
Type:	Logical
Initial value:	.F.

1.9.3.1.3 TFrPreview:lBtnEmail

if true the Email button will be show on the toolbar if present.

Scope:	Assignable
Type:	Logical
Initial value:	.F.

1.9.3.1.4 TFrPreview:lBtnExport

if true the export button will be show on the toolbar if present.

Scope:	Assignable
---------------	------------

Type:	Logical
Initial value:	.T.

1.9.3.1.5 TFrPreview:IBtnFind

if true the Find button will be show on the toolbar if present.

Scope:	Assignable
Type:	Logical
Initial value:	.T.

1.9.3.1.6 TFrPreview:IBtnNav

if true the navigation buttons will be shown on the toolbar if present.

Scope:	Assignable
Type:	Logical
Initial value:	.T.

1.9.3.1.7 TFrPreview:IBtnOpen

if true the Open button will be show on the toolbar if present.

Scope:	Assignable
Type:	Logical
Initial value:	.F.

1.9.3.1.8 TFrPreview:IBtnOutline

if true the Outline button will be show on the toolbar if present.

Scope:	Assignable
Type:	Logical
Initial value:	.F.

1.9.3.1.9 TFrPreview:IBtnPage

if true the Page settings button will be show on the toolbar if present.

Scope:	Assignable
Type:	Logical
Initial value:	.T.

1.9.3.1.10 TFrPreview:IBtnPDF

if true the export to PDF button will be show on the toolbar if present.

Scope:	Assignable
Type:	Logical
Initial value:	.T.

1.9.3.1.11 TFrPreview:IBtnPrint

if true the Print button will be show on the toolbar if present.

Scope:	Assignable
Type:	Logical
Initial value:	.T.

1.9.3.1.12 TFrPreview:IBtnSave

if true the Save button will be show on the toolbar if present.

Scope:	Assignable
Type:	Logical
Initial value:	.F.

1.9.3.1.13 TFrPreview:IBtnThumb

if true the Thumbnail button will be show on the toolbar if present.

Scope:	Assignable
Type:	Logical
Initial value:	.F.

1.9.3.1.14 TFrPreview:IBtnZoom

if true the zoom buttons will be shown on the toolbar if present.

Scope:	Assignable
Type:	Logical
Initial value:	.T.

1.9.3.1.15 TFrPreview:IStatusBar

If true the status bar will be shown.

Scope:	Assignable
Type:	Logical
Initial value:	.T.

1.9.3.1.16 TFrPreview:IToolbar

If true the Tool bar will be shown.

Scope:	Assignable
Type:	Logical
Initial value:	.T.

1.9.3.1.17 TFrPreview:nBorderStyle

Boder style.

Scope	Assignable
Type	Numeric
Initial value	bvETCHED
Possible values	bvNONE, bvRAISED, bvSUNKEN, bvBUMP, bvETCHED, bvFLAT

1.9.3.1.18 TFrPreview:nHeight

Height of the control.

Scope	Assignable
Type	Numeric
Initial value	100

1.9.3.1.19 TFrPreview:nPage

Report page number.

Scope	Assignable on run-time
Type	Numeric
Initial value	0

1.9.3.1.20 TFrPreview:nPageCount

Número total de páginas del informe.

Scope	Sólo lectura
Type	Numeric
Initial value	0

1.9.3.1.21 TFrPreview:nWidth

Width of the control.

Scope	Assignable
Type	Numeric
Initial value	100

1.9.3.1.22 TFrPreview:nZoomMode

Ordinal value on the Zoom combobox of the preview toolbar.

Scope	Assignable
Type	Numeric
Initial value	4

Possible values	1: 25%
	2: 50%
	3: 75%
	4: 100%
	5: 150%
	6: 200%
	7: Page width
	8: All page

1.9.3.1.23 TFrPreview:oReport

TFastReport object of the control.

Scope	Assignable
Type	Objeto
Initial value	NIL

1.9.3.2 TFrPreview:Methods

■ Constructor ■ Standard ■ Only after Create()

Type	Name
■	Close
■	Edit
■	Email
■	Export
■	ExportBIF
■	ExportBMP
■	ExportCSV
■	ExportEMF
■	ExportExcel
■	ExportGIF
■	ExportHTML
■	ExportJPEG
■	ExportMail
■	ExportODS
■	ExportODT
■	ExportPDF
■	ExportRTF
■	ExportTIFF
■	ExportTXT
■	Find
■	NavFirst
■	NavLast
■	NavNext

■	NavPrior
■	Open
■	Outline
■	Page
■	Pdf
■	Print
■	Save
■	Thumb
■	ZoomMinus
■	ZoomPlus

1.9.3.2.1 TFrPreview:Close

Closes the report (not the preview form).

Type	Only after Create
Parameters	None
Return value	NIL

1.9.3.2.2 TFrPreview:Edit

Sets Preview design mode.

Type	Only after Create
Parameters	None
Return value	NIL

Note: Only functional on the registered version of FastReport for Xailer.

1.9.3.2.3 TFrPreview:Email

Sends the report via Email.

Type	Only after Create
Parameters	None
Return value	NIL

1.9.3.2.4 TFrPreview:Export

Exports the report.

Type	Only after Create
Parameters	<nType> Filter type 1: PDF 2: HTML 3: RTF 4: CSV 5: Excel 6: BMP 7: JPEG 8: TIFF 9: GIF 10: TXT 11: Mail 12: ODS 13: ODT 14: EMF 15: BIF
Return value	NIL

1.9.3.2.5 TFrPreview:ExportBIF

Exports the report with BIF format.

Type	Only after Create
Parameters	None
Return value	NIL

1.9.3.2.6 TFrPreview:ExportBMP

Exports the report on BMP format.

Type	Only after Create
Parameters	None
Return value	NIL

1.9.3.2.7 TFrPreview:ExportCSV

Exports the report with CSV format.

Type	Only after Create
Parameters	None
Return value	NIL

1.9.3.2.8 TFrPreview:ExportEMF

Exports the report with EMF format.

Type	Only after Create
Parameters	None
Return value	NIL

1.9.3.2.9 TFrPreview:ExportExcel

Exports the report with Excel format.

Type	Only after Create
Parameters	None
Return value	NIL

1.9.3.2.10 TFrPreview:ExportGIF

Exports the report with GIF format.

Type	Only after Create
Parameters	None
Return value	NIL

1.9.3.2.11 TFrPreview:ExportHTML

Exports the report in HTML format.

Type	Only after Create
Parameters	None
Return value	NIL

1.9.3.2.12 TFrPreview:ExportJPEG

Exports the report with JPEG format.

Type	Only after Create
Parameters	None
Return value	NIL

1.9.3.2.13 TFrPreview:ExportMail

Sends the report via Email.

Type	Only after Create
Parameters	None
Return value	NIL

1.9.3.2.14 TFrPreview:ExportODS

Exports the report with ODS format (Open document spreadsheet).

Type	Only after Create
Parameters	None
Return value	NIL

1.9.3.2.15 TFrPreview:ExportODT

Exports the report with ODT format (Open document text).

Type	Only after Create
Parameters	None
Return value	NIL

1.9.3.2.16 TFrPreview:ExportPDF

Exports the report with PDF format.

Type	Only after Create
Parameters	None
Return value	NIL

1.9.3.2.17 TFrPreview:ExportRTF

Exports the report with RTF format.

Type	Only after Create
Parameters	None
Return value	NIL

1.9.3.2.18 TFrPreview:ExportTIFF

Exports the report with TIFF format.

Type	Only after Create
Parameters	None
Return value	NIL

1.9.3.2.19 TFrPreview:ExportTXT

Exports the report with TXT format.

Type	Only after Create
Parameters	None
Return value	NIL

1.9.3.2.20 TFrPreview:Find

Activates the Preview Find dialog.

Type	Only after Create
Parameters	None
Return value	NIL

1.9.3.2.21 TFrPreview:NavFirst

Navigates to first report page.

Type	Only after Create
Parameters	None
Return value	NIL

1.9.3.2.22 TFrPreview:NavLast

Navigates to last report page.

Type	Only after Create
Parameters	None
Return value	NIL

1.9.3.2.23 TFrPreview:NavNext

Navigates to next report page.

Type	Only after Create
Parameters	None
Return value	NIL

1.9.3.2.24 TFrPreview:NavPrior

Navigates to prior report page.

Type	Only after Create
Parameters	None
Return value	NIL

1.9.3.2.25 TFrPreview:Open

Activates de open file dialog to open a new report.

Type	Only after Create
Parameters	None
Return value	NIL

1.9.3.2.26 TFrPreview:Outline

Show the report outline.

Type	Only after Create
Parameters	None
Return value	NIL

1.9.3.2.27 TFrPreview:Page

Shows the report page settings.

Type	Only after Create
Parameters	None
Return value	NIL

1.9.3.2.28 TFrPreview:Print

Prints the report.

Type	Only after Create
Parameters	None
Return value	NIL

1.9.3.2.29 TFrPreview:Save

Saves the report.

Type	Only after Create
Parameters	None
Return value	NIL

1.9.3.2.30 TFrPreview:Thumb

Shows report thumbnails.

Type	Only after Create
Parameters	None
Return value	NIL

1.9.3.2.31 TFrPreview:ZoomMinus

Zoom minus.

Type	Only after Create
Parameters	None
Return value	NIL

1.9.3.2.32 TFrPreview:ZoomPlus

Zoom plus.

Type	Only after Create
Parameters	None
Return value	NIL

1.9.4 Sumatra PDF

Sumatra PDF reader is a great tool for viewing and printing PDF files. It is very fast, lightweight and its integration with Xailer is total, since it behaves like one more control within your application, which you can easily select from the IDE's own controls palette.

In order to use the control, your application needs to be able to access a single file named SumatraPdf.exe that you can find in the \xailer\samples\sumatra directory. You can see that in that same folder there is another executable with the name SumatraPdf31.exe, which corresponds to the previous version of Sumatra PDF viewer. Both have been incorporated, because the previous version is ridiculously large at 2,721 Kb which may be enough for the files you want to display. However, version 3.2 has a size of 14,017 Kb which is not very big considering everything it does.

Sumatra PDF viewer supports the following file types: PDF, EPUB, MOBI, CHM, XPS, DjVu, CBZ, and CBR. And the following extensions:

- PDF (.pdf), unencrypted EPUB (.epub), MOBI (.mobi)
- Fiction Wise (.fb2, .fb2z, .zfb2), Palm DOC format (.pdb)
- comic book files: (.cbz, .cbr, .cbr, .cb7z)
- DjVu (.djv, .djvu), Microsoft Compiled HTML Html (.chm)
- XPS (.xps), images (.jpg, .png, .gif, .webp, .tiff, tga, .j2k)

You can obtain further information on this link: <https://www.sumatrapdfreader.org>

The license under which this product is offered is: GNU General Public License v3, which in principle is compatible with the commercial use of the product as long as all the restrictions imposed by the license are followed. However, the Xailer team contacted the developer Krzysztof Kowalczyk to confirm it and below we reproduce his response that can even be seen in his forum:

Question from iozuniga (Xailer team)

Hello,

I am one of the main developers of Xailer which is a visual development environment for Xbase users, using the Harbour-project open source xBase compiler.

Our users, for a long time, have requested a control to view and print PDF documents. We have tried many alternatives, but none is comparable with Sumatra PDF. We would love to be able to distribute the Sumatra PDF binaries (with absolutely all the requirements of your license) along with our product and that our users can also distribute it with the software they make, also complying with all the requirements of the license.

Sorry if this is a stupid question, we are a little bit confused about the AGPLv3 and BSD mixing licenses, but we do not want to do anything not authorized.

Thanks in advance.

Response from Krzysztof Kowalczyk (Main developer of Sumatra PDF Reader)

If you don't modify the SumatraPDF, you can include it with your software.

If you do modify SumatraPDF code or use its code, you have to conform to AGPLv3 license, which means also providing the code of the derived/combined software.

SumatraPDF license is AGPLv3. Some code is under more permissive BSD license but the whole is governed by more restrictive AGPLv3.

1.9.5 TSumatraViewer

Control that allows viewing PDF files.

Hierarchy	TWinControl descendant
File	source\Sumatra.prg

1.9.5.1 TSumatraViewer:Properties

Read only
 Assignable
 Design assignable
 Run-time assignable^a

Scope	Name	Type	Initial value
<input type="checkbox"/>	cExeFile	Character	"sumatrapdf.exe"
<input type="checkbox"/>	cExePath	Character	""
<input type="checkbox"/>	cFileName	Character	""
<input type="checkbox"/>	cLastError	Character	""
<input type="checkbox"/>	lHideToolBar	Logical	.F.
<input type="checkbox"/>	lRestrictMode	Logical	.F.
<input type="checkbox"/>	nInitialPage	Numeric	0
<input type="checkbox"/>	nInitialView	Numeric	vmSinglePage
<input type="checkbox"/>	nInitialZoom	Numeric	izFitWidth

1.9.5.1.1 TSumatraViewer:cExeFile

Name of the Sumatra PDF viewer executable. By default, the same application directory will be used.

Scope	Assignable
Type	Character
Initial value	"sumatrapdf.exe"

1.9.5.1.2 TSumatraViewer:cExePath

Localization of the Sumatra PDF viewer executable. By default, the same application directory will be used. If the full path including the file is not indicated, it is assumed that this property contains the directory and the name of the executable is:

[SumatraPdf.Exe](#)

Scope	Assignable
Type	Character
Initial value	""

1.9.5.1.3 TSumatraViewer:cFileName

Name of the PDF file to open.

Scope	Run-time assignable
Type	Character
Initial value	""

1.9.5.1.4 TSumatraViewer:cLastError

Description of last error produced.

Scope	Read only
Type	Character
Initial value	""

1.9.5.1.5 TSumatraViewer:lHideToolBar

If true, the top tool bar is hided.

Scope	Assignable
Type	Logical
Initial value	.F.

1.9.5.1.6 TSumatraViewer:lRestrictMode

If true, the restrictive mode of the utility will be used, by which some features of the utility are restricted, such as:

- Access to the file system
- Access to the registry
- Access to the Internet

For more information see the following link:

<https://www.sumatrapdfreader.org/docs/Command-line-arguments.html>

Scope	Assignable
Type	Logical
Initial value	.F.

1.9.5.1.7 TSumatraViewer:nInitialPage

First page to show.

Scope	Assignable
Type	Numeric
Initial value	0

1.9.5.1.8 TSumatraViewer:nInitialView

Initial view.

For further information, consult:

<https://www.sumatrapdfreader.org/docs/Command-line-arguments.html>

Scope	Assignable
Type	Numeric
Initial value	izFitWidth
Possible values	izFitPage izFitWidth izFitContent izPercent10 izPercent25 izPercent50 izPercent100 izPercent125 izPercent150

1.9.5.1.9 TSumatraViewer:nInitialZoom

Initial zoom.

For further information consult:

<https://www.sumatrapdfreader.org/docs/Command-line-arguments.html>

Scope	Assignable
Type	Numeric
Initial value	vmSinglePage
Possible values	vmSinglePage vmContinousSinglePage vmFacing vmContinousFacing vmBookView vmContinousBookView

1.9.5.2 TSumatraViewer:Methods

■ Constructor ■ Standard

Type	Name
■	FileOpen

1.9.5.2.1 TSumatraViewer:FileOpen

Opens the PDF document. This method is equivalent to setting the cFileName property.

Type	Standard
Parameters	<cFileName> File name
Return value	<ISuccess> True if success

1.9.6 TSumatraPrinter

Control that allows printing PDF files.

Hierarchy TComponent descendant
File source\Sumatra.prg

1.9.6.1 TSumatraPrinter:Properties

■ Read only ■ Assignable ■ Design assignable ■ Run-time assignable

Scope	Name	Type	Initial value
■	cBin	Character	""
■	cExeFile	Character	"sumatrapdf.exe"
■	cExePath	Character	""
■	cFileName	Character	""
■	cLastError	Character	""
■	cPrinter	Character	""
■	cRange	Character	""
■	lExitWhenDone	Logical	.T.
■	lShowDialog	Logical	.F.
■	lSilent	Logical	.T.
■	nColor	Numeric	0
■	nCopies	Numeric	1
■	nDuplex	Numeric	pdDefault
■	nPageSize	Numeric	psDefault
■	nPagesType	Numeric	ptAll
■	nScale	Numeric	psDefault

1.9.6.1.1 TSumatraPrinter:cBin

Printer bin to use.

For further information:

<https://www.sumatrapdfreader.org/docs/Command-line-arguments.html>

Scope	Assignable
Type	Character
Initial value	""

1.9.6.1.2 TSumatraPrinter:cExeFile

Sumatra PDF viewer executable name. By default the same App directory will be used.

Scope	Assignable
Type	Character
Initial value	"sumatrapdf.exe"

1.9.6.1.3 TSumatraPrinter:cExePath

Sumatra PDF viewer executable path. By default, the App directory will be used. If no full path is set, is assumed that this property contains the directory name and the executable name is:

`SumatraPdf.Exe`

Scope	Assignable
Type	Character
Initial value	""

1.9.6.1.4 TSumatraPrinter:cPrinter

Printer name to use.

For further information:

<https://www.sumatrapdfreader.org/docs/Command-line-arguments.html>

Scope	Assignable
Type	Character
Initial value	""

1.9.6.1.5 TSumatraPrinter:cRange

Range of pages to print. You can indicate specific pages or ranges. For example:

"1-3,5,8-10"

From 1 to 3, 5 and from 8 to 10.

For further information:

<https://www.sumatrapdfreader.org/docs/Command-line-arguments.html>

Scope	Assignable
Type	Character
Initial value	""

1.9.6.1.6 TSumatraPrinter:!ExitWhenDone

If true Sumatra is unloaded after printing.

For further information:

<https://www.sumatrapdfreader.org/docs/Command-line-arguments.html>

Scope	Aszignable
Type	Logical
Initial value	.T.

1.9.6.1.7 TSumatraPrinter:!ShowDialog

If true, the print dialog will be shown.

For further information:

<https://www.sumatrapdfreader.org/docs/Command-line-arguments.html>

Scope	Assignable
Type	Logical
Initial value	.F.

1.9.6.1.8 TSumatraPrinter:!Silent

If true, the Sumatra App is not displayed during printing.

For further information:

<https://www.sumatrapdfreader.org/docs/Command-line-arguments.html>

Scope	Assignable
Type	Logical
Initial value	.T.

1.9.6.1.9 TSumatraPrinter:nColor

Impression type: color or black & white.

For further information:

<https://www.sumatrapdfreader.org/docs/Command-line-arguments.html>

Scope	Assignable
Type	Numeric
Initial value	pcDefault
Possible values	pcDefault pcColor pcMonochrome

1.9.6.1.10 TSumatraPrinter:nCopies

Number of copies

For further information:

<https://www.sumatrapdfreader.org/docs/Command-line-arguments.html>

Scope	Assignable
Type	Numeric
Initial value	1

1.9.6.1.11 TSumatraPrinter:nDuplex

Duplex printing.

For further information:

<https://www.sumatrapdfreader.org/docs/Command-line-arguments.html>

Scope	Assignable
Type	Numeric
Initial value	pdDefault
Possible values	pdDefault pdSimplex pdDuplex

pdDuplexShort
pdDuplexLong

1.9.6.1.12 TSumatraPrinter:nPageSize

Page size.

For further information:

<https://www.sumatrapdfreader.org/docs/Command-line-arguments.html>

Scope	Assignable
Type	Numeric
Initial value	psDefault
Possible values	psDefault psA4 psA5 psA3 psA2 psLetter

1.9.6.1.13 TSumatraPrinter:nPageType

Page type.

For further information:

<https://www.sumatrapdfreader.org/docs/Command-line-arguments.html>

Scope	Assignable
Type	Numeric
Initial value	ptAll
Possible values	ptAll ptEven ptOdd

1.9.6.1.14 TSumatraPrinter:nScale

Printing scale.

For further information:

<https://www.sumatrapdfreader.org/docs/Command-line-arguments.html>

Scope	Assignable
Type	Numeric

Initial value	psDefault
Possible values	psDefault psFit psShrink psNoScale

1.9.6.1.15 TSumatraPrinter:Methods

■ Constructor ■ Standard

Type	Name
■	Print


1.9.6.1.15.1 TSumatraPrinter:Print

Prints the PDF document. This method is equivalent to setting cFileName property.

Type	Standard
Parameters	<cFileName> File name
Return value	<ISuccess> True if success

1.9.7 PDF Sign tool

Tool to **digitally sign PDF** files. Uses *GNU Affero General Public License v3.0* which lets you use it commercially with some limitations:

 IcoDeveloper/PDFSign is licensed under the GNU Affero General Public License v3.0 Permissions of this strongest copyleft license are conditioned on making available complete source code of licensed works and modifications, which include larger works using a licensed work, under the same license. Copyright and license notices must be preserved. Contributors provide an express grant of patent rights. When a modified version is used to provide a service over a network, the complete source code of the modified version must be made available.	Permissions ✓ Commercial use ✓ Modification ✓ Distribution ✓ Patent use ✓ Private use	Limitations ✗ Liability ✗ Warranty	Conditions ⓘ License and copyright notice ⓘ State changes ⓘ Disclose source ⓘ Network use is distribution ⓘ Same license

The tool can be downloaded from:
<https://github.com/IcoDeveloper/PDFSign>

You may also find it at **Xailer\samples\sumatra** with the name [PdfSign13.exe](#)

Licence:
<https://github.com/IcoDeveloper/PDFSign/blob/master/LICENSE>

Recommended time servers:
<https://freetlsa.org/tsr>

Command line use:

```
pdfsign v1.3.0, (c) 2019 icomedias GmbH
powered by iTextSharp 5.5 Copyright (C) 1999-2018 by iText Group NV
Usage: pdfsign [OPTIONS]
Sign a PDF file using a signing certificate
```

Options:

```
-i, --infile=VALUE          PDF input file
-o, --outfile=VALUE         output file for signed PDF
-b, --backpage=VALUE       PDF file to append to infile before placing
                           signature (optional)
-c, --certfile=VALUE       PKCS12 signing certificate
-p, --password=VALUE       import password for signing certificate
  --thumbprint=VALUE       thumbprint for signing certificate from windows
                           store
  --store=VALUE            store for signing certificate from windows (
                           CurrentUser or LocalMachine (default
                           LocalMachine))
-r, --reason=VALUE         signature reason (gets embedded in signature)
-l, --location=VALUE       signature location (gets embedded in signature)
-t, --contact=VALUE        signature contact (gets embedded in signature)
-s, --show                 show signature (signature field visible), on: -s+
                           off: -s-, default on
  --page=VALUE            page of the document to place signature: 1..n,
                           last. default 1
  --template=VALUE        Template for the signature text. use \n for line
                           breaks, [name], [date] for substitution
  --dateformat=VALUE       format for [date] substitutuin when using template
--showvalidity            show signature validity (deprecated), on: -
                           showvalidity+ off: -showvalidity-, default off
--tsa=VALUE               URL of rfc3161 TSA (Time Stamping Authority)
--width=VALUE             signature width, default 180
--height=VALUE            signature height, default 80
--hsep=VALUE              horizontal seperation of signatures, default 10
--vsep=VALUE              vertical seperation of signatures, default 10
--hoffset=VALUE           horizontal offset of signatures, default 350
--voffset=VALUE           vertical offset of signatures, default 5
--cols=VALUE              number of signature columns, default 1
-m, --multi               allow multiple signatures, on: -m+, off: -m-,
                           default on
-h, -?, --help            show this help message and exit
```

Return Values:

```
0: Success
-1: Bad Command Line Option(s)
-2: Error processing signing certificate
-3: Error getting secret key
-4: Error getting certificate chain
-5: Error processing input file
-6: Error opening output file````
```

multiple signatures

Multiple signatures are supported; if you leave signature visibility turned on, additional signatures get seperate signature field names (Signature, Signature1, Signature2...) and are automatically positioned as a grid with --cols columns from left to right and bottom to top.

1.9.8 TPdfSignTool

This class allows the digital signature (**PKCS12** format) of PDF files.

Hierarchy
File

TComponent descendant
source\PdfSingTool.prg

1.9.8.1 TPdfSign:Properties

■ Read only ■ Assignable ■ Design assignable ■ Run-time assignable

Scope	Name	Type	Initial value
■	cCertFile	Character	""
■	cContact	Character	""
■	cError	Character	""
■	cInpuFile	Character	""
■	cOutputFile	Character	""
■	cPassword	Character	""
■	cReason	Character	""
■	cTimeStampAccount	Character	""
■	cTimeStampPassword	Character	""
■	cTimeStampURL	Character	""
■	IShowProcess	Logical	.F.
■	IShowResult	Logical	.F.
■	IShowSignature	Logical	.T.
■	nErroCode	Logical	.F.
■	nSignColumns	Numeric	1
■	nSignHeight	Numeric	80
■	nSignHorzOffset	Numeric	350
■	nSignHorzSep	Numeric	10
■	nSignVertOffset	Numeric	5
■	nSignVertSep	Numeric	10
■	nSignWidth	Numeric	180

1.9.8.1.1 TPdfSignTool:cCertFile

Digital sign certificate with PKCS12 format.

Scope	Assignable
Type	Character
Initial value	""

1.9.8.1.2 TPdfSignTool:cContact

Person name signing the document.

Scope	Assignable
Type	Character

Initial value	""
----------------------	----

1.9.8.1.3 TPdfSignTool:cError

Last error description.

Scope	Read only
Type	Character
Initial value	""

1.9.8.1.4 TPdfSignTool:cInputFile

Input filename to sign.

Scope	Assignable
Type	Character
Initial value	""

1.9.8.1.5 TPdfSignTool:cLocation

Location where the document is signed.

Scope	Assignable
Type	Character
Initial value	""

1.9.8.1.6 TPdfSignTool:cOutputFile

Output signed filename.

Scope	Assignable
Type	Character
Initial value	""

1.9.8.1.7 TPdfSignTool:cPassword

Certificate filename password.

Scope	Assignable
Type	Character
Initial value	""

1.9.8.1.8 TPdfSignTool:cReason

Reason the document is signed.

Scope	Assignable
Type	Character
Initial value	""

1.9.8.1.9 TPdfSignTool:cTimeStampAccount

User name for the timestamp service.

Scope	Assignable
Type	Character
Initial value	""

1.9.8.1.10 TPdfSignTool:cTimeStampPassword

Password for the timestamp service.

Scope	Assignable
Type	Character
Initial value	""

1.9.8.1.11 TPdfSignTool:cTimeStampURL

Timestamp server URL.

Scope	Assignable
Type	Character
Initial value	""

1.9.8.1.12 TPdfSignTool:cToolPath

Directory where **pdf_sign.exe** utility is located. Leave blank if is in the same application directory or in the system PATH.

Scope	Assignable
Type	Character
Initial value	""

1.9.8.1.13 TPdfSignTool:IMultiSign

True if the document can have multiple certificates.

Scope	Assignable
Type	Logical
Initial value	.F.

1.9.8.1.14 TPdfSignTool:IShowProcess

True to show the signing process on a console window.

Scope	Assignable
Type	Logical
Initial value	.F.

1.9.8.1.15 TPdfSignTool:IShowResult

True to show the document signed with the standard PDF viewer.

Scope	Assignable
Type	Logical
Initial value	.F.

1.9.8.1.16 TPdfSignTool:iShowSignature

True to show signature on the signed document.

Scope	Assignable
Type	Logical
Initial value	.T.

1.9.8.1.17 TPdfSignTool:nErrorCode

Last error code. Value zero means success.

Scope	Read only
Type	Numeric
Initial value	0

1.9.8.1.18 TPdfSignTool:nSignColumns

Number of digital signatures.

Scope	Assignable
Type	Numeric
Initial value	1

1.9.8.1.19 TPdfSignTool:nSignHeight

Height of digital signatures.

Scope	Assignable
Type	Numeric
Initial value	80

1.9.8.1.20 TPdfSignTool:nSignHorzOffset

Left margin to display signatures.

Scope	Assignable
Type	Numeric
Initial value	350

1.9.8.1.21 TPdfSignTool:nSignHorzSep

Horizontal separation between signatures.

Scope	Assignable
Type	Numeric
Initial value	10

1.9.8.1.22 TPdfSignTool:nSignVertOffset

Vertical margin from the bottom of the page to display signatures.

Scope	Assignable
Type	Numeric
Initial value	5

1.9.8.1.23 TPdfSignTool:nSignVertSep

Vertical separation between signatures.

Scope	Assignable
Type	Numeric
Initial value	10

1.9.8.1.24 TPdfSignTool:nSignWidth

Display width of digital signatures.

Scope	Assignable
Type	Numeric
Initial value	180

1.9.8.2 TPdfSignTool:Methods

■ Constructor ■ Standard

Type	Name
■	Run

1.9.8.2.1 TPdfSignTool:Run

Signs the document

Type	Standard
Parameters	None
Return value	<ISuccess> True if success

1.10 Multithread programming

Introduction

In programming, multitasking is the ability to perform different routines simultaneously, with the possibility of communication and synchronization between them. Within multitasking itself, we can distinguish two concepts derived from the same, but differ in their ways: multiprocessing and multithreading (or multithread).

The multi-tasking is applied to different programs running within a computer, and although there is the possibility of communication and synchronization between them, each process is separated from the rest by protection mechanisms of the operating system, and run in areas or completely separate memory spaces, so that they can not interact with each other by mistake. From our developer standpoint, we should be concerned how it really works, because it is the operating system that manages everything you need.

The multithreaded programming, unlike multiprocessing is to create different threads of code within the same program, sharing the same memory space as well as the mechanisms for communication and synchronization between them. Being all the threads within the same memory space, there is a significant risk of interactions between them, which if not handled correctly can lead to errors.

Fortunately, the operating system provides us with a number of mechanisms to control the execution of threads, and avoid situations that may compromise performance. However, it is we who must control all this, and pay attention to all the circumstances that may be conflicting and find ways to solve them. This may seem simple, but there are so many factors to consider, which often becomes very complicated, especially on current multicore processors.

In Xailer, we wanted to simplify this task a bit, adding some mechanisms that serve to keep everything more under control, and avoid the need to directly manage operating system resources.

Of course the Harbour itself also provides some mechanisms that help multithreaded programming, but in this paper we focus on Xailer, since our view, is simpler and fits better way to program all in Xailer .

Multithreaded programming support has been introduced into Xailer 3.1.

In previous versions of Xailer using the Harbour compiler, you can also use

multithreading, but using Harbour own functions, and with many limitations, since Xailer does not check any additional thread activity and all that task falls to the programmer.

Old version of Xailer that used xHarbour can only make use of the C-level multithreaded programming, very simple and limited, because although xHarbour supported multithreaded, it had some issues that made it very unstable and unreliable.

To use the multithreaded programming in Xailer you must include the **hbvmmt** library into your project. You'll need to go to the project properties, and under the heading libraries, uncheck the **hbvm** library and check **hbvmmt**.

How to create a thread

The easiest way to create a new thread is using the new component TThread. This component allows creating a new thread, as well as communicate with the new thread and the main thread and handle the synchronization between the two.

The execution of the second thread starts with the Run method (<uCode>, ...), where <uCode> can be a codeblock or a pointer to a function or procedure, and can pass all parameters needed. The Run () method runs the second thread and returns immediately. As the second thread can take a while to get started, there is no guarantee that the return from Run () method, the second thread is already created and running.

The second thread may be controlled using the following methods:

- [Pause](#)([<IWait>]) stops the thread execution, which will be on pause until you restart its execution or abandon it completely.
- [Resume](#)([<IWait>]) continues execution of a thread that is in a paused state.
- [Stop](#)([<IWait>]) leaves the thread execution. This abandonment occurs through a [BREAK](#) statement, so that if you use a [BEGIN SEQUENCE](#) / [END SEQUENCE](#) within the thread code, we have the opportunity to maintain control of the thread output, eg release resources used.
- [Quit](#)() leaves the thread execution immediately. This is done using the QUIT command within the thread, causing the complete closure of the VM corresponding to this thread and will have no ability to control the output thereof.

Methods [Pause](#)(), [Resume](#)() and [Stop](#)() support a logical type parameter, which indicates whether the method should wait until you have been processed, or if on the contrary, it should return immediately. The default value is `.F.`, which means that it returns immediately.

The [Quit](#)() method always waits for the second thread to finish its execution. In addition to the above methods, the class has a series of events that keep us informed about the execution of the thread:

- [OnStart](#)(<oSender>) fires right when the thread begins execution.
- [OnEnd](#)(<oSender>) occurs just when the thread terminates, either because the code has reached the end or when you call the methods [Stop](#)() or [Quit](#)() .

- `OnPause(<oSender>)` is executed when the thread enters pause mode, by calling the `Pause()`.
- `OnResume(<oSender>)` fires when the thread exits the sleep mode, by calling the `Resume()`.

The second thread

When you call the `Run()` method and runs the thread, this thread "gets" the Application variables, Screen, Printer and major AppData, and creates a new variable called `SelfThread` always available and contains the object of the corresponding `TThread` class. This public variable `SelfThread` is also available from the main thread, but its value is always `Nil`. This can be useful in certain circumstances to check if you are running the main thread (`SelfThread == Nil`) or if we are in a second thread (`== SelfThread <object>`).

From the second thread you may use the methods `Pause()`, `Resume()`, `Stop()` and `Quit()` from the object `SelfThread`, that will interact with itself.

Communication between the main thread and the second thread

Many times it is necessary to notify the main thread about something that happens in the second thread, and vice versa. Eg, from the second thread can go notifying the main thread on the progress of a task, so that the main thread increments a progress bar that shows the user the current state of the task.

To send notifications we have several methods:

- `Notify([<lWait>], [<nValue>], [<uData>])` sends data from the second thread to the main thread.
 - `<lWait>` indicates whether the call waits for the main thread response. If is `.T.`, then it will not return until the main thread has answered to its `OnNotify()` event. If is `.F.`, the method returns immediately. The default value is `.F.`
 - `<nValue>` is a numeric value (unsigned 32 bits) you want to send.
 - `<uData>` is any valid data in Harbour (a single value, a string, an array, an object, etc.).
- `NotifyThread([<lWait>], [<nValue>], [<uData>])` sends data from the main thread to the second thread.
 - `<lWait>` indicates whether the call waits for the main thread response. If is `.T.`, then it will not return until the main thread has answered to its `OnNotifyThread()` event. If is `.F.`, the method returns immediately. The default value is `.F.`
 - `<nValue>` is a numeric value (unsigned 32 bits) you want to send..
 - `<uData>` is any valid data in Harbour (a single value, a string, an array, an object, etc.).

On the opposite side, which receives notifications, these events are triggered:

- `OnNotify(<oSender>, <nValue>, <uData>)` received in the main thread when the second thread sends a message.
 - `<oSender>` is `TThread` class object that triggered the event.
 - `<nValue>` is a numeric value (unsigned 32 bits) that was sent from the method `Notify()` of the second thread.

- `<uData>` is the `<uData>` value sent from the `Notify()` method on the second thread.
- `OnNotifyThread(<oSender>, <nValue>, <uData>)` is received in the second thread when the main thread sends a message.
 - `<oSender>` is `TThread` class object that triggered the event..
 - `<nValue>` is a numeric value (unsigned 32 bits) that was sent from the method `NotifyThread()` of the second thread.
 - `<uData>` is the value `<uData>` sent from the `NotifyThread()` method on the main thread.

You can not use the method `Notify()` from the main thread, or method `NotifyThread()` from the second thread, putting `<lWait>` a `.T.`, and causing complete blockage of the program (deadlock), because the same thread could not answer his own event.

The second thread must call regularly the `ProcessMessages()` function, so you can receive notifications and events from the main thread. If `ProcessMessages()` is never called, the thread will be completely isolated, and not notified so I will not respond to requests from the main thread. In the same way, if the main thread is executing a loop or a long process `ProcessMessages()` should also be called to give the opportunity to process the notifications it receives the second thread.

When a second thread has completed a task but has to continue running, unfinished, waiting for a notification from the main thread, it can be paused by calling `SelfThread:Pause()` method, and it will not consume any CPU resources. To reactivate it you should call the method `SelfThread:Resume()` or should be the main thread which will call the `Resume()` method on the corresponding `TThread` Object.

If the second thread needs to wait for a space of time, you can also call `ProcessMessages(<nMilisecs>)` where `<nMilisecs>` is the time in milliseconds. If `<nMilisecs>` is 0 then it does not wait, and if it is -1 it waits indefinitely. The default value is 0. If you use `ProcessMessages()` in this way, there will not be any consume of CPU resources, and there will be no need to reactivate the thread calling `Resume()`, since it will automatically resume upon receipt of any message.

Synchronization between the main thread and the second thread

In addition to sending data or notifications between both sides of the thread, it is sometimes necessary to synchronize the execution of both, so that one of them wait for the other to arrive at a particular point or vice versa, the two do not coincide at a certain point program.

To match both threads at a given point, using the following methods:

- `WaitSignal([<nMilisecs>])` waits until a signal is given from the other thread. If the other thread sends the signal within the specified time (in milliseconds), the function returns the value `.T.`, while if it timed out without receiving the signal, the return value will be `.F.`. If `<nMilisecs>` is 0, then the method does not hold, and returns immediately returning values `.T.` o `.F.` depending on the signal has been issued by the other thread or not. If `<nMilisecs>` is -1, then the method will wait indefinitely. The default value is -1.
- `Signal()` emits a signal that will receive the other thread through the `WaitSignal()` method.

If, however, one wants to avoid both threads execute a piece of code at the same time (what is known as a critical section), using the following methods:

- `Lock([<nMilisecs>])` try to activate a lock within the specified time (in milliseconds). If the lock is achieved, the method returns `T.`, but if you have not achieved within the specified time (because the other thread has an active lock), then returns `F.` If `<nMilisecs>` is 0, then the method does not hold, and returns immediately returning values `.T.` o `.F.` depending on whether it has been able to lock or not. If `<nMilisecs>` is -1, then the method will wait indefinitely. The default value is -1 `<nMilisecs>`.
- `Unlock()` removes the lock that has been obtained with `Lock()`.

These two methods work much like the typical record locking on DBF files.

The debugger in multithreaded programs

Xailer Debugger has been modified to be able to debug multithreaded programs. The main thread can be debugged perfectly as before, but not the second ones. Any point of rupture or call `ALTD ()` to invoke the debugger is completely ignored. Perhaps in the future will be enlarged to support to second threads, but now there is this limitation.

On the other hand, when you invoke the debugger from the main thread, and there are other threads running, there is no guarantee that they will stop or continue running. Typically, if those threads are running PRG code (pcode), they will stop too, but not always will. The only thing that is certain is that if the second thread throws a notification to the main thread, they will stop his execution waiting for a reply, that on the other hand is stopped by the debugger. Thus, the second thread will be suspended until the main thread respond to the notification.

Multithreaded programming considerations

The multithreaded programming involves a series of circumstances that do not exist in the linear programming (race conditions, mutual exclusion, deadlocks, etc..). These potentially hazardous situations should be avoided as much as possible, trying to make the threads as independent as possible, and controlling the interaction between them through the mechanisms established for that purpose.

Whenever possible, avoid sharing variables and / or members of objects between threads. If we need to modify some value, and that the other thread know that value, it is best to use the notification system by `Notify()` and `OnNotify()`. In particular, there should never be modified `TThread` properties from the second thread. Likewise, the second thread receives the Application variables, `Screen`, `Printer` and `AppData`, but should be treated as read-only, and not to modify any of its properties. If you need to print from the second thread, you should create a second object of `TPrinter`.

If you need to share a variable or object between the main thread and the second thread, we have to create a critical section that allows us to modify that variable from one side each time. For this, we use the methods `Lock ()` and `Unlock ()` we have seen before. This can be useful, for example, modifying a value of `AppData`.

As a rule, from within a critical section you should not ever use other synchronization mechanisms and / or notification. Doing so may lead to a deadlock that is the situation that occurs when two threads are waiting for the other one to be unlocked. Eg suppose that from the main thread the `Lock()` method is called to start a critical section to update some variables. Earlier, the second thread had called `Lock()` and obtained the lock, so the main thread will wait until i gets the lock. Under these conditions, if the second thread calls the `Notify()` method it will not return until it has

been notified to the main thread, but it is stopped waiting to `Lock()`, so I can not answer. In the end, both lines have been blocked, waiting to one another indefinitely. The best way to avoid this situation is to make the critical sections as small and fast as possible, and never call another synchronization mechanisms and / or notification until it has come out of the critical section. On the other hand, it is convenient to use always a time limit for `WaitSignal()` methods and `Lock()` so that we can abort a task in the event of failure. In the same way, you should use no waiting notifications, setting the parameter `<lWait>` to `.F.`, whenever possible.

Should be avoided as much as possible the use of different synchronization mechanisms at the same time. Simultaneous use can easily lead to deadlock situations. Eg, when a thread calls its method `WaitSignal()` to wait for the other thread, if this second thread throws a notification `Notify()`, there will be a deadlock, since the first thread can not respond to the notification due it is waiting for a signal. When from a thread is executed a method of an object of another thread, it will be executed on the first execution space, so they will surely produce unwanted effects. Therefore, you should never execute code from another thread.

The GUI can not be used from a second thread. Should always be used from the main thread. When a second thread needs to show something to the user, it must notify the main thread, and this will be in charge of showing what the user needs. A typical case would be to increase a progress bar that shows the user the current state of a task. But the second thread should never access the progress bar. Instead, you must use the `Notify()` method, which will be collected by the main thread through the `OnNotify()` event and may increase the progress bar.

1.10.1 Harbour Multitasking

```
hb_threadStart( <@sStart()> | <bStart> | <cStart> [, <params, ...> ] ) -> <pThID>
```

Description:

Creates a new thread. Returns a pointer to the new created thread

Samples:

```
// using a function pointer
hb_ThreadStart( @function() )

// using a function name (it must be public)
hb_ThreadStart( "function" )

// using parameters
hb_ThreadStart( "function", 1, 2 3 )

// using codeblocks
hb_ThreadStart( {|| function() } )
```

```
hb_threadSelf() -> <pThID> | NIL
```

Description:

Returns the actual thread pointer. NIL is called from the main application thread.

```
hb_threadId( [ <pThID> ] ) -> <nThNo>
```

Description:

Returns the actual thread identifier (a number).

```
hb_threadJoin( <pThID> [, @<xRetCode> ] ) -> <LOK>
```

Description:

Waits until a thread is finished and on the second parameter which is passed by reference hold the its return value.

```
hb_threadDetach( <pThID> ) -> <LOK>
```

Description:

Indicates that the resources used by that thread must be liberated when the thread ends. It does not provoque the end of the thread, but onces used the hb_threadJoin() function can not be used.

```
hb_threadQuitRequest( <pThID> ) -> <LOK>
```

Description:

Tries to kill a thread.

```
hb_threadTerminateAll() -> NIL
```

Description:

Sends a QUIT message to all the second threads and waits untill all finish. It can only be called from the main application thread.

```
hb_threadWaitForAll() -> NIL
```

Description:

Waits until all the second threads finish. It can only be called from the main application thread.

```
hb_threadWait( <pThID> | <apThID>, [ <nTimeOut> ] ) => <nThInd> | <nThCount> | 0
```

Description:

Waits until all the selected threads finish (it can be an array of threads) . The second parameter sets the maximum wait time in seconds (decimals can be used). A value of -1 sets a indefinitely wait. Returns the first thread finished, the total number of threads finised or cero if no threads were closed.

```
hb_mutexCreate() -> <pMtx>
```

Description:

Permits to create a Mutex entity. The Valtype() of this variable is a pointer. A Mutex serves for varios purposes, but its main use of semaphore, but can also be used to subscribe to messages (signals).

```
hb_mutexLock( <pMtx> [, <nTimeOut> ] ) -> <lLocked>
```

Description:

Sets a Mutex as locked by the thread that called the function. This function only returns when the lock is achieved.

While a thread locks a Mutes any other thread that tries to lock the same mutex, will be blocked until the Mutex is unlocked. When the Mutes is unlocked, only one of the threads will gain access to the Mutex and the other threads will still wait. The second parameter sets the maximum time to wait in seconds (decimals can be used). A value of -1 or not setting the second parameters indicates to wait indefinitely.

```
hb_mutexUnlock( <pMtx> ) -> <lOK>
```

Description:

Unlocks a Mutex and makes it available for another thread

```
hb_mutexNotify( <pMtx> [, <xVal> ] ) -> NIL
```

Description:

Sends a notification. If one or more threads are waiting (subscribed), any of them is notified and leaves its wait state. Other threads subscribers are still waiting. There is no relationship between the subscription order and the order of notification. If no threads are waiting, the subscription will be pending in a queue and waiting for a thread to be subscribed. You can send information to the thread waiting for the subscription, using the second parameter of the function.

```
hb_mutexNotifyAll( <pMtx> [, <xVal> ] ) -> NIL
```

Description:

Sends a notification to all the waiting threads. If there are no waiting threads is like if the function has never been called. You can send information to the thread waiting for the subscription, using the second parameter of the function.

```
hb_mutexSubscribe( <pMtx>, [ <nTimeOut> ] [, @<xSubscribed> ] ) -> <lSubscribed>
```

Description:

Subscribes a thread to be advised of a notification. If there are pending notifications it exits immediately, if not it stuck waiting for a notification. You can specify a timeout (seconds), after which unlocks despite not receiving any notification.

The notifying thread, can submit information at the time of notification which will be received as the third parameter of the subscription function. Returns a boolean true if the notification occurred and false on end by timeout.

```
hb_mutexSubscribeNow( <pMtx>, [ <nTimeOut> ] [, @<xSubscribed> ] ) -> <lSubscribed>
```

Description:

The operation is similar to the function `Subscribe()`, the only difference is that before you begin, it clears any pending notification.

1.10.2 Futures or Promises

As of Xailer 7 you can use **Futures** that tremendously simplify multi-threaded programming, so its use is recommended.

According to Wikipedia, a future value (also called a future or a promise) is "A replacement for a result that is not yet available, usually because its computation has not yet finished, or its transfer over the network has not been completed." . It may seem like a somewhat cryptic definition, but it clearly indicates its functionality, although it does not mention how they work or what they are based on. The functionality of futures or promises is totally linked to multi-threaded programming, which can become tremendously complex due to the existing ease of making programming errors that crash our applications or make them act unexpectedly.

A future or promise is closely linked to the concept of asynchronous. That is, a code that we know when we execute it, but we don't know when it will return with a result. Let's think about a standard function; it always returns a result when the code inside it is finished executing. If said function has to access the Internet or perform heavy work, our application will be completely paralyzed until said code is finished. To avoid this problem, in classic multi-threaded programming we would create an asynchronous process through a thread:

```
TThread( ) :Run( bCode )
```

Where **bCode**, is for example, a block of code that will be executed when the thread starts.

Regardless of the processing time required by said code, the return from the execution of that line will be immediate. Therefore, it will be the responsibility of the programmer to create mechanisms to know when that code has finished and if it has been done correctly or not. The matter is complicated when, from said code that is executed in a second thread, we want to access memory areas that have not been created in that same thread, or even to access the screen in the broadest sense. The first rule of thumb to keep in mind is that the screen should never be accessed from a secondary thread and this is true for any development environment. It is not a limitation of Xailer, Harbour or Windows. Even if it seems to work, forget it, it's a real mirage, your application will end up with sporadic GPF sooner rather than later. The second rule of thumb is to avoid accessing memory areas created by the main thread or other threads, and if you do, establish the necessary mechanisms so that all the threads involved can access those same memory areas in an orderly manner and without blocking. In short, too many complications that prevent multi-threaded programming from being widely used.

Futures or promises tremendously simplify these inconveniences since they have mechanisms to solve these two problems, which are:

- Control of when the process in a second thread has terminated, either successfully or in error through a simple event that is fired when this occurs.
- Possibility of chaining asynchronous processes that are executed in cascade.
- Possibility of executing code in the main thread from the future itself, which allows, for example, to operate with the controls that are seen on the screen.

Xailer 7 incorporates several ways to implement futures:

- Via an **ASYNC** clause in the function or method definition
- Via estricta programación orientada a objetos

Using ASYNC:

Just add the **ASYNC** clause in the method or function definition. Something like that:

```
METHOD Btn1Click( oSender ) CLASS TForm1 ASYNC
```

Code that will be executed in the future must be included in **AWAIT** commands:

```
AWAIT INLINE { | |  
  FOR nFor := 1 TO 100  
    Sleep(10)  
  NEXT nFor  
  RETURN "Exit from first task"  
}
```

There can be more than one **AWAIT** statement in the same function or method. When the first **AWAIT** statement ends asynchronously, the execution of the next **AWAIT** code will proceed. This allows you to chain asynchronous processes in an extremely easy way.

```
FUNCTION MyTest(...) ASYNC  
  AWAIT INLINE { | |....}  
  AWAIT INLINE { | |....}  
  AWAIT INLINE { | |....}
```

Note that it is possible for each **AWAIT** statement to return a value, and furthermore it is also possible to know if the completed **AWAIT** has been successful or not. The private variable named

LastAwait picks up the task(TFutureTask) which has just finished processing. And therefore from the following **AWAIT** clause you can check if the task finished successfully and the return value:

```
LastAwait:nState (uncompleted, completed with value, completed with error)
LastAwait:ReturnValue
```

AWAIT supports defining code as an extended code block using the **AWAIT INLINE** expression. But you can also directly call a function with the following syntax:

```
AWAIT FUNCTION MyFunction( ... )
```

To control how and when the asynchronous process has finished you can create an additional **AWAIT INLINE** just to control it or simply use the **TFuture:OnComplete** event. If you're a little observant, you'll have noticed that apparently there is no variable that references that **TFuture** object that is mentioned. Xailer creates that variable for you with local scope when you use the **ASYNC** clause with the name **ThisFuture**. In fact, Xailer automatically creates a **TFuture** object in each function or method with the **ASYNC** clause and assigns it to said variable. Therefore, controlling the end of the asynchronous process would be as simple as doing something like this:

```
ThisFuture:OnComplete := { || Msginfo( LastAwait:ReturnValue ) }
```

We have already seen how to create and run asynchronous processes, even run with several cascading tasks, but we have not yet seen anything about how to run code on the main thread from the asynchronous task, such as any screen operation, which we have already seen. indicated that they cannot be done from asynchronous tasks. To do this, Xailer offers another command that is very easy to use and has a similar syntax to the **AWAIT** command, which is the **SYNCHRO** command. This command must only be used from **AWAIT INLINE** blocks or from **AWAIT FUNCTION** functions. For example:

```
AWAIT INLINE { ||
FOR nFor := 1 TO 100
  Sleep(10)
  SYNCHRO INLINE { || ::oProgressBar:nValue := nFor }
NEXT
RETURN "Exit from first task"
}
```

Notice how we update a progress bar from within the asynchronous task itself. Easier impossible. Just like the **AWAIT** command that can use a function with **AWAIT FUNCTION**, the **SYNCHRO** command also supports the **SYNCHRO FUNCTION** syntax in which I can develop all the complexity that the code might have.

Through object'oriented programming:

This is the recommended method if you have experience with Xailer or the object-oriented programming that Harbour offers. You will notice that it is basically the same as what we have explained so far, but without having to use a single command. The first step is to create an object of the **TFuture** class (operation carried out internally by the **ASYNC** clause):

```
LOCAL oFuture AS CLASS TFuture
oFuture := TFuture():New()
```

Then you have to add the **tasks** for said future. And we will do it with the **AddThreadTask**(bCode) method that receives as a parameter a block of code to be executed in said task:

```

LOCAL oTask AS CLASS TFutureTask
LOCAL bWork
bWork := { ||
FOR nFor := 1 TO 100
    Sleep(30)
NEXT
RETURN "Exit from first task"
}
oTask := oFuture:AddThreadTask( bWork )

```

In order to execute code in the main thread from the asynchronous task, we will use the **RunSynchroTask**(bCode) method and logically the calls to this method must be made from the task itself:

```

LOCAL oFuture AS CLASS TFuture
LOCAL oTask AS CLASS TFutureTask
LOCAL bWork
bWork := { ||
FOR nFor := 1 TO 100
    Sleep(30)
    oFuture:RunSynchroTask( {||::oProgressBar:nValue := nFor } )
NEXT
RETURN "Exit from first task"
}
oTask := oFuture:AddThreadTask( bWork )

```

Finally, to control the final result of the process:

```
oFuture:OnComplete := {|| .... }
```

An important point to make is the fact that asynchronous processes are **fired immediately** when the task is created. That is, **AWAIT INLINE** and its equivalent in POO **TFuture:AddThreadTask** (bCode) **do not wait**. They are responsible for launching the secondary thread. If there are several **AWAIT INLINES**, when they are executed, they are added to the list of tasks to be processed by the Future. If there are no pending tasks, the first **AWAIT INLINE** will be processed immediately. When this one finishes, the next one will be executed.

Everything can be a little more complicated, but not much. It is likely that a task will fail in execution. Is it possible to control it? It is! and in a very simple way. For that reason the object **ThisFuture** has an **OnError** event that receives the **TError** object and the **TFutureTask** as parameters. If from that event we return a true logical value, the next task in the future will be executed as if nothing had happened, otherwise, an execution error will be produced.

1.10.3 TThread

Main class for generating threads.

Hierarchy	TComponent descendant
File	\source\Thread.prg

1.10.3.1 TThread:Properties

■ Read only ■ Assignable □ Design assignable ■ Run-time assignable

Scope	Name	Type	Initial value
■	nStatus	Numeric	0

1.10.3.1.1 TThread:nStatus

Thread state.

Scope:	Read only		
Type:	Numeric		
Initial value:	0		
Possible values:	0	Finished	
	1	Running	
	2	Paused	
	-1	Stopped or waiting for termination	

1.10.3.2 TThread:Methods

■ Constructor ■ Standard □ Only after Create()

Type	Name
□	Lock
□	Notify
□	NotifyThread
□	Pause
□	Quit
□	Resume
□	Run
□	Signal
□	Stop
□	Unlock
□	WaitSignal

1.10.3.2.1 TThread:Lock

try to activate a lock within the specified time (in milliseconds). If the lock is achieved, the method returns `T.`, but if you have not achieved within the specified time (because the other thread has an active lock), then returns `F.` If `<nMilisecs>` is 0, then the method does not hold, and returns immediately returning values `.T.` or `.F.` depending on whether it has been able to lock or not. If `<nMilisecs>` is -1, then the method will wait indefinitely. The default value is -1 `<nMilisecs>`.

Type	Only after Create
Parameters	<code><nMilisecs></code>

	Milliseconds
Return value	True if success

1.10.3.2.2 TThread:Notify

Sends data from second thread to main thread.

Type	Only after Create
Parameters	<p>[<IWait>]: Indicates whether the call waits for the main thread response. If is <code>.T.</code>, then it will not return until the main thread has answered to its <code>OnNotify()</code> event. If is <code>.F.</code>, the method returns immediately. The default value is <code>.F.</code></p> <p>[<nValue>]: Is a numeric value (unsigned 32 bits) you want to send</p> <p>[<uData>]: Is any valid data in Harbour (a single value, a string, an array, an object, etc.)</p>
Return value	NIL

1.10.3.2.3 TThread:NotifyThread

Sends data from the main thread to the second thread.

Type	Only after Create
Parameters	<p>[<IWait>]: Indicates whether the call waits for the main thread response. If is <code>.T.</code>, then it will not return until the main thread has answered to its <code>OnNotifyThread()</code> event. If is <code>.F.</code>, the method returns immediately. The default value is <code>.F.</code></p> <p>[<nValue>]: Is a numeric value (unsigned 32 bits) you want to send</p> <p>[<uData>]: Is any valid data in Harbour (a single value, a string, an array, an object, etc.)</p>
Return value	NIL

1.10.3.2.4 TThread:Pause

Stops the thread execution, which will be on pause until you restart its execution or abandon it completely.

Type	Only after Create
Parameters	[<IWait>]:

	Indicates whether the call waits for a response. The default value is <code>.F.</code>
Return value	NIL

1.10.3.2.5 TThread:Quit

Leaves the thread execution immediately. This is done using the QUIT command within the thread, causing the complete closure of the VM corresponding to this thread and will have no ability to control the output thereof.

Type	Only after Create
Parameters	None
Return value	NIL

1.10.3.2.6 TThread:Resume

Restarts execution of a paused thread.

Type	Only after Create
Parameters	[<IWait>]: Indicates whether the call waits for a response. The default value is <code>.F.</code>
Return value	NIL

1.10.3.2.7 TThread:Run

Starts thread execution. The Run() method starts the thread and returns immediately. Since the new created thread may need some time to start, there is no guarantee that from returning from the method, the new thread is created and running.

Type	Only after Create
Parameters	<uCode, ...>: It can be a codeblock, a function pointer, a function name, and you can pass any parameter as necessary
Return value	pThread

1.10.3.2.8 TThread:Signal

Emits a signal that will receive the other thread through the `WaitSignal()` method. For further information consult the function `hb_MutexNotify()` Harbour multitasking chapter.

Type	Only after Create
Parameters	None
Return value	NIL

1.10.3.2.9 TThread:Stop

Leaves the thread execution. This abandonment occurs through a `BREAK` statement, so that if you use a `BEGIN SEQUENCE` / `END SEQUENCE` within the thread code, we have the opportunity to maintain control of the thread output, eg release resources used.

Type	Only after Create
Parameters	[<IWait>]: Indicates whether the call waits for a response. The default value is <code>.F.</code>
Return value	NIL

1.10.3.2.10 TThread:Unlock

Deletes the lock obtained through the `Lock()` method.

Type	Only after Create
Parameters	None
Return value	True if success

1.10.3.2.11 TThread:WaitSignal

Waits until a signal is given from the other thread. If the other thread sends the signal within the specified time (in milliseconds), the function returns the value `.T.`, while if it timed out without receiving the signal, the return value will be `.F.`. If `<nMilisecs>` is 0, then the method does not hold, and returns immediately returning values `.T.` or `.F.` depending on the signal has been issued by the other thread or not. If `<nMilisecs>` is -1, then the method will wait indefinitely. The default value is -1. For further information consult the function `hb_MutexSubscribe()` on Harbour Multitasking chapter.

Type	Only after Create
Parameters	<nMilisecs>

	Milliseconds
Return value	True if success

1.10.3.3 TThread:Events

Name
OnEnd
OnNotify
OnNotifyThread
OnPause
OnResume
OnStart

1.10.3.3.1 TThread:OnEnd

Event triggered when the thread terminates, either because the code has reached the end or when you call the methods [Stop\(\)](#) or [Quit\(\)](#).

Parameters:	<oSender> : Object that triggers the event.
Return value:	NIL

1.10.3.3.2 TThread:OnNotify

Event received on main thread when second threads sends a notification.

Parameters:	<oSender> : Object that triggers the event <nValue> : Numeric value (32 bits unsigned) sent from the Notify() method of second thread <uData> : <uData> value sent from the Notify() method of second thread
Return value:	NIL

1.10.3.3.3 TThread:OnNotifyThread

Event received on second thread from main thread through a notification.

Parameters:	<oSender> : Object that triggers the event
--------------------	--

<nValue>:	Numeric value (32 bits unsigned) sent from the <code>NotifyThread()</code> on main thread
<uData>:	<code><uData></code> value sent from the <code>NotifyThread()</code> method of main thread
Return value:	NIL

1.10.3.3.4 TThread:OnPause

Event triggered when thread goes to pause state using the `Pause` method.

Parameters:	<oSender>: Object that triggers the event.
Return value:	NIL

1.10.3.3.5 TThread:OnResume

Event triggered when the thread restarts from a pause state through the `Resume` method.

Parameters:	<oSender>: Object that triggers the event.
Return value:	NIL

1.10.3.3.6 TThread:OnStart

Event triggered when the thread starts its execution.

Parameters:	<oSender>: Object that triggers the event.
Return value:	NIL

1.11 TFuture

Main class for the use of Futures. See introduction to futures and promises in this chapter for more information.

Hierarchy TComponent descendant

File \source\Future.prg

1.11.1 TFuture:Properties

■ Read only ■ Assignable ■ Design assignable ■ Run-time assignable

Scope	Name	Type	Initial value
■	aThreadTasks	TFutureTask[]	{}
■	LastAwaitTask	TFutreTask	NIL
■	nState	Numeric	ftUNCOMPLETED

1.11.1.1 TFuture:aThreadTasks

TFutureTask objects array.

Scope:	read only
Type:	TFutureTask array
Initial value:	{}

1.11.1.2 TFuture>LastAwaitTask

last TFutureTask object proccsed.

Scope:	read only
Type:	Objeto TFutureTask
Initial value:	NIL

1.11.1.3 TFuture:nState

TFuture state.

Scope:	read only
Type:	Numeric
Initial value:	0
Possible values:	0 ftUNCOMPLETED 1 ftCOMPLETEDWITHVALUE 2ftCOMPLETEDWITHERROR

1.11.2 TFuture:Methods

■ Constructor ■ Standard ■ Only after Create()

Type	Name
■	AddThreadTask
■	RunSynchroTask / Synchronize
■	CreateFrom
■	CurrentTask
■	Stop
■	TaskCount

1.11.2.1 TFuture:AddThreadTask

Adds a task to TFuture.

Type	Only after create
Parameters	<bCode> Code-block to be executed
Return value	TFutureTask o NIL

1.11.2.2 TFuture:RunSynchroTask

Run a synchronous task immediately from the code of any TFutureTask .

Type	Only after create
Parameters	<bCode> synchronous code-block to execute
Return value	True if success

Ejemplo:

```

LOCAL oFuture AS CLASS TFuture
LOCAL oTask AS CLASS TFutureTask
LOCAL bWork
bWork := { ||
FOR nFor := 1 TO 100
  Sleep(30)
  oFuture:RunSynchroTask( { || ::oProgressBar:nValue := nFor } )
NEXT
RETURN "Exit from first task"
}
oTask := oFuture:AddThreadTask( bWork )

```


1.11.2.3 TFuture:CreateFrom

Fast constructor with basic parameters.

Type	Constructor
Parameters	<bTask> Code-block to run [<bComplete>] Code-block to run when completed [<bError>] Code-block to run in case of error
Return value	Reference to Self

1.11.2.4 TFuture:CurrentTask

Ordinal on aThreadTasks array which is processing.

Type	Only after create
Parameters	None
Return value	Ordinal number

1.11.2.5 TFuture:Stop

Suspends TFuture tasks.

Type	Only after create
Parameters	[<IForced>] By default .T. Ends TFuture processing . If IForced is true, causes the thread to be destroyed. Otherwise it simply waits for the current TFutureTask to finish.
Return value	NIL

1.11.2.6 TFuture:TaskCount

Returns the number of tasks on the TFuture object.

Type	Only after create
Parameters	None
Return value	Total tasks

1.11.3 TFuture:Events

Name
OnComplete
OnCompleteTask
OnError

1.11.3.1 TFuture:OnComplete

Event fired when the TFuture is completely executed.

Parameters:	<oSender>: Object that triggers the event. <ReturnValue>: Return value of the last TFutureTask processed
Return value:	NIL

1.11.3.2 TFuture:OnCompleteTask

Event fired when a TFutureTask is completed.

Parameters:	<oSender>: Object that triggers the event. <ReturnValue>: Return value from the TFutureTask <nTask>: Ordinal number in aThreadTasks
Return value:	NIL

1.11.3.3 TFuture:OnError

Event fired when a execution error is produced on TFutureTask task.

Parameters:	<oSender>: Object that triggers the event. <oError>: TError object
Return value:	If it returns a logical value of True, the error is ignored and the next task continues or the execution of the future is terminated.

1.12 TFutureTask

Class that represents each of the tasks that an object can execute in cascade by a TFuture.

Hierarchy TObject descendant
File \source\Future.prg

1.12.1 TFutureTask:Properties

■ Read only ■ Assignable ■ Design assignable ■ Run-time assignable

Scope	Name	Type	Initial value
■	nState	Numeric	ftUNCOMPLETED
■	oError	TError	NIL
■	oParent	TFuture	NIL
■	ReturnValue	Any	NIL

1.12.1.1 TFutureTask:nState

TFutureTask state.

Scope:	read only
Type:	Numeric
Initial value:	0
Possible values:	0 ftUNCOMPLETED 1 ftCOMPLETEDWITHVALUE 2 ftCOMPLETEDWITHERROR

1.12.1.2 TFutureTask:oError

TError object generated when an error is produced.

Scope:	read only
Type:	TError
Initial value:	NIL

1.12.1.3 TFutureTask:oParent

A reference to its TFuture owner.

Scope:	read only
---------------	-----------

Type:	TFuture
Initial value:	NIL

1.12.1.4 TFutureTask:ReturnValue

Returned value by the task when is completed.

Scope:	read only
Type:	Any
Initial value:	NIL

1.13 Animations

Animations are classes that allow you to set different types of animations for any visual control. Its operation is based on the use of Futures so the application does not freeze for the entire duration of the animation. For more information about Futures, see the chapter about Multi-tasking.

Demo of animations on Xailer's YouTube channel

The base class for all animations is TAnimation which offers all the necessary functionality to perform any type of animation. There are two classes that inherit from TAnimation that are dedicated to perform animation in a very easy way:

- **TAniNumProperty:** This class allows you to set the animation based on one of the properties of the control. .

Propiedades	Eventos
Cargo	
cProperty	nWidth
cVarName	oAniNumProperty1
lAutoReverse	.F.
lEnabled	.F.
lForceRefresh	.F.
lLoop	.F.
lReverse	.F.
lStartFromCurrent	.T.
nDelay	0
nDuration	1000
nInterpolation	aiELASTIC_OUT
nStartValue	0
nStopValue	300
nSyncsInSec	20
oControl	oEdit1

In this example you can perfectly understand their behavior. The object TAniNumProperty controls the property '**nWidth**' of the control '**oEdit1**' and it will modify this property from its current value (**lStartFromCurrent** to true) up to a value of 300 (**nStopValue**) and will perform it in a period of 1

second (**nDuration**). The entire process will begin as soon as the property **IEnabled** is set to true or execute its method **Start()**. As you can see, it is so easy to set up an animation in any control.

- **TAniControlSize**: This class allows you to set the animation based on the dimensions of the control. That is, its size defined by its properties: nLeft, nTop, nWidth and nHeight. Its use is basically restricted to resizing any control from an initial position and size to a final position and size.

Propiedades	Eventos
Cargo	
aStartValues	{ 0,0,0,0 }
aStopValues	{ 96,48,300,200 }
cVarName	oAniControlSize1
IAutoReverse	.F.
IEnabled	.F.
IForceRefresh	.F.
ILoop	.F.
IReverse	.F.
IStartFromCurrent	.T.
nDelay	0
nDuration	1000
nInterpolation	aiLINEAR
nSyncsInSec	20
oControl	oMemo1

In this example you can perfectly understand their behavior. The object TAniControlSize controls the dimensions of the control 'oMemo1' and it will modify these dimensions from their current value (**IStartFromCurrent** a True) up to a value of {96,48,300,200} (**aStopValues**) and will perform it in a period of 1 second (**nDuration**). The entire process will begin as soon as the property **IEnabled** is set to true or execute its method Start().

Both classes also have a number of very interesting properties, which are:

- **nInterpolation**: This property allows you to set the type of animation. The most basic form would be a linear interpolation (aiLINEAR) which is the only one existing in Xailer Personal and Xailer Professional. Xailer Enterprise has many more types: bounce, explosion, elastic and circular.
- **nDelay**: Allows you to set a delay in milliseconds before the start of the animation. This property is very useful when the **IEnabled** property is set to true in the IDE and therefore it is necessary that the form is loaded and displayed before the animation is executed.
- **ILoop**: If this property is set to true the animation will start again automatically when it finishes. And therefore the animation will continue until **IEnabled** becomes false.
- **IReverse**: If this property is set to true, the values of 'Start' and 'Stop' are exchanged.
- **IAutoReverse**: This property allows an automatic rollback to be performed at the end of a cycle, so that in the next cycle the opposite effect becomes true. This property in conjunction with the ILoop property set to true causes the classic zoom in- zoom out effect.
- **nSyncsInSec**: This property allows you to set the number of synchronizations to be made of the control per second. The higher this number is, in principle, the smoother and more fluid the movement will appear, but it's not like that. It depends on the speed of your PC and the work the animation has to do, as it may drag to other controls that are also visible.

1.13.1 TAnimation

Main class for the use of Animations. It is rare that you will need to use this class directly. It is usual to use one of its descendants: TAniNumProperty or TAniControlSize.

Hierarchy TComponent descendant
Files \source\Animations.prg
 \source\Animations2.prg

1.13.1.1 TAnimation:Propiedades

■ Read only ■ Assignable ■ Design assignable ■ Run-time assignable

Scope	Name	Type	Initial value
■	IAutoReverse	Logical	.F.
■	IEnabled	Logical	.F.
■	ILoop	Logical	.F.
■	IReverse	Logical	.F.
■	IStartFromCurrent	Logical	.F.
■	nDelay	Numeric	0
■	nDuration	Numeric	200
■	nInterpolation	Numeric	aiLINEAR
■	nSyncsInSec	Numeric	10

1.13.1.1.1 TAnimation:IAutoReverse

If true, it reverses the order of the values calculated for the animation for each execution cycle.

This property allows an automatic reversion to be made when a loop ends, so that in the next loop the opposite effect is performed. This property together with the property ILoop to true causes the classic zoom in- zoom out effect.

Scope:	Assignable
Type:	Logical
Initial value:	.F.

1.13.1.1.2 TAnimation:IEnabled

If true, start the animation.

Scope:	Assignable
---------------	------------

Type:	Logical
Initial value:	.F.

1.13.1.1.3 TAnimation:ILoop

If true, the animation is run continuously.

Scope:	Assignable
Type:	Logical
Initial value:	.F.

1.13.1.1.4 TAnimation:IReverse

If true, reverses the order of the values calculated for the animation. The start becomes the end and vice versa.

Scope:	Assignable
Type:	Logical
Initial value:	.F.

1.13.1.1.5 TAnimation:IStartFromCurrent

If true, the animation starts from the last processed position and not from the initial values it might have.

Scope:	Assignable
Type:	Logical
Initial value:	.F.

1.13.1.1.6 TAnimation:nDelay

Set a delay time in milliseconds before the start of the animation.

Scope:	Assignable
Type:	Numeric
Initial value:	0

1.13.1.1.7 TAnimation:nDuration

Animation duration in milliseconds.

Scope:	Assignable
Type:	Numeric
Initial value:	200

1.13.1.1.8 TAnimation:nInterpolation

This property allows you to set the type of animation. The most basic form would be a linear interpolation (**aiLINEAR**) which is the only one available in Xailer Personal and Xailer Professional. Xailer Enterprise has many more types: bounce, explosion, elastic and circular.

Scope:	Assignable
Type:	Numeric
Initial value:	aiLINEAR
Valores posibles:	aiLINEAR aiBOUNCE_IN aiBOUNCE_OUT aiEXPO_IN aiEXPO_OUT aiELASTIC_IN aiELASTIC_OUT aiCIRCLE_IN aiCIRCLE_OUT

1.13.1.1.9 TAnimation:nSynchInSec

This property allows you to set the number of synchronizations to be made of the control per second. The higher this number is, in principle, the smoother and more fluid the movement will appear, but it's not like that. It depends on the speed of your PC and the work the animation has to do, as it may drag to other controls that are also visible.

Scope:	Assignable
Type:	Numeric
Initial value:	10

1.13.1.2 TAnimation:Methods

■ Constructor ■ Standard ■ Only after Create()

Type	Name
■	Start
■	Stop

1.13.1.2.1 TAnimation:Start

Starts the animation.

Type	Only after Create
Parámetros	None
Return value	True if success

1.13.1.2.2 TAnimation:Stop

Stops the animation.

Type	Only after Create
Parámetros	None
Return value	True if success

1.13.1.3 TAnimation:Events

Name
OnFinish
OnProcess
OnStart

1.13.1.3.1 TAnimation:OnFinish

Event that is fired when the complete execution of the animation is finished.

Parameters:	<oSender>: Object that fires the event.
Return value:	NIL

1.13.1.3.2 TAnimation:OnProcess

Event that is fired at each repetition of the animation

Parameters:	<oSender>: Object that fires the event.
Return value:	NIL

1.13.1.3.3 TAnimation:OnStart

Event that is fired when the animation execution starts.

Parameters:	<oSender>: Object that triggers the event.
Return value:	NIL

1.13.2 TAniNumProperty

This class allows to set the animation based on one of the properties of any visual control. The properties it adds to the TAnimation instance are the following:

■ Read only ■ Assignable ■ Design assignable ■ Run-time assignable

Scope	Name	Type	Initial value
■	cProperty	Character	""
■	IForceRefresh	Logical	.F.
■	INcaRefresh	Logical	.F.
■	nStartValue	Numeric	0
■	nStopValue	Numeric	0
■	oControl	Object	NIL

1.13.2.1 TAniNumProperty:cProperty

Name of the property of the **oControl** object to be modified in the animation.

Scope:	Assignable
Type:	Character
Initial value:	""

1.13.2.2 TAniNumProperty:IForceRefresh

If true, a call to the **Refresh()** method of the **oControl** control is triggered after each animation repetition. It is only necessary to set it to true if you notice that the control is not updating correctly.

Scope:	Assignable
Type:	Logical
Initial value:	.F.

1.13.2.3 TAniNumProperty:INcaRefresh

If true, a call to the **Refresh()** method of the **oControl** no client area control is triggered after each animation repetition. It is only necessary to set it to true if you notice that the control is not updating correctly.

Scope:	Assignable
Type:	Logical
Initial value:	.F.

1.13.2.4 TAniNumProperty:nStartValue

cProperty initial value from the oControl property when animation starts.

Scope:	Assignable
Type:	Numeric
Initial value:	0

1.13.2.5 TAniNumProperty:nStopValue

cProperty end value of the oControl when animations ends.

Scope:	Assignable
Type:	Numeric
Initial value:	0

1.13.2.6 TAniNumProperty:oControl

Control in scope to be animated.

Scope:	Assignable
Type:	Objeto
Initial value:	NIL

1.13.3 TAniControlSize

This class allows you to set the animation based on the dimensions of the control. In other words, its size is determined by its properties: nLeft, nTop, nWidth and nHeight. Its use is basically restricted to resizing any control from an initial position and size to a final position and size. The properties it adds to the TAnimation instance are the following:

■ Read only ■ Assignable ■ Design assignable ■ Run-time assignable

Scope	Name	Type	Initial value
■	IForceRefresh	Logical	.F.
■	INcaRefresh	Logical	.F.
■	aStartValues	Array	{0,0,0,0}
■	aStopValues	Array	{0,0,0,0}
■	oControl	Objeto	NIL

1.13.3.1 TAniControlSize:IForceRefresh

If true, a call to the **Refresh()** method of the **oControl** control is triggered after each animation repetition. It is only necessary to set it to true if you notice that the control is not updating correctly.

Scope:	Assignable
Type:	Logical
Initial value:	.F.

1.13.3.2 TAniControlSize:INcaRefresh

If true, a call to the **Refresh()** method of the **oControl** no client area control is triggered after each animation repetition. It is only necessary to set it to true if you notice that the control is not updating correctly.

Scope:	Assignable
Type:	Logical
Initial value:	.F.

1.13.3.3 TAniControlSize:aStartValues

Array with the initial values of the control oControl coordinates.

Scope:	Assignable
Type:	Matriz
Initial value:	{0,0,0,0}

1.13.3.4 TAniControlSize:oControl

Control in scope to be animated.

Scope:	Assignable
Type:	Objeto
Initial value:	NIL

1.14 TAniControlSize:aStopValues

Array with the endl values of the control oControl coordinates

Scope:	Assignable
Type:	Matriz
Initial value:	{0,0,0,0}

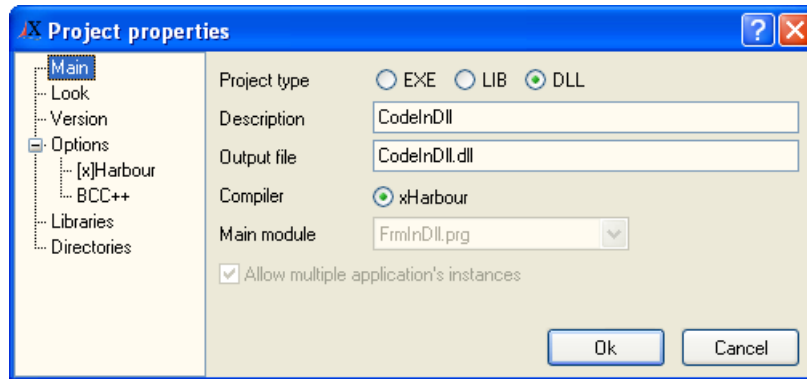
1.15 DLL creation

After Xailer 2 is possible to create DLLs, dynamic link libraries. As its name says, they are libraries that can be loaded on run time by the main application. When loaded, all its functions will be ready to use. The main benefit of the use of DLLs is that you can break any of your applications into different modules, so you can deliver only the modules you want and in case you need to send an update of the program you only need to send the modified or new modules.

The DLL libraries created by Xailer are libraries that only contain pcode (intermediate code) to be executed by the main application virtual machine. Therefore, this kind of libraries can only be used from programs written in Xailer.

The DLL libraries can have resources inside, like the main executable file. By default, the application will search for resources on the executable, but if the resource is not found, it will continue searching on all the loaded DLL libraries and in the same order they have been loaded.

To create a DLL library is necessary to indicate so on the Project properties dialog:



For the correct creation of the DLL is necessary the existence of the library XailerDLL.lib on the \lib directory.

In order to access to any DLL function from the main application you must load the library with the function **HB_LibLoad**(<CDLLFile>) which returns a pointer to that library, that it will be a non null value if the operation has succeeded.

If you need to unload the library you may use the function **HB_LibFree**(<nHandle>) which receives as unique parameter the pointer obtained by the previous function.

To access any function or class on that DLL library is necessary to indicate the compiler that the function will be loaded dynamically and you must the use the xHarbour **DYNAMIC** directive, in the same way you may use the REQUEST directive.

Here is a little sample showing its use:

```

DYNAMIC TFormInDLL

.....
.....

//-----

METHOD Button1Click( oSender ) CLASS TForm1

    LOCAL nHandle

    nHandle := HB_LibLoad( "CodeInDll.dll" )

    TFormInDLL():New(Self):ShowModal()

    HB_LibFree( nHandle )

RETURN Nil

```

You may find a complete sample of DLLs us on Xailer\Samples\Dlls directory.

Limitations:

However, the use of DLL libraries have some limitations that you should know:

- Is possible that when you load the library with the function `HB_LibLoad()` the debugger message window is displayed showing you a list of functions not found. Obviously, this windows is only shown when executed from the IDE. This list of functions, it refers, to basic low level functions (written in 'C') that presumably will be used by the DLL, but they are missing (or not exported) on the main executable. In case this functions are called from code inside the DLL a GPF (general protection fault error) is automatically produced. The only way to avoid this problem is to force the link of those functions and the classic `REQUEST` directive is not valid, since that only works for high level code. You must create dummy functions on the executable that call those missing functions. **This limitation only affects to your own 'C' routines written on the DLL itself. If you have not created any low level function inside the DLL you will not have any problem, even if the message with the missing functions is shown.**
- All the **high level functions** that you use inside the DLL must be linked on the main executable or developed on the library itself, on contrary you will receive the typical run time error message: 'Function not found'. To force that all these functions get linked on the final executable you must use the `xHarbour REQUEST` directive.
- You may overload any PRG function that exists on the main executable but it will only be executed when its called from inside the DLL.
- You may overload more than once the same PRG function on more than one DLL, but on that case, from the main executable, only the first loaded function will be executed.

1.16 Functions

In Xailer you will find few functions due it tries to offer an extensive use of the Object Oriented Programming. However, there are functions that due its simplicity it makes no sense to promote them to a class and in other cases they are function that only access to the Windows API function. For this kind of functions the documentation provided is basically the parameters they receive and the return values and suggest to the programmer to check the Windows API documentation to get more information.

Those functions need a deeper Windows API knowledge and however they are reserved for advanced users. You might not need to use those functions, but they are available if you want to use them.

1.16.1 Arithmetical bit operation functions

Name
Shl
Shr
lAnd
lOr
nAnd
nExclude
nNot
nOr
nXor

Windows API function accessible from Xailer:

- HIBYTE(nValue) --> nHi
- HILONG(nValue) --> nHi
- HIWORD(nValue) --> nHi
- LOBYTE(nValue) --> nLo
- LOLONG(nValue) --> nLo
- LOWORD(nValue) --> nLo
- MAKELONG(nLo, nHi) --> nLong
- MAKELONGLONG(nLo, nHi) --> nLongLong
- MAKELPARAM(nLo, nHi) --> nLparam
- MAKEWORD(nLo, nHi) --> nWord

1.16.1.1 Shl

Left shift bit operation.

Category:	Arithmetical bit operations
Parameters:	<nValue1>, ..., <nValueN> Values to process
Return value:	<nValue>
Module:	Core.c
See also:	Shr

1.16.1.2 Shr

Right shift bit operation.

Category:	Arithmetical bit operations
Parameters:	<nValue1>, ..., <nValueN> Values to process
Return value:	<nValue>
Module:	Core.c
See also:	Shl

1.16.1.3 IAnd

AND boolean operation over numeric values.

Category:	Arithmetical bit operations
Parameters:	<nValue1>, ..., <nValueN> Values to process
Return value:	<IValue>

Module:	Core.c
See also:	IOr

1.16.1.4 IOr

OR boolean operation over numeric values.

Category:	Arithmetical bit operations
Parameters:	<nValue1>, ..., <nValueN> Values to process
Return value:	<IValue>
Module:	Core.c
See also:	IAnd

1.16.1.5 IXOr

XOR boolean operation over numeric values.

Category:	Arithmetical bit operations
Parameters:	<nValue1>, ..., <nValueN> Values to process
Return value:	<IValue>
Module:	Core.c
See also:	IOr

1.16.1.6 nAnd

AND boolean operation over numeric values.

Category:	Arithmetical bit operations
Parameters:	<nValue1>, ..., <nValueN> Values to process
Return value:	<nValue>
Module:	Core.c
See also:	nOr, nNot, nExclude, nXor

1.16.1.7 nExclude

EXCLUDE numeric operation over numeric values. (it is the opposite than an nOr operation).

Category:	Arithmetical bit operations
Parameters:	<nValue1>, ..., <nValueN> Values to process
Return value:	<nValue>
Module:	Core.c
See also:	nOr, nAnd, nNot, nXor

1.16.1.8 nNot

NOT boolean operation over numeric values.

Category:	Arithmetical bit operations
Parameters:	<nValue1>, ..., <nValueN> Values to process
Return value:	<nValue>
Module:	Core.c
See also:	nOr, nAnd, nExclude, nXor

1.16.1.9 nOr

OR boolean operation over numeric values.

Category:	Arithmetical bit operations
Parameters:	<nValue1>, ..., <nValueN> Values to process
Return value:	<nValue>
Module:	Core.c
See also:	nAnd, nExclude, nNot, nXor

1.16.1.10 nXor

XOR boolean operation over numeric values.

Category:	Arithmetical bit operations
Parameters:	<nValue1>, ..., <nValueN> Values to process

Return value:	<nValue>
Module:	Core.c
See also:	nAnd, nExclude, nNot, nOr

1.16.2 Clipboard functions

Windows API function accessible from Xailer:

- ChangeClipboardChain(hWnd1, hWnd2) --> ISuccess
- CloseClipboard(hWnd) --> ISuccess
- CountClipboardFormats() --> nFormats
- EmptyClipboard(hWnd) --> ISuccess
- EnumClipboardFormats() --> aFormats
- GetClipboardData(nFormat) --> xValue
- GetClipboardFormatName(nFormat) --> cFormat
- GetClipboardOwner() --> hWnd
- GetClipboardSequenceNumber() --> hWnd
- GetClipboardViewer() --> hWnd
- GetOpenClipboardWindow() --> hWnd
- IsClipboardFormatAvailable(nFormat) --> IValue
- OpenClipboard(hWnd) --> ISuccess
- RegisterClipboardFormats(cFormat) --> nFormat
- SetClipboardData(nFormat, [cText | Handle])

1.16.3 Communications functions

Name
MySQL_Ping
SendWMCopyData

Windows API function accessible from Xailer:

- MakeIPAddress(n1, n2, n3, n4) --> IParam
- MakeIPRange(nLo, nHi) --> IParam

1.16.3.1 SendWMCopyData

Permits the communication between processes sending text messages between different applications.

Category:	Communications functions
Parameters:	<hWnd cWindowTitle> Window handle or window caption <cString> Text message [<nMsgId>]

	Message identifier
Return value:	<Nil>
Module:	Core.c
See also:	TForm:OnCopyData event

1.16.3.2 MySql_Ping

Checks MySql communications and may avoid disconnections due timeouts.

Category:	Communications functions
Parameters:	<hWnd> Window handle
Return value:	<nValue> 0: Connection alive or error
Module:	Not available
See also:	MySql info

1.16.4 Conversion functions

Name
AnsiToWide
Atranspose
DtoSql
DTToSql
Hex
HexToDec
HexToString
RGBInverse
RGBToHTML
StringToHex
StrMySql
StrSql
StrSQLite
WideToAnsi
W2Hex
XA_DateTimeZ2Local
XA_UrlEncode
XA_UrlDecode
XA_UrlDecodeToAnsi

Windows API function accessible from Xailer:

- AnsiToOem(cText)--> cText

- AnsiToUTF8(cText)--> cText
- B2Hex(nBit)--> cHex
- GetBValue(nRGB)--> nBlue
- GetGValue(nRGB)--> nGreen
- GetRValue(nRGB)--> nRed
- L2Hex(nLong)--> cHex
- LL2Hex(nLongLong)--> cHex
- OemToAnsi(cText)--> cText
- UTF8ToAnsi(cText)--> cText

1.16.4.1 AnsiToWide

Converts a ANSI string type to UNICODE format.

Category:	Conversion
Parameters:	<cAnsi>
Return value:	<cUnicode>
Module:	Util.c
See also:	WideToAnsi

1.16.4.2 Atranspose

Transposes an array.

Category:	Conversion
Parameters:	<aArray>
Return value:	<aArray>
Module:	Util.c

1.16.4.3 DtoSql

Converts a date value to a string type recognized by most SQL engines (AAAA-MM-DD) .

Category:	Conversion
Parameters:	<dDate>
Return value:	<cSqlDate>
Module:	Dataset.prg

1.16.4.4 DToSql

Converts a date-time value to a string type recognized by most SQL engines (AAAA-MM-DD HH:MM:SS).

Category:	Conversion
Parameters:	<tDateTime> Date-time
Return value:	<cSqlDateTime>
Module:	Util.prg
See also:	DToSql

1.16.4.5 Hex

Converts a number to hexadecimal format.

Category:	Conversion
Parameters:	<nValue>
Return value:	<cHex>
Module:	Util.c
See also:	HexToDec, HexToString, StringToHex, W2Hex

1.16.4.6 HexToDec

Converts an hexadecimal format number (saved in a string) to numeric format.

Category:	Conversion
Parameters:	<cHex>
Return value:	<nDec>
Module:	Util.c
See also:	Hex, HexToString, StringToHex, W2Hex

1.16.4.7 HexToString

Converts an hexadecimal format number (saved in a string) to decimal format as a string.

Category:	Conversion
Parameters:	<cHex>
Return value:	<cDec>

Module:	Util.c
See also:	Hex, HexToDec, StringToHex, W2Hex

1.16.4.8 RGBInverse

Gets the inverse color.

Category:	Conversion
Parameters:	<nColor> Color to convert
Return value:	<nInvColor> Intense color
Module:	Choosecolor.prg

1.16.4.9 RGBToHTML

Converts a RGB color format in HTML format.

Category:	Conversion
Parameters:	<nColor> Color to convert
Return value:	<cHtmlColor> Color in HTML format
Module:	Choosecolor.prg

1.16.4.10 StringToHex

Converts a number in decimal format (saved in a string) to hexadecimal format in a string.

Category:	Conversion
Parameters:	<cDec>
Return value:	<cHex>
Module:	Util.c
See also:	Hex, HexToDec, HexToString, W2Hex

1.16.4.11 StrMySql

Modifies the string to convert non allowed characters on SQL instructions under MySQL.

Category:	Conversion
Parameters:	<cString> String to convert
Return value:	<cString> String converted
Module:	Datasource.prg
See also:	StrSql

1.16.4.12 StrSql

Modifies the string to convert non allowed characters on SQL instructions.

Category:	Conversion
Parameters:	<cString> String to convert [<cCharsToEscape>] Characters to convert. By default "\\\""
Return value:	<cString> String converted
Module:	Datasource.prg
See also:	StrMySql, StrSQLite

1.16.4.13 StrSqlLite

Modifies the string to convert non allowed characters on SQL instructions under SQLite.

Category:	Conversion
Parameters:	<cString> String to convert
Return value:	<cString> String converted
Module:	Datasource.prg
See also:	StrSql

1.16.4.14 WideToAnsi

Converts a UNICODE string to ANSI format.

Category:	Conversion
Parameters:	<cUnicode>
Return value:	<cAnsi>
Module:	Util.c
See also:	AnsiToWide

1.16.4.15 W2Hex

Converts a WORD number type to hexadecimal format.

Category:	Conversion
Parameters:	<nValue>
Return value:	<cHex>
Module:	Util.c
See also:	Hex, HexToDec, HexToString, StringToHex

1.16.4.16 XA_DateTimeZ2Local

Converts a string with the format 0000-00-00T00:00:00.000Z with UTC date/time to DateTime value.

Category:	Conversion
Parameters:	<cString>
Return value:	<tDateTime>
Module:	Util.c

1.16.4.17 XA_UrlEncode

Encodes a URL string to URL-UTF8 format.

Note: URL encoding is a mechanism for translating unprintable or special characters to a universally accepted format by web servers and browsers.

Category:	Conversion
Parameters:	<cUrl>
Return value:	<cUrlUtf8>

Module:	Util.c
----------------	--------

1.16.4.18 XA_UrDecode

Decodes a URL string to URL-UTF8 format.

Note: URL encoding is a mechanism for translating unprintable or special characters to a universally accepted format by web servers and browsers.

Category:	Conversion
Parameters:	<cUrl>
Return value:	<cUrlUtf8>
Module:	Util.c

1.16.4.19 XA_UrDecodeToAnsi

Decodes a URL string to URL-Ansi format.

Note: URL encoding is a mechanism for translating unprintable or special characters to a universally accepted format by web servers and browsers.

Category:	Conversion
Parameters:	<cUrl>
Return value:	<cUrlAnsi>
Module:	Util.c

1.16.5 Date and time functions

Windows API function accessible from Xailer:

- SetDate(nDay, nMonth, nYear) --> ISuccess
- SetTime(nHours, nMinutes, nSeconds, nMillSeconds) --> ISuccess
- GetTimeZoneInformation() --> hTimeZoneInfo (hash) // Bias, StandardName, StandardDate, StandardBias, DaylightName, DayLightDate, DayLightBias
- XA_Local2UTC(dtLocal) --> dtUTC
- XA_UTC2Local(dtUTC) --> dtLocal
- XA_UnixTimeStamp2DateTime(nTime) --> tDateTime

1.16.6 Debug functions

Name
LogFile
LogDebug
LogDisplay
OutDebug
OutDebugDec
OutDebugHex

Windows API function accessible from Xailer:

- OutputDebugString(cText) --> NIL
- IsDebuggerPresent() --> IValue

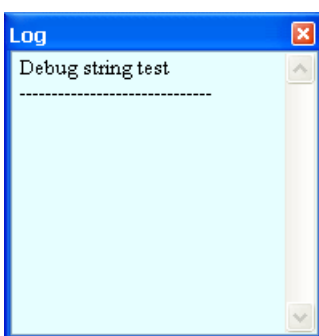
1.16.6.1 LogDebug

Show a message on the IDE debugger windows. This function only works in conjunction with the IDE.

Category:	Debug
Parameters:	<p1>, ..., <p9> Values to show
Return value:	NIL
Module:	Debug.prg
See also:	LogFile, OutDebugDec, OutDebug

1.16.6.2 LogDisplay

Shows a message on a window with the same format than the IDE debugger window. Once the window is shown, future calls to this function just appends the message to the window body text.



Category:	Debug
Parameters:	<p1>, ..., <p9> Values to show

Return value:	NIL
Module:	Debug.prg
See also:	LogFile, OutDebugDec, OutDebug

1.16.6.3 LogFile

Adds the text passes as parameter to the **LOG.TXT** file located in the current directory.

Category:	Debug
Parameters:	<cText> Text to add
Return value:	NIL
Module:	Util.c
See also:	OutDebug

1.16.6.4 OutDebug

Shows in the debug information windows the text passed as parameter.

The debug information window is visible only when the application is executed from the IDE. Any call to this function will not have any effect in the final executable.

The method used to show this information in a windows in an standard Windows API mechanism based on the OutputDebugString function.

Category:	Debug
Parameters:	<cText1>, ..., <cTextN> Texts to show
Return value:	NIL
Module:	Util.c
See also:	LogFile, OutDebugDec, OutDebugHex

1.16.6.5 OutDebugDec

Shows in the debug information windows the decimal number passed as parameter.

The debug information window is visible only when the application is executed from the IDE. Any call to this function will not have any effect in the final executable.

The method used to show this information in a windows in an standard Windows API mechanism based on the OutputDebugString function.

Category:	Debug
Parameters:	<nValue> Value to show
Return value:	NIL
Module:	Debug.prg
See also:	LogFile, OutDebug, OutDebugHex

1.16.6.6 OutDebugHex

Shows in the debug information windows the hexadecimal number passed as parameter.

The debug information window is visible only when the application is executed from the IDE. Any call to this function will not have any effect in the final executable.

The method used to show this information in a windows in an standard Windows API mechanism based on the OutputDebugString function.

Category:	Debug
Parameters:	<nValue> Value to show
Return value:	NIL
Module:	Debug.prg
See also:	LogFile, OutDebugDec, OutDebug

1.16.7 Dialog functions

Name
Calculator
ChooseColorDlg
ChooseFontDlg
CreateFontDlg
FileOpenDlg
FileSaveDlg
MsgAlert
MsgCancelOk
MsgDebug
MsgInfo
MsgNoYes
MsgOkCancel
MsgRetryCancel
MsgStop
MsgSysError
MsgYesNo

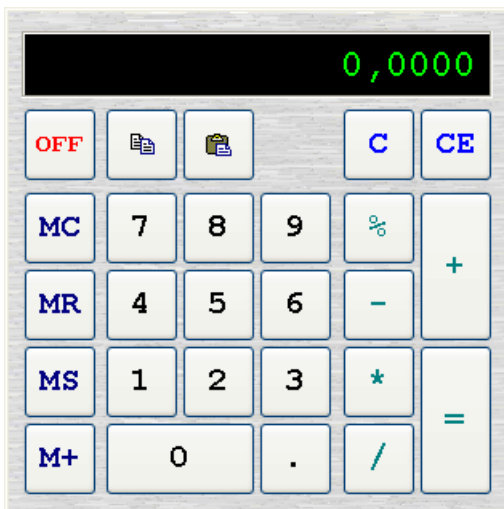
MsgYesNoCancel
 PageSetupDlg
 PickIconDlg
 PrintDlg
 RunFolder
 ShAboutDlg
 ShExitWindowsDlg
 ShFileProperty
 ShFindComputerDlg
 ShFindFilesDlg
 ShNetConnectionDlg
 ShRestartWindowsDlg

Windows API function accessible from Xailer:

- CommDlgExtendedError() --> nError
- MsgBeep(nType) --> NIL
- ShInvokePrinterCommand(oForm, nAction, cPrinterName, cVerb, IModal) --> NIL

1.16.7.1 Calculator

Shows the internal Xailer calculator.

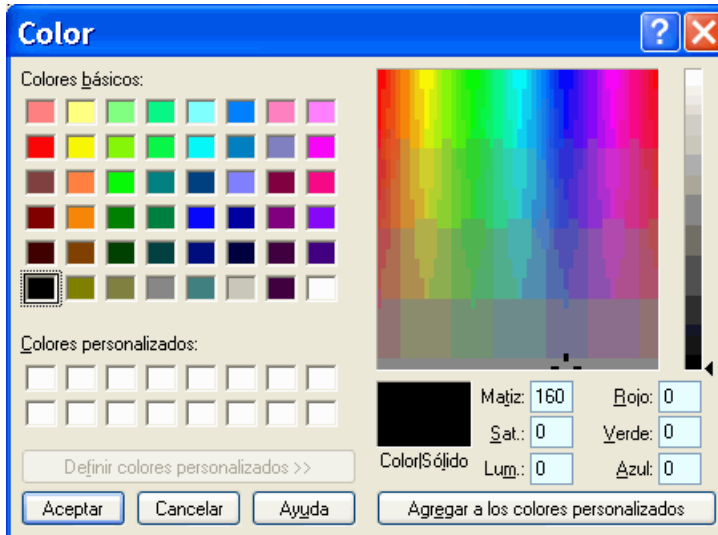


Category:	Dialogs
Parameters	[<nDecimals>]
:	Number of decimals to use. By default 4
Return value:	None
Module:	Calculator.prg

1.16.7.2 ChooseColorDlg

Shows the color selection dialog.

This function only instantiates a TChooseColorDlg object, initializes some properties and execute its Run method to take its updated parameters.



Category:	Dialogs
Parameters	[<oParent>] : Proprietary form. Default: desktop. [<@nColor>] Initial color. It is the same as the nColor property from the TChooseColorDlg object.
Return value:	<IOk> .T. if the user finished pressing the OK button
Module:	ChooseColor.prg
See also:	TChooseColorDlg

1.16.7.3 ChooseFontDlg

Shows the font selection dialog.

This function only instantiates a TChooseFont object, initializes some properties and execute its Run method to take its updated parameters.



Category: Dialogs

Parameters [<oParent>]

: Proprietary form. Default: desktop.

[<@cName>]

Font name. It corresponds to the cName property from the TChooseFont object

[<@nHeight>]

Font height. It corresponds to the nHeight property from the TChooseFont object

[<@nColor>]

Initial color. It corresponds to the nColor property from the TChooseFont object

[<IScreen>]

If it is .T. it will use only screen fonts. It corresponds to the IScreenFonts property from the TChooseFont object

[<IPrinter>]

If it is .T. it will use only printer fonts. It corresponds to the IPrinterFonts from the TChooseFont object

[<@nCharset>]

Font dataset. It corresponds to the nCharset property from the TChooseFont object

Return

value: <IOk>
.T. if the user finished pressing the OK button

Module: ChooseFont.prg

See also: TChooseFont, CreateFontDlg

1.16.7.4 CreateFontDlg

Shows the font selection dialog.

This function only instantiates a TChooseFont object, initializes some properties and execute its Run method to take its updated parameters and **it returns the new TFont object created.**

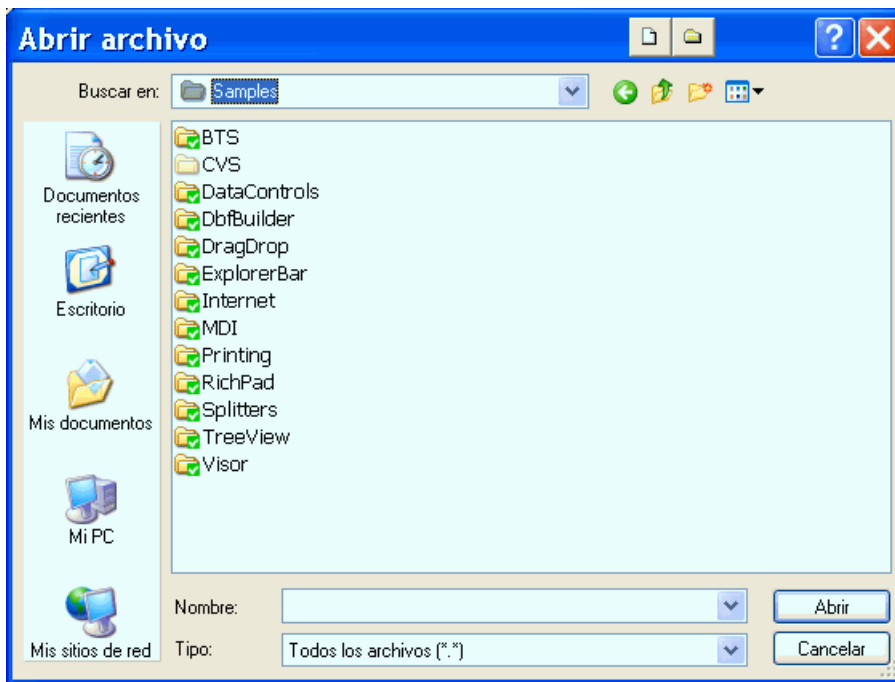


Category:	Dialogs
Parameters	<p>[<oParent>] : Proprietary form. Default: desktop.</p> <p>[<@cName>] Font name. It corresponds to the cName property from the TChooseFont object</p> <p>[<@nHeight>] Font height. It corresponds to the nHeight property from the TChooseFont object</p> <p>[<@nColor>] Initial color. It corresponds to the nColor property from the TChooseFont object</p> <p>[<IScreen>] If it is .T. it will use only screen fonts. It corresponds to the IScreenFonts property from the TChooseFont object</p> <p>[<IPrinter>] If it is .T. it will use only printer fonts. It corresponds to the IPrinterFonts from the TChooseFont object</p> <p>[<@nCharset>] Font dataset. It corresponds to the nCharset property from the TChooseFont object</p>
Return value:	<oFont> TFont object
Module:	ChooseFont.prg
See also:	TChooseFont, ChooseFontDlg

1.16.7.5 FileOpenDlg

Shows the file selection dialog.

This function only instantiates a TFileOpenDlg object, initializes some properties and execute its Run method to return the selected filename.

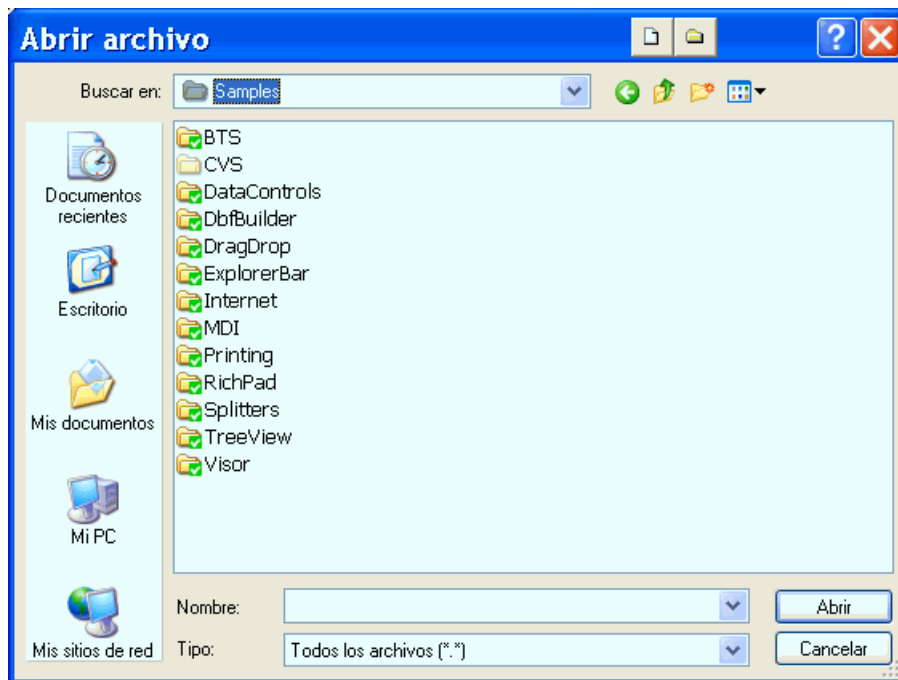


Category:	Dialogs
Parameters	[<oParent>] : Proprietary form. Default: desktop.
	[<cTitle>] Dialog title. It corresponds to the cTitle property from the TFileOpenDlg object
	[<cInitialDir>] Initial directory. It corresponds to the cInitialDir property from the TFileOpenDlg object
	[<cFilter>] File filter. It corresponds to the cFilter property from the TFileOpenDlg object
	[<nFilter>] Initial filter. It corresponds to the nFilterIndex property from the TFileOpenDlg object
Return value:	<cFile> Selected file
Module:	CommonDlg.prg
See also:	TFileOpenDlg, FileSaveDlg

1.16.7.6 FileSaveDlg

Shows the save file dialog.

This function only instantiates a TFileSaveDlg object, initializes some properties and execute its Run method to return the selected filename.



Category:	Dialogs
Parameters	[<oParent>] : Proprietary form. Default: desktop.
	[<cTitle>] Dialog title. It corresponds to the cTitle property from the TFileSaveDlg object
	[<InitialDir>] Initial directory. It corresponds to the clnitialDir property from the TFileSaveDlg object
	[<cFilter>] File filter. It corresponds to the cFilter property from the TFileSaveDlg object
	[<nFilter>] Initial filter. It corresponds to the nFilterIndex property from the TFileSaveDlg object
Return value:	<cFile> Selected file
Module:	CommonDlg.prg
See also:	TFileSaveDlg, FileOpenDlg

1.16.7.7 MsgAlert

Shows an modal alert message.



Category:	Dialogs
Parameters	<cMessage>
:	Message to show [<cTitle>] Dialog title. Default: the application title.
Return value:	NIL
Module:	MsgBox.c
See also:	Other Msg functions???, TMessageBox

1.16.7.8 MsgCancelOK

Shows a modal dialog with the "OK" and "Cancel" options where "Cancel" is the default option.



Category:	Dialogs
Parameters	<cMessage>
:	Message to show [<cTitle>] Dialog title. Default: the application title.
Return value:	<IOk> .T. if the user pressed "OK"
Module:	MsgBox.c
See also:	Other Msg functions???, TMessageBox

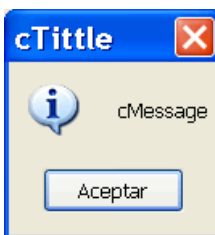
1.16.7.9 MsgBox

Shows a modal dialog with the "Debug" title.

Category:	Dialogs
Parameters :	<cMessage> Message to show
Return value:	NIL
Module:	MsgBox.c
See also:	Other MsgBox functions???, TMessageBox

1.16.7.10 MsgBoxInfo

Shows a modal information dialog.



Category:	Dialogs
Parameters :	<cMessage> Message to show [<cTitle>] Dialog title. Default: the application title.
Return value:	NIL
Module:	MsgBox.c
See also:	Other MsgBox functions???, TMessageBox

1.16.7.11 MsgNoYes

Shows a modal dialog with the options "Yes" and "No" where "No" is the default option.



Category:	Dialogs
Parameters	<cMessage>
:	Message to show [<cTitle>] Dialog title. Default: the application title.
Return value:	<IYes> .T. if the user pressed "Yes"
Module:	MsgBox.c
See also:	Other Msg functions??? (), TMessageBox

1.16.7.12 MsgOkCancel

Shows a modal dialog with the options "OK" and "Cancel", where "Cancel" is the default option.



Category:	Dialogs
Parameters	<cMessage>
:	Message to show [<cTitle>] Dialog title. Default: the application title.
Return value:	<IOk> .T. if the user pressed "OK"
Module:	MsgBox.c
See also:	Other Msg functions??? (), TMessageBox

1.16.7.13 MsgRetryCancel

Shows a modal dialog with the "Retry" and "Cancel" option where "Retry" is the default option.



Category:	Dialogs
Parameters	<cMessage> : Message to show [<cTitle>] Dialog title. Default: the application title.
Return value:	<IOk> .T. if the user pressed "Retry"
Module:	MsgBox.c
See also:	Other Msg functions??? (), TMessageBox

1.16.7.14 MsgStop

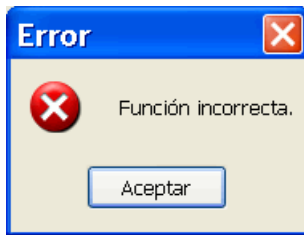
Shows a stop modal dialog.



Category:	Dialogs
Parameters	<cMessage> : Message to show [<cTitle>] Dialog title. Default: the application title.
Return value:	NIL
Module:	MsgBox.c
See also:	Other Msg functions??? (), TMessageBox

1.16.7.15 MsgSysError

Shows a modal dialog with extended information about the last Windows API error produced.



Category:	Dialogs
Parameters	<nError>
:	Error number
Return value:	NIL
Module:	MsgBox.c
See also:	Other Msg functions??? (), TMessageBox

1.16.7.16 MsgYesNo

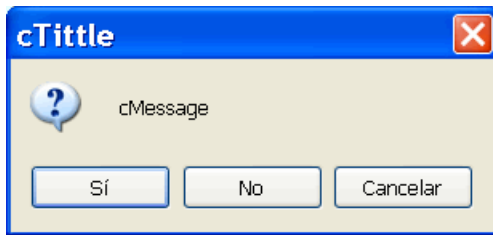
Shows a modal dialog with the "Yes" and "No" options, where "Yes" is the default option.



Category:	Dialogs
Parameters	<cMessage>
:	Message to show [<cTitle>] Dialog title. Default: the application title.
Return value:	<IYes> .T. if the user pressed "Yes"
Module:	MsgBox.c
See also:	Other Msg functions??? (), TMessageBox

1.16.7.17 MsgYesNoCancel

Shows a modal dialog with the "Yes", "No" and "Cancel" options.

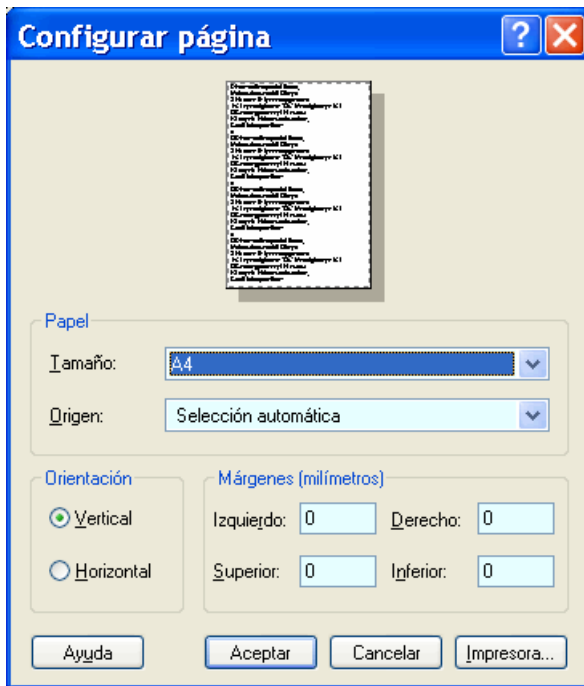


Category:	Dialogs
Parameters	<cMessage> Message to show
:	[<cTitle>] Dialog title. Default: the application title.
Return value:	<nRet> mrYES, mrNO, mrCANCEL
Module:	MsgBox.c
See also:	Other Msg functions??? (), TMessageBox

1.16.7.18 PageSetupDlg

Shows Printer page selection dialog.

This function only instantiates a TPageSetupDlg object, initializes some properties and execute its Run method to return with its updated parameters.



Category: Dialogs

Parameters [`<oParent>`]

: Proprietary form. Default: desktop.

[`<@aPaperSize>`]

Array with the page height and width. It corresponds to the `aPaperSize` property from the `TPageSetupDlg` object

[`<@aMargins>`]

Array with the printer margins. It corresponds to the `aMargins` property from the `TPageSetupDlg` object

[`<llInches>`]

.T. if the dimensions and coordinates provided are specified in inches. It corresponds to the `llInches` property from the object `TPageSetupDlg` object

Return `<!Ok>`

value: .T. if the user finished pressing the OK button

Module: PageDlg.prg

See also: TPageSetupDlg

1.16.7.19 PickIconDlg

Shows a file icon selection dialog.

This function only instantiates a `TPickIconDlg` object, initializes some properties and execute its `Run` method to return with its updated parameters.



Category:	Dialogs
Parameters	[<oParent>] Proprietary form. Default: desktop.
:	[<cFilename>] File with the icons. It corresponds to the cFilename property from the TPickIconDlg object
	[<@ nIndex>] Icon selected by default. It corresponds to the nIndex property from the TPickIconDlg dialog
Return value:	<IOk> .T. if the user finished pressing the OK button
Module:	PickIconDlg.prg
See also:	TPickIconDlg

1.16.7.20 PrintDlg

Shows a printer selection dialog.

This function only instantiates a TPrintDlg, object, initializes some properties and execute its Run method to return with its updated parameters.



Category: Dialogs

Parameters [<oParent>]

: Proprietary form. Default: desktop.

[<@nFromPage>]

Initial page. It corresponds to the nFromPage property from the TPrintDlg object

[<@nToPage>]

Final page. It corresponds to the nToPage property from the TPrintDlg object

[<@nCopies>]

Number of copies. It corresponds to the nCopies property from the TPrintDlg object

[<@ICollate>]

Page collated. It corresponds to the ICollate property from the TPrintDlg object

Return <IOk>

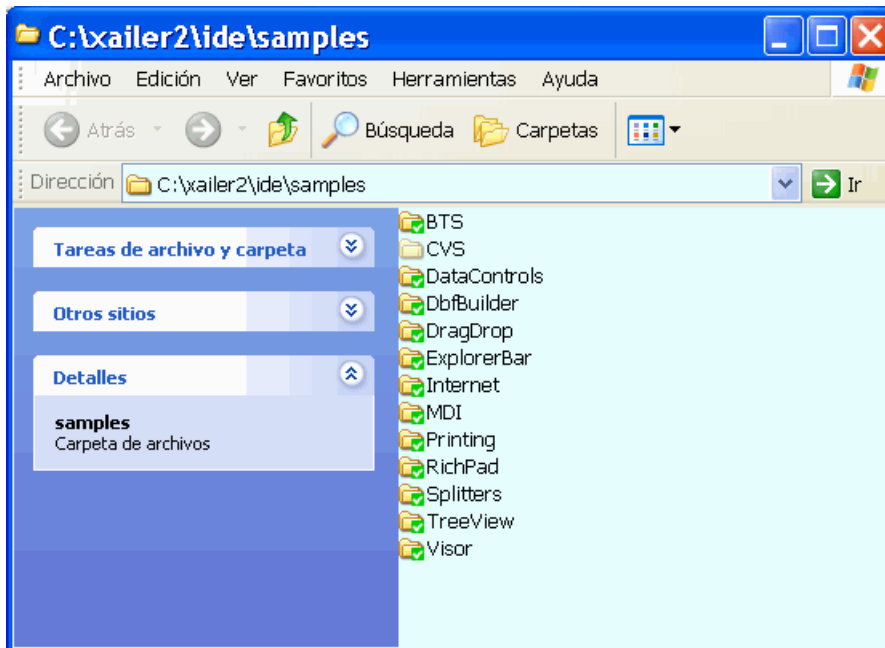
value: .T. if the user finished pressing the OK button

Module: PrintDlg.prg

See also: TPrintDlg

1.16.7.21 RunFolder

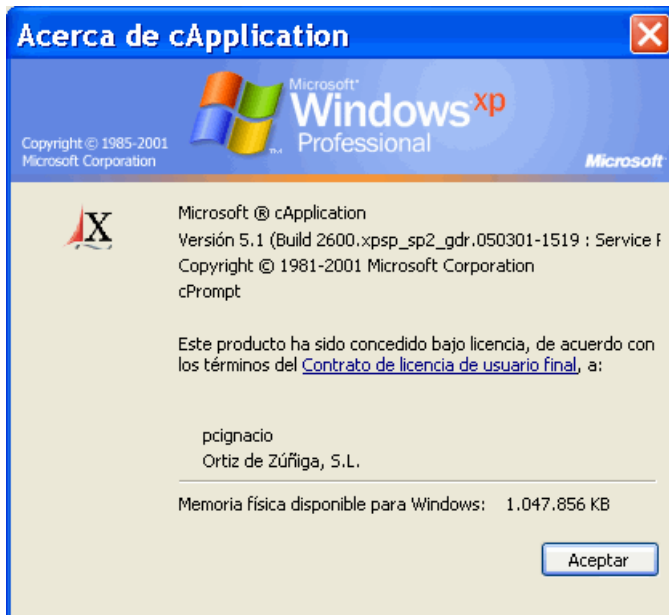
Shows a specific folder through the Windows file explorer.



Category:	Dialogs
Parameters	<cDirectory>
:	Directory to show
Return value:	NIL
Module:	ShFolder.prg
See also:	See also Sh* functions()

1.16.7.22 ShAboutDlg

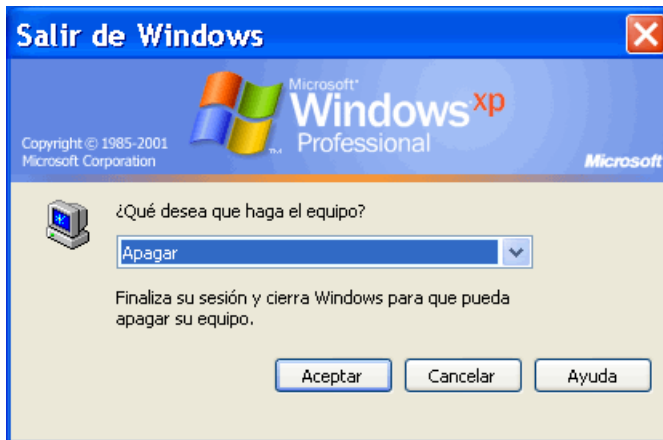
Shows the typical "About" windows screen.



Category:	Dialogs
Parameters	<oParent> Proprietary form
:	<cApplication> Application name that will be shown in the dialog
	<cPrompt> Message to show in the dialog body
	[<olcon>] Icon to show
Return value:	NIL
Module:	ShFolder.prg
See also:	See also Sh* functions()

1.16.7.23 ShExitWindowsDlg

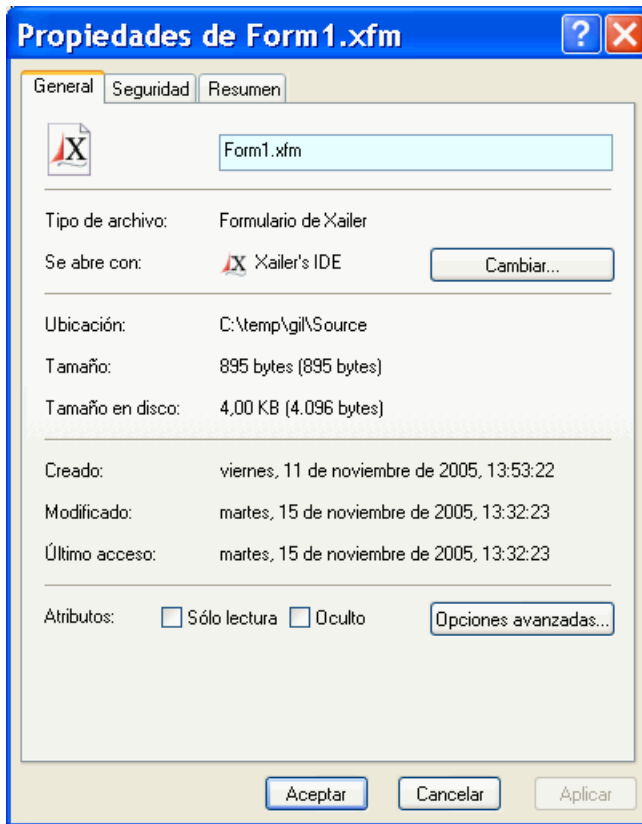
Shows the classic "Exit Windows" dialog.



Category:	Dialogs
Parameters	<oParent>
:	Proprietary form
Return value:	<IOk> Returns .T. if the user pressed "OK"
Module:	ShFolder.prg
See also:	See also Sh* functions()

1.16.7.24 ShFileProperty

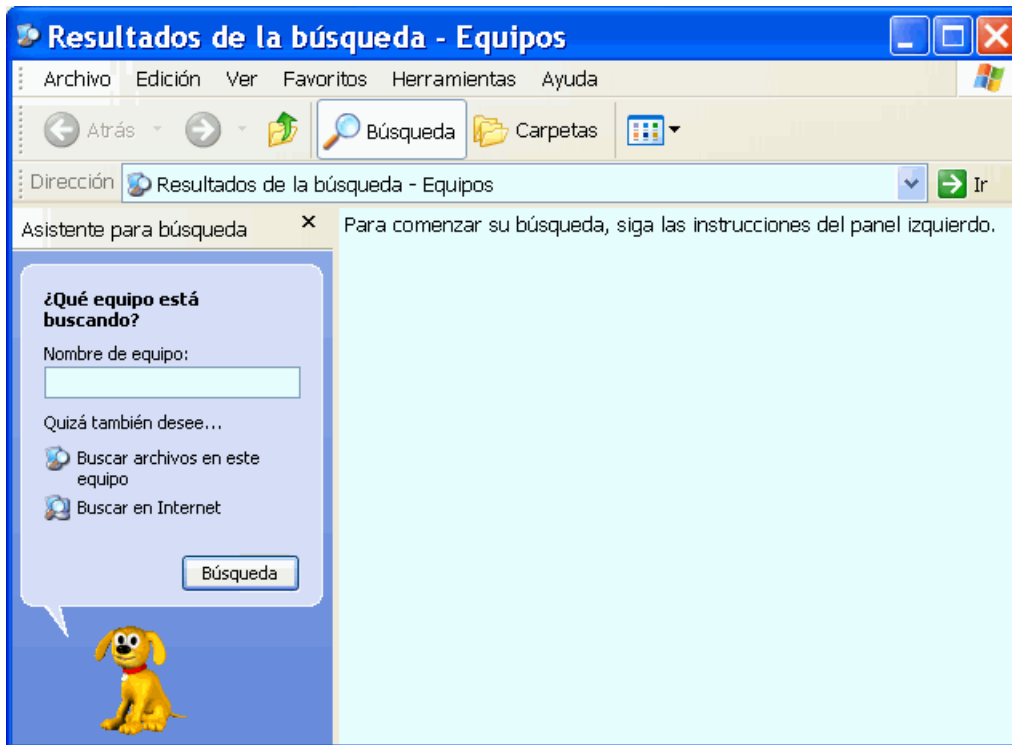
Shows the file properties through the classic windows dialog.



Category:	Dialogs
Parameters	<oParent>
:	Proprietary form
	<cFile>
	File to get the properties
Return value:	NIL
Module:	ShFolder.prg
See also:	See also Sh* functions()

1.16.7.25 ShFindComputerDlg

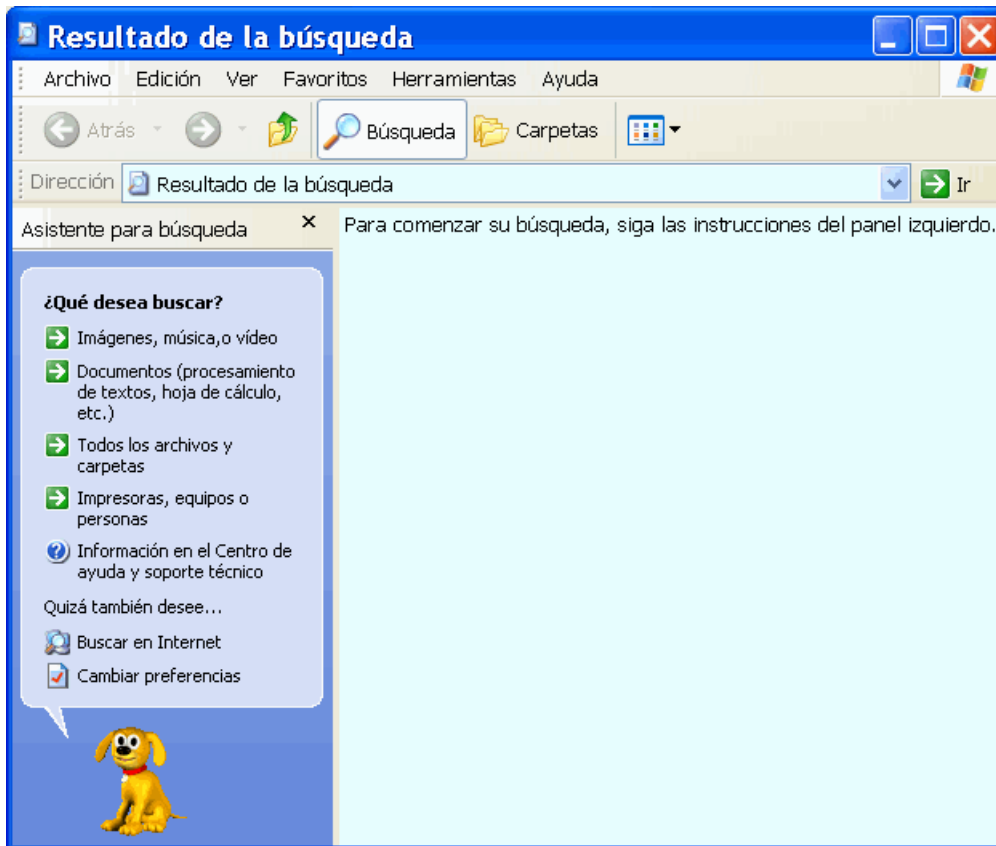
Shows the classic Windows computer search dialog.



Category:	Dialogs
Parameters	None
:	
Return value:	NIL
Module:	ShFolder.prg
See also:	See also Sh* functions()

1.16.7.26 ShFindFilesDlg

Shows the classic Windows file search dialog.



Category:	Dialogs
Parameters	None
:	
Return value:	NIL
Module:	ShFolder.prg
See also:	See also Sh* functions()

1.16.7.27 ShNetConnectionDlg

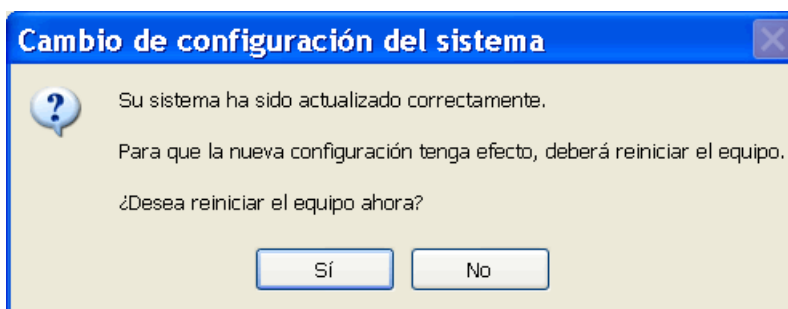
Shows the Windows "connect to network resources" dialog.



Category:	Dialogs
Parameters	<oParent> Proprietary form <cResourceName> File to get the properties <nResourceType> Resource type. A 1 value is used to connect to network drives
Return value:	<IOk> .T. if the connection with the resource was successful
Module:	ShelAppiDlg.prg
See also:	See also Sh* functions()

1.16.7.28 ShRestartWindowsDlg

Shows the classic Windows restart system dialog.



Category:	Dialogs
Parameters	<oParent> Proprietary form
:	<cPrompt> Text to show after the restart operation
	<nFlags> Exit method:
	<ul style="list-style-type: none"> • EWX_FORCE • EWX_LOGOFF • EWX_POWEROFF • EWX_REBOOT • EWX_SHUTDOWN
	For more information, please check the ExitWindowsEx function in the Windows API documentation.
Return value:	<IOk> .T. if the user pressed the "Yes" button
Module:	ShelAppiDlg.prg
See also:	See also Sh* functions()

1.16.8 DLLs and execution functions

Name
Execute
GetFileVersionInfo
IsOcxInstalled
IsRunning
KillProcess
XA_GpfHandler
XA_IsGDIPlus
XA_ResourceSearchOrder

Windows API function accessible from Xailer:

- DllFileOS(cDll) --> cFileOS
- DllFileType(cDll) --> cFileType
- DllFileVersion(cDll) --> cVersion
- DllRegisterSever(cDll) --> ISuccess
- DllUnRegisterSever(cDll) --> ISuccess
- DllVersion(cDll) --> { nHiMS, nLoMS, nHiLS, nLoLS, nFileOS, nFileType }
- DllVersionBuild(cDll) --> nBuild
- DllVersionMajor(cDll) --> nMajor
- DllVersionMinor(cDll) --> nMinor
- DllVersionRelease(cDll) --> nRelease
- FreeLibrary(hLib) --> nSuccess
- GetProcessHandle(cFilename) --> hProcess
- LoadLibrary(cDll) --> hLib

- ShellExecute(hWnd, cOper, cFile, cParam, cDir, nShowMode) --> nReturn
- ShellExecuteEx(hWnd, cFile, cOper, cParam, cDir, nShowMode) --> ISuccess
- WinExec(cFile, nMode) --> nError

1.16.8.1 Execute

Executes an external process.

Category:	DLLs & execution
Parameters:	<p><cFile> File to execute</p> <p>[<cDir>] Starting directory. If empty, it uses the application root directory. If there is not path on 'cFile' it uses <i>Application:CurDir</i></p> <p>[<IWait>] If it is .T. the application will wait until the process ends. By default .F.</p> <p>[<nShowMode>] External process show mode:</p> <ul style="list-style-type: none"> • SW_HIDE • SW_SHOWNORMAL • SW_SHOWMINIMIZED • SW_SHOWMAXIMIZED <p>By default SW_SHOWNORMAL</p> <p>[<@nProcessId>] Process ID to be assigned by the function.</p>
Return value:	<hProcess nExitCode> Process handle when IWait is .F.. Otherwise it returns the process exit code
Module:	Util.c
See also:	ShellExecute, ShellExecuteEx, WinExec

1.16.8.2 GetFileVersionInfo

Returns version information of an executable file (EXE) or a dynamic link library (DLL).

Category:	DLLs and execution
Parameters:	<p><cFile> File name</p> <p>[<cltem>] Data to retrieve. Possible values:</p> <ul style="list-style-type: none"> • CompanyName • FileDescription • FileVersion • InternalName • LegalCopyright

	<ul style="list-style-type: none"> • OriginalFilename • ProductName • ProductVersion • Build • LegalTrademarks
Return Value:	<cInfo> Version info
Module:	DLLFunc.prg

1.16.8.3 IsOcxInstalled

Returns TRUE if a Ocx control is installed.

Category:	DLLs & execution
Parameters:	<cClsId cName> Control name or identifier
Return value:	<IExists> TRUE if is installed
Module:	Internal

1.16.8.4 IsRunning

Returns TRUE if a process is running.

Category:	DLLs & execution
Parameters:	<hProcess> Handle to process
Return value:	<IState> TRUE if is running
Module:	Util.c
See also:	Execute, ShellExecute, ShellExecuteEx, WinExec

1.16.8.5 KillProcess

Kills an active process.

Category:	DLLs & execution
Parameters:	<hProcess> Handle to process
Return value:	<ISuccess>

	TRUE if success
Module:	Util.c
See also:	Execute, ShellExecute, ShellExecuteEx, WinExec

1.16.8.6 XA_GpfHandler

Intercepts GPFs giving the ability to change its default behavior which is to show a simple message with the call stack trace.

If a GPF is fired and a code-block is already set with this function. That code-block will be executed, but at the end the application always quits.

Category:	DLLs & execution
Parameters:	[<bCode>] Code-block to execute in case of GPF error. The code-block receives a unique parameter with the call stack as a string
Return value:	<bOldCode> Previous code-block
Module:	Internal

1.16.8.7 XA_isGDIPlus

Indicates if the GDI+ Microsoft library is installed.

Category:	DLLs & execution
Parameters:	None
Return value:	<ISuccess> TRUE if success
Module:	Internal

1.16.8.8 XA_ResourceSearchOrder

Sets the resource search order.

Category:	DLLs & execution
Parameters:	[<nOrder>] 1 Exe first 2 DLLs first
Return value:	<nPrevOrder>
Module:	Internal

1.16.9 EMF functions (Enhanced meta files)

Windows API function accessible from Xailer:

- CloseEnhMetaFile(hDC) --> hEmf
- CopyEnhMetaFile(hMeta, cFile) --> hEmf
- CreateEnhMetaFile(hDC, cFile, aRect, cDescription) --> hEmf
- DeleteEnhMetaFile(hEmf) --> nRetCode
- GetEnhMetaFile(cFile) --> hEmf
- PlayEnhMetaFile(hDC, hEmf) --> ISuccess

1.16.10 File functions

Name
ExistDir
FileDateTime
FileDateTimeLocal
FileDrive
FileExtension
FileFullName
FileHasDrive
FileHasExtension
FileHasPath
FilePath
FileSetExtension
FileSetName
FileSetPath
FileShortName
FileUnique
GetDrives
GetTempFileName
SetFileDateTime

Windows API function accessible from Xailer:

- CommitPrivateProfileFile(cFile) --> ISuccess
- CommitProfileFile() --> ISuccess
- CopyFile(cSourceFile, cTargetFile) --> ISuccess
- DeletePrivateProfileEntry(cSection, cEntry, cFileName) --> ISuccess
- DeletePrivateProfileSection(cSection, cFileName) --> ISuccess
- DeleteProfileEntry(cSection, cEntry) --> ISuccess
- DeleteProfileSection(cSection) --> ISuccess
- GetCurrentDirectory() --> cDir
- GetFileAttributes(cFile) -> nAttrib

- GetFolder(nId) --> cFolder
- GetFolderAdminTools(IDisplayName) --> cFolder
- GetFolderAltStartup(IDisplayName) --> cFolder
- GetFolderAppData(IDisplayName) --> cFolder
- GetFolderBitBucket(IDisplayName) --> cFolder
- GetFolderCDBurnArea(IDisplayName) --> cFolder
- GetFolderCommonAdminTools(IDisplayName) --> cFolder
- GetFolderCommonAltStartup(IDisplayName) --> cFolder
- GetFolderCommonAppData(IDisplayName) --> cFolder
- GetFolderCommonDesktopDirectory(IDisplayName) --> cFolder
- GetFolderCommonDocuments(IDisplayName) --> cFolder
- GetFolderCommonFavorites(IDisplayName) --> cFolder
- GetFolderCommonMusic(IDisplayName) --> cFolder
- GetFolderCommonOEMLinks(IDisplayName) --> cFolder
- GetFolderCommonPictures(IDisplayName) --> cFolder
- GetFolderCommonPrograms(IDisplayName) --> cFolder
- GetFolderCommonStartMenu(IDisplayName) --> cFolder
- GetFolderCommonStartup(IDisplayName) --> cFolder
- GetFolderCommonTemplates(IDisplayName) --> cFolder
- GetFolderCommonVideo(IDisplayName) --> cFolder
- GetFolderComputersNearMe(IDisplayName) --> cFolder
- GetFolderConnections(IDisplayName) --> cFolder
- GetFolderControls(IDisplayName) --> cFolder
- GetFolderCookies(IDisplayName) --> cFolder
- GetFolderDesktop(IDisplayName) --> cFolder
- GetFolderDesktopDirectory(IDisplayName) --> cFolder
- GetFolderDrives(IDisplayName) --> cFolder
- GetFolderFavorites(IDisplayName) --> cFolder
- GetFolderFonts(IDisplayName) --> cFolder
- GetFolderHistory(IDisplayName) --> cFolder
- GetFolderInternet(IDisplayName) --> cFolder
- GetFolderInternetCache(IDisplayName) --> cFolder
- GetFolderLocalAppData(IDisplayName) --> cFolder
- GetFolderMyDocuments(IDisplayName) --> cFolder
- GetFolderMyMusic(IDisplayName) --> cFolder
- GetFolderMyPictures(IDisplayName) --> cFolder
- GetFolderMyVideo(IDisplayName) --> cFolder
- GetFolderNetHood(IDisplayName) --> cFolder
- GetFolderNetWork(IDisplayName) --> cFolder
- GetFolderPersonal(IDisplayName) --> cFolder
- GetFolderPrintHood(IDisplayName) --> cFolder
- GetFolderPrinters(IDisplayName) --> cFolder
- GetFolderProfile(IDisplayName) --> cFolder
- GetFolderProgramFiles(IDisplayName) --> cFolder
- GetFolderProgramFilesCommon(IDisplayName) --> cFolder
- GetFolderProgramFilesCommonX86(IDisplayName) --> cFolder
- GetFolderProgramFilesX86(IDisplayName) --> cFolder
- GetFolderPrograms(IDisplayName) --> cFolder
- GetFolderRecent(IDisplayName) --> cFolder
- GetFolderResources(IDisplayName) --> cFolder

- GetFolderResourcesLocalized(IDisplayName) --> cFolder
- GetFolderSendTo(IDisplayName) --> cFolder
- GetFolderStartMenu(IDisplayName) --> cFolder
- GetFolderStartup(IDisplayName) --> cFolder
- GetFolderSystem(IDisplayName) --> cFolder
- GetFolderSystemX86(IDisplayName) --> cFolder
- GetFolderTemplates(IDisplayName) --> cFolder
- GetFolderWindows(IDisplayName) --> cFolder
- GetLongPathName(cFile) --> cLongFile
- GetPrivateProfileSection(cSection, cFileName, nBuffer) --> cText
- GetPrivateProfileSectionNames(cFileName, nBuffer) --> cText
- GetPrivateProfileString(cSection, cEntry, cValue, cFileName, nBuffer) --> cText
- GetPrivateProfileInt(cSection, cEntry, nValue, cFileName) --> nValue
- GetProfileInt(cSection, cEntry, nValue) --> nValue
- GetProfileSection(cSection, nBuffer) --> cText
- GetProfileString(cSection, cEntry, cValue, nBuffer) --> cText
- GetShortPathName(cFile) --> cShortFile
- GetSystemDirectory() --> cDir
- GetTempDirectory() --> cDir
- GetTempDirectoryLong() --> cDir
- GetWindowsDirectory() --> cDir
- SearchPath(cPath, cFile, cExtension) --> cFile
- SetFileAttributes(cFile, nAttrib) --> ISuccess
- WritePrivateProfileString(cSection, cEntry, cString, cFileName) --> ISuccess
- WriteProfileString(cSection, cEntry, cString) --> ISuccess

1.16.10.1 GetTempFileName

Returns a valid temporal file name. The file extension is always "TMP".

Category	File
Parameters	[<cPath>] : Directory. By default Windows temporal path
	[<cPrefix>] file prefix (only the first 3 characters are used). By default "TMP"
	[<nNumber>] Initial number to create the temporal file. By default 0 which means it will be generated randomly
Return value:	<cFileName> File name
Module:	Core.c
See also:	FileUnique

Note: For further information consult the API function **GetTempFileName()**. The main difference with the function FileUnique is that this last function always tries to erase the file so it gives you the chance to reuse old and orphan temporal files due a run time error or a power down. Logically if the temporal file is in use the function will not be able to delete it and it will return another valid name.

1.16.10.2 GetDrives

Return a list with all the drives of a specific type.

Category	File
Parameters	[<nType>]
:	Drive type: 0 Any 1 Drive with no root path 2 Removable drive 3 Fixed drive 4 Network drive 5 CD-ROM drive 6 RAM drive
Return value:	<aData> Drives array
Module:	Core.c

1.16.10.3 FileDateTime

Returns the modified date and time of a file.

Category	File
Parameters	<cFilename>
:	File name
Return value:	<dtDateTime> Date time value
Module:	Util.c

1.16.10.4 FileDateTimeLocal

Returns the modified **local** date and time of a file.

Category	File
Parameters	<cFilename>
:	File name
Return value:	<dtDateTime> Date time value
Module:	Util.c

1.16.10.5 SetFileDateTime

Sets local date and time of a file.

Category	File
Parameters	<cFilename> File name
:	<dtDateTime> Date time value
Return value:	True if success
Module:	Util.c

1.16.11 Graphic functions

Name
AngleArc
Arc
ArcTo
Chord
ColorBlend
Ellipse
ExtFloodFill
GetBmpFromColor
HeightToSize
Pie
SizeToHeight
XA_CaptureBitmap
XA_DrawRoundedRect
XA_FillRoundedRect
XA_GetColorization
XA_GradientFillH
XA_GradientFillV
XA_LoadImageAndResize
XA_Lightness
XA_Luminance
XA_PaintCross
XA_SetBmpDisabled
XA_UrlEncode
XA_UrlDecode
XA_UrlDecodeToAnsi

Windows API function accessible from Xailer:

- AddFontResource(cFile) --> nFontsAdded

- BeginPaint(hWnd, @ps) --> hDC
- CopyCursor(hCursor) --> hNewCursor
- CopyIcon(hIcon) --> hNewIcon
- CreateBrushIndirect(nStyle, nColor, nHatch) --> hBrush
- CreateCompatibleDC(hDC) --> hDC
- CreateDC(cDriver, cDevice, cOutput) --> hDC
- CreateFont(Height, Width, Escapement, Orientation, Weight, Style, Name, CharSet) --> hFont
- CreateHatchBrush(nStyle, nColor) --> hBrush
- CreatePatternBrush(hBitmap) --> hBrush
- CreatePen(nStyle, nWidth, nColor) --> hPen
- CreatePenIndirect(nStyle, nWidth, nColor) --> hPen
- CreateScalableFontResource(nHidden, cResFile, cFontFile, cFilePath) --> nError
- CreateSolidBrush(nColor) --> hBrush
- CursorArrow() --> NIL
- CursorHand() --> NIL
- CursorNS() --> NIL
- CursorWE() --> NIL
- CursorWait() --> NIL
- DeleteDC(hDC) --> ISuccess
- DeleteObject(hObject) --> ISuccess
- DestroyCursor(hCursor) --> ISuccess
- DestroyIcon(hIcon) --> ISuccess
- DrawBox(hDC, aRect, hPen) --> NIL
- DrawFocusRect(hDC, aRect) --> nResult
- DrawFrameControl(hDC, aRect, nType, nState) --> NIL
- DrawIcon(hDC, nX, nY, hIcon) --> ISuccess
- DrawIconEx(hDC, nX, nY, hIcon, nWidth, nHeight, nFrame, hBrush, nFlags) --> ISuccess
- DrawRaisedBox(hDC, aRect) --> NIL
- DrawStatusText(hDC, aRect, cText, nFlags) --> NIL
- DrawText(hDC, cText, aRect, nFormat) --> nHeight
- EndPaint(hWnd, ps) --> ISuccess
- ExtTextOut(hDC, nX, nY, nOptions, aRect, cText) --> nError
- FillRect(hDC, aRect, hBrush) --> ISuccess
- FrameRect(hDC, aRect, hBrush) --> ISuccess
- GetBrushOrgEx(hDC) --> { nX, nY }
- GetCursor() --> hCursor
- GetCursorPos(@nX, @nY) --> { nX, nY }
- GetDC(hWnd) --> hDC
- GetFontLanguageInfo(hDC) --> nRet
- GetFontNames(nType, hDC, lSort) --> aInfo
- GetIconInfo(hIcon) --> cInfo
- GetStockObject(nObject) --> nIdentifier
- GetSysColorBrush(nColor) --> nIDBrush
- GetTextExtentPoint32(hDC, cText) --> { nX, nY }
- GetTextSize(cText, hFont, Handle) --> {width, height}
- InvertRect(hDC, aRect) --> nResult
- LineTo(hDC, nX, nY, hPen) --> ISuccess
- LoadCursorFromFile(cFile) --> hCursor
- MoveTo(hDC, nX, nY) --> ISuccess
- PtInRect(aRect, aPoint) --> lIn

- Rectangle(hDC, aRect) --> ISuccess
- ReleaseDC(hWnd, hDC) --> nSuccess
- RemoveFontResource(cFile) --> ISuccess
- RoundRect(hDC, aRect, nWidthElip, nHeightElip) --> ISuccess
- SelectObject(hDC, hObject) --> hOldObject
- SetBlackPen(hDC) --> hOldPen
- SetBrushOrgEx(hDC, nX, nY) --> { nX, nY }
- SetCursor(hCursor) --> nOldCursor
- SetCursorPos(@nX, @nY) --> ISuccess
- SetPixel(hDC, nX, nY, nColor) --> nColor
- SetSystemCursor(hCursor, nId) --> ISuccess
- SetTextAlign(hDC, nMode) --> ISuccess
- SetWhitePen(hDC) --> hOldPen
- ShowCursor(IState) --> nDisplayCounter
- SizeToHeight(nSize, hDC) --> nHeight
- TextOut(hDC, nX, nY, cText) --> ISuccess

1.16.11.1 AngleArc

Draws a line segment and an arc.

Category:	Graphics
Parameters:	<nX>: Circle center X coordinate <nY>: Circle center Y coordinate <nRadius>: Circle radius <nStartAngle>: Start angle in grades realtive to X axe <nSweepAngle>: End angle in grades realtive to X axe
Return value:	ISuccess .T. if the operation is successful
Module:	Graphics.c
See also:	Arc, ArcTo

1.16.11.2 Arc

Draws an elliptical arc.

Category:	Graphics
Parameters:	<aRect>: Rectangle (Left, Top, Right, Bottom) coordinates <nXStart>: Radius initial X point that defines the arc

	<nYStart>: Radius initial Y point that defines the arc <nXEnd>: Radius final X point that defines the arc <nYEnd>: Radius final Y point that defines the arc
Return value:	ISuccess .T. if the operation is successful
Module:	Graphics.c
See also:	AngleArc, ArcTo

1.16.11.3 ArcTo

Draws an elliptical arc and updates hDC position.

Category:	Graphics
Parameters	<aRect>: Rectangle (Left, Top, Right, Bottom) coordinates <nXStart>: Radius initial X point that defines the arc <nYStart>: Radius initial Y point that defines the arc <nXEnd>: Radius final X point that defines the arc <nYEnd>: Radius final Y point that defines the arc
Return value:	ISuccess .T. if the operation is successful
Module:	Graphics.c
See also:	AngleArc, Arc

1.16.11.4 Chord

Draws a chord.

Category:	Graphics
Parameters	<aRect>: Rectangle (Left, Top, Right, Bottom) coordinates <nXRadial1>: Initial X point that defines the beginning of the area <nYRadial1>: Initial Y point that defines the beginning of the area <nXRadial2>:

	Final X point that defines the beginning of the area <nYRadial2> Final Y point that defines the beginning of the area [<oBrush>]: TBrush object to be used
Return value:	ISuccess .T. if the operation is successful
Module:	Graphics.c

1.16.11.5 ColorBlend

Calculates the intermediate color between two color.

Category:	Graphics
Parameters:	<nClrFrom> From color <nClrTo> To color <nAlpha> Transparency. From 0 to 255
Return value:	<nColor>
Module:	Graphics.c

1.16.11.6 Ellipse

Draws an ellipse.

Category:	Graphics
Parameters:	<hDC> Handle to DC <nLeft> Left coordinate <nTop> Top coordinate <nRight> Right coordinate <nBottom> Bottom coordinage
Return value:	<ISuccess> True if success
Module:	Graphics.c

1.16.11.7 ExtFloodFill

Fills an area with a specific color.

Category:	Graphics
Parameters:	<hDC> Handle to DC <nXStart> Initial X coordinate <nXEnd> Initial Y coordinate <nColor> Fill color <nType> 0: The fill area is bounded by the color specified by the nColor parameter. 1: The fill area is defined by the color that is specified by nColor. Filling continues outward in all directions as long as the color is encountered. This style is useful for filling areas with multicolored boundaries.
Return value:	<ISuccess> True if success
Module:	Graphics.c

1.16.11.8 GetBmpFromColor

Creates a square bitmap from a specific color. In order to use the bitmap on any control you must set INoFrame to false in order to manage correctly its transparency.

Category:	Graphics
Parameters:	<nColor> Color code <nSize> Bitmap height and width. By default 16 <INoFrame> If true the bitmap will not have any frame. By default false.
Return value:	<hBitmap> Bitmap handle
Module:	Graphics.c

1.16.11.9 HeightToSize

Converts height size in pixels to the font size.

Category:	Graphics
------------------	----------

Parameters	<nHeight>
:	Height in pixels
Return value:	<nSize> Font size
Module:	Font.prg
See also:	SizeToHeight

1.16.11.1 Pie

Draws a pie.

Category:	Graphics
Parameters	<aRect> : Rectangle (Left, Top, Right, Bottom) coordinates
:	<nXRadial1> : Initial X point that defines the beginning of the area
	<nYRadial1> : Initial Y point that defines the beginning of the area
	<nXRadial2> : Final X point that defines the beginning of the area
	<nYRadial2> : Final Y point that defines the beginning of the area
	[<oBrush>] : TBrush object to be used
Return value:	ISuccess .T. if the operation is successful
Module:	Graphics.c

1.16.11.1 SizeToHeight

Converts the font size to height in pixels.

Category:	Graphics
Parameters:	<nSize> Font size
Return value:	<nHeight> Height in pixels
Module:	Font.prg
See also:	HeightToSize

1.16.11.1:XA_CaptureBitmap

Creates a bitmap from a specific window area.

Category:	Graphics
Parameters:	<hWnd> Window handle [<aRect> <nLeft>, <nTop>, <nRight>, <nBottom>] Window area. By default windows client area
Return value:	<hBitmap> Bitmap handle
Module:	Internal

1.16.11.1:XA_DrawRoundedRect

Draws a rounded rectangle.

Category:	Graphics
Parameters:	<hDC> Handle to context <rect> Rectangle coordinates <nRadius> Border radius <nColor> Color <nTransparency> Transparency level (1-255). By default 255 <nPenSize> Pen size. By default 2
Return value:	NIL
Module:	Internal

1.16.11.1:XA_FillRoundedRect

Fills a rounded rectangle.

Category:	Graphics
Parameters:	<hDC> Handle to context <rect> Rectangle coordinates <nRadius> Border radius <nColor>

	Color <nTransparency> Transparency level (1-255). By default 255
Return value:	NIL
Module:	Internal

1.16.11.1\XA_GetColorization

Default color of the system established on Windows screen properties. Available on Windows Vista or later.

Category:	Graphics
Parameters:	None
Return value:	<nValue> From 0 to 255
Module:	Graphics.c

1.16.11.1\XA_GradientFillH

Paints a rectangle with an horizontal gradient.

Category:	Graphics
Parameters	<hDC> : Handle to the device context <aRect> : Rectangle to paint <nClrFrom> : Initial color <nClrTo> : End color
Return value:	NIL
Module:	Graphics.c
See also:	XA_GradientFillV

1.16.11.1\XA_GradientFillV

Paints a rectangle with a vertical gradient.

Category:	Graphics
Parameters	<hDC>

:	Handle to the device context <aRect> Rectangle to paint <nClrFrom> Initial color <nClrTo> End color
Return value:	NIL
Module:	Graphics.c
See also:	XA_GradientFillH

1.16.11.1 XA_LoadImageAndResize

Retorna un handle a bitmap en base a un fichero o stream de datos y con una resolución específica.

Categoría:	Gráficos
Parámetros:	<clmageData> Fichero o fuente de datos <nWidth> Ancho del bitmap <nHeight> Alto del bitmap <ITransparent> Verdadero si fondo a de hacerse transparente [<nBgColor>] Color de fondo cuando ITransparent igual a .F.
Valor Retorno:	<hBitmap>
Módulo:	Interno

1.16.11.1 XA_Lightness

Color lightness. It is not a very fast function. If speed is important it is preferable to use XA_Luminance.

Category:	Graphics
Parameters:	<nColor> Color
Return value:	<nValue> Percentage from 0 to 100
Module:	Util.c
See also:	XA_Luminance

1.16.11.2:XA_Luminance

Color luminance. If speed is not important you should consider use XA_Lightness.

Category:	Graphics
Parameters:	<nColor> Color
Return value:	<nValue> From 0 to 255
Module:	Util.c
See also:	XA_Lightness

1.16.11.2:XA_PaintCross

Paints a cross.

Category:	Graphics
Parameters:	<hDC> Handle to context <rect> Rectangle coordinates <nColor> Color <nTransparency> Transparency level (1-255). By default 255
Return value:	NIL
Module:	Internal

1.16.11.2:XA_SetBmpDisableMode

Sets the way images on disabled control will be painted.

Category:	Graphics
Parameters :	<nMode>: dmSUNKEN: Sunken efect. Default mode. dmGRAYED: Shades of gray
Return value:	NIL
Module:	Internal

The effect dmGRAYED is only available on XP operating system or above and the screen mode is 32 bits. The images should be also of 8 bytes type(256 colors or 32 bits); images of 16 or 24 can not be converted. On those cases the dmSUNKEN style will be used.

This function only runs when is Xailer who calculates de disabled images. In case the user has given disabled images to the control, those will take precedence.

1.16.12 Keyboard functions

Name

XA_SetKey

Windows API function accessible from Xailer:

- GetAsyncKeyState(nKey) --> nState
- GetKeyState(nKey) --> nState
- IGetAsyncKeyState(nKey) --> IState
- IGetKeyState(nKey) --> IState
- RegisterHotKey(hWnd, nHotKey, nFlags, nVK) --> nError
- SetKeyState(nKey, IOnOff) --> NIL // nKey := VK_CAPITAL, VK_NUMLOCK y VK_SCROLL
- UnRegisterHotKey(hWnd, nHotKey) --> nError

1.16.12.1 XA_SetKey

[x]Harbour/Clipper SET KEY command support. Permits to se a code-block that will be evaluated when a specific key is pushed by the user in any part of the application.

Category:	Keyboard
Parameters:	<p><nKey> Pushed key</p> <p><nFlag> Keyboard state flags, can be a OR combination of the following values: FSHIFT, FCONTROL, FALT</p> <p><bCode> Code-block to evaluate</p>
Return value:	None
Module:	Internal

Sample:

```
XA_SetKey( Asc( "C" ), FALT, { || Calculator() } )
```

1.16.13 Menu functions

Windows API function accessible from Xailer:

- AppendMenu(hWnd, nFlags, hItem, cText) --> ISuccess

- CheckMenuItem(hMenu, nPos, nFlags) --> nPrevState
- CheckMenuRadioItem(hMenu, nFirst, nLast, nCheck, nFlags) --> nError
- CreateMenu() --> hMenu
- CreatePopupMenu() --> hMenu
- DeleteMenu(hMenu, hItem, nFlags) --> ISuccess
- DestroyMenu(hMenu) --> ISuccess
- DrawMenuBar(hWnd) --> ISuccess
- EnableMenuItem(hMenu, nPos, nFlags) --> nPrevState
- EndMenu() --> ISuccess
- GetMenu(hWnd) --> hMenu
- GetMenuItemCount(hMenu) --> nItems
- GetMenuItemID(hMenu, nPos) --> nId
- GetMenuState(hMenu, nPos, nFlags) --> nState
- GetSubMenu(hMenu, nPos) --> hMenu
- GetSystemMenu(hWnd, IReset) --> hMenu
- HiliteMenuItem(hWnd, hMenu, nPos, nFlags) --> nError
- InsertMenu(hMenu, nPos, nFlags, hItem, cText) --> ISuccess
- InsertMenuItem(hMenu, hBefItem, nPosMeaning, nType, nState, nId, nSubMenu, nChecked, nUnChecked, cType) --> ISuccess
- IsMenu(hMenu) --> IValue
- RemoveMenu(hMenu, nPos, nFlags) --> ISuccess
- SetMenu(hWnd, hMenu) --> ISuccess
- SetMenuDefaultItem(hMenu, nPos, nFlags) --> nError
- SetMenuHeight(hMenu, nPixels) --> ISuccess
- SetMenuInfo(hMenu, nStyle, hBkGnd, nData) --> ISuccess
- SetMenuItemInfo(hMenu, hItem, nPosMeaning, nType, nState, nId, nSubMenu, nChecked, nUnChecked, cType) --> ISuccess
- TrackPopupMenu(hMenu, nFlags, nX, nY, hWnd, [{ nLeft, nTop, nRight, nBottom }]) --> nError
- TrackPopupMenuEx(hMenu, nFlags, nX, nY, hWnd, [{ nLeft, nTop, nRight, nBottom }]) --> nError

1.16.14 Multi-language functions

Name
GetWindowsLanguage
i18n
i18n_Language
LT
LT2
SetLanguage

The multi-language functions allow to develop applications that can support several languages at the time. It works in this way:

- All the label applications needed to be language sensitive, should be assigned manually with the LT or LT2 functions, normally in the control creation, or in the OnInitialize event in the form.
- All the label for every language must be saved as resources in the StringTables format.
- You must create a StringTable for every language, using the same identifiers for every label. For example:


```
#define MSG_WARNING 33000

STRINGTABLE LANGUAGE 10,0 // Spanish
{
MSG_WARNING ";ATENCIÓN!"
MSG_PAGE "Página %1 de %2"
}

STRINGTABLE LANGUAGE 9,0 // English
{
MSG_WARNING "WARNING!"
MSG_PAGE "Page %1 of %2"
}

STRINGTABLE LANGUAGE 7, 0 // German
{
MSG_WARNING "WARNUNG!"
MSG_PAGE "Seite %1 von %2"
}
```

Check that every StringTable has identified the language.

- To use it in your application you only need to use the LT function passing as parameter the resource identifier to be used.

```
METHOD Label1Create( oSender ) CLASS TForm1
```

```
oSender:cText := LT( MSG_WARNING )
```

```
RETURN Nil
```

The LT2 function is identical than the LT function, but it eliminates any "&" found in the text. Remember that the "&" can be used as keyboard shortcut and it is possible that you want to show the same label sometimes with the shortcut and sometime without it.

The LT and LT2 function also accept parameters. You only need to specify inside the string expressions like %1, %2, ..., %N, that will be replaced by the indicates parameters in the function. For example:

```
LT( MSG_PAGE, "1", "2" )
```

1.16.14.1 GetWindowsLanguage

Returns the current Windows language identifier used.

Category:	Multi-language
Parameters	None
:	
Return value:	<nLangID>
Module:	Internal
See also:	SetLanguage

1.16.14.2 i18n

Returns the translated text of a literal string established by the function `i18n_Language`.

Category:	Multi-language
Parameters:	<cText> String to translate
Return value:	<cText> Translated string
Module:	i18n.prg
See also:	i18n_Language

The function **i18n()** uses as source the a file with the extension `i18n` which is created automatically by the iDE when the **F3** key is pushed on any `cText` type property of the object inspector. The IDE builds automatically a text file with all the text translations. Is the user responsibility to include that file as a resource on the final executable as RCDATA with the following instruction::

```
i18n RCDATA test.i18n
```

The resource name must b "i18n" in order to work properly.

With the function `i18n_Language()` you can se the language to use.

For further information consult the sample `\samples\i18n`

1.16.14.3 i18n_Language

Establishes the language to be used by `i18n()` function.

Category:	Multi-language
Parameters:	<nLangID> New language identifier
Return value:	<nOldLangID> Old language identifier
Module:	i18n.prg
See also:	i18n

1.16.14.4 LT

Returns the text from a string resource for the current windows language used in the computer.

Category:	Multi-language
Parameters	<nID>
:	String resource identifier
	<cPar1, ..., cParN>
	Parameters to replace in the string resource
Return value:	<cText>
	String text for the current language selected
Module:	Internal
See also:	LT2

The **LT** function accepts parameters. You only need to input in the string the expressions %1, %2,....%N that will be replaced for the indicated parameters in the functions. For example:

```
STRINGTABLE LANGUAGE 10,0 // Spanish
{
MSG_PAGE      "Página %1 de %2"
}
```

```
LT( MSG_PAGE, "1", "2" )
```

If the function does not find the resource for the current selected language, it will use English as default language.

1.16.14.5 LT2

Returns the text from a string resource for the current Windows language used in the computer. It suppresses the possible '&' characters. this functions is identical to LT but it eliminates all the 'possible &' symbols in the string.

Category:	Multi-language
Parameters	<nID>
:	String resource identifier
	<cPar1, ..., cParN>
	Parameters to replace in the string resource
Return value:	<cText>
	String text for the current selected language
Module:	Internal
See also:	LT

The **LT** function accepts parameters. You only need to input in the string the expressions %1, %2,....%N that will be replaced for the indicated parameters in the functions. For example:

```
STRINGTABLE LANGUAGE 10,0 // Spanish
{
```

```
MSG_PAGE      "Página %1 de %2"
}

LT2( MSG_PAGE, "1", "2" )
```

If the function does not find the resource for the current selected language, it will use English as default language.

1.16.14.6 SetLanguage

Establishes the language to be used by the application. By default, the language is established by the operating system and it is possible to get it through the `GetWindowsLanguage` function.

Category:	Multi-language
Parameters	<nLangID>
:	New language identifier
Return value:	<nOldLangID>
	Old language identifier
Module:	Internal
See also:	GetWindowsLanguage

1.16.15 Multimedia functions

Windows API function accessible from Xailer:

- `Beep([<nToneFreq>], [nDuration]) --> Nil`
- `SndPlaySound(<cFilename>, <lAsynchronous>) --> ISuccess`
- `PlaySound(<cFilename>, <hModule>, <nFlags>) --> ISuccess`

1.16.16 ODBC functions

Windows API function accessible from Xailer:

- `SqlFetch(hStmt) --> nRetCode`
- `SqlFetchScroll(hStmt, nOrient, nOffset) --> nRetCode`
- `SqlGetData(hStmt, nField, @nLen, @cBuffer) --> nRetCode`
- `SqlGetValue(hStmt, nField, nType, @nLen, nDec, lDateAsString) --> Value`
- `SqlPrimaryKeys(hStmt, cTableName) --> nRetCode`

1.16.17 Printing functions

Windows API function accessible from Xailer:

- PrinterAbortDoc(hDC) --> ISuccess
- PrinterCreateDC() --> hDC
- PrinterDeleteDC(hDC) --> ISuccess
- PrinterEndDoc(hDC, lFreeDC) --> ISuccess
- PrinterEndPage(hDC) --> ISuccess
- PrinterResetDC(hDC) --> hDC
- PrinterStartDoc(hDC, cJobName, cOutput) --> ISuccess
- PrinterStartPage(hDC) --> ISuccess

1.16.18 Resource functions

Windows API function accessible from Xailer:

- EnumResourceNames(<hModule>, <cType> | <nType>) --> aResourceNames
- ExtractIcon([hInstance], cld | nld, nIndex) --> hIcon
- FindResource(hModule, cName, cType) --> hRes
- FindResourceEx(hModule, cName, cType, nLanguage) --> hRes
- LoadCursor([hInstance], cld | nld) --> hCursor
- LoadIcon([hInstance], cld | nld) --> hIcon
- LoadImage([hInstance], cld, nType, nWidth, nHeight, nFlags) --> cText
- LoadResource(hModule, hRes) --> hMem
- LoadString([hInstance], nld) --> cText
- LockResource(hMem) --> pMem
- MAKEINTRESOURCE(nInteger) --> nRes
- SizeOfResource(hModule, hRes) --> nSize

1.16.19 String functions

Name
Ajoin
Decrypt
Encrypt
IsAllDigits
IsCharAlphaNumeric
IsStringNumeric
ReadLine
ReplaceChars
ToString
XA_Aes256Decrypt
XA_Aes256Encrypt
XA_LineCount
XA_RemoveTags
XA_RTFTToTxt

1.16.19.1 Ajoin

Joins all the strings in an array with a specific separator string.

Category:	Strings
Parameters:	<aStrings> String array to process [<cSeparator>] Separator to use. By default CRLF
Return value:	<cText>
Module:	Utilprg.prg

1.16.19.2 Decrypt

Decrypts a string encrypted with EnCrypt.

Category:	Strings
Parameters:	<cText> Text to process <cKey> Encrypt key
Return value:	<cText>
Module:	Core.c
See also:	EnCrypt

1.16.19.3 Encrypt

Encrypts a string

Category:	Strings
Parameters:	<cText> Text to process <cKey> Encrypt key
Return value:	<cText>
Module:	Core.c
See also:	DeCrypt

1.16.19.4 IsAllDigits

Determines if all characters of a string are numeric type.

Category:	Strings
Parameters:	<cString> String [<IDecPoint>] If true, the decimal point will also be treated as a numeric char. By default false
Return value:	<IValue>
Module:	Util.c
See also:	IsStringNumeric, IsStringNumeric

1.16.19.5 IsCharAlphaNumeric

Determines if a character is alphanumeric type.

Category:	Strings
Parameters:	<nChar> Character
Return value:	<IValue>
Module:	Util.c
See also:	IsStringNumeric, IsAllDigits

1.16.19.6 IsStringNumeric

Determines if all the character string elements are numeric type, decimal point, plus or minus sign.

Category:	Strings
Parameters:	<cString> Text to search
Return value:	<IValue>
Module:	Util.c
See also:	IsCharAlphaNumeric, IsAllDigits

1.16.19.7 ReadLine

Reads a line of a string from a specific position. **nFrom** marks the starting position and in output receives the last position searched. Is a lot fastest than Memoline() on large strings.

Category:	Strings
Parameters:	<cString> String to read <@nFrom> Starting position [<cSep>] Separator. By default CRLF
Return value:	<cLine>
Module:	Utilprg.prg

Sample:

LOCAL nAt

DO WHILE .T.

 cLine := ReadLine(cText, @nAt)

 ...
ENDDO

1.16.19.8 ReplaceChars

Replaces a set of characters in a string with a new character set. For example:

```
Replace( ....., "áéíóú", "aeiou" )
```

Category:	Strings
Parameters:	<cString> String to read <cSearch> Search set [<cReplace>] Replace set
Return value:	<cString>
Module:	Utilprg.prg

1.16.19.9 ToString

Converts any type value to character type.

Category:	Strings
Parameters:	<xValue> Value to cast [<cDefault>] Default value for NIL values. By default the string 'NIL'
Return value:	<cValue>
Module:	Util.c

1.16.19.10 XA_Aes256Decrypt

Decrypts a string encrypted with the XA_Aes256Encrypt.

Category:	Strings
Parameters:	<cText> Text to process. The string length must be multiple of 16. It is a requirement of the encryption algorithm itself. <cKey> Encrypt key
Return value:	<cText>
Module:	internal
See also:	XA_Aes256Encrypt

1.16.19.11 XA_Aes256Encrypt

Encrypts a string with the AES 256 encryption system.

Category:	Strings
Parameters:	<cText> Text to process. The string length must be multiple of 16. It is a requirement of the encryption algorithm itself <cKey> Encrypt key
Return value:	<cText>
Module:	internal
See also:	XA_Aes256Decrypt

1.16.19.1:XA_LineCount

Returns the number of lines in the string. Takes care on escape sequences.

Category:	Strings
Parameters:	<cString> String to read
Return value:	<nLines>
Module:	Utilprg.prg

1.16.19.1:XA_RemoveTags

Remove HTML tags from a string

Category:	Strings
Parameters:	<cString> string to remove HTML tags
Return value:	<cString>
Module:	Util.c

1.16.19.1:XA_RTFToTxt

Converts a file or RTF string to TXT string.

Category:	Strings
Parameters:	<cFile cString> RTF file or string
Return value:	<cString>
Module:	Internal

1.16.20 String Lists functions

Xailer has a set of quick search functions for String lists, ignoring case doing exact match. Is recommend its use on those cases where the native [x]Harbour Ascan() is too slow.

Xailer can handle two different types of String lists:

- **WordList:**

Saves the string list on a tree structure. With small string lists the relation between the memory consumption and the list size is big, but it decreases quickly when the list grows since the tree nodes are shared by all the strings.

Only ASCII characters with codes between 0x20 and 0x5f are permitted (after being turned to uppercase). No duplicates are allowed.

Is specially indicated for big word lists (over 1000 words). With small lists (less than 100) is better to use SortedList.

The management functions for this type of lists are:

Name
XA_WLCreate
XA_WLAdd
XA_WLSearch
XA_WLFree

- **Sorted List:**

Saves the word list on a sorted list structure. The memory consumption is lineal using 8 bytes for every word. But requires that the original array where the words have been obtained remains active since the structure only contains pointers to strings from the original array.

Does not permit to include new elements to the list. If you need to include a new word you will need to recreate the list. Admits duplicates and the search inside the list is of binary type (dichotomous).

Is specially indicated for small word lists (less than 100). With big lists (over 1000 words) is better to use WordList.

The management functions for this type of lists are:

Name
XA_SLCreate
XA_SLSearch
XA_SLFree

1.16.20.1 XA_WLCreate

Creates an structure to save the word list. Every word is taken from the array **<aList>**, and its value will be its position on the array.

Category:	String list
Parameters:	<p>[<aList>]: Array with the word list. If NIL or an empty array the list will be created empty.</p> <p>[<nFrom>]: First element of the array to include. By default the first element of the array.</p> <p>[<nSize>]: Number of elements from the array to include counting from <nFrom>. By default all the elements until the end of the array.</p> <p>[<nSubarrayPos>]: Only functional with multidimension arrays, and</p>

	indicates the position of the words inside each subarray. For example: To include the table field names from an array returned by DbStruct(), <nSubarrayPos> should be 1.
Return value:	<hWordList> : Handle to be used with the rest of functions of WordList.
Module:	Internal
See also:	XA_WLAdd, XA_WLSearch, XA_WLFree

1.16.20.2 XA_WLAdd

Adds a new word to the list. If the word already exists, it will not be included and the return value will be FALSE.

Category:	String list
Parameters :	<hWordList> : Handle returned by XA_WLCreate(). <cString> : Word to add. <nValue> : Value of the word in the list.
Return value:	<ISuccess> : True if success.
Module:	Internal
See also:	XA_WLCreate, XA_WLSearch, XA_WLFree

1.16.20.3 XA_WLSearch

Searchs a word on the list.

Category:	String lists
Parameters :	<hWordList> : Handle returned by XA_WLCreate(). <cString nPos > : Word to search or position in the list
Return value:	<nPos cString> : If a string is passed as a parameter, then it returns the position of the word on the list. If the word does not exist, <nPos> will be 0. On the contrary, if a number is passed as a parameter it will return the string correspondent to that position.
Module:	Internal
See also:	XA_WLCreate, XA_WLAdd, XA_WLFree

1.16.20.4 XA_WLFree

Destroys the word list structure and frees its assigned memory.

Category:	String lists
Parameters :	<hWordList>: Handle returned by XA_WLCreate().
Return value:	None
Module:	Internal
See also:	XA_WLCreate, XA_WLAdd, XA_WLSearch

1.16.20.5 XA_SLCreate

Creates an structure to save the word list. Evey word is taken from the array **<aList>**, and its value will be its position on the array.

Category:	String list
Parameters :	<aList>: Array with the word list. [<nFrom>]: First element of the array to include. By default the first element of the array. [<nSize>]: Number of elements from the array to include counting from <nFrom> . By default all the elements until the end of the array. [<nSubarrayPos>]: Only functional with multidimension arrays, and indicates the position of the words inside each subarray. For example: To include the table field names from an array returned by DbStruct(), <nSubarrayPos> should be 1.
Return value:	<hWordList>: Handle to be used with the rest of functions of SortedList.
Módulo:	Internal
Seea also:	XA_SLSearch, XA_SLFree

1.16.20.6 XA_SLSearch

Searchs a word on the list.

Category:	String lists
Parameters :	<hWordList>: Handle returned by XA_SLCreate(). <cString>: Word to search.
Valor Returno:	<nValue>: Value of the word on the list. If the word does not exist, <nValue> will be 0
Module:	Internal
See also:	XA_SLCreate, XA_SLFree

1.16.20.7 XA_SLFree

Destroys the word list structure and frees its assigned memory.

Category:	String lists
Parameters :	<hWordList>: Handle returned by XA_SLCreate().
Return value:	None
Module:	Internal
See also:	XA_SLCreate, XA_SLSearch

1.16.21 System information functions

Name
EnumWindows
GetComputerName
GetEnvironmentVariable
GetUserName
GetVersionEx
GetWinPlatform
IsElevated
IsUserAnAdmin
SetEnvironmentVariable
SHGetFolder
XA_IsWin7
XA_IsWin8
XA_IsWin9x

XA_IsWin10
XA_IsWin11
XA_IsWinNT
XA_IsWinVista
XA_IsWow64
XA_IsVistaOrHigher
XA_IsXPOrHigher
XailerVersion
Class TSysVer

1.16.21.1 EnumWindows

Returns an array with the handle of all the active windows.

Category:	System information
Parameters:	None
Return value:	<aWinHandles>
Module:	Util.c

1.16.21.2 GetComputerName

Returns the NETBIOS computer name.

Category:	System information
Parameters:	None
Return value:	<cName>
Module:	Util.c
See also:	GetUserName

1.16.21.3 GetEnvironmentVariable

Returns the value of any environment variable

Category:	System information
Parameters:	<cVar> Environment variable to be retrieved
Return value:	<cValue>
Module:	Util.c
See also:	SetEnvironmentVariable

1.16.21.4 GetUserName

Returns the user name of the application.

Category:	System information
Parameters:	None
Return value:	<cName>
Module:	Util.c
See also:	GetComputerName

1.16.21.5 GetVersionEx

Returns in an array the Windows version that is being used.

Category:	System information
Parameters:	None
Return value:	<{ nMajor + nMinor, nBuild, nPlatform, cVersion }>
Module:	Core.c
See also:	XailerVersion

1.16.21.6 GetWinPlatform

Returns the platform number from the Windows version that is being used.

Category:	System information
Parameters:	None
Return value:	<nPlatform>
Module:	Application.prg
See also:	XailerVersion

1.16.21.7 IsElevated

Returns .T. if the application is running on administrator mode.

Category:	System information
Parameters:	None
Return value:	<IValue>
Module:	Util.c

1.16.21.8 IsUserAnAdmin

Returns .T. if the user running the application has administrative privileges. In Windows Vista and Windows 7 you must execute the application with administrative privileges for this function to return true.

Category:	System information
Parameters:	None
Return value:	<IValue>
Module:	Util.c

1.16.21.9 SetEnvironmentVariable

Sets the value of any environment variable

Category:	System information
Parameters:	<cVar> Environment variable to be set
Return value:	<cValue>
Module:	Util.c
See also:	GetEnvironmentVariable

1.16.21.1 SHGetFolder

Returns the path or name of a specific system folder.

Category:	System information
Parameters:	<nCLSID> Folder identifier <IDisplayName> If true returns the path, if false returns the name
Return value:	<cValue>
Module:	Internal

Sample:

```
#include "shlobj.api"
```

```
SHGetFolder( CSIDL_PERSONAL ) --> C:\Documents and Settings\Mis documentos  
SHGetFolder( CSIDL_PERSONAL, .T. ) --> Mis documentos
```

nCLSID identifiers from ShlObj.Api:

```

#define CSIDL_DESKTOP 0 // <desktop>
#define CSIDL_INTERNET 1 // Internet Explorer (icon on desktop)
#define CSIDL_PROGRAMS 2 // Start Menu\Programs
#define CSIDL_CONTROLS 3 // My Computer\Control Panel
#define CSIDL_PRINTERS 4 // My Computer\Printers
#define CSIDL_PERSONAL 5 // My Documents
#define CSIDL_FAVORITES 6 // <user name>\Favorites
#define CSIDL_STARTUP 7 // Start Menu\Programs\Startup
#define CSIDL_RECENT 8 // <user name>\Recent
#define CSIDL_SENDTO 9 // <user name>\SendTo
#define CSIDL_BITBUCKET 10 // <desktop>\Recycle Bin
#define CSIDL_STARTMENU 11 // <user name>\Start Menu
#define CSIDL_MYDOCUMENTS 12 // logical "My Documents" desktop icon
#define CSIDL_MYMUSIC 13 // "My Music" folder
#define CSIDL_MYVIDEO 14 // "My Videos" folder
#define CSIDL_DESKTOPDIRECTORY 16 // <user name>\Desktop
#define CSIDL_DRIVES 17 // My Computer
#define CSIDL_NETWORK 18 // Network Neighborhood
#define CSIDL_NETHOOD 19 // <user name>\nethood
#define CSIDL_FONTS 20 // windows\fonts
#define CSIDL_TEMPLATES 21
#define CSIDL_COMMON_STARTMENU 22 // All Users\Start Menu
#define CSIDL_COMMON_PROGRAMS 23 // All Users\Programs
#define CSIDL_COMMON_STARTUP 24 // All Users\Startup
#define CSIDL_COMMON_DESKTOPDIRECTORY 25 // All Users\Desktop
#define CSIDL_APPDATA 26 // <user name>\Application Data
#define CSIDL_PRINTHOOD 27 // <user name>\PrintHood
#define CSIDL_LOCAL_APPDATA 28 // <user name>\Local
// Settings\Applicaiton Data (non roaming)
#define CSIDL_ALTSTARTUP 29 // non localized startup
#define CSIDL_COMMON_ALTSTARTUP 30 // non localized common startup
#define CSIDL_COMMON_FAVORITES 31
#define CSIDL_INTERNET_CACHE 32
#define CSIDL_COOKIES 33
#define CSIDL_HISTORY 34
#define CSIDL_COMMON_APPDATA 35 // All Users\Application Data
#define CSIDL_WINDOWS 36 // GetWindowsDirectory()
#define CSIDL_SYSTEM 37 // GetSystemDirectory()
#define CSIDL_PROGRAM_FILES 38 // C:\Program Files
#define CSIDL_MYPICTURES 39 // C:\Program Files\My Pictures
#define CSIDL_PROFILE 40 // USERPROFILE
#define CSIDL_SYSTEMX86 41 // x86 system directory on RISC
#define CSIDL_PROGRAM_FILESX86 42 // x86 C:\Program Files on RISC
#define CSIDL_PROGRAM_FILES_COMMON 43 // C:\Program Files\Common
#define CSIDL_PROGRAM_FILES_COMMONX86 44 // x86 Program Files\Common on RISC
#define CSIDL_COMMON_TEMPLATES 45 // All Users\Templates
#define CSIDL_COMMON_DOCUMENTS 46 // All Users\Documents
#define CSIDL_COMMON_ADMINTOOLS 47 // All Users\Start
// Menu\Programs\Administrative Tools
#define CSIDL_ADMINTOOLS 48 // <user name>\Start
// Menu\Programs\Administrative Tools
#define CSIDL_CONNECTIONS 49 // Network and Dial-up Connections
#define CSIDL_COMMON_MUSIC 53 // All Users\My Music
#define CSIDL_COMMON_PICTURES 54 // All Users\My Pictures
#define CSIDL_COMMON_VIDEO 55 // All Users\My Video
#define CSIDL_RESOURCES 56 // Resource Directory
#define CSIDL_RESOURCES_LOCALIZED 57 // Localized Resource Directory
#define CSIDL_COMMON_OEM_LINKS 58 // Links to All Users OEM specific
// apps
#define CSIDL_CDBURN_AREA 59 // USERPROFILE\Local
// Settings\Application Data\Microsoft\CD
// Burning
#define CSIDL_COMPUTERSNEARME 61 // Computers Near Me (computered from
// Workgroup membership)

#define CSIDL_FLAG_CREATE 32768 // combine with CSIDL_ value to force

```

```
        folder creation in SHGetFolderPath()
#define CSIDL_FLAG_DONT_VERIFY    16384    // combine with CSIDL_ value to return
                                     an unverified folder path
#define CSIDL_FLAG_NO_ALIAS      4096     // combine with CSIDL_ value to insure
                                     non-alias versions of the pidl
#define CSIDL_FLAG_PER_USER_INIT 2048     // combine with CSIDL_ value to
                                     indicate per-user init (eg. upgrade)
#define CSIDL_FLAG_MASK          65280    // mask for all possible flag values
```

1.16.21.1:XA_IsWin7

Returns .T. if the system is Windows7.

Category:	System information
Parameters:	None
Return value:	<IValue>
Module:	Util.c
See also:	XailerVersion

1.16.21.1:XA_IsWin8

Returns .T. if the system is Windows8.

Category:	System information
Parameters:	None
Return value:	<IValue>
Module:	Util.c
See also:	XailerVersion

1.16.21.1:XA_IsWin9x

Returns .T. if the system is Windows 98 or Windows ME.

Category:	System information
Parameters:	None
Return value:	<IValue>
Module:	Util.c
See also:	XailerVersion

1.16.21.1\XA_IsWin10

Returns .T. if the system is Windows 10.

Category:	System information
Parameters:	None
Return value:	<IValue>
Module:	Util.c
See also:	XailerVersion

1.16.21.1\XA_IsWin11

Returns .T. if the system is Windows 11.

Category:	System information
Parameters:	None
Return value:	<IValue>
Module:	Util.c
See also:	XailerVersion

1.16.21.1\XA_IsWinNT

Returns .T. if the system used is Windows NT or later.

Category:	System information
Parameters:	None
Return value:	<IValue>
Module:	Util.c
See also:	XailerVersion

1.16.21.1\XA_IsWow64

Returns .T. if the system used is a 64 bits OS and the application is running on a 32 bits subsystem.

Category:	System information
Parameters:	None
Return value:	<IValue>
Module:	Util.c

See also: XailerVersion

1.16.21.1!XA_IsWinVista

Returns .T. if the system used is Windows Vista.

Category:	System information
Parameters:	None
Return value:	<IValue>
Module:	Util.c
See also:	XailerVersion

1.16.21.1!XA_IsVistaOrHigher

Returns .T. if the system used is Windows Vista or higher.

Category:	System information
Parameters:	None
Return value:	<IValue>
Module:	Util.c
See also:	XailerVersion

1.16.21.2!XA_IsXPOrHigher

Returns .T. if the system used is Windows XP or higher.

Category:	System information
Parameters:	None
Return value:	<IValue>
Module:	Util.c
See also:	XailerVersion

1.16.21.2!XailerVersion

Returns information about the Xailer version used.

Category:	System information
------------------	--------------------

Parameters:	<nAction> Information type. It can be: <ul style="list-style-type: none"> • xaVerVersion • xaVerCopyright • xaVerCompiler • xaVerCCompiler • xaVerPlatform • xaVerMajor • xaVerMinor • xaVerRevision • xaVerBuild • xaVerWeb • xaVerMail • xaVerNews
Return value:	<clnfo>
Module:	Core.c
See also:	GetVersionEx

1.16.21.2:TSysVer class

The class **TSysVer** gives important system information. Most of it is already available with the XA_Is???() functions, however, this class gives further information that can be very useful.

Propiedades de sólo lectura
IlsWin2000
IlsWin7
IlsWin8
IlsWin81
IlsWin98
IlsWinHomeServer
IlsWinMe
IlsWinServer2003
IlsWinServer2003R2
IlsWinServer2008
IlsWinServer2008R2
IlsWinServer2012
IlsWinServer2012R2
IlsWinVista
IlsWinXP
IlsWinXP64
IlsWow64
nBuildNumber
nMajorVersion

nMinorVersion
nPlatformId
nProcessorArch
nProductInfo
nProductType
nServicePackMajor
nServicePackMinor
nSuiteMask

1.16.22 Useful functions

Name

XA_VarsEqual

1.16.22.1 XA_VarsEqual

Check if the **value** of two variables are the same avoiding runtime errors. Does the comparison at the low level for speed.

Category:	Util
Parameters:	<xValue1> First value <xValue2> Second value
Return value:	True if equal
Module:	Util.c

1.16.23 Timers & Threads functions

Windows API function accessible from Xailer:

- GetTickCount() --> nCount
- GetWindowThreadProcessId(hWnd) --> hProcessID
- KillTimer(hWnd, hTimer) --> ISuccess
- QueryPerformanceCounter() --> nCount
- QueryPerformanceFrequency() --> nFrec
- SetTimer(hWnd, nId, nTimeOut) --> hTimer
- Sleep(nMilliseconds) --> NIL

1.16.24 Windows functions

Name

GetActiveForm

ProcessMessages

Windows API function accessible from Xailer:

- AnimateWindow(hWnd, nMilliseconds, nFlags) --> ISuccess
- CenterMDIWindow(hWnd) --> NIL
- CenterWindow(hWnd, hWndBase) --> NIL
- ClientToScreen(hWnd, @[aPoint | nX, nY]) --> aPoint
- CreateCaret(hWnd, hBitmap, nWidth, nHeight) --> nError
- CreateWindow(cClassName, cWinName, nStyle, nExStyle, nX, nY, nWidth, nHeight, hWndParent, hMenu, hInstance) --> hWnd
- DestroyWindow(hWnd) --> ISuccess
- EnableScrollBar(Handle, nType, nArrows) --> ISuccess
- EnableWindow(hWnd, IState) --> ISuccess
- GetActiveWindow() --> hWnd
- GetCapture() --> hWnd
- GetClientRect(hWnd, @[aRect | nLeft, nTop, nRight, nBottom]) --> aRect
- GetDesktopWindow() --> hWnd
- GetDeviceCaps(hDC, nIndex) --> nValue
- GetFocus() --> hWnd
- GetForegroundWindow() --> hWnd
- GetLastError() --> nError
- GetParent(hWnd) --> hWnd
- GetParent(hWnd) --> hWndParent
- GetScrollInfo(Handle, nType) --> { nMin, nMax, nPage, nPos, nTrackPos }
- GetScrollPage(Handle, nType) --> nPage
- GetScrollPos(Handle, nType) --> nActualPos
- GetScrollRange(Handle, nType) --> { nMin, nMax }
- GetSysColor(nIndex) --> nColor
- GetSystemMetrics(nIndex) --> nValue
- GetWindow(hWnd, nRelation) --> hWnd
- GetWindowLong(hWnd, nIndex) --> nLong
- GetWindowRect(hWnd, @[aRect | nLeft, nTop, nRight, nBottom]) --> aRect
- GetWindowText(hWnd) --> cText
- InvalidateRect(hWnd, {left, top, right, bottom} | NIL, IEraser) --> ISuccess
- IsChild(hParentWnd, hWnd) --> IValue
- IsIconic(hWnd) --> IState
- IsWindow(hWnd) --> ISuccess
- IsWindowEnabled(hWnd) --> IState
- IsWindowVisible(hWnd) --> IValue
- IsZoomed(hWnd) --> IState
- LockWindowUpdate(hWnd) --> ISuccess
- MoveWindow(hWnd, nX, nY, nWidth, nHeight, IRepaint) --> ISuccess
- MulDiv(nNumber, nNumerator, nDenominator) --> nValue
- PostMessage(hWnd, nWParam, nLParam) --> nResult
- PostQuitMessage(nExitCode) --> NIL
- ReleaseCapture() --> ISuccess

- ScreenToClient(hWnd, @[aPoint | nX, nY]) --> aPoint
- SendMessage(hWnd, nWParam, nLParam) --> nResult
- SetActiveWindow(hWnd) --> ISuccess
- SetBkColor(hDC, nColor) --> nOldBkColor
- SetBkMode(hDC, nMode) --> nOldBkMode
- SetCapture(hWnd) --> nOldCapture
- SetErrMode(nMode) --> nOldMode
- SetFocus(hWnd) --> nOldFocus
- SetForegroundWindow(hWnd) --> ISuccess
- SetParent(hWndChild, hWndParent) --> ISuccess
- SetScrollInfo(Handle, nType, nMin, nMax, nPage, nPos, IDisableNoScroll, IRedraw) --> nActualPos
- SetScrollPage(Handle, nType, nPage, IDisableNoScroll, IRedraw) --> nActualPos
- SetScrollPos(Handle, nType, nPos, IDisableNoScroll, IRedraw) --> nActualPos
- SetScrollRange(Handle, nType, nMin, nMax, IDisableNoScroll, IRedraw) --> nActualPos
- SetTextColor(hDC, nColor) --> nOldTextColor
- SetWindowLong(hWnd, nIndex, nLong) --> nOldLong
- SetWindowPos(hWnd, hWndAfter, nX, nY, nWidth, nHeight, nFlags) --> ISuccess
- SetWindowText(hWnd, cText) --> ISuccess
- ShowCaret(hWnd) --> ISuccess
- ShowScrollBar(Handle, nType, IShow) --> ISuccess
- ShowWindow(hWnd, nShowMode) --> ISuccess
- TrackMouseEvent(hWnd, nFlags, nTimeOut) --> nSuccess
- UpdateWindow(hWnd) --> ISuccess
- ValidateRect(hWnd, {left, top, right, bottom} | NIL, IEraser) --> ISuccess

1.16.24.1 GetActiveForm

Returns the active form object.

Category:	Windows
Parameters	None
:	
Return value:	<oForm>
Module:	Application.prg
See also:	TApplication

1.16.24.2 ProcessMessages

This function permits that all windows process the Windows message queue.

The operating system interacts with all windows through messages that it sends. Those windows receive those messages and act in consequence. Normally this process is completely automatic, with no user request. However, on very large programming loops than consume a lot of CPU is possible that the rest of windows do not have the chance to process their messages that the operating system is sending, like the painting messages for example, on those cases you can

force that those messages get processed calling the ProcessMessages function.

Category:	Windows
Parameters	None
:	
Return value:	NIL
Module:	Internal

Is rare that you need to call this function, except when you execute very intensive loops. In any case you should be extremely cautious when using this function since it may produce undesirable effects on your own application. As a sample we point out some typical errors that are generated when using this function:

- Processes that are triggered with a button click. Es typical to forget to put that button disable or even the complete form. If you make calls to ProcessMessages(), then the user will be able to click again that same button, which probably will produce undesirable effects.
- Processes that affect a database tables that are also showed in some other windows. If on your working loop you are navigating through a table and you call the function ProcessMessages() you should be aware that the table should not be used by any Browser on any active windows even if they are disabled, because the Browse painting routine will provoke a change on the on the table record pointer, that will surely generate equally undesirable effects.

Sample:

```
DO WHILE !Eof()
  n++
  IF n > 100
    ProcessMessages()
    n := 0
  ENDIF
  .....
  .....
ENDDO
```

1.17 Commands

Xailer has its own commands for control creation. However ist use is not recommended since there is no need to do so if you use the form designer included within the IDE, except when you need to create the forms and controls directly from code. The only exception are the basic report engine commands whose documentation can be found on the report engine area documentation.

All the form and controls commands are defined on the file \INCLUDE\CONTROLS.CH, and its definition is really self explanatory, normally coincinding the clauses of the controls with the object properties.

Below is included the complete CONTROLS.CH file. See the documentation of the corresponding class for further information.

```
//----- MENUS -----
```

```

#xcommand MENU [ <oMenu> ] [ OF <oForm> ] [ <popup: POPUP> ] [ MESSAGE <cMsg> ] => ;
    [ <oMenu> := ] XA_MenuBegin( [ <oForm> ], [ <.popup> ], [ <cMsg> ] )

#xcommand MENUADD [ <oMenu> ] [ OF <oForm> ] [ <popup: POPUP> ] [ MESSAGE <cMsg> ] => ;
    [ <oMenu> := ] XA_MenuAdd( [ <oMenu> ], [ <oForm> ], [ <.popup> ], [ <cMsg> ] )

#xcommand MENUITEM [ <oItem> <x: CAPTION,TEXT> ] <cText> [ <action: ACTION, ONCLICK, ON
CLICK> <OnClick> ] [ MENU <oSubMenu> ] ;
    [ <lDisabled: DISABLED> ] ;
    [ <lChecked: CHECKED> ] ;
    [ <lGrayed: GRAYED> ] ;
    [ <lHilited: HILITED> ] ;
    [ <lDefault: DEFAULT> ] ;
    [ WHEN <bWhen> ] ;
    [ BITMAP <cBitmap> ] ;
    [ MESSAGE <cMsg> ] ;
    [ <lRight: RIGHT> ] ;
    => ;
    [ <oItem> := ] XA_MenuItem( <cText>, [ { |oSender,oMenu|<OnClick> } ],
[ <oSubMenu> ], ;
        !<.lDisabled.>, <.lChecked.>, <.lGrayed.>, <.lHilited.>, ;
        <.lDefault.>, [ <cBitmap> ], [ <{bWhen}> ], [ <cMsg> ], [ <.lRight.>
] )

#xcommand SEPARATOR => XA_MenuSeparator()

#xcommand ENDMENU => XA_MenuEnd()

#xcommand ACTIVATE POPUP <oPopup> [OF <oParent> ] [AT <nRow>, <nCol> ] => ;
    TrackPopupMenu( <oPopup>:Handle, TPM_RETURNCMD, [ <nCol> ], [ <nRow> ],
[ <oParent>:Handle ] )

//----- FORM -----
#xcommand DEFINE FORM [ <oForm> ] [ OF <oParent> ] [ <x: TITLE,CAPTION,TEXT> <cText> ] ;
    [ FROM <nTop>, <nLeft> ] [ SIZE <nWidth>, <nHeight> ] ;
    [ BORDERSTYLE <nBorderStyle> ] [ ICON <oIcon> ] ;
    [ <lHelpIcon: HELPICON > ] ;
    [ COLOR <nClrText>, <nClrPane> ] ;
    => ;
    WITH OBJECT [ <oForm> := ] TForm():New( <oParent> ) ;
        ;:SetBounds( [ <nLeft> ], [ <nTop> ], [ <nWidth> ], [ <nHeight> ], .f. )
;

    [ ; :cText := <cText> ] ;
    [ ; :nBorderStyle := <nBorderStyle> ] ;
    [ ; :oIcon := <oIcon> ] ;
    [ ; :lHelpIcon := <.lHelpIcon.> ] ;
    [ ; :nClrText := <nClrText> ] ;
    [ ; :nClrPane := <nClrPane> ] ;
    ;:Create() ;
;END

#xcommand ACTIVATE FORM <oForm> [ <mx: MAXIMIZED> ] [ <mn: MINIMIZED> ] ;
    [ <lCenter: CENTER, CENTERED> ] ;
    => ;
    <oForm>:Show( Iif( <.mx.>, SW_MAXIMIZE, ;
        Iif( <.mn.>, SW_MINIMIZE, SW_NORMAL ) ), ;
        <.lCenter.> )

#xcommand ACTIVATE FORM <oForm> [ <mx: MAXIMIZED> ] [ <mn: MINIMIZED> ] MODAL ;
    [ <lCenter: CENTER, CENTERED> ] ;
    => ;
    <oForm>:ShowModal( Iif( <.mx.>, SW_MAXIMIZE, ;
        Iif( <.mn.>, SW_MINIMIZE, SW_NORMAL ) ), ;
        <.lCenter.> )

```

```

//----- PANEL -----
#xcommand @ <nTop>, <nLeft> PANEL [ <oPanel> ] [ <x: CAPTION,TEXT> <cText> ] [ OF
<oParent> ] ;
    [ SIZE <nWidth>, <nHeight> ] ;
    [ <lSunken: SUNKEN > ] ;
    [ <lRaised: RAISED > ] ;
    [ COLOR <nClrText>, <nClrPane> ] ;
    [ ALIGN <nAlign> ] ;
    [ ANCHORS <nAnchors> ] ;
    [ <lNoXPLOOK: NOXPLOOK> ] ;
=> ;
    WITH OBJECT [ <oPanel> := ] TPanel():New( [ <oParent> ] ) ;
    [; :cText := <cText> ] ;
    [; :SetBounds( [<nLeft>], [<nTop>], [<nWidth>], [<nHeight>], .f. )
;

    [; :nBevelOuter := bv<lSunken> ] ;
    [; :nBevelOuter := bv<lRaised> ] ;
    [; :nClrText := <nClrText> ] ;
    [; :nClrPane := <nClrPane> ] ;
    [; :nAlign := <nAlign> ] ;
    [; :nAnchors := <nAnchors> ] ;
    [; :lXPLOOK := !<lNoXPLOOK> ] ;
    [; :Create() ] ;
;END

//----- LABEL -----
#xcommand @ <nTop>, <nLeft> LABEL <cText> [ VAR <oLabel> ] [ OF <oParent> ] ;
    [ <lSize:SIZE> <nWidth>, <nHeight> ] ;
    [ <lAutoSize: AUTOSIZE > ] ;
    [ <lBorder: BORDER > ] ;
    [ <lHalfSunken: HALFSUNKEN > ] ;
    [ <lHalfRaised: HALFRAISED > ] ;
    [ <lSunken: SUNKEN > ] ;
    [ <lRaised: RAISED > ] ;
    [ <lEtched: ETCHED > ] ;
    [ <lBump: BUMP > ] ;
    [ <lFlat: FLAT > ] ;
    [ ALIGNMENT <nAlignment> ] ;
    [ VALIGNMENT <nVAlignment> ] ;
    [ COLOR <nClrText>, <nClrPane> ] ;
    [ <lTransparent: _TRANSPARENT > ] ;
    [ <lNoTransparent: NOTTRANSPARENT > ] ;
    [ <lMultiline: MULTILINE > ] ;
=> ;
    WITH OBJECT [ <oLabel> := ] TLabel():New( <oParent> ) ;
    [; :cText := <cText> ] ;
    [; :SetBounds( [<nLeft>], [<nTop>], [<nWidth>], [<nHeight>], .f. )
;

    [; :lAutoSize := !<lSize.> ] ;
    [; :lAutoSize := <lAutoSize.> ] ;
    [; :nBorderStyle := bv<lBorder> ] ;
    [; :nBorderStyle := bv<lHalfSunken> ] ;
    [; :nBorderStyle := bv<lHalfRaised> ] ;
    [; :nBorderStyle := bv<lSunken> ] ;
    [; :nBorderStyle := bv<lRaised> ] ;
    [; :nBorderStyle := bv<lEtched> ] ;
    [; :nBorderStyle := bv<lBump> ] ;
    [; :nBorderStyle := bv<lFlat> ] ;
    [; :nAlignment := <nAlignment> ] ;
    [; :nVAlignment := <nVAlignment> ] ;
    [; :nClrText := <nClrText> ] ;
    [; :nClrPane := <nClrPane> ] ;
    [; :lTransparent := <lTransparent.> ] ;
    [; :lTransparent := !<lNoTransparent.> ] ;

```

```

        [; :lMultiline := <.lMultiline.> ] ;
        ;:Create() ;
    ;END

#xcommand @ <nTop>, <nLeft> LABEL <oLabel> <x: CAPTION,TEXT> <cText> [ <moreclauses,...>
] ;

=> ;
    @ <nTop>, <nLeft> LABEL <cText> VAR <oLabel> [ <moreclauses> ]

//----- BUTTON -----

#xcommand @ <nTop>, <nLeft> BUTTON [ <oBtn> ] [ OF <oParent> ] ;
[ SIZE <nWidth>, <nHeight> ] ;
[ <x: CAPTION,TEXT,PROMPT> <cText> ] ;
[ <x: ON CLICK, ACTION> <bOnClick> ] ;
[ <lDefault: DEFAULT > ] ;
[ <lCancel: CANCEL > ] ;
[ <lDisabled: DISABLED > ] ;
[ MODALRESULT <nModalResult> ] ;
[ MESSAGE <cMsg> ] ;
=> ;
WITH OBJECT [ <oBtn> := ] TButton():New( <oParent> ) ;
[; :cText := <cText> ] ;
[; :SetBounds( [<nLeft>], [<nTop>], [<nWidth>], [<nHeight>], .f. )
;

[; :OnClick := { | oSender | <bOnClick> } ] ;
[; :lDefault := <.lDefault.> ] ;
[; :lCancel := <.lCancel.> ] ;
[; :lEnabled := !<.lDisabled.> ] ;
[; :nModalResult := <nModalResult> ] ;
[; :cMessage := <cMsg> ] ;
;:Create() ;
;END

//----- BTNBMP -----

#xcommand @ <nTop>, <nLeft> BTNBMP [ <oBtn> ] [ OF <oParent> ] ;
[ SIZE <nWidth>, <nHeight> ] ;
[ <x: BITMAP, BITMAPS, IMAGES> <oBitmap> ] ;
[ <x: CAPTION,TEXT,PROMPT> <cText> ] ;
[ <x: ON CLICK, ACTION> <bOnClick> ] ;
[ <lDefault: DEFAULT > ] ;
[ <lCancel: CANCEL > ] ;
[ <lDisabled: DISABLED > ] ;
[ MODALRESULT <nModalResult> ] ;
[ MESSAGE <cMsg> ] ;
=> ;
WITH OBJECT [ <oBtn> := ] TBtnBmp():New( <oParent> ) ;
[; :oBitmaps := <oBitmap> ] ;
[; :cText := <cText> ] ;
[; :SetBounds( [<nLeft>], [<nTop>], [<nWidth>], [<nHeight>], .f. )
;

[; :OnClick := { | oSender | <bOnClick> } ] ;
[; :lDefault := <.lDefault.> ] ;
[; :lCancel := <.lCancel.> ] ;
[; :lEnabled := !<.lDisabled.> ] ;
[; :nModalResult := <nModalResult> ] ;
[; :cMessage := <cMsg> ] ;
;:Create() ;
;END

//----- IMAGE -----

#xcommand @ <nTop>, <nLeft> IMAGE [ <oImage> ] [ OF <oParent> ] ;
[ SIZE <nWidth>, <nHeight> ] ;
[ <x: FILE, NAME, RESNAME, RESOURCE> <cResName> ] ;

```

```

[ <lNone:      NONE > ] ;
[ <lRaised:    RAISED > ] ;
[ <lSunken:    SUNKEN > ] ;
[ <lBump:      BUMP > ] ;
[ <lEtched:    ETCHED > ] ;
[ <lFlat:      FLAT > ] ;
[ <lNoTabStop: NOTABSTOP > ] ;
[ <lTransparent: TRANSPARENT > ] ;
[ <lNoTransparent: NOTRANSPARENT > ] ;
[ COLOR <nClrPane> ] ;
=> ;
WITH OBJECT [ <oImage> := ] TImage():New( <oParent> ) ;
[; :oPicture := <cResName> ] ;
  ;:SetBounds( [<nLeft>], [<nTop>], [<nWidth>],[<nHeight>], .f. )
;

[; :nBorderStyle := bv<lNone> ] ;
[; :nBorderStyle := bv<lRaised> ] ;
[; :nBorderStyle := bv<lSunken> ] ;
[; :nBorderStyle := bv<lBump> ] ;
[; :nBorderStyle := bv<lEtched> ] ;
[; :nBorderStyle := bv<lFlat> ] ;
[; :nBorderStyle := bv<lFlat> ] ;
[; :lTabStop      := !<lNoTabStop.> ] ;
[; :lTransparent := <lTransparent.> ] ;
[; :lTransparent := !<lNoTransparent.> ] ;
[; :nClrPane := <nClrPane> ] ;
;:Create() ;
;END

//----- SYSANIMATE -----

#xcommand @ <nTop>, <nLeft> SYSANIMATE [ <oSysAnimate> ] [ OF <oParent> ] ;
[ SIZE <nWidth>, <nHeight> ] ;
[ <x: FILE, AVI> <cAviName> ] ;
[ <x: COMMONAVI> <cCommonAvi> ] ;
[ <lAutoPlay: AUTOPLAY > ] ;
[ <lCentered: CENTERED > ] ;
=> ;
WITH OBJECT [ <oSysAnimate> := ] TSysAnimate():New( <oParent> ) ;
  ;:SetBounds( [<nLeft>], [<nTop>], [<nWidth>], [<nHeight>], .f. )
;

[; :cName := <cAviName> ] ;
[; :nCommonAvi := avi<cCommonAvi> ] ;
[; :lAutoPlay := <lAutoPlay.> ] ;
[; :lCentered := <lCentered.> ] ;
;:Create() ;
;END

//----- PROGRESSBAR -----

#xcommand @ <nTop>, <nLeft> PROGRESSBAR [ <oProgressBar> ] [ OF <oParent> ] ;
[ SIZE <nWidth>, <nHeight> ] ;
[ COLOR <nClrText>, <nClrPane> ] ;
[ VALUE <nValue> ] ;
[ RANGE <nMinValue>, <nMaxValue> ] ;
[ <lSmooth: SMOOTH > ] ;
[ <lVertical: VERTICAL > ] ;
=> ;
WITH OBJECT [ <oProgressBar> := ] TProgressBar():New( <oParent> )
;
  ;:SetBounds( [<nLeft>], [<nTop>], [<nWidth>], [<nHeight>], .f. )
;

[; :nClrText := <nClrText> ] ;
[; :nClrPane := <nClrPane> ] ;
[; :nValue := <nValue> ] ;
[; :nMin := <nMinValue> ] ;

```

```

        [; :nMax      := <nMaxValue> ] ;
        [; :lSmooth := <.lSmooth.> ] ;
        [; :nOrientation:= Iif( <.lVertical.>, orVertical, orHorizontal )
;
        ;:Create() ;
;END

//----- CHECKBOX -----

#xcommand @ <nTop>, <nLeft> CHECKBOX [ <oChk> ] [ OF <oParent> ] ;
[ SIZE <nWidth>, <nHeight> ] ;
[ <x: CAPTION,TEXT,PROMPT> <cText> ] ;
[ <lDisabled: DISABLED > ] ;
[ <lChecked: CHECKED > ] ;
[ <lTransparent: _TRANSPARENT > ] ;
[ MESSAGE <cMsg> ] ;
=> ;
WITH OBJECT [ <oChk> := ] TCheckBox():New( <oParent> ) ;
[; :cText := <cText> ] ;
;:SetBounds( [<nLeft>], [<nTop>], [<nWidth>], [<nHeight>], .f. )
;

[; :lEnabled := !<.lDisabled.> ] ;
[; :lChecked := <.lChecked.> ] ;
[; :lTransparent := <.lTransparent.> ] ;
[; :cMessage := <cMsg> ] ;
;:Create() ;
;END

//----- RADIOBUTTON -----

#xcommand @ <nTop>, <nLeft> RADIO [ <oRadio> ] [ OF <oParent> ] ;
[ SIZE <nWidth>, <nHeight> ] ;
[ <x: CAPTION,TEXT,PROMPT> <cText> ] ;
[ <lDisabled: DISABLED > ] ;
[ <lChecked: CHECKED > ] ;
[ <lGroup: GROUP > ] ;
[ <lTransparent: _TRANSPARENT > ] ;
[ MESSAGE <cMsg> ] ;
=> ;
WITH OBJECT [ <oRadio> := ] TRadio():New( <oParent> ) ;
[; :cText := <cText> ] ;
;:SetBounds( [<nLeft>], [<nTop>], [<nWidth>], [<nHeight>], .f. )
;

[; :lEnabled := !<.lDisabled.> ] ;
[; :lChecked := <.lChecked.> ] ;
[; :lGroup := <.lGroup.> ] ;
[; :lTransparent := <.lTransparent.> ] ;
[; :cMessage := <cMsg> ] ;
;:Create() ;
;END

//----- GROUPBOX -----

#xcommand @ <nTop>, <nLeft> GROUPBOX [ <oGroup> ] [ OF <oParent> ] ;
[ SIZE <nWidth>, <nHeight> ] ;
[ <x: CAPTION,TEXT,PROMPT> <cText> ] ;
[ <lDisabled: DISABLED > ] ;
=> ;
WITH OBJECT [ <oGroup> := ] TGroupBox():New( <oParent> ) ;
[; :cText := <cText> ] ;
;:SetBounds( [<nLeft>], [<nTop>], [<nWidth>], [<nHeight>], .f. )
;

[; :lEnabled := !<.lDisabled.> ] ;
;:Create() ;
;END

```

```
//----- EDIT -----
#xcommand @ <nTop>, <nLeft> EDIT [ <oEdit> ] [ OF <oParent> ] ;
    [ SIZE <nWidth>, <nHeight> ] ;
    [ <x: CAPTION,TEXT,PROMPT> <cText> ] ;
    [ <lNoBorder: NOBORDER > ] ;
    [ <lReadOnly: READONLY > ] ;
    [ <lPassword: PASSWORD > ] ;
    [ <lNoHScroll: NOHSCROLL > ] ;
    [ COLOR <nClrText>, <nClrPane> ] ;
    [ <lDisabled: DISABLED > ] ;
    [ MESSAGE <cMsg> ] ;
    [ FONT <oFont> ] ;
    [ ALIGNMENT <nAlignment> ] ;
=> ;
WITH OBJECT [ <oEdit> := ] TEdit():New( <oParent> ) ;
    ;:SetBounds( [<nLeft>], [<nTop>], [<nWidth>], [<nHeight>], .f. )
;

    [; :cText := <cText> ] ;
    [; :lBorder := !<lNoBorder.> ] ;
    [; :lReadOnly := <lReadOnly.> ] ;
    [; :lPassword := <lPassword.> ] ;
    [; :lHScroll := !<lNoHScroll> ] ;
    [; :nClrText := <nClrText> ] ;
    [; :nClrPane := <nClrPane> ] ;
    [; :lEnabled := !<lDisabled.> ] ;
    [; :cMessage := <cMsg> ] ;
    [; :nAlignment := <nAlignment> ] ;
    ;:Create() ;
    [; :SetFont( <oFont> ) ] ;
;END

//----- MASKEDIT -----
#xcommand @ <nTop>, <nLeft> MASKEDIT [ <oMaskEdit> ] [ OF <oParent> ] ;
    [ SIZE <nWidth>, <nHeight> ] ;
    [ <x: CAPTION,TEXT,PROMPT> <cText> ] ;
    [ <p: PICTURE> <cPicture> ] ;
    [ <lNoBorder: NOBORDER > ] ;
    [ <lReadOnly: READONLY > ] ;
    [ <lNoHScroll: NOHSCROLL > ] ;
    [ COLOR <nClrText>, <nClrPane> ] ;
    [ <lDisabled: DISABLED > ] ;
    [ MESSAGE <cMsg> ] ;
    [ FONT <oFont> ] ;
    [ ALIGNMENT <nAlignment> ] ;
=> ;
WITH OBJECT [ <oMaskEdit> := ] TMaskEdit():New( <oParent> ) ;
    ;:SetBounds( [<nLeft>], [<nTop>], [<nWidth>], [<nHeight>], .f. )
;

    [; :cText := <cText> ] ;
    [; :cPicture := <cPicture> ] ;
    [; :lBorder := !<lNoBorder.> ] ;
    [; :lReadOnly := <lReadOnly.> ] ;
    [; :lHScroll := !<lNoHScroll> ] ;
    [; :nClrText := <nClrText> ] ;
    [; :nClrPane := <nClrPane> ] ;
    [; :lEnabled := !<lDisabled.> ] ;
    [; :cMessage := <cMsg> ] ;
    [; :nAlignment := <nAlignment> ] ;
    ;:Create() ;
    [; :SetFont( <oFont> ) ] ;
;END

//----- DATEEDIT -----
```



```

#xcommand @ <nTop>, <nLeft> DATEEDIT [ <oDateEdit> ] [ OF <oParent> ] ;
    [ SIZE <nWidth>, <nHeight> ] ;
    [ <x: CAPTION,TEXT,PROMPT> <cText> ] ;
    [ <lNoBorder: NOBORDER > ] ;
    [ <lReadOnly: READONLY > ] ;
    [ <lNoHScroll: NOHSCROLL > ] ;
    [ COLOR <nClrText>, <nClrPane> ] ;
    [ <lDisabled: DISABLED > ] ;
    [ MESSAGE <cMsg> ] ;
    [ FONT <oFont> ] ;
=> ;
    WITH OBJECT [ <oDateEdit> := ] TDateEdit():New( <oParent> ) ;
        ;:SetBounds( [<nLeft>], [<nTop>], [<nWidth>], [<nHeight>], .f. )
;

    [; :cText := <cText> ] ;
    [; :lBorder := !<lNoBorder.> ] ;
    [; :lReadOnly := <lReadOnly.> ] ;
    [; :lHScroll := !<lNoHScroll> ] ;
    [; :nClrText := <nClrText> ] ;
    [; :nClrPane := <nClrPane> ] ;
    [; :lEnabled := !<lDisabled.> ] ;
    [; :cMessage := <cMsg> ] ;
    ;:Create() ;
    [; :SetFont( <oFont> ) ] ;
;END

//----- MEMO -----
#xcommand @ <nTop>, <nLeft> MEMO [ <oMemo> ] [ OF <oParent> ] ;
    [ SIZE <nWidth>, <nHeight> ] ;
    [ <x: CAPTION,TEXT,PROMPT> <cText> ] ;
    [ <lNoBorder: NOBORDER > ] ;
    [ <lReadOnly: READONLY > ] ;
    [ <lHScroll: HSCROLL > ] ;
    [ <lNoVScroll: NOVSCROLL > ] ;
    [ COLOR <nClrText>, <nClrPane> ] ;
    [ <lDisabled: DISABLED > ] ;
    [ MESSAGE <cMsg> ] ;
    [ FONT <oFont> ] ;
=> ;
    WITH OBJECT [ <oMemo> := ] TMemo():New( <oParent> ) ;
        [; :cText := <cText> ] ;
        ;:SetBounds( [<nLeft>], [<nTop>], [<nWidth>], [<nHeight>], .f. )
;

    [; :lBorder := !<lNoBorder.> ] ;
    [; :lReadOnly := <lReadOnly.> ] ;
    [; :lHScroll := <lHScroll> ] ;
    [; :lVScroll := !<lNoVScroll> ] ;
    [; :nClrText := <nClrText> ] ;
    [; :nClrPane := <nClrPane> ] ;
    [; :lEnabled := !<lDisabled.> ] ;
    [; :cMessage := <cMsg> ] ;
    ;:Create() ;
    [; :SetFont( <oFont> ) ] ;
;END

//----- RICEDIT -----
#xcommand @ <nTop>, <nLeft> RICEDIT [ <oRTF> ] [ OF <oParent> ] ;
    [ SIZE <nWidth>, <nHeight> ] ;
    [ <x: CAPTION,TEXT,PROMPT> <cText> ] ;
    [ <file: FILE, FILENAME> <cFile> ] ;
    [ RTFSIZE <nSize> ] ;
    [ FORMAT <nFormat> ] ;
    [ <lNoBorder: NOBORDER> ] ;

```

```

[ <lReadOnly: READONLY> ] ;
[ <lHScroll: HSCROLL> ] ;
[ <lNoScroll: NOSCROLL> ] ;
[ <lDisabled: DISABLED> ] ;
[ MESSAGE <cMsg> ] ;
=> ;
WITH OBJECT [ <oRTF> := ] TRichEdit():New( <oParent> ) ;
[; :cText := <cText> ] ;
  ;:SetBounds( [<nLeft>], [<nTop>], [<nWidth>], [<nHeight>], .f. )
;

[; :cFile := <cFile> ] ;
[; :nMaxLenght := <nSize> ] ;
[; :nFormat := <nFormat> ] ;
[; :lBorder := !<.lNoBorder.> ] ;
[; :lReadOnly := <.lReadOnly.> ] ;
[; :lHScroll := <.lHScroll> ] ;
[; :lScroll := !<.lNoScroll> ] ;
[; :lEnabled := !<.lDisabled.> ] ;
[; :cMessage := <cMsg> ] ;
;:Create() ;
;END

//----- COMBOBOX -----

#xcommand @ <nTop>, <nLeft> COMBOBOX [ <oCombo> ] [ OF <oParent> ] ;
[ SIZE <nWidth>, <nHeight> ] ;
[ ITEMS <aItems,...> ] ;
[ COLOR <nClrText>, <nClrPane> ] ;
[ STYLE <nStyle> ] ;
[ <lDisabled: DISABLED > ] ;
[ MESSAGE <cMsg> ] ;
[ FONT <oFont> ] ;
=> ;
WITH OBJECT [ <oCombo> := ] TComboBox():New( <oParent> ) ;
  ;:SetBounds( [<nLeft>], [<nTop>], [<nWidth>], [<nHeight>], .f. )
;

[; :nClrText := <nClrText> ] ;
[; :nClrPane := <nClrPane> ] ;
[; :nStyle := <nStyle> ] ;
[; :lEnabled := !<.lDisabled.> ] ;
[; :cMessage := <cMsg> ] ;
;:Create() ;
[; :SetItems( {<aItems>} ) ] ;
[; :SetFont( <oFont> ) ] ;
;END

//----- LISTBOX -----

#xcommand @ <nTop>, <nLeft> LISTBOX [ <oLbx> ] [ OF <oParent> ] ;
[ SIZE <nWidth>, <nHeight> ] ;
[ ITEMS <aItems,...> ] ;
[ COLOR <nClrText>, <nClrPane> ] ;
[ STYLE <nStyle> ] ;
[ <lDisabled: DISABLED > ] ;
[ <lHScroll: HSCROLL > ] ;
[ MESSAGE <cMsg> ] ;
[ FONT <oFont> ] ;
=> ;
WITH OBJECT [ <oLbx> := ] TListBox():New( <oParent> ) ;
  ;:SetBounds( [<nLeft>], [<nTop>], [<nWidth>], [<nHeight>], .f. )
;

[; :nClrText := <nClrText> ] ;
[; :nClrPane := <nClrPane> ] ;
[; :nStyle := <nStyle> ] ;
[; :lEnabled := !<.lDisabled.> ] ;

```

```

        [; :lHScroll := <.lHScroll.> ] ;
        [; :cMessage := <cMsg> ] ;
        [; :aItems := <aItems> ] ;
        ;:Create() ;
        [; :SetFont( <oFont> ) ] ;
;END

//----- ARRAYBROWSE -----

#xcommand @ <nTop>, <nLeft> ARRAYBROWSE [ <oArrBrw> ] [ OF <oParent> ] ;
    [ SIZE <nWidth>, <nHeight> ] ;
    [ ITEMS <aItems,...> ] ;
    [ HEADERS <aHeads,...> ] ;
    [ COLOR <nClrText>, <nClrPane> ] ;
    [ STYLE <nStyle> ] ;
    [ <lDisabled: DISABLED > ] ;
    [ <lHScroll: HSCROLL > ] ;
    [ MESSAGE <cMsg> ] ;
    [ FONT <oFont> ] ;
=> ;
    WITH OBJECT [ <oArrBrw> := ] TArrayBrowse():New( <oParent> ) ;
        ;:SetBounds( [<nLeft>], [<nTop>], [<nWidth>], [<nHeight>], .f. )
;

        [; :nClrText := <nClrText> ] ;
        [; :nClrPane := <nClrPane> ] ;
        [; :nStyle := <nStyle> ] ;
        [; :lEnabled := !<.lDisabled.> ] ;
        [; :lHScroll := <.lHScroll.> ] ;
        [; :cMessage := <cMsg> ] ;
        ;:Create() ;
        [; :SetFont( <oFont> ) ] ;
        [; :SetArray( <aItems>, <aHeads> ) ] ;
;END

//----- TREEVIEW -----

#xcommand @ <nTop>, <nLeft> TREEVIEW [ <oTree> ] [ OF <oParent> ] ;
    [ SIZE <nWidth>, <nHeight> ] ;
    [ COLOR <nClrText>, <nClrPane> ] ;
    [ <lDisabled: DISABLED > ] ;
    [ MESSAGE <cMsg> ] ;
    [ FONT <oFont> ] ;
=> ;
    WITH OBJECT [ <oTree> := ] TTreeView():New( <oParent> ) ;
        ;:SetBounds( [<nLeft>], [<nTop>], [<nWidth>], [<nHeight>], .f. )
;

        [; :nClrText := <nClrText> ] ;
        [; :nClrPane := <nClrPane> ] ;
        [; :lEnabled := !<.lDisabled.> ] ;
        [; :cMessage := <cMsg> ] ;
        ;:Create() ;
        [; :SetFont( <oFont> ) ] ;
;END

//----- SPLITTER -----

#xcommand @ <nTop>, <nLeft> SPLITTER [ <oSplitter> ] [ OF <oParent> ] ;
    [ SIZE <nWidth>, <nHeight> ] ;
    [ MINSIZE <nMinSize> ] ;
    [ ALIGN <nAlign> ] ;
    [ <lBorder: BORDER > ] ;
    [ <lSunken: SUNKEN > ] ;
    [ <lRaised: RAISED > ] ;
    [ <lEtched: ETCHED > ] ;
    [ <lBump: BUMP > ] ;
    [ <lFlat: FLAT > ] ;

```

```

[ COLOR <nClrPane> ] ;
=> ;
WITH OBJECT [ <oSplitter> := ] TSplitter():New( <oParent> ) ;
    ;:SetBounds( [ <nLeft> ], [ <nTop> ], [ <nWidth> ], [ <nHeight> ], .f. )
;

[; :nMinSize      := <nMinSize> ] ;
[; :nAlign        := <nAlign> ] ;
[; :nBorderStyle := bv<lBorder> ] ;
[; :nBorderStyle := bv<lSunken> ] ;
[; :nBorderStyle := bv<lRaised> ] ;
[; :nBorderStyle := bv<lEtched> ] ;
[; :nBorderStyle := bv<lBump> ] ;
[; :nBorderStyle := bv<lFlat> ] ;
[; :nClrPane      := <nClrPane> ] ;
;:Create() ;
;END

//----- TIMER -----

#xcommand DEFINE TIMER [ <oTimer> ] [ OF <oParent> ] ;
[ INTERVAL <nInterval> ] ;
[ <x: ON CLICK, ACTION> <bOnTimer> ] ;
=> ;
WITH OBJECT [ <oTimer> := ] TTimer():New( <oParent> ) ;
    [; :nInterval:= <nInterval> ] ;
    [; :OnTimer := { | oSender | <bOnTimer> } ] ;
;:Create() ;
;END

#xcommand ACTIVATE TIMER <oTimer> => <oTimer>:Enable()
#xcommand DEACTIVATE TIMER <oTimer> => <oTimer>:Disable()

//----- FONT -----

#xcommand DEFINE FONT [ <oFont> ] ;
[ NAME <cName> ] ;
[ SIZE <nWidth>, <nHeight> ] ;
[ <lBold: BOLD> ] ;
[ <lItalic: ITALIC> ] ;
[ <lUnderline: UNDERLINE> ] ;
[ ESCAPEMENT <nEscapement> ] ;
=> ;
WITH OBJECT [ <oFont> := ] TFont():New() ;
    [; :nWidth := <nWidth> ] ;
    [; :nHeight := <nHeight> ] ;
;:Create( [ <cName> ] ) ;
[; :lBold := <.lBold.> ] ;
[; :lItalic := <.lItalic.> ] ;
[; :lUnderline := <.lUnderline.> ] ;
[; :nEscapement := <nEscapement> ] ;
;END

//-----

```

1.18 More Help

Xailer uses as programming language CA-Clipper, probably the more powerful and used dialect of Xbase. The documentation included with CA-Clipper is indispensable tool for any person who wishes to begin with Xailer, except the predefined classes which are useless. If you never been a CA-Clipper user you may find documentation about it on this link:

<http://www.itlnet.net/programming/program/Reference/c53g01c/menu.html>

CA-Clipper as a commercial product become discontinued and only a 16 bits version is available, for that reason the xBase community decided to create a substitute for CA-Clipper but for 32 bits, and not only for Microsoft environments, but also for Linux, MacOS and OS2 among others. Actually there are various products that offer the same functionality that the original CA-Clipper on 32 bits and with even more power, functionality and speed. **Harbour** and **xHarbour** are two examples of them. Both products maintain complete compatibility with the classic CA-Clipper but also offer great enhancements. They are free 'Open Source' products that can be freely used by any end user.

Xailer uses **Harbour** as programming language. You may find documentation and support on the following links:

<https://harbour.github.io/>
[http://en.wikipedia.org/wiki/Harbour_\(software\)](http://en.wikipedia.org/wiki/Harbour_(software))

1.19 Appendix

1.19.1 Colors

To indicate the color, you can use several ways to do it:

- Using the RGB(Red, Green, Blue) function where the Red, Green and Blue parameters are numbers between 0 and 255. For example RGB(0, 0, 0) is the black color and RGB(255, 255, 255) is the white color.
- Using a predefined Xailer colors:

cltGray	0xC0C0C0
clDkGray	0x808080
clAliceBlue	0xFFFF8F0
clAntiqueWhite	0xD7EBFA
clAquamarine	0xD4FF7F
clAzure	0xFFFFF0
clBeige	0xDC5F5
clBisque	0xC4E4FF
clBlanchedAlmond	0xCDFFFF
clBlueViolet	0xE22B8A
clBrown	0x2A2AA5
clBurlyWood	0x87B8DE
clCadetBlue	0xA09E5F
clChartreuse	0x00FF7F
clChocolate	0x1E69D2
clCoral	0x507FFF
clCornflowerBlue	0xED9564
clCornsilk	0xDC8FF
clCrimson	0x3C14DC
clCyan	0xFFFF00
clDarkBlue	0x8B0000
clDarkCyan	0x8B8B00
clDarkGoldenRod	0x0B86B8
clDarkGray	0xA9A9A9
clDarkGreen	0x006400
clDarkKhaki	0x6BB7BD

clDarkMagenta	0x8B008B
clDarkOliveGreen	0x2F6B55
clDarkOrange	0x008CFF
clDarkOrchid	0xCC3299

- Or using the system predefined colors:

clScrollBar	GetSysColor(COLOR_SCROLLBAR)
clBackground	GetSysColor(COLOR_BACKGROUND)
clActiveCaption	GetSysColor(COLOR_ACTIVECAPTION)
clInactiveCaption	GetSysColor(COLOR_INACTIVECAPTION)
clMenu	GetSysColor(COLOR_MENU)
clWindow	GetSysColor(COLOR_WINDOW)
clWindowFrame	GetSysColor(COLOR_WINDOWFRAME)
clMenuText	GetSysColor(COLOR_MENUTEXT)
clWindowText	GetSysColor(COLOR_WINDOWTEXT)
clCaptionText	GetSysColor(COLOR_CAPTIONTEXT)
clActiveBorder	GetSysColor(COLOR_ACTIVEBORDER)
clInactiveBorder	GetSysColor(COLOR_INACTIVEBORDER)
clAppWorkSpace	GetSysColor(COLOR_APPWORKSPACE)
clHighlight	GetSysColor(COLOR_HIGHLIGHT)
clHighlightText	GetSysColor(COLOR_HIGHLIGHTTEXT)
clBtnFace	GetSysColor(COLOR_BTNFACE)
clBtnShadow	GetSysColor(COLOR_BTNshadow)
clGrayText	GetSysColor(COLOR_GRAYTEXT)
clBtnText	GetSysColor(COLOR_BTNTEXT)
clInactiveCaptionText	GetSysColor(COLOR_INACTIVECAPTIONTEXT)
clBtnHighlight	GetSysColor(COLOR_BTNHIGHLIGHT)
cl3DDkShadow	GetSysColor(COLOR_3DDKSHADOW)
cl3DLight	GetSysColor(COLOR_3DLIGHT)
clInfoText	GetSysColor(COLOR_INFOTEXT)
clInfoBk	GetSysColor(COLOR_INFOBK)
clHotLight	GetSysColor(COLOR_HOTLIGHT)
clGradientActiveCaption	GetSysColor(COLOR_GRADIENTACTIVECAPTION)
clGradientInactiveCaption	GetSysColor(COLOR_GRADIENTINACTIVECAPTION)

1.19.2 Keyboard Shortcuts

Code editor:

Regular keystrokes are (besides navigation and selection keystrokes):

Ctrl + A	Select all text
Ctrl + B	Expandir abbreviation
Ctrl + C	Copy
Ctrl + Shift + C	Comments / Uncomments current line or text block
Ctrl + D	Duplicate actual line
Ctrl + E	Selects the text that completes the current bracket at the cursor position
Ctrl + F	Search text

Ctrl + Shift + F	Search text on all project files
Ctrl + G	Go to line number
Ctrl + I	Inverts current assignment. Instructions of type: <a> := are converted to := <a>
Ctrl + L	Delete actual line
Ctrl + N	New module
Ctrl + O	Open file
Ctrl + R	Search and replace text
Ctrl + S	Save the document
Ctrl + T	Swap current line with the one above
Ctrl + V	Paste
Ctrl + X	Cut
Ctrl + Y	Redo
Ctrl + Z	Undo
Ctrl + +	Increase text size
Ctrl + -	Decrease text size
Ctrl + *	Copies the actual object inspector item, including its value, to the code editor
Ctrl + /	Reset text size
Ctrl + F1	Help on current token
Ctrl + F2	Set text mark
F2	Go to next mark
Shift + F2	Go to previous mark
Ctrl + Shift + F2	Remove all marks
F3	Search next occurrence
Tab	Indent selection
Shift + Tab	Unindent selection
Alt + '-'	Colapse block
Alt + '+'	Expand block
Alt + Up	Go to beginning of block or to previous block
Alt + Down	Go to next block
Ctrl + Alt + '-'	Colapse all blocks
Ctrl + Alt + '+'	Expand all blocks
Shift + End + Del	Delete from actual cursor position to the end of line
Alt + Mouse text selection	Columnar text selection
Ctrl + F7	Moves to next message on messages area
Ctrl + Shift + F7	Moves to previous message on messages area
Ctrl + Shift + Up	Moves selected text up
Ctrl + Shift + Down	Moves selected text down

Form editor:

F2	Selects <i>cText</i> property on object inspector
----	---

Object inspector:

Regular keystrokes are (besides navigation and selection keystrokes):

Ctrl + Intro	Show edition dialog for members with a orange triangle
F2	Opens edition dialog of the element or shows its enumerated values
Ctrl+ *	Inserts on the code editor the active element with the following syntax:

```
::oControl:xMember :=
```

Messages window:

Regular keystrokes are (besides navigation and selection keystrokes):

Ctrl + Supr	Deletes actual line
-------------	---------------------

1.19.3 Mouse Shortcuts

Code editor:

Ctrl+Click on folding buttons	Expands the current and any lower folding
Shift+Click en los puntos de folding	Colaps the current and any lower folding
Ctrl+Click on left margin	Sets or unsets Bookmarks
Shift+Click on left margin	Sets or unsets Break points
Alt + Mouse select	Selects text in columnar style
Doble Click on editor tab	Selects file in project inspector

1.19.4 Special Characters

From the Object inspector there are some characters that can not be typed directly, like a simple 'New Line'. On those cases you may use the following special characters:

CR (carriage return)	\r
LF (new line)	\n
TAB (tab)	\t
NULO (Null character)	\0 (zero)
SLASH	\\

1.19.5 Predefined resources






Predefined resources are a collection of icons, cursors and images that Xailer uses internally on certain situations that shows forms like the Preview printing form. This resources can be also accesed by the programmer for its own use.

Icons:





















XA_ICO_XAILER



Cursors:

XA_CUR_DRAG	
XA_CUR_SPLITTERH	
XA_CUR_SPLITTERV	
XA_CUR_ZOOMIN	
XA_CUR_ZOOMOUT	

Bitmaps:

XA_BMP_DNARROW	
XA_BMP_EXIT	
XA_BMP_FIRST	
XA_BMP_NEXT	
XA_BMP_PREV	
XA_BMP_LAST	
XA_BMP_INSERT	
XA_BMP_DELETE	
XA_BMP_EDIT	
XA_BMP_OK	
XA_BMP_CANCEL	
XA_BMP_REFRESH	
XA_BMP_OK16	
XA_BMP_CANCEL16	
XA_BMP_OPENTYPE	
XA_BMP_TRUETYPE	
XA_BMP_COPY	
XA_BMP_PASTE	
XA_BMP_METALTEXTURE	
XA_BMP_PRINTER	

Índice

- : -

:SetMaster 1541

- A -

aAccels 695

aAlignment 1496

aArguments 109

aArrayData 441

aAttachments 720, 781

aBCC 720

AbortDoc 136

aButtons 1133

aCC 720

aChecked 942

aClrKeys 967

aColors 319

aCols 407, 812

aColumns 513, 1472

aComponents 89

aControls 77

aCustomColors 1079

aData 883, 1342, 1490, 1545

aDataSet 1181

aDatasets 1509

aDays 870

aDaysSelected 871

Add 1029, 1439

AddArray 1515

AddBCC 727

AddBitmap 612

AddButton 650

AddCalcField 1285

AddCategory 1515

AddCC 727

AddCol 418, 829

AddColumn 520, 1481

AddData 121

AddDataset 1183, 1515

AddDbf 1516

AddField 1343

AddFile 1462

AddFunction 1516

AddGroup 521, 1482

AddIcon 613, 1029, 1034

AddIdxFile 1322

AddImage 419, 634, 674, 947

AddItem 261, 284, 295, 326, 363, 371, 381, 498, 521, 554, 613, 635, 650, 674, 688, 890, 928, 948, 990, 1011, 1023

AddLinkedControl 1286

AddMember 1427

AddMenu 300

AddNew 1286

AddParam 1357

AddReceipt 727

AddRequestHeader 762

AddRow 442

AddSection 213

AddSeparator 295, 364, 372, 651

AddSupraCol 829

AddTag 1322

AddTempTag 1323

AddThreadTask 1608

AddTool 155, 248

AddUserVar 1287

AddVariable 1517

aDescription 1555

aDisplayCols 813

Adjust 277, 419, 1482

AdjustCols 419, 829

AdjustColWidth 830

aDropFiles 71

aEditListBound 449, 851

aEditListText 449, 851

aFields 1277, 1544, 1545

aFileMask 1446, 1457

aFiles 1447, 1457

aFonts 1472, 1497

aForms 109

aFrom 1109

aGridData 813

aGroupCols 865

aGroups 513, 1473

aHolidays 539

aldxFiles 1318

almages 482
altems 257, 280, 293, 324, 360, 367, 379, 392,
479, 482, 489, 493, 513, 534, 551, 609, 630, 645,
667, 685, 799, 884, 906, 923, 942, 983, 1001, 1008,
1020, 1437
altemsBound 480, 1378, 1381, 1398, 1401
Ajoin 1686
aKeys 968
Alias 1323
Alignment 274, 920
aLine 1497
aLinkedControls 1277
AllowColHiding 814
AlternateColor 815
AltSortChildren 689
aMails 751
aMargins 561, 1140
aMinMargins 1140
aMonitors 148
aMonths 871
AngleArc 167, 1670
Animations 1612
AnsiToWide 1629
aPaperSize 1140
aPenPos 163
aPicture 1490
Append 291, 420, 830, 1324
ApplyLevels 344
AppWindow 90
Arc 168, 1670
ArcTo 168, 1671
Area 1324
aReceipts 721, 781
AreGroups 1482
aRelationFields 1539
aRenamed 1109
aResults 1355
Arithmetical bit operation functions 1623
Arrange 522
aSDays 871
aSelected 407, 813, 884, 942
aSMonths 871
aSQLParams 1355, 1366
Assign 1287
AsString 1424
aStartValues 1620

aStopValues 1621
aThreadTasks 1607
aTitle 1490
aTo 1109
aTooltips 1410
aTotal 1500
Atranspose 1629
Attachments 742
AttemptConnect 762
Attributes 136
aTypes 225
aUserFields 1278
Authorize 750
Average 866

- B -

BackLine 1482
Beep 1137
BeginPaint 53, 1011
BeginTrans 1204, 1212, 1222, 1231, 1242, 1253
bFooter 1500
bFor 1473
bGroup 1500
bHeader 1500
BlackWhite 344
Blank 1273, 1287
Bof 1288
Bookmark 420
BrowseComputers 1076
BrowseFile 1076
BrowseFolder 1076
BrowsePrinters 1077
bSkip 1473
Buffer 1268
BuildSQLSt 1183
BulkExecute 1253
bWhile 1473

- C -

cAccount 782
cAddress 721
cAgent 759
Calc 1517

Calculator 1638
cAlias 446, 1318
Cancel 713, 717, 1213, 1288
CancelEdit 420, 522, 830
CanClear 578
CanCopy 270, 579
CanCut 271, 579
CanEdit 1288, 1352
CanGoBack 704, 731
CanGoForward 704, 731
CanonicalizeUrl 763
CanPaste 271, 579
CanPrintToPdf 136
CanRedo 579
CanUndo 271, 580
Cargo 32
CategoriesList 1517
cAuthor 1556
cBalloon 38
cBCC 742, 747
cBin 1577
cBody 721
cBuffer 711
cCachePath 730
cCC 742, 747
cCertFile 1583
cCharset 721
cClassname 1056
cClientId 747
cCommandLine 1051
cComment 1457
cComments 109
cCompany 110
cConnect 1181, 1191, 1200, 1207, 1239, 1249, 1262
cContact 1583
cCopyright 110
cCreationTime 1122
cCryptKey 1249
cCSDVersion 239
cCustomFormat 484, 642
cDatabase 1220, 1229, 1250
cDataField 1371
cDefaultExt 1099, 1115
cDescription 110, 190, 1065
cDeviceId 1453
cDirectory 110, 1171, 1447
cDisplayName 1065, 1268
cDllPath 1509
CdoMail:Properties 741
cDrive 1130
cDriver 1191, 1196, 1197, 1200
cDSN 1262
cError 712, 715, 1584
cErrorMsg 1051
cExeFile 1573, 1577
cExePath 1573, 1577
cExtra 722
cFieldName 450
cFileMask 1123
cFileName 111, 204, 212, 561, 1099, 1123, 1147, 1432, 1447, 1458, 1509, 1574
cFilenameOnly 1100
cFilePath 1100
cFilter 450, 505, 852, 937, 1100
cFindText 561
cFontName 562
cFooter 450, 532, 852
cFRLicense 1510
cFrom 742
cFromAddress 782
cFromName 782
cFullFilename 1101
cFullName 1269
cFullPath 1123
cGrandTotal 1473
cGroup 964
CharFromPos 580
cHeader 451, 532, 852, 937
CheckAll 948
CheckConnection 763
cHint 917, 1381
ChooseColorDlg 1084, 1639
ChooseFontDlg 1096, 1639
Chord 168, 1671
cHost 1220, 1229
cHtml 974
cHtmlColor 1079
cIniFile 1437
cIniSection 1438

cInitialDir 1101
cInitString 1556
cInputFile 1584
cJobName 1474
cLabel 1131
cLanguage 748, 974, 1510
cLastAccessTime 1123
cLastError 1250, 1278, 1447, 1458, 1510, 1574
cLastMail 751
cLastWriteTime 1123
CleanUserData 702, 704
Clear 291, 580, 1518
ClearAllFilters 420, 929
ClearData 1518
ClearFilter 830, 929, 1289
ClearHistory 732
ClearMasterDetail 1518
ClearOptions 1555
ClearRelations 1324
ClearScope 1325
ClearTicks 664
ClearVariables 1518
Click 252, 255, 305, 657, 978
Clipboard functions 1627
Clipper 1716
cLocale 730
cLocalFile 715
cLocation 1584
Clone 161, 188, 198, 224, 229, 1289, 1325, 1344, 1427
Close 99, 179, 192, 206, 210, 625, 713, 717, 754, 763, 788, 791, 1289, 1357, 1564
CloseDevTools 732
CloseFile 763, 773
CloseKey 1045
CloseProcess 1184
CloseRequest 764
CloseURL 764
cMaxTime 643
cMessage 71, 293, 297, 742, 782, 1158, 1171, 1442
cMessageCancel 1158
cMessageId 748
cMinTime 643
cMsgNoFilter 447, 1368
cName 156, 187, 191, 195, 209, 220, 226, 623, 782, 788, 791, 1041, 1065, 1086, 1269, 1278, 1318, 1356, 1360, 1539
cNullValue 451
cOkBtnText 1070
ColAtIndex 421
ColAtPos 421, 831
cOldValue 1500
Collapse 689, 1023
CollapseAll 675, 1012
ColorBlend 1672
ColorDlg 580
colors 1717
ColPos 421
ColumnSeparator 816
ColWithHeader 421, 831
Command 774
Comment 136
Commit 214, 1290
CommitAll 1194, 1264
CommitTrans 1204, 1213, 1222, 1231, 1242, 1253
Communications functions 1627
Compare 199
Complete 1352
CompleteSeek 1325
ConfirmZoneCrossing 764
Connect 764, 1045, 1213, 1222, 1232, 1242, 1254
ConnectTag 1213
Continue 1326
Conversion functions 1628
Copy 192, 581
Count 754
CountBinSources 137
CountFormats 180
CountPapers 137
cOutputFile 1584
cPageNumbers 1553
cPageTotal 1474
cPassword 722, 743, 752, 760, 783, 1200, 1207, 1220, 1229, 1239, 1263, 1319, 1447, 1458, 1556, 1585
cPasswordChar 265
cPdfFilename 1474
cPhpModule 1250
cPicture 356, 358, 451, 852
cPort 124

- cPrinter 1553, 1577
 - cPrinterName 124
 - cProcess 1279
 - cPrompt 322
 - cProperty 1618
 - cRange 1578
 - cReason 1585
 - Create 34, 53, 137, 159, 162, 169, 189, 192, 199, 206, 210, 214, 221, 224, 230, 247, 248, 296, 300, 330, 364, 389, 531, 533, 536, 558, 619, 625, 640, 656, 690, 728, 773, 785, 999, 1005, 1029, 1035, 1059, 1062, 1127, 1162, 1174, 1184, 1290, 1358, 1363, 1434, 1444
 - CreateDC 153
 - CreateDirectory 773
 - CreateFont 1094
 - CreateFontDlg 1097, 1641
 - CreateFrom 1609
 - CreateFromHandle 157
 - CreateIndirect 230
 - CreateKey 1045
 - CreateNew 157
 - CreateTable 1194, 1223, 1232
 - cRecSelectorHeader 813
 - cRemotelFile 716
 - cReplace 562, 1165
 - cRequestedFilter 701
 - cReslco 1511
 - cResName 1511
 - cRoot 1070
 - Crop 344
 - cRuntimeDir 701
 - cSampleText 331
 - cSecretId 748
 - cSelect 1350, 1366
 - cServer 722, 743, 752, 788, 791
 - cShortcut 799
 - cShortFileName 1124
 - cShortPath 1124
 - cSortEval 451
 - cSoundFile 1133
 - cStatusText 1071
 - cSubject 722, 743, 748, 783
 - cSysName 239
 - cSysVer 239
 - cTable 1269
 - cTableName 1367
 - cTempDir 1546
 - cText 39, 280, 298, 328, 360, 491, 505, 529, 534, 555, 562, 609, 616, 639, 654, 685, 748, 906, 942, 996, 1003, 1020, 1070, 1110, 1117, 1133, 1158, 1166, 1423
 - cTextEmpty 1443
 - cTime 643, 783
 - cTimeStampAccount 1585
 - cTimeStampPassword 1585
 - cTimeStampURL 1585
 - cTitle 1071, 1101, 1115, 1134, 1158, 1171, 1511
 - cTo 743, 749
 - cToken 749
 - cToolBar 975
 - cToolPath 1586
 - cToolTip 39, 452, 505, 535, 616, 639, 654, 853, 937, 996, 1003, 1020, 1033
 - cTooltipTitle 40
 - cTotalPict 1491
 - cTradeMarks 111
 - cType 1269
 - cURL 336
 - CurrentDir 116, 1254
 - CurrentDrive 117
 - CurrentTask 1609
 - cUser 723, 744, 760, 1201, 1208, 1220, 1230, 1263
 - cUserDataDir 701
 - cUsername 752
 - Cut 581
 - cValue 1501
 - cVersion 111
 - cWinNTProduct 240
 - cWorkingDir 1052
 - cXALicense 1511
 - cXLSDData 1474
- D -**
- DataCol 1370
 - DataControls:Introduction 1174
 - Date functions 1634
 - dCreationDate 1124
 - dDate 783
 - Debug functions 1635

Dec 308
DecimalSeparator 117
Decrypt 1686
DefaultBin 126
DefDataExtension 1194, 1204
DefIdxExtension 1195, 1197, 1198, 1205
DefMemoExtension 1195
DefValue 1270
DelCol 422, 831
DelColumn 522, 1482
DelDataset 1184, 1519
DelError 1290, 1519
DelErrors 1184
Delete 365, 378, 422, 491, 537, 559, 641, 657, 690, 755, 831, 901, 999, 1006, 1023, 1291, 1326, 1439
DeleteAllCookies 704
DeleteAllItems 522
DeleteCategory 1519
DeleteChilds 690, 1024
Deleted 1327
DeleteDC 153
DeleteEntry 214
DeleteFile 774, 1254
DeleteFiles 1450
DeleteItem 262, 284, 381, 498, 614, 635, 651, 891, 929, 948, 990
DeleteItems 284, 382, 499, 614, 635, 675, 690, 891, 930, 948, 991, 1012, 1024, 1439
DeleteKey 1045
DeleteSection 215
DeleteSections 215
DeleteTag 1327
DeleteValue 1046
DeleteVariable 1519
DelGroup 523, 1483
DelLinkedControl 1291
DelProcess 1185
DelRow 442
DelTable 1195, 1214, 1223, 1232, 1243, 1264
DelTag 1327
DelTempTag 1328
DelTool 248
Descripcion del color en formato HTML 1084
Destroy 34, 54, 122, 137, 158, 162, 189, 193, 200, 211, 221, 224, 230, 296, 301, 620, 626, 728, 765, 786, 1030, 1035, 1046, 1062, 1128, 1162
DeviceInfo 1454
Devices 1455
Dialog functions 1637
Directory 774
Disable 54, 492, 1062
Disconnect 1046, 1214, 1223, 1233, 1243, 1254
dLastAccessDate 1124
dLastWriteDate 1125
DLL creation 1621
DLLs 1621
DLLs and execution functions 1660
dMaxDate 485, 539
dMinDate 485, 539
DocumentSetup 138
DoDefault 296
DoExport 1520
DownloadFile 775
DrawPicture 169
DrawRichEdit 170
Drive 1432
DriverName 138
dToday 539
DtoSql 1629
DTToSql 1630
dValue 485, 539, 872

- E -

Edit 422, 462, 537, 690, 832, 1291, 1564
EditFilter 499
EditLabel 1012
Ellipse 170, 1672
Email 1564
EMF functions 1664
Empty 180
EmptyUndoBuffer 271, 581
Enable 55, 492, 1062
Encrypt 1243, 1686
End 35, 620, 1053, 1292
EndDoc 138
EndEditLabelNow 675, 1012
EndLine 1483
EndPage 138

- EndPoint 55, 1012
- EndProcess 1185
- EnsureVisible 537, 691, 1024
- EnumFormatNames 181
- EnumFormats 180
- EnumKeys 1046
- EnumSectionEntries 215
- EnumSectionValues 215
- EnumValues 1047
- EnumWindows 1695
- Eof 1292
- ErrorDialog 765
- EventAssigned 35
- Eventos 1503
- Execute 1075, 1082, 1094, 1106, 1113, 1120, 1131, 1138, 1145, 1155, 1162, 1167, 1173, 1214, 1224, 1233, 1243, 1255, 1265, 1451, 1462, 1661
- ExecuteScript 705
- Expand 691, 1024
- ExpandAll 675, 1013
- ExpandPartial 691
- Export 1565
- ExportBIF 1565
- ExportBMP 1565
- ExportCSV 1566
- ExportEMF 1566
- ExportExcel 1566
- ExportGIF 1566
- ExportHTML 1566
- ExportJPEG 1567
- ExportMail 1567
- ExportODS 1567
- ExportODT 1567
- ExportPDF 1568
- ExportRTF 1568
- ExportTIFF 1568
- ExportTXT 1568
- Extension 1433
- ExtFloodFill 170, 1673
- F -**
- FastAddNew 1292
- FastEdit 1293
- FastReport 1504
- FastReport Register 1508
- FastUpdate 1294
- FieldCount 1294
- FieldDec 1295, 1343
- FieldDisplayName 1295
- FieldExist 1295
- FieldGet 1274, 1295
- FieldGetByName 1296
- FieldIsNull 1352
- FieldLen 1296
- FieldName 1296
- FieldNames 1297
- FieldPos 1297
- FieldPut 1274, 1297
- FieldSQLType 1297
- FieldType 1298
- FieldValue 1520
- File 1215, 1224, 1233, 1244, 1255, 1265
- File functions 1664
- FileDateTime 1667
- FileDateTimeLocal 1667
- FileOpen 1576
- FileOpenDlg 1108, 1642
- FileOpenImageDlg 1107
- FileSaveDlg 1116, 1643
- FileSaveImageDlg 1116
- FilesCount 1450
- FilesInfo 1450
- FilesSize 1451, 1462
- FileLock 1328
- FillRect 171
- Filter 1298, 1345
- FilterEval 452, 853
- FileUnlock 1328
- Find 1128, 1568
- FindDlg 581
- FindNext 1128
- FindText 582
- FirstChild 1024
- Firsts steps 6
- FirstVisible 949
- FlipH 344
- FlipV 345
- Flush 1047
- FontDlg 582
- ForceDir 1255

ForcedSeek 1329
ForceltemVisible 949
Found 1329
Free 35
FullName 1433
Func:FileOpenImageDlg 1107
Func:FileSaveImageDlg 1116
Function:FileOpenDlg 1108
Function:FileSaveDlg 1116
Functions 1623
Futures 1597

- G -

GetAccountNames 786
GetActiveForm 1705
GetAllItems 676, 1013
GetArray 216
GetBinary 1047
GetBinSourceNames 139
GetBinSourceTypes 139
GetBitmap 1030, 1059
GetBmpFromColor 1673
GetBullet 582
GetCaptureDateTime 231
GetCatalogs 1215, 1224, 1234, 1256
GetCol 582
GetComputerName 1695
GetConnectedState 765
GetCount 676, 1013
GetCurrentDirectory 775
GetCurSel 284
GetData 181
GetDataRow 832
GetDate 216, 1048
GetDefaults 1427
GetDrives 1667
GetDropFiles 182
GetEntry 216
GetEnvironmentVariable 1695
GetErrorDescription 766, 1425
GetErrors 1185, 1521
GetEventMask 583
GetFile 775
GetFileSize 766, 776
GetFileVersionInfo 1661
GetFirstVisibleLine 583
GetFocused 523
GetFormatName 182
GetHorzExtent 285
GetHtml 975
GetIcon 1030, 1059
GetIconIndex 1059
GetIconName 1060
GetIconSize 1031
GetIconType 1060
GetImageCount 1031
GetIndent 584
GetItemAtPos 1013
GetItemByHandle 676
GetItemByID 1014
GetItemByName 676, 1014
GetItemByPos 372
GetItemCount 373
GetItemPos 373, 1014
GetKeys 1428
GetLine 583
GetLineCount 583
GetLineFromChar 584
GetLineIndex 584
GetLogical 217, 1048
GetMainMenu 301
GetMember 1429
GetMessageCount 792
GetModify 584
GetName 1428
GetNextSibling 691, 1024
GetNumbering 585
GetNumeric 217, 1048
GetNumTicks 664
GetOptions 585
GetOwner 183
GetPaperNames 139
GetPaperSizes 139
GetPaperTypes 140
GetParagraphSpacing 585
GetPicture 1425
GetPictureHandle 231
GetPos 585
GetPrevSibling 692, 1025

GetPrimaryKeys 1353
GetPrinterDefault 140
GetPrinterFonts 140
GetPrinterNames 140
GetPrintHeight 586
GetProperty 1521
GetRecord 1299
GetRedoName 586
GetRequest 766
GetResolutions 141
GetRow 442, 586, 1299, 1345
GetRowHash 1300
GetRows 1300
GetRowsHash 1300
GetRTF 586
GetRTFRange 587
GetScope 1329
GetSection 217
GetSectionEx 217
GetSections 218
GetSectionsEx 218
GetSel 272, 587
GetSelectedItem 677, 1015
GetSelection 523, 587
GetSelectionType 588
GetSellItems 285
GetSelText 272
GetSiblingControl 56
GetString 1048
GetStructFrom 1345
GetTables 1215, 1224, 1234, 1244, 1256, 1265
GetTags 1330
GetTempFileName 1666
GetText 183, 588
GetTextHeight 200
GetTextMode 588
GetTextRange 588
GetTextSize 200
GetTextWidth 200
GetThumbnailSize 231
GetTick 665
GetTickPos 665
GetTicks 665
GetTitle 705
GetTypographicOptions 589
GetUndoName 589
GetUserName 1696
GetValue 443
GetValues 1428
GetVariable 1522
GetVersion 766
GetVersionEx 1696
GetVisibleCount 523, 677
GetWindow 183
GetWindowsLanguage 1681
GetWinPlatform 1696
GetZoom 589
GetZoomLevel 732
GoBack 705, 732
GoBottom 422, 832, 1301
GoDown 422, 832
GoFirstControl 79
GoForward 705, 733
GoHPos 423
GoLeft 423, 833
GoLeftMost 423, 833
GoNextControl 56, 80
GoPageDown 423
GoPageUp 424
GoPrevControl 56, 80
GoRight 424, 833
GoRightMost 424, 833
Goto 424, 1301
GotoLine 589
GoTop 425, 833, 1301
GoUp 425
Graphic functions 1668
GrayScale 345
GroupOn 834

- H -

Handle 36, 149, 686
Harbour 1716
Harbour multitasking 1594
HasChanged 1274
HasChilds 1025
HasData 184
HasDocument 733
HasDrive 1433

HasExtension 1433
HasMessages 792
HasPassword 1451
HasPath 1434
HasText 590
hBitmap 160
hConnect 1201
hCurrentKey 1043
hDC 125, 149, 163, 1150
Header 867
HeightToSize 1673
Help 1716
HelpID 38, 1080, 1086, 1101, 1118, 1133, 1141, 1150, 1166
Hex 1630
HexToDec 1630
HexToString 1630
Hide 57, 1035
HideBalloon 57, 1035
HidePopup 733
HideTooltip 57
HitTest 327, 373, 499, 524, 677, 834, 891, 930
HitTestCol 834
hNewDC 191
HotTrack 924
hPalette 226
hProcess 1052
hSession 760
hThread 1052
HTMLToRTF 590

- | -

i18n 1682
i18n_Language 1682
ImageSize 1455
Inc 309
InitFrom 1302
Initial 1361
Initialize 117, 122, 137, 153
InitValue 1270
InsCol 425, 834
Insert 425, 835
InsertBitmap 590
InsertControl 80
InsertFile 591
InsertItem 262, 285, 365, 373, 382, 499, 524, 636, 652, 677, 692, 891, 930, 949, 991, 1015, 1025
InsertObject 591
InsertObjectFromFile 591
InsertPageBreak 591
InsertPicture 592
InsertRTF 592
InsertSeparator 652
InsertTable 592
InsertText 593
InsRow 443
Install 1066
Installed 786
Introduction 6, 1174
IsActive 901, 1541
IsAllDigits 1687
IsBitmap 233
IsBof 426
IsCancel 1162
IsCharAlphaNumeric 1687
IsComplete 1353
IsDir 1256
IsDuplex 141
IsEditable 426
IsElevated 1696
IsEmpty 184, 525, 678, 931, 1015
IsEnhMetafile 233
IsEof 426
IsError 1185
IsExpanded 693
IsFilter 835, 931
IsFiltered 443
IsIcon 233
IsInputParam 1363
IsInstalled 706, 1066
IsItemVisible 949
IsLoaded 1541
IsMetafile 234
IsMultipleEdit 426
IsOcXInstalled 1662
IsOnEdit 835
IsOpen 788, 792
IsOutputParam 1363
IsPicture 234

IsPopupVisible 733
IsPrinting 141
IsPrivateFormat 184
IsReadOnly 1302
IsRecordLocked 1330
IsRTF 593
IsRunning 1042, 1054, 1066, 1662
IsSelected 901
IsSelection 593
IsSingleEdit 835
IsSorted 836
IsStringNumeric 1687
IsSystemFormat 184
IsUserAnAdmin 1697
IsWinVista 242
ItemChecked 950
ItemHot 950
ItemPushed 950
ItemSelected 950

- J -

JobTitle 124

- K -

Keyboard functions 1679
Keyboard shortcuts 1718
KeyCount 426, 1302
KeyGoto 1303
KeyNo 427, 1303
Kill 1054
KillProcess 1662

- L -

IAborted 125, 1110
IAbortOnErrors 1181, 1343, 1512
IActive 89, 623
IAddColor 320
IAdsLocking 1201
IAlignKeepSpace 40
IAlignLeft 514
IAlignShiftOnRotate 40
IAllFonts 1087

IAllowAppend 407, 814
IAllowColHiding 407
IAllowColSizing 407
IAllowColSwapping 408, 814
IAllowDelete 408, 814
IAllowDevTools 701
IAllowEdit 408, 815, 1549
IAllowInsert 408, 815
IAllowMultiSelect 1102
IAllowRowSizing 408
IAllowScripts 701
IAllowSizing 814
IAllowsNull 1270
IAllowSort 815
IAllowUndo 1110
IAlt 509
IAlwaysTip 645
IAnd 1624
IAnyColor 1080
IAppModal 1134
IArchive 1125
IArrowKeys 696
IAskPassword 784
IAskRecipients 784
LastAwaitTask 1607
LastChild 1025
LastError 1185
LastVisible 951
IAsynFetch 1351
IAuth 723
IAuthenticate 744
IAutoArrange 514
IAutoBuddy 696
IAutoCommit 1263
IAutoFit 339
IAutoHScroll 257, 289, 667
IAutoLand 1475
IAutoOpen 1198
IAutoOrder 441, 1368
IAutoPlay 623
IAutoReconnect 1220, 1230
IAutoRemove 1438
IAutoRename 1110
IAutoReverse 1614

IAutoSave 409, 816, 1373, 1375, 1378, 1381, 1384, 1386, 1388, 1390, 1391, 1393, 1399, 1402, 1404, 1406, 1408, 1416, 1418, 1421, 1438
IAutoScale 111
IAutoScroll 82, 544
IAutoSeek 1368
IAutoSelect 266, 480
IAutoSize 275, 350, 367, 646, 654
IAutoSort 280
IAutoTicks 659
IAutoTime 1159
IAutoVScroll 289
IBallons 646
IBandBorders 551
IBold 195, 562, 686, 1087
IBorder 72, 266, 280, 493, 509, 514, 540, 551, 563, 668
IBreak 556
IBrowseComputer 1071
IBtnCancel 1410
IBtnClose 1558
IBtnDelete 1410
IBtnEdit 1410, 1558
IBtnEmail 1558
IBtnExport 1558
IBtnFind 1559
IBtnFirst 1411
IBtnInsert 1411
IBtnLast 1411
IBtnNav 1559
IBtnNext 1411
IBtnOpen 1559
IBtnOutline 1559
IBtnPage 1560
IBtnPDF 1560
IBtnPrevious 1412
IBtnPrint 1560
IBtnRefresh 1412
IBtnSave 1560
IBtnThumb 1560
IBtnUpdate 1412
IBtnZoom 1561
IBusy 112
IButton 895
IButtons 494
IButtonStyle 630
ICalcDirectory 1171
ICancel 249, 805, 1475
ICancelDrag 494
ICancelTrack 494
ICanDrag 854, 924, 937
ICanEdit 855
ICanFilter 453, 855, 938
ICanHide 855, 924, 938
ICanHot 855, 938
ICanResize 855, 938
ICanSort 856, 938
ICardBorder 884
ICardShadow 884
ICenter 1071, 1080, 1087, 1102, 1118, 1134, 1141, 1151, 1159, 1166
ICentered 90, 623
ICheckBox 485
ICheckBoxes 514, 668, 942
IChecked 253, 298, 303, 360, 535, 654, 686, 902, 964, 977, 1020
IChildEdge 556
ICloseBox 90
IColDividerComplete 409
ICollapsible 361
ICollapsed 865
ICollate 125, 1151, 1553
IComboFont 331
ICompactPath 1159
ICompressed 1556
ICompressed 1125
IConnectAsync 1208
IConnected 712, 716, 753, 1182
IContextMenu 702
IConvertNulls 1547
ICreated 32
ICreatePrompt 1102
ICtrl 510
IDataMultiLine 856
IDateAsString 1239, 1263
IDBAutosize 1397
IDblCikToggle 552
IDefault 250, 298, 806
IDefMinMargins 1141
IDelFRVars 1512
IDelOnRead 753
IDigest 753

- IDirectory 1125
- IDirectShortCuts 90
- IDisableNoScroll 306, 563
- IDisplayErrors 1182, 1279, 1356
- IDontAddToRecent 1102
- IDontGoBelowDomain 1071
- IDoubleBuffer 515, 668
- IDoubleBuffered 1549
- IDoublePass 1547
- IDoubleQuotes 1241
- IDown 1118
- IDownsLeft 659
- IDpiAware 112
- IDragAcceptFiles 72
- IDragDrop 494
- IDragDropItem 668
- IDropAnimation 907, 913
- IDropped 258, 799, 907
- IDropShowDOW 913
- IDropShowFirstOfGroup 913
- IDropShowLines 913
- IEditable 480, 1270, 1374, 1376, 1378, 1382, 1384, 1386, 1388, 1390, 1392, 1393, 1399, 1402, 1404, 1406, 1408, 1417, 1419, 1421
- IEditBmp 331
- IEditBox 1072
- IEditLabels 515, 668, 1009
- IEditMode 409
- IEffects 1087
- LeftPos 856
- IEject 1501
- Len 593, 1429
- IEnabled 41, 298, 361, 639, 655, 996, 1002, 1061, 1614
- IEnableSizing 1103
- IEncrypted 1125
- IExact 401
- IExitWhenDone 1578
- IExpanded 328, 361, 1021
- IExtendedUI 258, 1072
- IFadeInOutExp 669
- IFastAppend 816
- IFastEdit 409, 816, 846
- IFileMustExist 1103
- IFilesOnly 1111
- IFillShape 971
- IFilterBar 409, 495, 817, 924
- IFindCase 563
- IFindDown 563
- IFindWord 564
- IFixedBmp 556
- IFixedLength 659
- IFixedOrder 552
- IFixedPitch 332, 1087
- IFixedSize 556
- IFixedWidth 630, 983
- IFlat 316, 495, 646
- IFlatButton 630
- IFocused 41
- IFooter 410, 817, 865
- IForceCreate 1073
- IForceFontExist 1088
- IForceIconLeft 631
- IForceLabelLeft 631
- IForceRefresh 1618, 1620
- IForceShowHidden 1103
- IFullDrag 495
- IFullOpen 1080
- IFullPath 1442
- IFullRowSelect 515, 669
- IFullSize 610
- IGapFirstCard 885
- IGmailOptions 744
- IGridLines 515
- IGripper 403, 556
- IGroup 655
- IGroupView 515
- IHasButtons 669
- IHasLines 669, 1009
- IHeader 410, 817
- IHeaderDragDrop 516
- IHelpButton 1134
- IHelpIcon 91
- IHidden 1126
- IHideOnClose 91
- IHidePrintToFile 1151
- IHideReadOnly 1103
- IHideScrollbars 82, 817, 885, 943
- IHideToolBar 1574
- IHighLiteToday 872
- IHilited 299

IHot 564
IHotBorder 868
IHotTrack 495, 631, 670, 696, 817, 885, 895, 943, 968
IHotTrackSelect 516
IHScroll 266, 280, 289, 410, 564
IHtml 564, 744, 749
IHtmlText 943
IIgnoreDevByZero 1547
IIncludeFiles 1072
IIncludePath 1448, 1460
IIncludeURLS 1072
LinesLeft 1483
LineTo 171
IInInches 1142
IInstaled 744
IIntegralHeight 281
List 755
IIsWin10 240
IIsWin2000 241
IIsWin7 240
IIsWin98 240
IIsWinHomeServer 242
IIsWinMe 241
IIsWinServer2003 241
IIsWinServer2016 242
IIsWinServer2019 242
IIsWinXP 241
IIsWinXP64 241
IIsWow64 243
IItalic 195, 564, 1088
IKeyExact 1319
ILabelTip 516
ILabelWrap 516
ILargeIcons 1057
ILegacyBehaviour 806
ILine 865
ILinesAtRoot 670
IList 646
ILoadOnDemand 1539
ILog 723
ILoggedOn 784
ILoop 1615
IMargins 1142
IMarquee 546, 957, 961
IMasksComPath 1461
IMatchCase 1118, 1166
IMaximizeBox 91
IMaximized 1550
IMenuOnRight 401
IMinimizeBox 92
IMinMargins 1142
IModal 92, 204, 1159, 1550
IModBorder 956
IModified 1270
IModify 266
IMsgAuto 92
IMultiColumn 281
IMultiLine 250, 253, 276, 303, 351, 361, 631, 800, 806, 856, 895, 938, 943, 1009
IMultipleSel 281, 818, 885, 944, 950
IMultiSelect 540
IMultiSign 1586
INcaRefresh 1619, 1620
INew 1319
INewDlgStyle 1073
INewSession 784
INoBrowse 1171
INoCancel 1159
INocase 401, 1119, 1166
INoChangeDir 1104
INoColumnHeader 517
INoCompare 1351
INoConfirmation 1111
INoConfirmMkDir 1111
INoCopySecurityAttrib 1111
INoDefault 1172
INoDereferenceLinks 1104
INoErrorUI 1112
INoFaceSel 1088
INoLabel 1172
INoMargins 1142
INoMinimize 1160
INoNetwork 1143
INoNetworkButton 1104
INoNewFolderButton 1073
INoOrientation 1143
INoPageNums 1151
INoPaper 1143
INoPrinter 1144

INoProgress 1160
INormal 1126
INoScript 1088
INoSelection 1152
INoSeparateMem 1172
INoSimulations 1089
INoSizeSel 1089
INoStyleSel 1089
INoTestFileCreate 1115
INoThousands 696
INoThumb 659
INoTicks 660
INoTime 1160
INoToday 540
INoTodayCircle 540
INoUpDow 1119
INoValidate 1104
INoVectorFonts 1089
INoVertFonts 1089
INoWarning 1144, 1152
INoWholeWord 1119, 1167
INumeric 267
Load 159, 193, 211, 231, 626, 1274, 1304
LoadCommon 626
LoadFile 594
LoadFPFile 1522
LoadFromFile 158, 189, 211, 232, 626, 1522
LoadFromResource 159, 189, 211, 232, 627, 1522
LoadFromStream 233
LoadFromString 1523
LoadLangFile 1523
LoadRTF 594
LoadStyleSheetFromFile 1523
LoadStyleSheetFromString 1523
Locate 1304, 1330
Location 141
Lock 1601
IOffline 1126
LogDebug 1635
LogDisplay 1635
LogFile 1636
Logoff 786
Logon 787
IOnAppend 1304
IOnEdit 1305
IOpen 1279, 1356
IOpened 396
IOptimize 1192
IOr 1625
IOutlineExpand 1550
IOutlineVisible 1550
IOverlay 1058
IOverwrite 1461
IOverWritePrompt 1115
IOwnBorder 313
IOwnerDraw 617
IPageNums 1152
IPageTotal 1475
IParam 686
IParentFont 41, 453, 474, 895
IParentIcon 92
IPassive 772
IPassword 267
IPathMustExist 1105
IPathStyle 276
IPDF 565
IPhantomTab 983
IPictureCacheInFile 1550
IPlay 312
IPrefix 1442
IPreventFullOpen 1080
IPreview 125, 1475
IPreviewModal 125, 1475
IPrimaryKey 1271
IPrinterFonts 1090
IPrintfEmpty 1547
IPrintSetup 1152
IPrintToFile 1152
IPrintWithNext 126, 1476
IPromptIfError 1208
IProtected 565
IPushed 250, 800
IPushLike 254, 303
IRE30 565
IRE40 565
IRE50 566
IRE60 566
IRE70 566
IRE80 566
IReadOnly 267, 567, 1105, 1126, 1280

IReadToCache 1241
IReceipt 724, 784
IRecordPointer 474
IRecordSelector 410
IRecurse 1461
IRedraw 42
IResize 495, 925
IRestrictMode 1574
IReturnDC 1153
IReturnFSAncestors 1073
IReturnFSDirs 1073
IReturnIC 1153
IReuseDel 1320
IReverse 1554
IReversed 660
IRGBInit 1081
IRichTooltip 670
IRight 299
IRightJustify 631
IRightsCheck 1201
IRightText 1135
IRowHeightButtons 818
IRule 567
ISameFrameSize 983
IScalable 226
IScalableOnly 1090
IScreenFonts 1090
IScriptsOnly 1090
IScrollBarAlways 281
IScrollOpposite 632
ISelected 1021, 1058
ISelection 1153
ISelRange 660
ISeparator 299, 655, 1442
ISetForeground 1135
ISetNullIfEmpty 1271
IShareable 1074
IShareAware 1105
IShared 1320
IShellSize 1058
IShift 510
IShortCuts 944
IShowActive 944, 984
IShowCellSelected 818
IShowDialog 1554, 1578
IShowDOW 872
IShowFirstOfGroup 872
IShowHandlers 342
IShowHelp 1081, 1091, 1119, 1144, 1153, 1167
IShowLines 872
IShowMessages 567
IShowPercent 961
IShowProcess 1586
IShowResult 1586
IShowSelAlways 517, 670
IShowSignature 1587
IShowSystemBalloon 267
IShowTabsMenu 984
IShowText 320
IShowTopics 205
ISilent 1112, 1578
ISilentMode 1547
ISimple 610
ISimpleProgress 1112
ISingleExpand 670, 1009
ISingleInstance 112
ISingleSel 517
ISizeGrip 610
ISmallCaps 567
ISmooth 339, 547
ISolidColor 1081
ISort 258
ISpecial 328
ISSL 745
IStartFromCurrent 1615
IStatusBar 702, 1561
IStretch 896
IStrikeOut 195, 568, 1091
ISubMenu 1443
ISubScript 568
ISuccess 716
ISummary 1476
ISupraHeader 818
ISupScript 568
ISyncBuddy 697
ISynchroData 1512
ISysMenu 93
ISystem 1127
LT 1683
LT2 1683

- ITabCloseButtons 632, 984
 - ITabsCanMove 985
 - ITabStop 72, 339, 367, 380, 392, 410, 496, 632, 985
 - ITaskModal 1135
 - ITemporary 1127
 - IText 1443
 - ITinyIntAsBoolean 1250
 - IToolbar 1561
 - ITooltips 660, 819, 1010, 1074
 - ITooltips 517, 671
 - ITopMost 1135
 - ITotal 1476, 1491
 - ITouch 149
 - ITransparent 164, 226, 254, 276, 303, 312, 314, 324, 339, 351, 368, 377, 380, 392, 399, 490, 568, 624, 632, 647, 661, 806, 819, 857, 873, 902, 904, 965, 968, 977, 985, 1003
 - ITransparentBody 633
 - ITransparentGrid 819
 - ITrimSpaces 357
 - ITrueTypeOnly 1091
 - IUnderline 195, 336, 569, 1091
 - IUndisclosedRecipients 723
 - IUnicode 1461
 - IUpdLinked 1280
 - IUpDown 486
 - IURLDetect 569
 - IUseDictionary 1202
 - IUseFileAttributes 1057
 - IUseFileCache 1548
 - IUseReturn 113, 819
 - IUseTabStops 282
 - IUseTimer 624
 - IUseUI 785
 - IValidate 1074
 - IVarHeight 553
 - IVertical 278
 - IVisible 42, 329, 454, 557, 654, 655, 656, 657, 658, 857, 896, 939, 997, 1002, 1033
 - IVSCroll 289, 411, 569
 - IWait 1041
 - IWantMapping 1112
 - IWantReturn 289
 - IWantTab 290
 - IWeekNumbers 541
 - IWholeWord 1119, 1167
 - IWinKey 510
 - IWord 570
 - IWordBreak 496
 - IWordEx 570
 - IWordSel 571
 - IWordWrap 571
 - IWordX 570
 - IWrap 697
 - IWrite 570
 - IWysiwyg 1091
 - IXLSNumeric 1476
 - IXOr 1625
 - IXPLook 316, 383
 - IZeroFill 357
- M -**
- MailFiles 755
 - MailFrom 755
 - MailHtml 756
 - MailSubject 756
 - MailText 756
 - MailTo 756
 - MakeThumbNail 234
 - Max 867
 - Maximize 100
 - MemberList 1429
 - Menu functions 1679
 - Min 867
 - Minimize 100
 - MinimumRowHeight 836
 - MkDir 1259
 - Modified 1305, 1429
 - ModifyItem 262, 285, 951
 - Mouse shortcuts 1720
 - MouseColPos 427
 - MouseRowPos 427
 - MoveAfter 693, 1026
 - MoveCol 427
 - MoveDown 693, 1026
 - MoveInto 693
 - MovePos 594
 - Movetab 991
 - MoveTo 172

MoveUp 694, 1026
 MsgAlert 1644
 MsgCancelOK 1644
 MsgDebug 1645
 MsgInfo 1645
 MsgNoYes 1646
 MsgOkCancel 1646
 MsgRetryCancel 1647
 MsgStop 1647
 MsgSysError 1648
 MsgYesNo 1648
 MsgYesNoCancel 1649
 Multi-language functions 1680
 Multithread programming 1589
 MyIP 1259
 MyLocation 1259
 MySql_Ping 1628

- N -

nAccelTime 697
 nAction 1113
 nAffectedRecords 1264
 nAffectedRows 1208, 1221, 1230, 1241, 1251
 nAlign 42, 324, 552, 610, 647, 896, 1412
 nAlignMarginBottom 43
 nAlignMarginLeft 43
 nAlignMarginRight 43
 nAlignMarginTop 44
 nAlignment 250, 254, 268, 276, 304, 351, 454, 474, 486, 506, 530, 571, 617, 800, 857, 897, 939, 997, 1003, 1491
 nAlignWeight 44, 897
 nAlignWeightSibling 45
 nAnchors 46
 nAnd 1625
 nArrayAt 441
 nArrayOrgAt 443
 nArrayPos 444
 nAttachAs 724
 nAttributes 1209
 nAuth 724
 NavFirst 1569
 Navigate 706, 733
 NavigateEx 706
 NavLast 1569
 NavNext 1569
 NavPrior 1569
 nBase 697
 nBevellInner 384
 nBevelOuter 384
 nBitsPerPixel 149
 nBkGndMarginX 78
 nBkGndMarginY 78
 nBkGndMode 78
 nBmpAlignment 454, 474, 857, 997, 1004
 nBmpHeight 317, 800, 807
 nBmpMargin 455, 475, 807, 925
 nBmpWidth 317, 801, 807
 nBorderRadius 807
 nBorderSides 47
 nBorderStyle 93, 276, 312, 314, 340, 351, 368, 380, 392, 400, 403, 411, 611, 617, 647, 702, 730, 968, 1561
 nBottomMargin 1476
 nBrightness 342
 nBtnHeight 645, 646, 647, 648, 649, 650, 651, 652, 653, 1410, 1411, 1412, 1413, 1414, 1415, 1416
 nBtnWidth 648
 nBuddyAlign 348, 697
 nBuddyOffset 349
 nBuffer 213, 772
 nBufferSize 1448, 1458
 nBuild 113
 nBuildNumber 243
 nButtons 1551
 nCaptionHeight 150
 nCaptionSize 396
 nCardClrAct 886
 nCardClrHot 885
 nCardClrLink 886
 nCardClrPane 886
 nCardClrSel 886
 nCardGap 887
 nCardInnerGap 887
 nCardMargin 887
 nCardSize 887
 nCharCase 268
 nCharSet 196, 332, 1092
 nCharSize 1491
 nCharType 1202
 nCircleGap 961

- nCircleSize 961
- nClickItem 888
- nClickPos 888
- nClientHeight 48, 150
- nClientLeft 48, 83, 150
- nClientTop 48, 83, 150
- nClientWidth 48, 151
- nClrAltPane 411
- nClrBarBack 980
- nClrBarFore 980
- nClrBorder 801, 873
- nClrCircle 962
- nClrColorization 902, 917, 956, 965, 977
- nClrDaysDisabled 873
- nClrDivider 820, 925
- nClrDropBorder 913
- nClrDropDaysDisabled 914
- nClrDropHot 914
- nClrDropHotPane 907
- nClrDropHotText 907
- nClrDropPane 908
- nClrDropSelPane 908
- nClrDropSelText 908
- nClrDropText 908
- nClrDropTextHeader 914
- nClrDropToday 914
- nClrEditPane 925
- nClrGlow 333
- nClrHeaderDivider 820
- nClrHeaderEditPane 820
- nClrHeaderEditPaneFocus 820
- nClrHeaderEditText 821
- nClrHeaderEditTextFocus 821
- nClrHeaderHotPane 821
- nClrHeaderHotText 821
- nClrHeaderPane 822
- nClrHeaderSelPane 822
- nClrHeaderText 822
- nClrHot 873
- nClrHotPane 822, 926, 944
- nClrHotText 823, 926, 945
- nClrInnerBorder 334
- nClrKeys 969
- nClrLine 671
- nClrLined 1477
- nClrLink 352, 897
- nClrOuterBorder 334
- nClrOver 336
- nClrPane 49, 164, 258, 282, 306, 325, 334, 340, 342, 358, 368, 392, 403, 411, 455, 475, 510, 518, 541, 571, 611, 671, 801, 823, 858, 868, 904, 926, 1492
- nClrPaneDisabled 268
- nClrPaneEnd 77
- nClrPaneFocus 268
- nClrPanelItems 801
- nClrPen 905
- nClrSelection 874
- nClrSelFocus 672
- nClrSelFocusPane 412
- nClrSelFocusText 412
- nClrSelItem 672
- nClrSelPane 412, 823, 926, 945
- nClrSelText 412, 823, 945
- nClrShine 334
- nClrText 49, 335, 336, 358, 368, 412, 455, 475, 486, 491, 518, 541, 572, 671, 824, 897, 927, 1004, 1021, 1501
- nClrTextDisabled 269, 903, 965
- nClrTextFocus 269
- nClrTextHeader 874
- nClrTextHot 369
- nClrTitleEnd 396
- nClrTitlePane 487, 542
- nClrTitleStart 396
- nClrTitleText 397, 487, 542
- nClrToday 874
- nClrTrailingText 487, 542
- nColDividerStyle 413
- nColor 126, 161, 223, 320, 1081, 1092, 1498, 1579
- nColorBoxSize 320
- nColorPlanes 151
- nColSel 413
- nColsWidth 282
- nColumn 290, 898
- nColumns 393
- nCommonAVI 622, 1160
- nCompressionLevel 1458
- nConnectionFlags 1202
- nConnTimeOut 1209
- nContrast 342

nConverter 572
nCopies 126, 1154, 1554, 1579
nCount 1148, 1448, 1459
nCounter 1477, 1501
nCrLfText 639, 997
nCrLfTextHot 639
nCursorLocation 1209, 1280
nCursorType 1281
nDataAlignment 858
nDataBmpAlignment 858
nDataCol 858
nDataFont 1492
nDataLines 413
nDataRows 413
nDataType 1378, 1382, 1399, 1402
nDataVAlignment 859
nDBType 1251
nDec 1271
nDefaultButton 1136
nDelay 1615
nDirection 1361
nDisplayMode 874
nDropItemHeight 909
nDropMargin 909
nDropNumberOfWeeks 915
nDroppedHeight 258, 859, 909, 915
nDroppedWidth 259, 859, 909, 915
nDuplex 127, 1579
nDuration 1616
nEditLength 455, 859
nEditMode 413, 824
nEditSelEnd 259
nEditSelStart 259
nEditType 456
NeedNewPage 1484
nEffects 802
Negative 345
nEncoding 724
nError 749
nErrorCode 1587
New 33, 57, 159, 169, 206, 214, 224, 389, 620, 757, 1029, 1036, 1063, 1163, 1430, 1434
NewDataSet 1195
NewFromDC 172
NewLine 1484
NewPage 1484
NewProcess 1186
nExclude 1626
Next 382
nExtractBufferSize 1448, 1459
nFileSize 1127
nFileType 1203
nFilterEditStyle 861, 939
nFilterIndex 1105
nFilterTimeOut 927
nFirstFrame 624
nFitInPage 387
nFlatStyle 317
nFont 1502
nFontColor 572
nFontHeight 332
nFontSize 572
nFontType 332
nFooterAlign 532
nFooterHeight 414, 824, 1502
nFooterImage 457
nFormat 573
nFormatType 487
nFormType 94
nFrame 985
nFrameClrPane 920
nFrameClrPen 920
nFrameEllipseX 920
nFrameEllipseY 921
nFrameRect 921
nFrameSize 921
nFreeze 414
nFromPage 1154
nGamma 343
nGap 1010
nGradient 77
nGridStyle 1477
nGroup 535
nGroupLine 1477
nHeaderAlign 532
nHeaderHeight 414, 824, 1502
nHeaderImage 457

nHeight 49, 151, 156, 191, 196, 209, 226, 251, 254, 259, 269, 274, 282, 290, 304, 314, 335, 340, 359, 369, 380, 383, 393, 400, 403, 414, 457, 475, 488, 497, 511, 518, 542, 545, 547, 611, 633, 644, 648, 672, 703, 730, 802, 869, 888, 922, 957, 962, 969, 972, 977, 979, 986, 1027, 1092, 1498, 1562
nHighLight 573
nHorzExtent 260, 282
nHotItem 888
nHotOpacity 869
nHotPos 889
nHotStyle 369
nIconIndex 1074
nIconSize 397
nID 1021
nImage 299, 362, 491, 506, 530, 535, 557, 617, 640, 655, 687, 861, 939, 998, 1004, 1021, 1033
nImageCount 227
nImageDisabled 362
nImageIndex 227
nImageMode 1454
nImageType 227
nIncrement 698
nIndent 536, 648, 672
nIndentWidth 483
nIndex 260, 283, 380, 393, 458, 506, 530, 533, 536, 557, 633, 656, 687, 824, 861, 889, 910, 945, 986, 1022, 1148, 1271
nInitialPage 1575
nInitialView 1575
nInitialZoom 1575
nInsertId 1251
nInterpolation 1616
nInterval 1061
nIsolationLevel 1210
nItem 329, 377, 640, 898, 998, 1004
nItemHeight 369, 946
nItemPadding 986
nKey 511
nKeyNo 414
nLastError 725, 760, 785, 1044, 1082, 1092, 1131, 1144, 1154, 1281, 1423, 1449, 1459, 1512
nLastFrame 624
nLayout 969
nLeft 49
nLeftMargin 370, 1477
nLen 415, 1272
nLine 290
nLineCount 291
nLineGap 921
nLineHeight 320, 1478
nLineSize 661
nLineSpacing 352, 573, 808, 903, 965
nLineWidth 972
nLoaded 1540
nLockScheme 1192
nLockType 1282
nLogicalHeight 227
nLogicalWidth 228
nMajorVersion 243
nMapMode 164
nMargin 317, 370, 393, 802, 825, 946, 1010
nMarginHor 927
nMargins 352
nMarginVer 927
nMarqueeSpeed 547, 958, 962
nMarqueeStyle 415
nMax 307, 547, 661, 698, 958, 962, 980, 1161
nMaxHeight 94
nMaxItems 1438
nMaxLength 260, 269, 574
nMaxMemSize 1548
nMaxPage 1154
nMaxPos 404
nMaxPosX 94
nMaxPosY 94
nMaxRecords 1282
nMaxRecsOnDesign 1540
nMaxSelCount 543
nMaxUndo 574
nMaxWidth 95, 1443
nMemoryRAM 244
nMemoSize 1193
nMemoType 1193
nMenu 1444
nMenuPos 1444
nMin 307, 548, 661, 698, 958, 963, 980
nMinHeight 95, 611
nMinorVersion 244
nMinPage 1154
nMinPos 404
nMinSize 404

nMinTabWidth 633
nMinWidth 95, 618
nModalResult 95, 251, 808
nMode 1211
nMonitors 151
nNewSilentMode 1548
nNot 1626
nNumbering 574
nNumberOfWeeks 875
nNumStyle 574
nOpacity 314, 325, 352, 972
NoPageDraw 1143
nOr 1626
nOrder 506, 1502
nOrientation 127, 196, 307, 318, 394, 397, 545, 548, 633, 662, 698, 802, 808, 946, 958, 981, 986
NoSortHeader 517
Notify 1602
NotifyThread 1602
nOutlineWidth 1551
nPage 308, 386, 1478, 1562
nPageCount 1562
nPageNumber 127
nPageSize 662, 1580
nPageType 1580
nPageWidth 575
nPaperLength 128
nPaperSizeType 128
nPaperWidth 130
nPDFPrinter 130
nPenColor 278, 315
nPenSize 278, 315, 905
nPenStyle 278, 315
nPhantomImage 987
nPixelsPerInch 151
nPlatformID 244
nPort 725, 745, 752, 760, 1221, 1230, 1251
nPos 272, 308, 458
nPosition 545
nPostSave 848
nPreviewShowMode 131
nPrinterIndex 131
nPrintPages 1554
nPrinttQuality 131
nPriority 726
nProcessorArch 245
nProductType 245
nPushedOpacity 869
nRadius 973
nRcvBufSize 753
nReadBufferSize 1449, 1460
nRecSelectorWidth 825
nRelease 113
nReleaseID 245
nRepeat 624
nResizeStyle 618
nResolution 1454
nResourceAVI 1161
nResult 1136
nRoot 1075
nRow 1478
nRowDividerStyle 415
nRowHeight 415, 825
nRowIncrement 825
nRowSel 416
nSaturation 343
nScale 114, 1580
nScrollIncrement 82, 352, 370, 545
nScrollShowDelay 82
nSelect 927
nSelectCol 826
nSelectionMode 875
nSelEnd 662
nSellImage 687, 1022
nSelStart 662
nSendTimeOut 1252
nSeparator 1478
nServerType 1203
nService 761
nServicePackMajor 246
nServicePackMinor 246
nShapeType 973
nShowMode 96, 1052
nSiblingWeight 898
nSignColumns 1587
nSignHeight 1587
nSignHorzOffset 1587
nSignHorzSep 1588
nSignVertOffset 1588
nSignVertSep 1588

nSignWidth 1588
nSize 197, 898, 998, 1005, 1093, 1361
nSizeMax 1093
nSizeMin 1093
nSkip 1305
nSort 458, 507, 940
nSpacing 325, 394
nSpanMode 1460
nSQLDec 1272
nSQLLen 1272
nSQLType 1272
nStartValue 1619
nState 533, 1607, 1611
nStatus 131, 712, 716, 1065, 1601
nStep 548, 959
nStopValue 1619
nStyle 161, 197, 222, 260, 293, 322, 803, 910, 918, 969, 1093, 1136
nSuiteMask 246
nSynclnSec 1616
nTableType 1198
nTabsClrPane 987
nTabsClrPaneHot 987
nTabsClrPaneSel 987
nTabsClrPen 988
nTabsClrText 988
nTabsClrTextHot 988
nTabsClrTextSel 988
nTabsMargin 989
nTabsSize 989
nTextAlignment 165
nTextGap 922
nTextPad 899
nTextVAlignment 165, 922
nTickFreq 663
nTickStyle 663
nTimeFrame 228
nTimeout 220, 726, 753, 761, 1136, 1182, 1193, 1221, 1230
nTimeoutResult 1137
nTipSide 663
nTitleAlignment 1492
nTitleClrPane 1493
nTitleClrText 1493
nTitleDnLine 1478
nTitleFont 1493
nTitleHeight 1493
nTitleUpLine 1479
nTooltipIcon 50
nTop 50
nToPage 1155
nTopIndex 261, 283
nTopMargin 370, 1479
nTotal 1493
nTotalFont 1494
nTotalLine 1479
nTransferType 772
nTransparency 343
nTries 726
nType 205, 228, 362, 618, 726, 899, 1131, 1361, 1423
nUlineColor 575
nUlineType 575
Numerate 1442
nUnits 575
nVAlignment 251, 254, 277, 304, 349, 353, 397, 803, 809, 861, 899, 903, 940, 966
nValue 548, 663, 699, 959, 963, 981, 1161
nVersion 114
nVertices 973
nView 518
nVirtualHeight 83
nVirtualWidth 83
nWeight 197, 1093
nWidth 50, 152, 156, 191, 198, 209, 223, 228, 252, 255, 261, 270, 275, 283, 291, 304, 308, 315, 318, 321, 326, 335, 340, 359, 371, 381, 384, 394, 400, 404, 416, 459, 476, 488, 497, 507, 511, 519, 530, 543, 545, 549, 553, 576, 618, 634, 644, 648, 664, 673, 703, 731, 803, 869, 889, 922, 940, 959, 963, 970, 973, 977, 981, 989, 1028, 1094, 1494, 1498, 1562
nWinPlatform 114
nWinVer 114
nWriteBufferSize 1449, 1460
nXor 1626
nZoom 341, 1552
nZoomFactor 388, 703
nZoomH 228
nZoomIndex 386
nZoomMode 1562
nZoomV 229

- O -

- oActiveControl 78, 96, 115
- oActiveForm 115
- oBitmap 322
- oBitmaps 318, 803, 809, 918
- Object oriented programming 26
- oBkGnd 79
- oBrush 51
- oBuddy 349, 699
- oCancelButton 96
- oCanvas 132
- oConnection 1212
- oControl 329, 558, 619, 1619, 1621
- oCursor 51, 900
- oCursorArrow 152
- oCursorDropNo 152
- oCursorDropYes 152
- oDataField 459, 481, 1371, 1374, 1376, 1379, 1382, 1385, 1386, 1388, 1390, 1392, 1394, 1395, 1397, 1400, 1403, 1404, 1406, 1408, 1417, 1419, 1421
- oDataGrid 826
- oDataset 481, 1273, 1369, 1370, 1372, 1374, 1376, 1379, 1383, 1385, 1387, 1389, 1391, 1392, 1394, 1395, 1397, 1400, 1403, 1405, 1407, 1409, 1413, 1417, 1419, 1422, 1479, 1544
- oDataSource 1283, 1356
- ODBC functions 1684
- oDefaultButton 96
- oDevice 165, 198, 223, 1480
- oDsMaster 1540
- oEdit 1010
- oEditCol 826
- oEditFilter 497, 862, 940
- oEmf 388
- oEngineOptions 1513
- oError 1611
- oExportInfo 132
- oFieldByName 1305
- oFont 51, 115, 166, 294, 461, 476, 900
- oFontDOW 875
- oFontHeader 875
- oFontShortcut 804, 970
- oFontTitle 398
- oFooter 416, 460, 826, 1480
- oFooterCol 862
- oForm 51
- oGridCol 940
- oGridEdit 459, 862
- oGroup 826
- oHeader 416, 460, 827, 876, 1480
- oHeaderCol 862
- oHelp 115
- olcon 97, 116, 329, 612, 1137, 1172
- oImageList 293, 301, 371, 416, 483, 497, 553, 612, 634, 649, 673, 827, 889, 910, 928, 947, 989, 1033
- oImageListDis 649
- oImageListHeader 519
- oImageListHot 649
- oImageListLarge 519
- oImageListSmall 519
- oLabelEx 890
- oMainForm 116
- oMDIClient 97
- oMDIMenu 97
- oMenu 79, 402, 656, 809, 1034, 1444
- oMsgBar 98
- OnAcceleratorKeyPressed 708
- OnActivate 102
- OnAddNew 1312
- OnAddressChange 735
- OnAddToTotal 1494
- OnAfterCreated 736
- OnAfterLoad 1532, 1542
- OnAfterPrint 1532
- OnAfterPrintReport 1532
- OnAppend 432, 839
- OnAuthCredentials 736
- OnBalloonClick 1038
- OnBalloonHide 1038
- OnBalloonShow 1038
- OnBalloonTimeout 1039
- OnBeforeConnect 1532
- OnBeforeCreate 63, 1532
- OnBeforeLoad 1533
- OnBeforePopup 736
- OnBeforePrint 1533
- OnBeforeResourceLoad 736
- OnBeginDoc 1533
- OnBeginDrag 63, 500

- OnBeginDragItem 680
OnBeginLabelEdit 526, 680, 1016
OnBeginTrack 501
OnBof 432
OnBookMark 433
OnBtnClick 323, 464, 919
OnBtnGenPdf 1533
OnBtnPrint 1534
OnBtnSendMail 1534
OnCalcRect 992
OnCalcVirtualSize 952
OnCancel 1164, 1312
OnCancelEdit 839
OnCardChange 892
OnCardClick 892
OnCardDbClick 893
OnCardGapPaint 893
OnCardPaint 893
OnCardSelection 894
OnChange 256, 263, 273, 287, 305, 309, 394, 433, 488, 511, 543, 550, 607, 637, 644, 666, 699, 839, 876, 903, 952, 959, 963, 966, 978, 981, 993, 1077, 1486
OnChangeD 953, 1486
OnChangeDisk 1452, 1463
OnChangeFile 1452, 1463
OnChangeFocus 102
OnChangeHeight 433
OnChangeRect 347
OnChangeSelected 839, 953
OnChangeSelection 434
OnChangeView 877
OnChangeWidth 464
OnChanging 637, 993
OnChar 73
OnCheckDate 877
OnCheckState 74, 301
OnCheckStateChanged 526, 680, 953
OnClearClick 919
OnClick 64, 252, 256, 302, 305, 337, 366, 395, 434, 477, 501, 507, 526, 614, 621, 658, 681, 804, 840, 905, 953, 966, 979, 994, 1000, 1007, 1016, 1445
OnClickCancel 1413
OnClickDelete 1414
OnClickEdit 1414
OnClickFirst 1414
OnClickInsert 1414
OnClickLast 1415
OnClickNext 1415
OnClickObject 1534
OnClickPrevious 1415
OnClickRecSelector 840
OnClickRefresh 1415
OnClickUpdate 1416
OnClose 103, 122, 398, 628, 737, 770, 789, 793, 1000, 1007, 1082, 1095, 1106, 1121, 1145, 1156, 1164, 1168, 1312, 1359, 1542
OnClosed 398
OnCloseTab 637, 993
OnCloseUp 263, 488, 911, 916
OnColumnClick 527
OnCommand 778
OnComplete 779, 1354, 1610
OnCompleteTask 1610
OnConnect 713, 717, 771, 1188
OnConnected 1189
OnContentLoading 708
OnContextMenu 64, 607, 840
OnCopyData 103
OnCreate 65, 103, 221, 464, 1056, 1189, 1312, 1359
OnCustomDraw 810
OnDataClick 465
OnDatadBClick 465
OnDataRClick 466
OnDbClick 264, 288, 435, 477, 501, 508, 841, 954, 1017, 1039
OnDbClickObject 1534
OnDeactivate 104
OnDelete 434, 842
OnDeleteItem 527
OnDesignLoadReport 1535
OnDesignSaveReport 1535
OnDestroy 65
OnDestroyed 65
OnDirectory 779
OnDisconnect 714, 718, 1189
OnDisconnected 1189
OnDispTooltip 66, 841
OnDividerDbClick 502, 508, 932
OnDocumentTitleChanged 708

OnDown 309
OnDownloadComplete 737
OnDownloadData 737
OnDownloadFile 738
OnDownloadStart 738
OnDragOver 66
OnDragOverItem 681
OnDrawActiveRow 842
OnDrawCell 466, 527, 842
OnDrawFooter 843
OnDrawGroupFooter 844
OnDrawGroupHeader 844
OnDrawHeader 845
OnDrawItem 615, 877, 911, 915, 932, 954, 1017
OnDrawRecSelector 845
OnDrawRow 435
OnDrawTab 994
OnDrawTabsBackground 995
OnDrop 804
OnDropCreate 911, 916
OnDropDown 264, 489, 653, 912, 916
OnDropFiles 75
OnEdit 435, 467, 846, 1313
OnEditChange 264
OnEditCol 846
OnEditColBtnClick 846
OnEditUpdate 264
OnEnd 1054, 1486, 1605
OnEndDrag 67, 502
OnEndDragItem 681
OnEndGroup 1487
OnEndLabelEdit 528, 682, 1018
OnEndLine 1487
OnEndPage 1487
OnEndTrack 502
OnEnter 74
OnEof 436
OnError 714, 718, 729, 779, 1189, 1610
OnExecute 1173, 1190
OnExecuteScriptCompleted 709
OnExit 74, 468
oNext 876
OnFail 222
OnFilterBtnClick 502, 933
OnFilterChange 436, 503, 933
OnFilterEdit 503, 934
OnFilterEnd 847
OnFilterStart 847
OnFilterTimeout 934
OnFind 1120, 1129, 1169
OnFinish 1617
OnFirst 1542
OnFirstPage 389
OnFooterClick 468, 847
OnFooterDbClick 468
OnFooterRClick 469, 847
OnGetData 469, 478, 771
OnGetTooltipText 1018
OnGetValue 470, 1276, 1535
OnGoBottom 436
OnGoTop 437
OnGridEditCreate 470
OnHeaderClick 471, 848
OnHeaderDbClick 471
OnHeaderRClick 471, 848
OnHelp 104, 120, 1083, 1095, 1106, 1121, 1146, 1156, 1168
OnHelpClick 104
OnHide 67
OnHistoryChanged 709
OnHScroll 608
OnInitialize 105, 1056
OnInitialized 1077
OnInsert 437, 848
OnInsertPicture 608
OnItemBeginDrag 934
OnItemChanged 503, 528, 934
OnItemChanging 504, 935
OnItemClick 935
OnItemEndDrag 935
OnItemExpand 327
OnItemExpanded 682, 1019
OnItemExpanding 682, 1019
OnItemRClick 936
OnKeyClick 970
OnKeyCount 437
OnKeyDown 75, 528, 683
OnKeyNo 437
OnKeyUp 76
OnLastPage 389

- OnLButtonDown 67, 1039
OnLButtonUp 68, 1039
OnLinkClick 353
OnLoadEnd 739
OnLoadError 739
OnLoadStart 739
OnMenuClick 810
OnMenuSelect 105
OnMouseMove 68, 1040
OnMouseOverObject 1536
OnMove 106, 405
OnNavigationCompleted 709
OnNavigationStarting 709
OnNavStateChange 740
OnNewItem 438
OnNewMail 758
OnNewTab 995
OnNewWindowRequested 710
OnNext 1543
OnNextPage 390
OnNotify 1605
OnNotifyThread 1605
OnOpen 398, 628, 771, 790, 793, 1313, 1359, 1543
OnOpened 399
OnPageChanged 1536
OnPageDown 310
OnPageUp 310
OnPaint 69
OnPainted 69
OnPastEof 438
OnPause 1606
OnPostAppend 1341
OnPostChange 1313
OnPostClose 1314, 1360
OnPostDelete 1315
OnPostEdit 438, 472
OnPostEnd 1488
OnPostGroup 1488
OnPostOpen 1315, 1360
OnPostPage 1488
OnPostSave 1315
OnPowerEvent 1068
OnPreAppend 1341
OnPreChange 1315, 1488
OnPreDelete 1316
OnPreSave 1317
OnPreview 147, 1536
OnPreviousPage 390
OnPrintData 1495
OnPrintFooter 1503
OnPrintHeader 1503
OnPrinting 608
OnPrintLine 1498
OnPrintPage 1536
OnPrintReport 1537
OnPrintTitle 1495
OnPrintTotal 1496, 1503
OnPrior 1543
OnProcess 1617
OnProgress 779, 1452, 1463, 1537
OnProgressStart 1537
OnProgressStop 1538
OnProtocolExecution 740
OnQueryEndSession 120
OnRButtonDbClick 1040
OnRButtonDown 70, 1041
OnRButtonUp 70, 1040
OnRClick 337, 439, 478, 504, 508, 529, 615, 621, 638, 683, 995, 1000, 1007
OnRead 794, 1055
OnRebuild 936
OnReplace 1169
OnReplaceAll 1169
OnRequestResize 608
OnReset 1164
OnResume 1606
OnRun 1068
OnSave 849
OnSaveCol 849
OnScan 1456
OnScrollDown 84, 353, 374
OnScrollLeft 84
OnScrollPageDown 85, 354, 375
OnScrollPageLeft 85
OnScrollPageRight 85
OnScrollPageUp 86, 354, 375
OnScrollRight 86
OnScrollUp 86, 354, 375
OnSeek 439

OnSelChanged 683
OnSelChanging 684
OnSelect 543, 878, 955
OnSetValue 1276
OnShow 70, 1083, 1096, 1107, 1138, 1146, 1157, 1164
OnShutDown 1068
OnSimpleMode 615
OnSingleExpand 684
OnSize 106
OnSkip 440
OnSort 445
OnSourceChanged 710
OnStart 628, 780, 1055, 1129, 1606, 1618
OnStartGroup 1488
OnStartLine 1489
OnStartPage 1489
OnStatusMessage 740
OnStop 628, 1129
OnThumbPos 310
OnThumbPosH 87
OnThumbPosV 87, 355, 376
OnThumbTrack 311
OnThumbTrackH 87
OnThumbTrackV 87, 355, 376
OnTimeOut 758
OnTimer 1063
OnTitleChange 741
OnTotalDnLoadInfo 714, 718
OnTotalLengthInfo 714, 718
OnTrack 504
OnUp 311
OnUpdate 849, 1317
OnValidate 1078, 1261
OnWebResourceRequested 710
OnWrite 790
OnZoomChanged 390
OnZoomIn 390
OnZoomOut 391
OOP 26
oOptionList 362
oPageSetupDlg 134
oParent 33, 52, 294, 329, 461, 476, 531, 533, 536, 687, 900, 998, 1005, 1022, 1028, 1034, 1061, 1145, 1148, 1155, 1480, 1611
Open 166, 185, 757, 767, 789, 792, 1306, 1346, 1358, 1480, 1570
OpenFile 776
OpenFileRead 776
OpenFileWrite 776
OpenKey 1049
OpenProcess 1186
OpenRequest 767
OpenURL 768
oPicture 313, 326, 341
oPopup 98, 576
oPrev 876
oPreviewDC 134
oPreviewOptions 1513
oPrevWindow 98
oPrinter 388
oPrintOptions 1513
oPrnDlg 134
OrdBagName 1331
OrdCount 1331
OrdFor 1331
OrdKey 1332
OrdKeyVal 1332
OrdName 1332
OrdNumber 1333
OrdSetFocus 1333
oRecords 1283
oRecSel 417
oRecSelector 827
oReport 1494, 1498, 1502, 1540, 1563
oReportOptions 1513
OrgRecno 1346
oSeek 417
oSelectCol 827
oSqlProc 1362
oSubMenu 300
oSupra 828
oSupraCol 862
oTitle 1481
oTreeView 688, 1022
OutDebug 1636
OutDebugDec 1636
OutDebugHex 1637
Outline 1570
OutputParams 1358

oWndOwner 134

- P -

Pack 1333
Page 1570
PageDlg 595
PageSetup 142
PageSetupDlg 1649
Paint 159, 193, 212, 236
PaperPhysicalSize 142, 154
PaperRes 142, 154
PaperSize 142, 154
ParamByName 1358
Password 1192
Paste 595
PasteSpecial 595
Path 1434
Pause 1602
Pdf 1568
PDF Sign tool 1581
PhysicalOffset 143
PickIconDlg 1650
PictureDlg 595
Pie 172, 1674
PixelsToMapMode 173
PixelToTwips 173
Play 194, 627
PointToPixels 173
Polaroid 345
Polygon 174
PopUp 206
Port 143
PosFromChar 596
PostMsg 58
PostRequest 768
Predefined resources 1720
PrepareReport 1524
Preview 143, 596
PreviewClose 1524
PreviewControl 143
PreviewNow 1485
Previous 382
Print 596, 1425, 1524, 1570, 1581
PrintAtCol 1484

PrintBox 597
PrintDlg 597, 1651
PrinterNames 124
Printing functions 1685
PrintProcessor 144
PrintSetup 597
ProcessMessages 1705
Promises 1597
Properties 1216
PutFile 777

- Q -

Query 1216, 1225, 1234, 1244, 1256, 1266
QueryArray 1205, 1216, 1225, 1234, 1245, 1257, 1266
QueryArrayHash 1225, 1235, 1245, 1257
QueryInfo 768
QueryMemDataset 1186
QueryReport 1217, 1226, 1235, 1245, 1257
QueryRow 1205, 1217, 1226, 1235, 1246, 1258, 1266
QueryRowHash 1226, 1236, 1246, 1258
QueryValue 1206, 1217, 1227, 1236, 1246, 1258, 1267
Quit 1603

- R -

Read 793, 1440
ReadFile 769
ReadLine 1688
ReadPop3Asyn 758
Recall 1334
RecCount 892, 951, 1306
RecLock 1334
RecNo 1307
RecordHasChanged 1353
Rectangle 174
RectToPixels 174
RecUnLock 1335
Redo 598
Redraw 58, 80
Refill 428
Refresh 59, 296, 330, 428, 1036, 1307, 1396, 1398, 1440, 1445, 1541

RefreshCurrent 428, 1307
RegisterFormat 185
Reindex 1335
ReleaseState 1307, 1336, 1346
Reload 706, 734
Remove 1031
RemoveAll 1031
RemoveControl 81
RemoveDirectory 777
RenameFile 777
RenameSection 218
Replace 292, 598
ReplaceAll 598
ReplaceChars 1688
ReplaceDlg 599
Replacelcon 1032
Report Generator 1464
Report Generator:Commands 1465
Reports 1464
RequestState 73, 81
Reset 428, 1163, 1430
Resolution 144, 154
Resource functions 1685
Resources 1720
Restore 100
RestoreExpandState 678
RestoreFromText 201
RestoreState 100, 428, 836, 931, 992, 1308, 1336, 1347
RestoreVariables 1524
Resume 1603
Retrieve 757
RetrieveAll 757
ReturnValue 1359, 1612
RGBInverse 1631
RGBToBGR 346
RGBToHTML 1084, 1631
RightPos 863
RollBackAll 1267
RollBackTrans 1206, 1218, 1227, 1237, 1247, 1259
Root 1043
Rotate 346
RoundRect 175
RTFTToHTML 599
RTFTToPDF 599
RTFTToWord 599
RTFTToWordX 600
RTFTToWrite 600
Run 144, 713, 717, 1042, 1053, 1066, 1075, 1082, 1094, 1106, 1113, 1120, 1131, 1138, 1145, 1148, 1155, 1162, 1167, 1173, 1358, 1451, 1462, 1485, 1589, 1603
RunFolder 1653
RunProc 1187
RunProcDirect 1187
RunSynchroTask 1608

- S -

Save 160, 237, 346, 1336, 1570
SaveExpandState 679
SaveFile 600
SaveFrom 1308
SaveState 101, 429, 836, 931, 992, 1308, 1336, 1347
SaveToArray 1337
SaveToFile 1524
SaveToFPFile 1525
SaveToStream 237, 346
SaveToString 1525
SaveToText 201
SaveVariables 1525
Scan 1455
Search 1348
Seek 429, 627, 1338
Select 378, 429, 462, 492, 537, 694, 1006, 1026, 1338
SelectAll 272, 430, 600
SelectCellFromPoint 430
SelectCol 430
Selected 1000, 1006
SelectedCol 430
SelectItem 679, 1015
SelectNone 273, 430, 601
SelectPen 176
SelectRoot 679, 1016
SelectString 262, 286
Selet 641, 999
Send 728, 745, 750, 787
SendMail 1260
SendMsg 59

SendRequest 769
SendWMCopyData 1627
SepFile 145
Sepia 347
ServerName 145
SetADOCConnectStr 1525
SetAlignment 462
SetArray 444
SetBarColor 549
SetBKColor 549
SetBkImage 525
SetBounds 60, 1552
SetBullet 601
SetColorMatrix 347
SetContent 707
SetCurrentDirectory 777
SetCurSel 286
SetData 186
SetDate 1049
SetDbf 447
SetDropFiles 186
SetEntry 219
SetEnvironmentVariable 1697
SetEventMask 601
SetExpandable 694
SetExtension 1434
SetFile 207
SetFileDateTime 1668
SetFilter 1309, 1345
SetFirstVisible 951
SetFocus 60, 101
SetFont 1095
SetFormatSettings 1526
SetHideSelection 601
SetHorzExtent 286
SetHtml 976
SetHyperLinkTooltips 602
SetIcon 1036, 1528
SetIconSize 1032
SetIndent 602
SetIndex 636
SetIniFile 1440
SetInterval 1063
SetItem 641
SetItems 1440
SetLanguage 1684
SetLicense 1528
SetLogical 1049
SetMenu 297, 1036
SetModify 602
SetName 1435
SetNumbering 603
SetNumeric 1050
SetOptions 603
SetOverlayIcon 118
SetParagraphSpacing 603
SetPath 1435
SetPos 604
SetPrimaryKeys 1353
SetProgress 118
SetProperty 1529
SetProxyData 769
SetReadTimeout 793
SetRelation 1339
SetRow 444
SetScope 1339
SetSection 219
SetSel 273, 287, 604
SetSellItems 287
SetString 1050
SetText 187, 604, 734
SetTextMode 604
SetTick 665
SetTimeOut 770
SetToolInfo 155, 249
SetToolTip 1037
SetTypes 238
SetTypographyOptions 605
Setup 145
SetValue 445, 1050, 1430
SetValues 1430
SetVariable 1530
SetVirtualHostNameToFolder 707
SetZoom 605
SetZoomLevel 734
ShAboutDlg 1654
ShareName 145
ShExitWindowsDlg 1655
ShFileProperty 1656
ShFindComputerDlg 1657

ShFindFilesDlg 1658
 SHGetFolder 1697
 Shift 14
 Shl 1624
 ShNetConnectionDlg 1659
 ShortName 1435
 Show 60, 101, 1037
 ShowBalloon 61, 1037
 ShowContext 207
 ShowDevTools 734
 ShowErrorList 1187
 ShowFinder 207
 ShowIndex 207
 ShowKeyword 208
 ShowModal 101
 ShowPopupMenu 61
 ShowPreparedReport 1530
 ShowReport 1530
 ShowSearch 208
 ShowTOC 208
 ShowTooltip 62
 ShowTopic 208
 Shr 1624
 ShRestartWindowsDlg 1659
 Signal 1604
 SimpleReport:Classes 1471
 SizeToHeight 1674
 Skip 1309
 Sort 525, 837, 1310, 1340, 1348
 SortChildren 694, 1026
 SortSection 219
 Special Characters 1720
 SqlInsert 1187, 1349
 SqlNextResult 1227, 1237
 SqlUpdate 1188, 1349
 Start 1067, 1617
 StartDoc 146
 StartDocPdf 146
 StartLine 1485
 StartPage 146
 State 1218
 Steplt 550
 Stop 627, 707, 1042, 1063, 1067, 1163, 1604, 1609, 1617
 StopGroupTyping 605

StopLoad 735
 StopReport 1530
 Stored procedures 1354
 StretchCols 431, 837
 String functions 1685
 String Lists functions 1690
 StringToHex 1631
 StrMySql 1632
 StrSql 1632
 StrSqlLite 1632
 Struct 1340
 Sum 867
 Sumatra PDF 1571
 SumTotal 866
 SwapCols 431
 SwapItems 932
 System information functions 1694
 SystemChanged 219

- T -

Table 1218, 1228, 1237, 1247, 1260
 TADODataSource 1206
 TADODataSource:BeginTrans 1212
 TADODataSource:Cancel 1213
 TADODataSource:cConnect 1207
 TADODataSource:CommitTrans 1213
 TADODataSource:Connect 1213
 TAdoDataSource:ConnectTag 1213
 TADODataSource:cPassword 1207
 TADODataSource:cUser 1208
 TADODataSource:DelTable 1214
 TADODataSource:Disconnect 1214
 TADODataSource:Execute 1214
 TADODataSource:File 1215
 TADODataSource:GetCatalogs 1215
 TADODataSource:GetTables 1215
 TADODataSource:IConnectAsync 1208
 TADODataSource:IPromptIfError 1208
 TADODataSource:Methods 1212
 TADODataSource:nAffectedRows 1208
 TADODataSource:nAttributes 1209
 TADODataSource:nConnTimeOut 1209
 TADODataSource:nCursorLocation 1209
 TADODataSource:nIsolationLevel 1210

- TADODataSource:nMode 1211
- TADODataSource:oConnection 1212
- TADODataSource:Properties 1207, 1216
- TADODataSource:Query 1216
- TADODataSource:QueryArray 1216
- TADODataSource:QueryReport 1217
- TADODataSource:QueryRow 1217
- TADODataSource:QueryValue 1217
- TADODataSource:RollBackTrans 1218
- TADODataSource:State 1218
- TADODataSource:Table 1218
- TADODataSourceVersion 1219
- TAdsDataSource 1198
- TAdsDataSource:BeginTrans 1204
- TAdsDataSource:cConnect 1200
- TAdsDataSource:cDriver 1200
- TAdsDataSource:CommitTrans 1204
- TAdsDataSource:cPassword 1200
- TAdsDataSource:cUser 1201
- TAdsDataSource:DefDataExtension 1204
- TAdsDataSource:DefIdxExtension 1205
- TAdsDataSource:hConnect 1201
- TAdsDataSource:IAdsLocking 1201
- TAdsDataSource:IRightsCheck 1201
- TAdsDataSource:IUseDictionary 1202
- TAdsDataSource:Methods 1203
- TAdsDataSource:nCharType 1202
- TAdsDataSource:nConnectionFlags 1202
- TAdsDataSource:nFileType 1203
- TAdsDataSource:nServerType 1203
- TAdsDataSource:Properties 1199
- TAdsDataSource:QueryArray 1205
- TAdsDataSource:QueryRow 1205
- TAdsDataSource:QueryValue 1206
- TAdsDataSource:RollBackTrans 1206
- TAniControlSize 1620
- TAniControlSize:aStartValues 1620
- TAniControlSize:aStopValues 1621
- TAniControlSize:IForceRefresh 1620
- TAniControlSize:INcaRefresh 1620
- TAniControlSize:oControl 1621
- TAnimatedGif 311
- TAnimatedGif:IPlay 312
- TAnimatedGif:ITransparent 312
- TAnimatedGif:nBorderStyle 312
- TAnimatedGif:oPicture 313
- TAnimatedGif:Properties 311
- TAnimation 1614
- TAnimation:Events 1617
- TAnimation:IAutoReverse 1614
- TAnimation:IEnabled 1614
- TAnimation:ILoop 1615
- TAnimation:IStartFromCurrent 1615
- TAnimation:Methods 1616
- TAnimation:nDelay 1615
- TAnimation:nDuration 1616
- TAnimation:nInterpolation 1616
- TAnimation:nSyncInSec 1616
- TAnimation:OnFinish 1617
- TAnimation:OnProcess 1617
- TAnimation:OnStart 1618
- TAnimation:Properties 1614
- TAnimation:Start 1617
- TAnimation:Stop 1617
- TAniNumProperty 1618
- TAniNumProperty:cProperty 1618
- TAniNumProperty:IForceRefresh 1618
- TAniNumProperty:INcaRefresh 1619
- TAniNumProperty:nStartValue 1619
- TAniNumProperty:nStopValue 1619
- TAniNumProperty:oControl 1619
- TAppData 121
- TAppData:AddData 121
- TAppData:Destroy 122
- TAppData:Events 122
- TAppData:Initialize 122
- TAppData:Methods 121
- TAppData:OnClose 122
- TApplication 107
- TApplication:aArguments 109
- TApplication:aForms 109
- TApplication:cComments 109
- TApplication:cCompany 110
- TApplication:cCopyright 110
- TApplication:cDescription 110
- TApplication:cDirectory 110
- TApplication:cFileName 111
- TApplication:cTitle 111
- TApplication:cTradeMarks 111
- TApplication:CurrentDir 116

TApplication:CurrentDrive 117
TApplication:cVersion 111
TApplication:DecimalSeparator 117
TApplication:Events 107, 119
TApplication:Inititalize 117
TApplication:IAutoScale 111
TApplication:IBusy 112
TApplication:IDpiAware 112
TApplication:ISingleInstance 112
TApplication:IUseReturn 113
TApplication:MessageBox 118
TApplication:Methods 107, 116
TApplication:nBuild 113
TApplication:nRelease 113
TApplication:nScale 114
TApplication:nVersion 114
TApplication:nWinPlatform 114
TApplication:nWinVer 114
TApplication:oActiveControl 115
TApplication:oActiveForm 115
TApplication:oFont 115
TApplication:oHelp 115
TApplication:olcon 116
TApplication:oMainForm 116
TApplication:OnHelp 120
TApplication:OnQueryEndSession 120
TApplication:Properties 107, 108
TApplication:Run 117
TApplication:SetOverlayIcon 118
TApplication:SetProgress 118
TApplication:Terminate 119
TArrayBrowse 440
TArrayBrowse:aArrayData 441
TArrayBrowse:AddRow 442
TArrayBrowse:DelRow 442
TArrayBrowse:Events 445
TArrayBrowse:GetRow 442
TArrayBrowse:GetValue 443
TArrayBrowse:InsRow 443
TArrayBrowse:IsFiltered 443
TArrayBrowse:IAutoOrder 441
TArrayBrowse:Methods 441
TArrayBrowse:nArrayAt 441
TArrayBrowse:nArrayOrgAt 443
TArrayBrowse:nArrayPos 444
TArrayBrowse:OnSort 445
TArrayBrowse:Properties 440
TArrayBrowse:SetArray 444
TArrayBrowse:SetRow 444
TArrayBrowse:SetValue 445
TArrayBrowse:Zap 445
TaskCount 1609
TAsyncDownload 711
TAsyncDownload:Cancel 713
TAsyncDownload:cBuffer 711
TAsyncDownload:cError 712
TAsyncDownload:Close 713
TAsyncDownload:Events 713
TAsyncDownload:IConnected 712
TAsyncDownload:Methods 713
TAsyncDownload:nStatus 712
TAsyncDownload:OnConnect 713
TAsyncDownload:OnDisconnect 714
TAsyncDownload:OnError 714
TAsyncDownload:OnTotalDnLoadInfo 714
TAsyncDownload:OnTotalLengthInfo 714
TAsyncDownload:Properties 711
TAsyncDownload:Run 713
TAsynFileDownload 715
TAsynFileDownload:Cancel 717
TAsynFileDownload:cError 715
TAsynFileDownload:cLocalFile 715
TAsynFileDownload:Close 717
TAsynFileDownload:cRemoteFile 716
TAsynFileDownload:Eventos 717
TAsynFileDownload:IConnected 716
TAsynFileDownload:ISuccess 716
TAsynFileDownload:Methods 716
TAsynFileDownload:nStatus 716
TAsynFileDownload:OnConnect 717
TAsynFileDownload:OnDisconnect 718
TAsynFileDownload:OnError 718
TAsynFileDownload:OnTotalDnLoadInfo 718
TAsynFileDownload:OnTotalLengthInfo 718
TAsynFileDownload:Properties 715
TAsynFileDownload:Run 717
TBalloon 154
TBalloon:AddTool 155
TBalloon:Metodos 155
TBalloon:SetToolInfo 155

TBarcode 1422
TBarcode:AsString 1424
TBarcode:cText 1423
TBarcode:GetErrorDescription 1425
TBarcode:GetPicture 1425
TBarcode:Methods 1424
TBarcode:nLastError 1423
TBarcode:nType 1423
TBarcode:Print 1425
TBarcode:Properties 1423
TBevel 313
TBevel:IOwnBorder 313
TBevel:ITransparent 314
TBevel:nBorderStyle 314
TBevel:nHeight 314
TBevel:nOpacity 314
TBevel:nPenColor 315
TBevel:nPenSize 315
TBevel:nPenStyle 315
TBevel:nWidth 315
TBevel:Properties 313
TBitmap 156
TBitmap:Create 159
TBitmap:CreateFromHandle 157
TBitmap:CreateNew 157
TBitmap:Destroy 158
TBitmap:Load 159
TBitmap:LoadFromFile 158
TBitmap:LoadFromResource 159
TBitmap:Methods 157
TBitmap:New 159
TBitmap:nHeight 156
TBitmap:nWidth 156
TBitmap:Paint 159
TBitmap:Properties 156
TBitmap:Save 160
TBlatMail 719
TBlatMail:aAttachments 720
TBlatMail:aBCC 720
TBlatMail:aCC 720
TBlatMail:AddBCC 727
TBlatMail:AddCC 727
TBlatMail:AddReceipt 727
TBlatMail:aReceipts 721
TBlatMail:cAddress 721
TBlatMail:cBody 721
TBlatMail:cCharset 721
TBlatMail:cExtra 722
TBlatMail:cPassword 722
TBlatMail:Create 728
TBlatMail:cServer 722
TBlatMail:cSubject 722
TBlatMail:cUser 723
TBlatMail:Destroy 728
TBlatMail:Events 728
TBlatMail:IAuth 723
TBlatMail:ILog 723
TBlatMail:IReceipt 724
TBlatMail:IUndisclosedRecipients 723
TBlatMail:Methods 727
TBlatMail:nAttachAs 724
TBlatMail:nAuth 724
TBlatMail:nEncoding 724
TBlatMail:nLastError 725
TBlatMail:nPort 725
TBlatMail:nPriority 726
TBlatMail:nTimeout 726
TBlatMail:nTries 726
TBlatMail:nType 726
TBlatMail:OnError 729
TBlatMail:Properties 719
TBlatMail:Send 728
TBrowse 405
TBrowse:aCols 407
TBrowse:AddCol 418
TBrowse:AddImage 419
TBrowse:Adjust 419
TBrowse:AdjustCols 419
TBrowse:Append 420
TBrowse:aSelected 407
TBrowse:Bookmark 420
TBrowse:CancelEdit 420
TBrowse:ClearAllFilters 420
TBrowse:ColAtIndex 421
TBrowse:ColAtPos 421
TBrowse:ColPos 421
TBrowse:ColWithHeader 421
TBrowse:DelCol 422
TBrowse>Delete 422
TBrowse>Edit 422

TBrowse:Events 432
TBrowse:GoBottom 422
TBrowse:GoDown 422
TBrowse:GoHPos 423
TBrowse:GoLeft 423
TBrowse:GoLeftMost 423
TBrowse:GoPageDown 423
TBrowse:GoPageUp 424
TBrowse:GoRight 424
TBrowse:GoRightMost 424
TBrowse:Goto 424
TBrowse:GoTop 425
TBrowse:GoUp 425
TBrowse:InsCol 425
TBrowse:Insert 425
TBrowse:IsBof 426
TBrowse:IsEditable 426
TBrowse:IsEof 426
TBrowse:IsMultipleEdit 426
TBrowse:KeyCount 426
TBrowse:KeyNo 427
TBrowse:IAllowAppend 407
TBrowse:IAllowColHiding 407
TBrowse:IAllowColSizing 407
TBrowse:IAllowColSwapping 408
TBrowse:IAllowDelete 408
TBrowse:IAllowEdit 408
TBrowse:IAllowInsert 408
TBrowse:IAllowRowSizing 408
TBrowse:IAutoSave 409
TBrowse:IColDividerComplete 409
TBrowse:IEditMode 409
TBrowse:IFastEdit 409
TBrowse:IFilterBar 409
TBrowse:IFooter 410
TBrowse:IHeader 410
TBrowse:IHScroll 410
TBrowse:IRecordSelector 410
TBrowse:ITabStop 410
TBrowse:IVScroll 411
TBrowse:Methods 417
TBrowse:MouseColPos 427
TBrowse:MouseRowPos 427
TBrowse:MoveCol 427
TBrowse:nBorderStyle 411
TBrowse:nClrAltPane 411
TBrowse:nClrPane 411
TBrowse:nClrSelFocusPane 412
TBrowse:nClrSelFocusText 412
TBrowse:nClrSelPane 412
TBrowse:nClrSelText 412
TBrowse:nClrText 412
TBrowse:nColDividerStyle 413
TBrowse:nColSel 413
TBrowse:nDataLines 413
TBrowse:nDataRows 413
TBrowse:nEditMode 413
TBrowse:nFooterHeight 414
TBrowse:nFreeze 414
TBrowse:nHeaderHeight 414
TBrowse:nHeight 414
TBrowse:nKeyNo 414
TBrowse:nLen 415
TBrowse:nMarqueeStyle 415
TBrowse:nRowDividerStyle 415
TBrowse:nRowHeight 415
TBrowse:nRowSel 416
TBrowse:nWidth 416
TBrowse:oFooter 416
TBrowse:oHeader 416
TBrowse:olmageList 416
TBrowse:OnAppend 432
TBrowse:OnBof 432
TBrowse:OnBookMark 433
TBrowse:OnChange 433
TBrowse:OnChangeHeight 433
TBrowse:OnChangeSelection 434
TBrowse:OnClick 434
TBrowse:OnDbClick 435
TBrowse:OnDelete 434
TBrowse:OnDrawRow 435
TBrowse:OnEdit 435
TBrowse:OnEof 436
TBrowse:OnFilterChange 436
TBrowse:OnGoBottom 436
TBrowse:OnGoTop 437
TBrowse:OnInsert 437
TBrowse:OnKeyCount 437
TBrowse:OnKeyNo 437
TBrowse:OnNewItem 438

- TBrowse:OnPastEof 438
- TBrowse:OnPostEdit 438
- TBrowse:OnRClick 439
- TBrowse:OnSeek 439
- TBrowse:OnSkip 440
- TBrowse:oRecSel 417
- TBrowse:oSeek 417
- TBrowse:Properties 405
- TBrowse:Refill 428
- TBrowse:Refresh 428
- TBrowse:RefreshCurrent 428
- TBrowse:Reset 428
- TBrowse:RestoreState 428
- TBrowse:SaveState 429
- TBrowse:Seek 429
- TBrowse>Select 429
- TBrowse>SelectAll 430
- TBrowse>SelectCellFromPoint 430
- TBrowse>SelectCol 430
- TBrowse:SelectedCol 430
- TBrowse>SelectNone 430
- TBrowse:StretchCols 431
- TBrowse:SwapCols 431
- TBrowse:ToExcel 431
- TBrowseForFolderDlg 1069
- TBrowseForFolderDlg:BrowseComputers 1076
- TBrowseForFolderDlg:BrowseFile 1076
- TBrowseForFolderDlg:BrowseFolder 1076
- TBrowseForFolderDlg:BrowsePrinters 1077
- TBrowseForFolderDlg:cOkBtnText 1070
- TBrowseForFolderDlg:cRoot 1070
- TBrowseForFolderDlg:cStatusText 1071
- TBrowseForFolderDlg:cText 1070
- TBrowseForFolderDlg:cTitle 1071
- TBrowseForFolderDlg:Events 1077
- TBrowseForFolderDlg:Execute 1075
- TBrowseForFolderDlg:IBrowseComputer 1071
- TBrowseForFolderDlg:ICenter 1071
- TBrowseForFolderDlg:IDontGoBelowDomain 1071
- TBrowseForFolderDlg:IEditBox 1072
- TBrowseForFolderDlg:IExtendedUI 1072
- TBrowseForFolderDlg:IForceCreate 1073
- TBrowseForFolderDlg:IIncludeFiles 1072
- TBrowseForFolderDlg:IIncludeURLS 1072
- TBrowseForFolderDlg:INewDlgStyle 1073
- TBrowseForFolderDlg:INoNewFolderButton 1073
- TBrowseForFolderDlg:IReturnFSAncestors 1073
- TBrowseForFolderDlg:IReturnFSDirs 1073
- TBrowseForFolderDlg:IShareable 1074
- TBrowseForFolderDlg:ITooltips 1074
- TBrowseForFolderDlg:IValidate 1074
- TBrowseForFolderDlg:Methods 1075
- TBrowseForFolderDlg:nIconIndex 1074
- TBrowseForFolderDlg:nRoot 1075
- TBrowseForFolderDlg:OnChange 1077
- TBrowseForFolderDlg:OnInitialized 1077
- TBrowseForFolderDlg:OnValidate 1078
- TBrowseForFolderDlg:Properties 1069
- TBrowseForFolderDlg:Run 1075
- TBrowseMod 810
- TBrowseMod:aCols 812
- TBrowseMod:AddCol 829
- TBrowseMod:AddSupraCol 829
- TBrowseMod:aDisplayCols 813
- TBrowseMod:AdjustCols 829
- TBrowseMod:AdjustColWidth 830
- TBrowseMod:aGridData 813
- TBrowseMod:Append 830
- TBrowseMod:aSelected 813
- TBrowseMod:CancelEdit 830
- TBrowseMod:ClearFilter 830
- TBrowseMod:ColAtPos 831
- TBrowseMod:ColWithHeader 831
- TBrowseMod:cRecSelectorHeader 813
- TBrowseMod:DelCol 831
- TBrowseMod>Delete 831
- TBrowseMod>Edit 832
- TBrowseMod:Events 838
- TBrowseMod:GetDataRow 832
- TBrowseMod:GoBottom 832
- TBrowseMod:GoDown 832
- TBrowseMod:GoLeft 833
- TBrowseMod:GoLeftMost 833
- TBrowseMod:GoRight 833
- TBrowseMod:GoRightMost 833
- TBrowseMod:GoTop 833
- TBrowseMod:GroupOn 834
- TBrowseMod:HitTest 834
- TBrowseMod:HitTestCol 834
- TBrowseMod:InsCol 834

TBrowseMod:Insert 835
TBrowseMod:IsFilter 835
TBrowseMod:IsOnEdit 835
TBrowseMod:IsSingleEdit 835
TBrowseMod:IsSorted 836
TBrowseMod:IAllowAppend 814
TBrowseMod:IAllowColHiding 814
TBrowseMod:IAllowColSwapping 814
TBrowseMod:IAllowDelete 814
TBrowseMod:IAllowEdit 815
TBrowseMod:IAllowInsert 815
TBrowseMod:IAllowSizing 814
TBrowseMod:IAllowSort 815
TBrowseMod:IAlternateColor 815
TBrowseMod:IAutoSave 816
TBrowseMod:IColumnSeparator 816
TBrowseMod:IFastAppend 816
TBrowseMod:IFastEdit 816
TBrowseMod:IFilterBar 817
TBrowseMod:IFooter 817
TBrowseMod:IHeader 817
TBrowseMod:IHideScrollBars 817
TBrowseMod:IHotTrack 817
TBrowseMod:IMultipleSel 818
TBrowseMod:IRowHeightButtons 818
TBrowseMod:IShowCellSelected 818
TBrowseMod:ISupraHeader 818
TBrowseMod:ITooltips 819
TBrowseMod:ITransparent 819
TBrowseMod:ITransparentGrid 819
TBrowseMod:IUseReturn 819
TBrowseMod:Methods 828
TBrowseMod:MinimumRowHeight 836
TBrowseMod:nClrDivider 820
TBrowseMod:nClrHeaderDivider 820
TBrowseMod:nClrHeaderEditPane 820
TBrowseMod:nClrHeaderEditPaneFocus 820
TBrowseMod:nClrHeaderEditText 821
TBrowseMod:nClrHeaderEditTextFocus 821
TBrowseMod:nClrHeaderHotPane 821
TBrowseMod:nClrHeaderHotText 821
TBrowseMod:nClrHeaderPane 822
TBrowseMod:nClrHeaderSelPane 822
TBrowseMod:nClrHeaderText 822
TBrowseMod:nClrHotPane 822
TBrowseMod:nClrHotText 823
TBrowseMod:nClrPane 823
TBrowseMod:nClrSelPane 823
TBrowseMod:nClrSelText 823
TBrowseMod:nClrText 824
TBrowseMod:nEditMode 824
TBrowseMod:nFooterHeight 824
TBrowseMod:nHeaderHeight 824
TBrowseMod:nIndex 824
TBrowseMod:nMargin 825
TBrowseMod:nRecSelectorWidth 825
TBrowseMod:nRowHeight 825
TBrowseMod:nRowIncrement 825
TBrowseMod:nSelectCol 826
TBrowseMod:oDataGrid 826
TBrowseMod:oEditCol 826
TBrowseMod:oFooter 826
TBrowseMod:oGroup 826
TBrowseMod:oHeader 827
TBrowseMod:olmageList 827
TBrowseMod:OnAppend 839
TBrowseMod:OnCancelEdit 839
TBrowseMod:OnChange 839
TBrowseMod:OnChangeSelected 839
TBrowseMod:OnClick 840
TBrowseMod:OnContextMenu 840
TBrowseMod:OnDbClick 841
TBrowseMod:OnDelete 842
TBrowseMod:OnDispTooltip 841
TBrowseMod:OnDrawActiveRow 842
TBrowseMod:OnDrawCell 842
TBrowseMod:OnDrawFooter 843
TBrowseMod:OnDrawGroupFooter 844
TBrowseMod:OnDrawGroupHeader 844
TBrowseMod:OnDrawHeader 845
TBrowseMod:OnDrawRecSelector 845
TBrowseMod:OnEdit 846
TBrowseMod:OnEditCol 846
TBrowseMod:OnEditColBtnClick 846
TBrowseMod:OnFilterEnd 847
TBrowseMod:OnFilterStart 847
TBrowseMod:OnFooterClick 847
TBrowseMod:OnFooterRClick 847
TBrowseMod:OnHeaderClick 848
TBrowseMod:OnHeaderRClick 848

- TBrowseMod:OnInsert 848
- TBrowseMod:OnPostSave 848
- TBrowseMod:OnSave 849
- TBrowseMod:OnSaveCol 849
- TBrowseMod:OnUpdate 849
- TBrowseMod:oRecSelector 827
- TBrowseMod:oSelectCol 827
- TBrowseMod:oSupra 828
- TBrowseMod:Propiedades 811
- TBrowseMod:RestoreState 836
- TBrowseMod:SaveState 836
- TBrowseMod:Sort 837
- TBrowseMod:StretchCols 837
- TBrowseMod:ToExcel 837
- TBrowseMod:UnSort 837
- TBrush 160
- TBrush:Clone 161
- TBrush:Create 162
- TBrush:Destroy 162
- TBrush:hBitmap 160
- TBrush:Methods 161
- TBrush:nColor 161
- TBrush:nStyle 161
- TBrush:Properties 160
- TBrwColMod:aEditListBound 851
- TBrwColMod:aEditListText 851
- TBrwColMod:cFilter 852
- TBrwColMod:cFooter 852
- TBrwColMod:cHeader 852
- TBrwColMod:cPicture 852
- TBrwColMod:cTooltip 853
- TBrwColMod:FilterEval 853
- TBrwColMod:ICanDrag 854
- TBrwColMod:ICanEdit 855
- TBrwColMod:ICanFilter 855
- TBrwColMod:ICanHide 855
- TBrwColMod:ICanHot 855
- TBrwColMod:ICanResize 855
- TBrwColMod:ICanSort 856
- TBrwColMod:IDataMultiLine 856
- TBrwColMod:LeftPos 856
- TBrwColMod:IMultiLine 856
- TBrwColMod:ITransparent 857
- TBrwColMod:IVisible 857
- TBrwColMod:Methods 863
- TBrwColMod:nAlignment 857
- TBrwColMod:nBmpAlignment 857
- TBrwColMod:nClrPane 858
- TBrwColMod:nDataAlignment 858
- TBrwColMod:nDataBmpAlignment 858
- TBrwColMod:nDataCol 858
- TBrwColMod:nDataVAlignment 859
- TBrwColMod:nDroppedHeight 859
- TBrwColMod:nDroppedWidth 859
- TBrwColMod:nEditLength 859
- TBrwColMod:nFilterEditStyle 861
- TBrwColMod:nImage 861
- TBrwColMod:nIndex 861
- TBrwColMod:nVAlignment 861
- TBrwColMod:oEditFilter 862
- TBrwColMod:oFooterCol 862
- TBrwColMod:oGridEdit 862
- TBrwColMod:oHeaderCol 862
- TBrwColMod:oSupraCol 862
- TBrwColMod:Properties 850
- TBrwColMod:RightPos 863
- TBrwColMod:uDefaultValue 863
- TBrwColMod:Value 863
- TBrwColumn 447
- TBrwColumn:aEditListBound 449
- TBrwColumn:aEditListText 449
- TBrwColumn:cFieldName 450
- TBrwColumn:cFilter 450
- TBrwColumn:cFooter 450
- TBrwColumn:cHeader 451
- TBrwColumn:cNullValue 451
- TBrwColumn:cPicture 451
- TBrwColumn:cSortEval 451
- TBrwColumn:cTooltip 452
- TBrwColumn:Edit 462
- TBrwColumn:Events 463
- TBrwColumn:FilterEval 452
- TBrwColumn:ICanFilter 453
- TBrwColumn:IParentFont 453
- TBrwColumn:IVisible 454
- TBrwColumn:Methods 461
- TBrwColumn:nAlignment 454
- TBrwColumn:nBmpAlignment 454
- TBrwColumn:nBmpMargin 455
- TBrwColumn:nClrPane 455

TBrwColumn:nClrText 455
TBrwColumn:nEditLength 455
TBrwColumn:nEditType 456
TBrwColumn:nFooterImage 457
TBrwColumn:nHeaderImage 457
TBrwColumn:nHeight 457
TBrwColumn:nIndex 458
TBrwColumn:nPos 458
TBrwColumn:nSort 458
TBrwColumn:nWidth 459
TBrwColumn:oDataField 459
TBrwColumn:oFont 461
TBrwColumn:oFooter 460
TBrwColumn:oGridEdit 459
TBrwColumn:oHeader 460
TBrwColumn:OnBtnClick 464
TBrwColumn:OnChangeWidth 464
TBrwColumn:OnCreate 464
TBrwColumn:OnDataClick 465
TBrwColumn:OnDatadBLClick 465
TBrwColumn:OnDataRClick 466
TBrwColumn:OnDrawCell 466
TBrwColumn:OnEdit 467
TBrwColumn:OnExit 468
TBrwColumn:OnFooterClick 468
TBrwColumn:OnFooterDbClick 468
TBrwColumn:OnFooterRClick 469
TBrwColumn:OnGetData 469
TBrwColumn:OnGetValue 470
TBrwColumn:OnGridEditCreate 470
TBrwColumn:OnHeaderClick 471
TBrwColumn:OnHeaderDbClick 471
TBrwColumn:OnHeaderRClick 471
TBrwColumn:OnPostEdit 472
TBrwColumn:oParent 461
TBrwColumn:Properties 448
TBrwColumn:SetAlignment 462
TBrwColumn:uDefaultValue 461
TBrwColumn:Value 463
TBrwColumnSelect 462
TBrwGroup 864
TBrwGroup:aGroupCols 865
TBrwGroup:lCollapsed 865
TBrwGroup:lFooter 865
TBrwGroup:lLine 865
TBrwGroup:Methods 865
TBrwGroup:Properties 864
TBrwGroup:SumTotal 866
TBrwGroupSt:Average 866
TBrwGroupSt:Header 867
TBrwGroupSt:Max 867
TBrwGroupSt:Methods 866
TBrwGroupSt:Min 867
TBrwGroupSt:Sum 867
TBrwModGrid 864
TBrwRec:nBmpAlignment 474
TBrwRecSel 473
TBrwRecSel:Events 477
TBrwRecSel:lParentFont 474
TBrwRecSel:lRecordPointer 474
TBrwRecSel:nAlignment 474
TBrwRecSel:nBmpMargin 475
TBrwRecSel:nClrPane 475
TBrwRecSel:nClrText 475
TBrwRecSel:nHeight 475
TBrwRecSel:nWidth 476
TBrwRecSel:oFont 476
TBrwRecSel:OnClick 477
TBrwRecSel:OnDbClick 477
TBrwRecSel:OnGetData 478
TBrwRecSel:OnRClick 478
TBrwRecSel:oParent 476
TBrwRecSel:Properties 473
TBtnBmp 316
TBtnBmp:lFlat 316
TBtnBmp:lXPLook 316
TBtnBmp:nBmpHeight 317
TBtnBmp:nBmpWidth 317
TBtnBmp:nFlatStyle 317
TBtnBmp:nMargin 317
TBtnBmp:nOrientation 318
TBtnBmp:nWidth 318
TBtnBmp:oBitmaps 318
TBtnBmp:Properties 316
TBtnPanelMod 868
TBtnPanelMod:lHotBorder 868
TBtnPanelMod:nClrPane 868
TBtnPanelMod:nHeight 869
TBtnPanelMod:nHotOpacity 869
TBtnPanelMod:nPushedOpacity 869

- TBtnPanelMod:nWidth 869
- TBtnPanelMod:Properties 868
- TButton 249
- TButton:Click 252
- TButton:Events 252
- TButton:ICancel 249
- TButton:IDefault 250
- TButton:IMultiLine 250
- TButton:IPushed 250
- TButton:Methods 252
- TButton:nAlignment 250
- TButton:nHeight 251
- TButton:nModalResult 251
- TButton:nVAlignment 251
- TButton:nWidth 252
- TButton:OnClick 252
- TButton:Properties 249
- TButtonEx 798
- TButtonEx:alters 799
- TButtonEx:cShortCut 799
- TButtonEx:Events 804
- TButtonEx:IDropped 799
- TButtonEx:IMultiLine 800
- TButtonEx:IPushed 800
- TButtonEx:nAlignment 800
- TButtonEx:nBmpHeight 800
- TButtonEx:nBmpWidth 801
- TButtonEx:nClrBorder 801
- TButtonEx:nClrPane 801
- TButtonEx:nClrPanelItems 801
- TButtonEx:nEffects 802
- TButtonEx:nHeight 802
- TButtonEx:nMargin 802
- TButtonEx:nOrientation 802
- TButtonEx:nStyle 803
- TButtonEx:nVAlignment 803
- TButtonEx:nWidth 803
- TButtonEx:oBitmaps 803
- TButtonEx:oFontShortCut 804
- TButtonEx:OnClick 804
- TButtonEx:OnDrop 804
- TButtonEx:Properties 798
- TButtonMod 805
- TButtonMod:Events 810
- TButtonMod:ICancel 805
- TButtonMod:IDefault 806
- TButtonMod:ILegacyBehaviour 806
- TButtonMod:IMultiLine 806
- TButtonMod:ITransparent 806
- TButtonMod:nBmpHeight 807
- TButtonMod:nBmpMargin 807
- TButtonMod:nBmpWidth 807
- TButtonMod:nBorderRadius 807
- TButtonMod:nLineSpacing 808
- TButtonMod:nModalResult 808
- TButtonMod:nOrientation 808
- TButtonMod:nVAlignment 809
- TButtonMod:oBitmaps 809
- TButtonMod:oMenu 809
- TButtonMod:OnCustomDraw 810
- TButtonMod:OnMenuClick 810
- TButtonMod:Properties 805
- TCalendarMod 869
- TCalendarMod:aDays 870
- TCalendarMod:aDaysSelected 871
- TCalendarMod:aMonths 871
- TCalendarMod:aSDays 871
- TCalendarMod:aSMonths 871
- TCalendarMod:dValue 872
- TCalendarMod:Events 876
- TCalendarMod:IHighLiteToday 872
- TCalendarMod:IShowDOW 872
- TCalendarMod:IShowFirstOfGroup 872
- TCalendarMod:IShowLines 872
- TCalendarMod:ITransparent 873
- TCalendarMod:nClrBorder 873
- TCalendarMod:nClrDaysDisabled 873
- TCalendarMod:nClrHot 873
- TCalendarMod:nClrSelection 874
- TCalendarMod:nClrTextHeader 874
- TCalendarMod:nClrToday 874
- TCalendarMod:nDisplayMode 874
- TCalendarMod:nNumberOfWeeks 875
- TCalendarMod:nSelectionMode 875
- TCalendarMod:oFontDOW 875
- TCalendarMod:oFontHeader 875
- TCalendarMod:oHeader 876
- TCalendarMod:OnChange 876
- TCalendarMod:OnChangeView 877
- TCalendarMod:OnCheckDate 877

TCalendarMod:OnDrawItem 877
TCalendarMod:oNext 876
TCalendarMod:OnSelect 878
TCalendarMod:oPrev 876
TCalendarMod:Properties 870
TCanvas:hDC 163
TCanvas 162
TCanvas:AngleArc 167
TCanvas:aPenPos 163
TCanvas:Arc 168
TCanvas:ArcTo 168
TCanvas:Chord 168
TCanvas:Create 169
TCanvas:DrawPicture 169
TCanvas:DrawRichEdit 170
TCanvas:Ellipse 170
TCanvas:ExtFloodFill 170
TCanvas:FillRect 171
TCanvas:LineTo 171
TCanvas:lTransparent 164
TCanvas:Methods 166
TCanvas:MoveTo 172
TCanvas:nClrPane 164
TCanvas:New 169
TCanvas:NewFromDC 172
TCanvas:nMapMode 164
TCanvas:nTextAlignment 165
TCanvas:nTextVAlignment 165
TCanvas:oDevice 165
TCanvas:oFont 166
TCanvas:oPen 166
TCanvas:Pie 172
TCanvas:PixelsToMapMode 173
TCanvas:PixelToTwips 173
TCanvas:PointToPixels 173
TCanvas:Polygon 174
TCanvas:Properties 163
TCanvas:Rectangle 174
TCanvas:RectToPixels 174
TCanvas:RoundRect 175
TCanvas:SelectFont 175
TCanvas:SelectPen 176
TCanvas:TextExtent 176
TCanvas:TextHeight 176
TCanvas:TextOut 177
TCanvas:TextRect 178
TCanvas:TextWidth 178
TCanvasTextLines 177
TCardBox 878
TCardBox:aData 883
TCardBox:AddItem 890
TCardBox:altems 884
TCardBox:aSelected 884
TCardBox:DeleteItem 891
TCardBox:DeleteItems 891
TCardBox:Events 892
TCardBox:HitTest 891
TCardBox:InsertItem 891
TCardBox:Introduction 879
TCardBox:lCardBorder 884
TCardBox:lCardShadow 884
TCardBox:lGapFirstCard 885
TCardBox:lHideScrollBars 885
TCardBox:lHotTrack 885
TCardBox:lMultipleSel 885
TCardBox:Methods 890
TCardBox:nCardClrAct 886
TCardBox:nCardClrHot 885
TCardBox:nCardClrLink 886
TCardBox:nCardClrPane 886
TCardBox:nCardClrSel 886
TCardBox:nCardGap 887
TCardBox:nCardInnerGap 887
TCardBox:nCardMargin 887
TCardBox:nCardSize 887
TCardBox:nClickItem 888
TCardBox:nClickPos 888
TCardBox:nHeight 888
TCardBox:nHotItem 888
TCardBox:nHotPos 889
TCardBox:nIndex 889
TCardBox:nWidth 889
TCardBox:olImageList 889
TCardBox:oLabelEx 890
TCardBox:OnCardChange 892
TCardBox:OnCardClick 892
TCardBox:OnCardDbClick 893
TCardBox:OnCardGapPaint 893
TCardBox:OnCardPaint 893
TCardBox:OnCardSelection 894

- TCardBody:Properties 883
- TCardBody:RecCount 892
- TCardBodyItem 894
- TCardBodyItem:IsActive 901
- TCardBodyItem:IButton 895
- TCardBodyItem:IHotTrack 895
- TCardBodyItem:IMultiLine 895
- TCardBodyItem:IParentFont 895
- TCardBodyItem:IStretch 896
- TCardBodyItem:IVisible 896
- TCardBodyItem:Methods 901
- TCardBodyItem:nAlign 896
- TCardBodyItem:nAlignment 897
- TCardBodyItem:nAlignWeight 897
- TCardBodyItem:nClrLink 897
- TCardBodyItem:nClrText 897
- TCardBodyItem:nColumn 898
- TCardBodyItem:nItem 898
- TCardBodyItem:nSiblingWeight 898
- TCardBodyItem:nSize 898
- TCardBodyItem:nTextPad 899
- TCardBodyItem:nType 899
- TCardBodyItem:nVAlignment 899
- TCardBodyItem:oCursor 900
- TCardBodyItem:oFont 900
- TCardBodyItem:oParent 900
- TCardBodyItem:Properties 894
- TCdoMail 741
- TCdoMail:aAttachments 742
- TCdoMail:cCC 742
- TCdoMail:cFrom 742
- TCdoMail:cMessage 742
- TCdoMail:cPassword 743
- TCdoMail:cServer 743
- TCdoMail:cSubject 743
- TCdoMail:cTO 743
- TCdoMail:cUser 744
- TCdoMail:IAuthenticate 744
- TCdoMail:IGmailOptions 744
- TCdoMail:IHtml 744
- TCdoMail:IInstaled 744
- TCdoMail:ISSL 745
- TCdoMail:Methods 745
- TCdoMail:nPort 745
- TCdoMail:Send 745
- TCdoMailcBCC 742
- TCdxDataSource 1197
- TCdxDataSource:cDriver 1197
- TCdxDataSource:DefIdxExtension 1198
- TCdxDataSource:IAutoOpen 1198
- TCdxDataSource:Methods 1198
- TCdxDataSource:nTableType 1198
- TCdxDataSource:Properties 1197
- TCefBrowser 729
- TCefBrowser:CanGoBack 731
- TCefBrowser:CanGoForward 731
- TCefBrowser:cCachePath 730
- TCefBrowser:ClearHistory 732
- TCefBrowser:cLocale 730
- TCefBrowser:CloseDevTools 732
- TCefBrowser:Events 735
- TCefBrowser:GetZoomLevel 732
- TCefBrowser:GoBack 732
- TCefBrowser:GoForward 733
- TCefBrowser:HasDocument 733
- TCefBrowser:HidePopup 733
- TCefBrowser:IsPopupVisible 733
- TCefBrowser:Methods 731
- TCefBrowser:Navigate 733
- TCefBrowser:nBorderStyle 730
- TCefBrowser:nHeight 730
- TCefBrowser:nWidth 731
- TCefBrowser:OnAddressChange 735
- TCefBrowser:OnAfterCreated 736
- TCefBrowser:OnAuthCredentials 736
- TCefBrowser:OnBeforePopup 736
- TCefBrowser:OnBeforeResourceLoad 736
- TCefBrowser:OnClose 737
- TCefBrowser:OnDownloadComplete 737
- TCefBrowser:OnDownloadData 737
- TCefBrowser:OnDownloadFile 738
- TCefBrowser:OnDownloadStart 738
- TCefBrowser:OnLoadEnd 739
- TCefBrowser:OnLoadError 739
- TCefBrowser:OnLoadStart 739
- TCefBrowser:OnNavStateChange 740
- TCefBrowser:OnProtocolExecution 740
- TCefBrowser:OnStatusMessage 740
- TCefBrowser:OnTitleChange 741
- TCefBrowser:Properties 729

TCefBrowser:Reload 734
TCefBrowser:SetText 734
TCefBrowser:SetZoomLevel 734
TCefBrowser:ShowDevTools 734
TCefBrowser:StopLoad 735
TCheckBox 253
TCheckBox:Click 255
TCheckBox:Events 256
TCheckBox:IChecked 253
TCheckBox:IMultiLine 253
TCheckBox:IPushLike 254
TCheckBox:ITransparent 254
TCheckBox:Methods 255
TCheckBox:nAlignment 254
TCheckBox:nHeight 254
TCheckBox:nVAlignment 254
TCheckBox:nWidth 255
TCheckBox:OnChange 256
TCheckBox:OnClick 256
TCheckBox:Properties 253
TCheckBox:Toggle 255
TCheckBoxMod 901
TCheckBoxMod:Events 903
TCheckBoxMod:IChecked 902
TCheckBoxMod:ITransparent 902
TCheckBoxMod:nClrColorization 902
TCheckBoxMod:nClrTextDisabled 903
TCheckBoxMod:nLineSpacing 903
TCheckBoxMod:nVAlignment 903
TCheckBoxMod:OnChange 903
TCheckBoxMod:Properties 902
TChooseColorDlg 1078
TChooseColorDlg:aCustomColors 1079
TChooseColorDlg:cHtmlColor 1079
TChooseColorDlg:Events 1082
TChooseColorDlg:Execute 1082
TChooseColorDlg:Functions 1083
TChooseColorDlg:HelpID 1080
TChooseColorDlg:IAnyColor 1080
TChooseColorDlg:ICenter 1080
TChooseColorDlg:IFullOpen 1080
TChooseColorDlg:IPreventFullOpen 1080
TChooseColorDlg:IRGBInit 1081
TChooseColorDlg:IShowHelp 1081
TChooseColorDlg:ISolidColor 1081
TChooseColorDlg:Methods 1082
TChooseColorDlg:nColor 1081
TChooseColorDlg:nLastError 1082
TChooseColorDlg:OnClose 1082
TChooseColorDlg:OnHelp 1083
TChooseColorDlg:OnShow 1083
TChooseColorDlg:Run 1082
TChooseColorDlg:Properties 1079
TChooseFont:SetFont 1095
TChooseFontDlg 1085
TChooseFontDlg:cName 1086
TChooseFontDlg:CreateFont 1094
TChooseFontDlg:Events 1095
TChooseFontDlg:Execute 1094
TChooseFontDlg:Functions 1096
TChooseFontDlg:HelpID 1086
TChooseFontDlg:IAllFonts 1087
TChooseFontDlg:IBold 1087
TChooseFontDlg:ICenter 1087
TChooseFontDlg:IEffects 1087
TChooseFontDlg:IFixedPitch 1087
TChooseFontDlg:IForceFontExist 1088
TChooseFontDlg:IIItalic 1088
TChooseFontDlg:INoFaceSel 1088
TChooseFontDlg:INoScript 1088
TChooseFontDlg:INoSimulations 1089
TChooseFontDlg:INoSizeSel 1089
TChooseFontDlg:INoStyleSel 1089
TChooseFontDlg:INoVectorFonts 1089
TChooseFontDlg:INoVertFonts 1089
TChooseFontDlg:IPrinterFonts 1090
TChooseFontDlg:IScalableOnly 1090
TChooseFontDlg:IScreenFonts 1090
TChooseFontDlg:IScriptsOnly 1090
TChooseFontDlg:IShowHelp 1091
TChooseFontDlg:IStrikeOut 1091
TChooseFontDlg:ITrueTypeOnly 1091
TChooseFontDlg:IUnderline 1091
TChooseFontDlg:IWysiwyg 1091
TChooseFontDlg:Methods 1094
TChooseFontDlg:nCharSet 1092
TChooseFontDlg:nColor 1092
TChooseFontDlg:nHeight 1092
TChooseFontDlg:nLastError 1092
TChooseFontDlg:nSize 1093

- TChooseFontDlg:nSizeMax 1093
- TChooseFontDlg:nSizeMin 1093
- TChooseFontDlg:nStyle 1093
- TChooseFontDlg:nWeight 1093
- TChooseFontDlg:nWidth 1094
- TChooseFontDlg:OnClose 1095
- TChooseFontDlg:OnHelp 1095
- TChooseFontDlg:OnShow 1096
- TChooseFontDlg:Properties 1085
- TChooseFontDlg:Run 1094
- TCircularImage 904
- TCircularImage:Events 905
- TCircularImage:ITransparent 904
- TCircularImage:nClrPane 904
- TCircularImage:nClrPen 905
- TCircularImage:nPenSize 905
- TCircularImage:OnClick 905
- TCircularImage:Properties 904
- TClipboard 179
- TClipboard:Close 179
- TClipboard:CountFormats 180
- TClipboard:Empty 180
- TClipboard:EnumFormatNames 181
- TClipboard:EnumFormats 180
- TClipboard:GetData 181
- TClipboard:GetDropFiles 182
- TClipboard:GetFormatName 182
- TClipboard:GetOwner 183
- TClipboard:GetText 183
- TClipboard:GetWindow 183
- TClipboard:HasData 184
- TClipboard:IsEmpty 184
- TClipboard:IsPrivateFormat 184
- TClipboard:IsSystemFormat 184
- TClipboard:Methods 179
- TClipboard:Open 185
- TClipboard:RegisterFormat 185
- TClipboard:SetData 186
- TClipboard:SetDropFiles 186
- TClipboard:SetText 187
- TColorComboBox 319
- TColorComboBox:aColors 319
- TColorComboBox:IAddColor 320
- TColorComboBox:IShowText 320
- TColorComboBox:nColor 320
- TColorComboBox:nColorBoxSize 320
- TColorComboBox:nLineHeight 320
- TColorComboBox:Properties 319
- TComboBox 256
- TComboBox:AddItem 261
- TComboBox:altems 257
- TComboBox:DeleteItem 262
- TComboBox:Events 263
- TComboBox:InsertItem 262
- TComboBox:IAutoHScroll 257
- TComboBox:IDropped 258
- TComboBox:IExtendedUI 258
- TComboBox:ISort 258
- TComboBox:Methods 261
- TComboBox:ModifyItem 262
- TComboBox:nClrPane 258
- TComboBox:nDroppedHeight 258
- TComboBox:nDroppedWidth 259
- TComboBox:nEditSelEnd 259
- TComboBox:nEditSelStart 259
- TComboBox:nHeight 259
- TComboBox:nHorzExtent 260
- TComboBox:nIndex 260
- TComboBox:nMaxLength 260
- TComboBox:nStyle 260
- TComboBox:nTopIndex 261
- TComboBox:nWidth 261
- TComboBox:OnChange 263
- TComboBox:OnCloseUp 263
- TComboBox:OnDbClick 264
- TComboBox:OnDropDown 264
- TComboBox:OnEditChange 264
- TComboBox:OnEditUpdate 264
- TComboBox:Properties 256
- TComboBox>SelectString 262
- TComboBoxEx 481
- TComboBoxEx:almages 482
- TComboBoxEx:altems 482
- TComboBoxEx:nIndentWidth 483
- TComboBoxEx:olmageList 483
- TComboBoxEx:Properties 482
- TComboBoxMod 905
- TComboBoxMod:altems 906
- TComboBoxMod:cText 906
- TComboBoxMod:Events 910

TComboBoxMod:IDropAnimation	907	TControl:GoNextControl	56
TComboBoxMod:IDropped	907	TControl:GoPreviControl	56
TComboBoxMod:nClrDropHotPane	907	TControl:HelpID	38
TComboBoxMod:nClrDropHotText	907	TControl:Hide	57
TComboBoxMod:nClrDropPane	908	TControl:HideBalloon	57
TComboBoxMod:nClrDropSelPane	908	TControl:HideTooltip	57
TComboBoxMod:nClrDropSelText	908	TControl:IAlignKeepSpace	40
TComboBoxMod:nClrDropText	908	TControl:IAlignShiftOnRotate	40
TComboBoxMod:nDropItemHeight	909	TControl:IEnabled	41
TComboBoxMod:nDropMargin	909	TControl:IFocused	41
TComboBoxMod:nDroppedHeight	909	TControl:IParentFont	41
TComboBoxMod:nDroppedWidth	909	TControl:IRedraw	42
TComboBoxMod:nIndex	910	TControl:IVisible	42
TComboBoxMod:nStyle	910	TControl:Methods	52
TComboBoxMod:olmageList	910	TControl:nAlign	42
TComboBoxMod:OnCloseUp	911	TControl:nAlignMarginBottom	43
TComboBoxMod:OnDrawItem	911	TControl:nAlignMarginLeft	43
TComboBoxMod:OnDropCreate	911	TControl:nAlignMarginRight	43
TComboBoxMod:OnDropDown	912	TControl:nAlignMarginTop	44
TComboBoxMod:Properties	906	TControl:nAlignWeight	44
TComponent	32	TControl:nAlignWeightSibling	45
TComponent:Cargo	32	TControl:nAnchors	46
TComponent:Create	34	TControl:nBorderSides	47
TComponent:Destroy	34	TControl:nClientHeight	48
TComponent:End	35	TControl:nClientLeft	48
TComponent:EventAssigned	35	TControl:nClientTop	48
TComponent:Free	35	TControl:nClientWidth	48
TComponent:ICreated	32	TControl:nClrPane	49
TComponent:Methods	33	TControl:nClrText	49
TComponent:New	33	TControl:New	57
TComponent:oParent	33	TControl:nHeight	49
TComponent:Properties	32	TControl:nLeft	49
TControl	37	TControl:nTooltipIcon	50
TControl:BeginPaint	53	TControl:nTop	50
TControl:cBalloon	38	TControl:nWidth	50
TControl:Create	53	TControl:oBrush	51
TControl:cText	39	TControl:oCursor	51
TControl:cTooltip	39	TControl:oFont	51
TControl:cTooltipTitle	40	TControl:oForm	51
TControl:Destroy	54	TControl:OnBeforeCreate	63
TControl:Disable	54	TControl:OnBeginDrag	63
TControl:Enable	55	TControl:OnClick	64
TControl:EndPaint	55	TControl:OnContextMenu	64
TControl:Events	63	TControl:OnCreate	65
TControl:GetSiblingControl	56	TControl:OnDestroy	65

- TControl:OnDestroyed 65
- TControl:OnDispTooltip 66
- TControl:OnDragOver 66
- TControl:OnEndDrag 67
- TControl:OnHide 67
- TControl:OnLButtonDown 67
- TControl:OnLButtonUp 68
- TControl:OnMouseMove 68
- TControl:OnPaint 69
- TControl:OnPainted 69
- TControl:OnRButtonDown 70
- TControl:OnRButtonUp 70
- TControl:OnShow 70
- TControl:oParent 52
- TControl:PostMsg 58
- TControl:Properties 37
- TControl:Redraw 58
- TControl:SendMsg 59
- TControl:SetBounds 60
- TControl:SetFocus 60
- TControl:Show 60
- TControl:ShowBalloon 61
- TControl:ShowPopupMenu 61
- TControl:ShowTooltip 62
- TControl:Update 62
- TControlRefresh 59
- TCursor 187
- TCursor:Clone 188
- TCursor:cName 187
- TCursor:Create 189
- TCursor:Destroy 189
- TCursor:LoadFromFile 189
- TCursor:LoadFromResource 189
- TCursor:Methods 188
- TCursor:Properties 187
- TDataField 1267
- TDataField:Blank 1273
- TDataField:Buffer 1268
- TDataField:cDisplayName 1268
- TDataField:cFullName 1269
- TDataField:cName 1269
- TDataField:cTable 1269
- TDataField:cType 1269
- TDataField:DefValue 1270
- TDataField:Events 1276
- TDataField:FieldGet 1274
- TDataField:FieldPut 1274
- TDataField:HasChanged 1274
- TDataField:InitValue 1270
- TDataField:IAllowsNull 1270
- TDataField:IEditable 1270
- TDataField:IModified 1270
- TDataField:Load 1274
- TDataField:IPrimaryKey 1271
- TDataField:ISetNullIfEmpty 1271
- TDataField:Methods 1273
- TDataField:nDec 1271
- TDataField:nIndex 1271
- TDataField:nLen 1272
- TDataField:nSQLDec 1272
- TDataField:nSQLLen 1272
- TDataField:nSQLType 1272
- TDataField:oDataSet 1273
- TDataField:OnGetValue 1276
- TDataField:OnSetValue 1276
- TDataField:Properties 1268
- TDataField:Valtype 1275
- TDataField:Value 1273
- TDataField:VarGet 1275
- TDataField:VarPut 1275
- TDataSet 1276
- TDataSet:AddCalcField 1285
- TDataSet:AddLinkedControl 1286
- TDataSet:AddNew 1286
- TDataSet:AddUserVar 1287
- TDataSet:aFields 1277
- TDataSet:aLinkedControls 1277
- TDataSet:Assign 1287
- TDataSet:aUserFields 1278
- TDataSet:Blank 1287
- TDataSet:Bof 1288
- TDataSet:Cancel 1288
- TDataSet:CanEdit 1288
- TDataSet:cLastError 1278
- TDataSet:ClearFilter 1289
- TDataSet:Clone 1289
- TDataSet:Close 1289
- TDataSet:cName 1278
- TDataSet:Commit 1290
- TDataSet:cProcess 1279

TDataSet:Create 1290
TDataSet:DelError 1290
TDataSet>Delete 1291
TDataSet:DelLinkedControl 1291
TDataSet>Edit 1291
TDataSet:End 1292
TDataSet:Eof 1292
TDataSet:Events 1311
TDataSet:FastAddNew 1292
TDataSet:FastEdit 1293
TDataSet:FastUpdate 1294
TDataSet:FieldCount 1294
TDataSet:FieldDec 1295
TDataSet:FieldDisplayName 1295
TDataSet:FieldExist 1295
TDataSet:FieldGet 1295
TDataSet:FieldGetByName 1296
TDataSet:FieldLen 1296
TDataSet:FieldName 1296
TDataSet:FieldNames 1297
TDataSet:FieldPos 1297
TDataSet:FieldPut 1297
TDataSet:FieldSQLType 1297
TDataSet:FieldType 1298
TDataSet:Filter 1298
TDataSet:GetRecord 1299
TDataSet:GetRow 1299
TDataSet:GetRowHash 1300
TDataSet:GetRows 1300
TDataSet:GetRowsHash 1300
TDataSet:GoBottom 1301
TDataSet:Goto 1301
TDataSet:GoTop 1301
TDataSet:InitFrom 1302
TDataSet:IsReadOnly 1302
TDataSet:KeyCount 1302
TDataSet:KeyGoto 1303
TDataSet:KeyNo 1303
TDataSet:IDisplayErrors 1279
TDataSet:Load 1304
TDataSet:Locate 1304
TDataSet:LOnAppend 1304
TDataSet:LOnEdit 1305
TDataSet:LOpen 1279
TDataSet:LOnReadOnly 1280
TDataSet:IUpdLinked 1280
TDataSet:Methods 1283
TDataSet:Modified 1305
TDataSet:nCursorLocation 1280
TDataSet:nCursorType 1281
TDataSet:nLastError 1281
TDataSet:nLockType 1282
TDataSet:nMaxRecords 1282
TDataSet:nSkip 1305
TDataSet:oDataSource 1283
TDataSet:oFieldByName 1305
TDataSet:OnAddNew 1312
TDataSet:OnCancel 1312
TDataSet:OnClose 1312
TDataSet:OnCreate 1312
TDataSet:OnEdit 1313
TDataSet:OnOpen 1313
TDataSet:OnPostChange 1313
TDataSet:OnPostClose 1314
TDataSet:OnPostDelete 1315
TDataSet:OnPostOpen 1315
TDataSet:OnPostSave 1315
TDataSet:OnPreChange 1315
TDataSet:OnPreDelete 1316
TDataSet:OnPreSave 1317
TDataSet:OnUpdate 1317
TDataSet:Open 1306
TDataSet:oRecords 1283
TDataSet:Properties 1277
TDataSet:RecCount 1306
TDataSet:RecNo 1307
TDataSet:Refresh 1307
TDataSet:RefreshCurrent 1307
TDataSet:ReleaseState 1307
TDataSet:RestoreState 1308
TDataSet:SaveFrom 1308
TDataSet:SaveState 1308
TDataSet:SetFilter 1309
TDataSet:Skip 1309
TDataSet:Sort 1310
TDataSet:Update 1310
TDataSet:UpdateControls 1310
TDataSet:VarGet 1311
TDataSet:VarPut 1311
TDataSource 1180

- TDataSource:aDataSet 1181
- TDataSource:AddDataSet 1183
- TDataSource:BuildSQLSt 1183
- TDataSource:cConnect 1181
- TDataSource:CloseProcess 1184
- TDataSource:Create 1184
- TDataSource:DelDataSet 1184
- TDataSource:DelErrors 1184
- TDataSource:DelProcess 1185
- TDataSource:EndProcess 1185
- TDataSource:Events 1188
- TDataSource:GetErrors 1185
- TDataSource:IsError 1185
- TDataSource:IAbortOnErrors 1181
- TDataSource:LastError 1185
- TDataSource:IConnected 1182
- TDataSource:IDisplayErrors 1182
- TDataSource:Methods 1182
- TDataSource:NewProcess 1186
- TDataSource:nTimeOut 1182
- TDataSource:OnConnect 1188
- TDataSource:OnConnected 1189
- TDataSource:OnCreate 1189
- TDataSource:OnDisconnect 1189
- TDataSource:OnDisconnected 1189
- TDataSource:OnError 1189
- TDataSource:OnExecute 1190
- TDataSource:OpenProcess 1186
- TDataSource:Properties 1180
- TDataSource:QueryMemDataset 1186
- TDataSource:RunProc 1187
- TDataSource:RunProcDirect 1187
- TDataSource:ShowErrorList 1187
- TDataSource:SqlInsert 1187
- TDataSource:SqlUpdate 1188
- TDateEdit 321
- TDateEdit:nWidth 321
- TDateEdit:Properties 321
- TDateEdit:Value 321
- TDateEditMod 912
- TDateEditMod:Events 915
- TDateEditMod:IDropAnimation 913
- TDateEditMod:IDropShowDOW 913
- TDateEditMod:IDropShowFirstOfGroup 913
- TDateEditMod:IDropShowLines 913
- TDateEditMod:nClrDropBorder 913
- TDateEditMod:nClrDropDaysDisabled 914
- TDateEditMod:nClrDropHot 914
- TDateEditMod:nClrDropTextHeader 914
- TDateEditMod:nClrDropToday 914
- TDateEditMod:nDropNumerOfWeeks 915
- TDateEditMod:nDroppedHeight 915
- TDateEditMod:nDroppedWidth 915
- TDateEditMod:OnCloseUp 916
- TDateEditMod:OnDrawItem 915
- TDateEditMod:OnDropCreate 916
- TDateEditMod:OnDropDown 916
- TDateEditMod:Properties 912
- TDatePicker 483
- TDatePicker:cCustomFormat 484
- TDatePicker:dMaxDate 485
- TDatePicker:dMinDate 485
- TDatePicker:dValue 485
- TDatePicker:Events 488
- TDatePicker:ICheckBox 485
- TDatePicker:IUpDown 486
- TDatePicker:nAlignment 486
- TDatePicker:nClrText 486
- TDatePicker:nClrTitlePane 487
- TDatePicker:nClrTitleText 487
- TDatePicker:nClrTrailingText 487
- TDatePicker:nFormatType 487
- TDatePicker:nHeight 488
- TDatePicker:nWidth 488
- TDatePicker:OnChange 488
- TDatePicker:OnCloseUp 488
- TDatePicker:OnDropDown 489
- TDatePicker:Properties 483
- TDBBrowse 1367
- TDBBrowse:cMsgNoFilter 1368
- TDBBrowse:IAutoOrder 1368
- TDBBrowse:IAutoSeek 1368
- TDBBrowse:oDataSet 1369
- TDBBrowse:Properties 1368
- TDBBrowseMod 1369
- TDbBrowseMod:DataCol 1370
- TDbBrowseMod:Methods 1370
- TDBBrowseMod:oDataSet 1370
- TDBBrowseMod:Propiedades 1370
- TDBBrwColMod:cDataField 1371

TDBBrwColMod:oDataField 1371
TDBBrwColMod:Properties 1371
TDBCardBox 1371
TDBCardBox:oDataSet 1372
TDBCardBox:Properties 1372
TDBCheckBox 1373
TDBCheckBox:IAutoSave 1373
TDBCheckBox:IEditable 1374
TDBCheckBox:Methods 1375
TDBCheckBox:oDataField 1374
TDBCheckBox:oDataSet 1374
TDBCheckBox:Properties 1373
TDBCheckBox:Value 1375
TDBCheckBoxMod 1375
TDBCheckBoxMod:IAutoSave 1375
TDBCheckBoxMod:IEditable 1376
TDBCheckBoxMod:Methods 1377
TDBCheckBoxMod:oDataField 1376
TDBCheckBoxMod:oDataSet 1376
TDBCheckBoxMod:Properties 1375
TDBCheckBoxMod:Value 1377
TDBComboBox 1377
TDBComboBox:altemsBound 1378
TDBComboBox:cHint 1381
TDBComboBox:IAutoSave 1378
TDBComboBox:IEditable 1378
TDBComboBox:Methods 1380
TDBComboBox:nDataType 1378
TDBComboBox:oDataField 1379
TDBComboBox:oDataSet 1379
TDBComboBox:Properties 1377
TDBComboBox:Value 1380
TDBComboBoxMod 1380
TDBComboBoxMod:altemsBound 1381
TDBComboBoxMod:IAutoSave 1381
TDBComboBoxMod:IEditable 1382
TDBComboBoxMod:Methods 1383
TDBComboBoxMod:nDataType 1382
TDBComboBoxMod:oDataField 1382
TDBComboBoxMod:oDataSet 1383
TDBComboBoxMod:Propiedades 1380
TDBComboBoxMod:Value 1383
TDBDateEdit 1384
TDBDateEdit:IAutoSave 1384
TDBDateEdit:IEditable 1384
TDBDateEdit:oDataField 1385
TDBDateEdit:oDataSet 1385
TDBDateEdit:Properties 1384
TDBDateEditMod 1385
TDBDateEditMod:IAutoSave 1386
TDBDateEditMod:IEditable 1386
TDBDateEditMod:oDataField 1386
TDBDateEditMod:oDataSet 1387
TDBDateEditMod:Properties 1386
TDBDatePicker 1387
TDBDatePicker:IAutoSave 1388
TDBDatePicker:IEditable 1388
TDBDatePicker:Methods 1389
TDBDatePicker:oDataField 1388
TDBDatePicker:oDataSet 1389
TDBDatePicker:Properties 1387
TDBDatePicker:Value 1389
TDBEdit 1389
TDBEdit:IAutoSave 1390
TDBEdit:IEditable 1390
TDBEdit:oDataField 1390
TDBEdit:oDataSet 1391
TDBEdit:Properties 1390
TDBEditBtn 1391
TDBEditBtn:IEditable 1392
TDBEditBtn:oDataField 1392
TDBEditBtn:oDataSet 1392
TDBEditBtn:Properties 1391
TDBEditBtnIAutoSave 1391
TDBEditMod 1393
TDBEditMod:IAutoSave 1393
TDBEditMod:IEditable 1393
TDBEditMod:oDataField 1394
TDBEditMod:oDataSet 1394
TDBEditMod:Properties 1393
TDbfBrowse 446
TDbfBrowse:cAlias 446
TDbfBrowse:cMsgNoFilter 447
TDbfBrowse:Methods 447
TDbfBrowse:Properties 446
TDbfBrowse:SetDbf 447
TDbfDataSet 1317
TDbfDataSet:AddIdxFile 1322
TDbfDataSet:AddTag 1322
TDbfDataSet:AddTempTag 1323

- TDbfDataSet:aldxFiles 1318
- TDbfDataSet:Alias 1323
- TDbfDataSet:Append 1324
- TDbfDataSet:Area 1324
- TDbfDataSet:cAlias 1318
- TDbfDataSet:ClearRelations 1324
- TDbfDataSet:ClearScope 1325
- TDbfDataSet:Clone 1325
- TDbfDataSet:cName 1318
- TDbfDataSet:CompleteSeek 1325
- TDbfDataSet:Continue 1326
- TDbfDataSet:cPassword 1319
- TDbfDataSet>Delete 1326
- TDbfDataSet:Deleted 1327
- TDbfDataSet>DeleteTag 1327
- TDbfDataSet:DelTag 1327
- TDbfDataSet:DelTempTag 1328
- TDbfDataSet:Events 1341
- TDbfDataSet:FillLock 1328
- TDbfDataSet:FillUnLock 1328
- TDbfDataSet:ForcedSeek 1329
- TDbfDataSet:Found 1329
- TDbfDataSet:GetScope 1329
- TDbfDataSet:GetTags 1330
- TDbfDataSet:IsRecordLocked 1330
- TDbfDataSet:IKeyExact 1319
- TDbfDataSet:INew 1319
- TDbfDataSet:Locate 1330
- TDbfDataSet:IReUseDel 1320
- TDbfDataSet:IShared 1320
- TDbfDataSet:Methods 1321
- TDbfDataSet:OnPostAppend 1341
- TDbfDataSet:OnPreAppend 1341
- TDbfDataSet:OrdBagName 1331
- TDbfDataSet:OrdCount 1331
- TDbfDataSet:OrdFor 1331
- TDbfDataSet:OrdKey 1332
- TDbfDataSet:OrdKeyVal 1332
- TDbfDataSet:OrdName 1332
- TDbfDataSet:OrdNumber 1333
- TDbfDataSet:OrdSetFocus 1333
- TDbfDataSet:Pack 1333
- TDbfDataSet:Properties 1317
- TDbfDataSet:Recall 1334
- TDbfDataSet:RecLock 1334
- TDbfDataSet:RecUnLock 1335
- TDbfDataSet:Reindex 1335
- TDbfDataSet:ReleaseState 1336
- TDbfDataSet:RestoreState 1336
- TDbfDataSet:Save 1336
- TDbfDataSet:SaveState 1336
- TDbfDataSet:SaveToArray 1337
- TDbfDataSet:Seek 1338
- TDbfDataSet>Select 1338
- TDbfDataSet:SetRelation 1339
- TDbfDataSet:SetScope 1339
- TDbfDataSet:Sort 1340
- TDbfDataSet:Struct 1340
- TDbfDataSet:Zap 1340
- TDbfDataSource 1190
- TDbfDataSource:cConnect 1191
- TDbfDataSource:cDriver 1191
- TDbfDataSource:CommitAll 1194
- TDbfDataSource:cPassword 1192
- TDbfDataSource:CreateTable 1194
- TDbfDataSource:DefDataExtension 1194
- TDbfDataSource:DefIdxExtension 1195
- TDbfDataSource:DefMemoExtension 1195
- TDbfDataSource:DelTable 1195
- TDbfDataSource:IOptimize 1192
- TDbfDataSource:Methods 1193
- TDbfDataSource:NewDataSet 1195
- TDbfDataSource:nLockScheme 1192
- TDbfDataSource:nMemoSize 1193
- TDbfDataSource:nMemoType 1193
- TDbfDataSource:nTimeout 1193
- TDbfDataSource:Properties 1191
- TDBHyperLink 1394
- TDBHyperLink:Methods 1395
- TDBHyperLink:oDataField 1395
- TDBHyperLink:oDataSet 1395
- TDBHyperLink:Properties 1395
- TDBHyperLink:Refresh 1396
- TDBLabel 1396
- TDBLabel:IAutoSize 1397
- TDBLabel:Methods 1397
- TDBLabel:oDataField 1397
- TDBLabel:oDataSet 1397
- TDBLabel:Properties 1396
- TDBLabel:Refresh 1398

TDBListBox 1398
TDBListBox:altemsBound 1398
TDBListBox:IAutoSave 1399
TDBListBox:IEditable 1399
TDBListBox:Methods 1400
TDBListBox:nDataType 1399
TDBListBox:oDataField 1400
TDBListBox:oDataSet 1400
TDBListBox:Properties 1398
TDBListBox:Value 1401
TDBListBoxMod 1401
TDBListBoxMod:altemsBound 1401
TDBListBoxMod:IAutoSave 1402
TDBListBoxMod:IEditable 1402
TDBListBoxMod:nDataType 1402
TDBListBoxMod:oDataField 1403
TDBListBoxMod:oDataSet 1403
TDBListBoxMod:Properties 1401
TDBMaskEdit 1403
TDBMaskEdit:IAutoSave 1404
TDBMaskEdit:IEditable 1404
TDBMaskEdit:oDataField 1404
TDBMaskEdit:oDataSet 1405
TDBMaskEdit:Properties 1404
TDBMemo 1405
TDBMemo:IAutoSave 1406
TDBMemo:IEditable 1406
TDBMemo:oDataField 1406
TDBMemo:oDataSet 1407
TDBMemo:Properties 1405
TDBMemoMod 1407
TDBMemoMod:IAutoSave 1408
TDBMemoMod:IEditable 1408
TDBMemoMod:oDataField 1408
TDBMemoMod:oDataSet 1409
TDBMemoMod:Properties 1407
TDBNavigator 1409
TDBNavigator:aTooltips 1410
TDBNavigator:Events 1413
TDBNavigator:IBtnCancel 1410
TDBNavigator:IBtnDelete 1410
TDBNavigator:IBtnEdit 1410
TDBNavigator:IBtnFirst 1411
TDBNavigator:IBtnInsert 1411
TDBNavigator:IBtnLast 1411
TDBNavigator:IBtnNext 1411
TDBNavigator:IBtnPrevious 1412
TDBNavigator:IBtnRefresh 1412
TDBNavigator:IBtnUpdate 1412
TDBNavigator:nAlign 1412
TDBNavigator:oDataSet 1413
TDBNavigator:OnClickCancel 1413
TDBNavigator:OnClickDelete 1414
TDBNavigator:OnClickEdit 1414
TDBNavigator:OnClickFirst 1414
TDBNavigator:OnClickInsert 1414
TDBNavigator:OnClickLast 1415
TDBNavigator:OnClickNext 1415
TDBNavigator:OnClickPrevious 1415
TDBNavigator:OnClickRefresh 1415
TDBNavigator:OnClickUpdate 1416
TDBNavigator:Properties 1409
TDBRadioMenu 1416
TDBRadioMenu:IAutoSave 1416
TDBRadioMenu:IEditable 1417
TDBRadioMenu:Methods 1418
TDBRadioMenu:oDataField 1417
TDBRadioMenu:oDataSet 1417
TDBRadioMenu:Properties 1416
TDBRadioMenu:Value 1418
TDBSwitch 1420
TDBSwitch:IAutoSave 1421
TDBSwitch:IEditable 1421
TDBSwitch:Methods 1422
TDBSwitch:oDataField 1421
TDBSwitch:oDataSet 1422
TDBSwitch:Properties 1421
TDBSwitch:Value 1422
TDBTimePicker 1418
TDBTimePicker:IAutoSave 1418
TDBTimePicker:IEditable 1419
TDBTimePicker:Methods 1420
TDBTimePicker:oDataField 1419
TDBTimePicker:oDataSet 1419
TDBTimePicker:Properties 1418
TDBTimePicker:Value 1420
TEdiMod:Properties 917
TEdit 265
TEdit:CanCopy 270
TEdit:CanCut 271

TEdit:CanPaste 271
TEdit:CanUndo 271
TEdit:cPasswordChar 265
TEdit:EmptyUndoBuffer 271
TEdit:Events 273
TEdit:GetSel 272
TEdit:GetSelText 272
TEdit:IAutoSelect 266
TEdit:IBorder 266
TEdit:IHScroll 266
TEdit:IModify 266
TEdit:INumeric 267
TEdit:IPassword 267
TEdit:IReadOnly 267
TEdit:IShowSystemBalloon 267
TEdit:Methods 270
TEdit:nAlignment 268
TEdit:nCharCase 268
TEdit:nClrPaneDisabled 268
TEdit:nClrPaneFocus 268
TEdit:nClrTextDisabled 269
TEdit:nClrTextFocus 269
TEdit:nHeight 269
TEdit:nMaxLength 269
TEdit:nPos 272
TEdit:nWidth 270
TEdit:OnChange 273
TEdit:Properties 265
TEdit>SelectAll 272
TEdit>SelectNone 273
TEdit:SetSel 273
TEdit:Value 270
TEditBtn 322
TEditBtn:cPrompt 322
TEditBtn:Events 323
TEditBtn:nStyle 322
TEditBtn:oBitmap 322
TEditBtn:OnBtnClick 323
TEditBtn:Properties 322
TEditForGrid 864
TEditMod 917
TEditMod:cHint 917
TEditMod:Events 918
TEditMod:nClrColorization 917
TEditMod:nStyle 918
TEditMod:oBitmaps 918
TEditMod:OnBtnClick 919
TEditMod:OnClearClick 919
TEnhMetafile 190
TEnhMetafile:cDescription 190
TEnhMetafile:Close 192
TEnhMetafile:cName 191
TEnhMetafile:Copy 192
TEnhMetafile:Create 192
TEnhMetafile:Destroy 193
TEnhMetafile:hNewDC 191
TEnhMetafile:Load 193
TEnhMetafile:Methods 191
TEnhMetafile:nHeight 191
TEnhMetafile:nWidth 191
TEnhMetafile:Paint 193
TEnhMetafile:Play 194
TEnhMetafile:Properties 190
TE ExplorerBar 323
TE ExplorerBar:AddItem 326
TE ExplorerBar:altItems 324
TE ExplorerBar:Events 327
TE ExplorerBar:HitTest 327
TE ExplorerBar:ITransparent 324
TE ExplorerBar:Methods 326
TE ExplorerBar:nAlign 324
TE ExplorerBar:nClrPane 325
TE ExplorerBar:nOpacity 325
TE ExplorerBar:nSpacing 325
TE ExplorerBar:nWidth 326
TE ExplorerBar:OnItemExpand 327
TE ExplorerBar:oPicture 326
TE ExplorerBar:Properties 324
TE ExplorerGroup 327
TE ExplorerGroup:Create 330
TE ExplorerGroup:cText 328
TE ExplorerGroup:IExpanded 328
TE ExplorerGroup:ISpecial 328
TE ExplorerGroup:IVisible 329
TE ExplorerGroup:Methods 330
TE ExplorerGroup:nItem 329
TE ExplorerGroup:oControl 329
TE ExplorerGroup:olcon 329
TE ExplorerGroup:oParent 329
TE ExplorerGroup:Properties 328

TExplorerGroup:Refresh 330
TExStruct:Methods 1426
TExStruct 1425
TExStruct:AddMember 1427
TExStruct:Clone 1427
TExStruct:GetDefaults 1427
TExStruct:GetKeys 1428
TExStruct:GetMember 1429
TExStruct:GetName 1428
TExStruct:GetValues 1428
TExStruct:Len 1429
TExStruct:MemberList 1429
TExStruct:Modified 1429
TExStruct:New 1430
TExStruct:Reset 1430
TExStruct:SetValue 1430
TExStruct:SetValues 1430
TExStruct:ToJson 1431
TextExtent 176
TextHeight 176
TextLines 177
TextOut 177
TextRect 178
TextWidth 178
TFArrayDataset:aData 1545
TFastReport 1508
TFastReport:aDatasets 1509
TFastReport:AddCategory 1515
TFastReport:AddDataset 1515
TFastReport:AddDbf 1516
TFastReport:AddFunction 1516
TFastReport:AddVariable 1517
TFastReport:Calc 1517
TFastReport:CategoriesList 1517
TFastReport:cDllPath 1509
TFastReport:cFileName 1509
TFastReport:cFRLicense 1510
TFastReport:cLanguage 1510
TFastReport:cLastError 1510
TFastReport:Clear 1518
TFastReport:ClearData 1518
TFastReport:ClearMasterDetail 1518
TFastReport:ClearVariables 1518
TFastReport:cReslco 1511
TFastReport:cResName 1511
TFastReport:cTitle 1511
TFastReport:cXALicense 1511
TFastReport:DelDataset 1519
TFastReport:DelError 1519
TFastReport>DeleteCategory 1519
TFastReport>DeleteVariable 1519
TFastReport:DoExport 1520
TFastReport:Events 1531
TFastReport:FieldValue 1520
TFastReport:GetErrors 1521
TFastReport:GetProperty 1521
TFastReport:GetVariable 1522
TFastReport:IAbortOnErrors 1512
TFastReport:IDelFRVars 1512
TFastReport:LoadFPFile 1522
TFastReport:LoadFromFile 1522
TFastReport:LoadFromResource 1522
TFastReport:LoadFromString 1523
TFastReport:LoadLangFile 1523
TFastReport:LoadStyleSheetFromFile 1523
TFastReport:LoadStyleSheetFromString 1523
TFastReport:ISynchroData 1512
TFastReport:Methods 1514
TFastReport:nLastError 1512
TFastReport:oEngineOptions 1513
TFastReport:OnAfterLoad 1532
TFastReport:OnAfterPrint 1532
TFastReport:OnAfterPrintReport 1532
TFastReport:OnBeforeConnect 1532
TFastReport:OnBeforeCreate 1532
TFastReport:OnBeforeLoad 1533
TFastReport:OnBeforePrint 1533
TFastReport:OnBeginDoc 1533
TFastReport:OnBtnGenPdf 1533
TFastReport:OnBtnPrint 1534
TFastReport:OnBtnSendMail 1534
TFastReport:OnClickObject 1534
TFastReport:OnDbClickObject 1534
TFastReport:OnDesignLoadReport 1535
TFastReport:OnDesignSaveReport 1535
TFastReport:OnGetValue 1535
TFastReport:OnMouseOverObject 1536
TFastReport:OnPageChanged 1536
TFastReport:OnPreview 1536
TFastReport:OnPrintPage 1536

TFastReport:OnPrintReport 1537
TFastReport:OnProgress 1537
TFastReport:OnProgressStart 1537
TFastReport:OnProgressStop 1538
TFastReport:oPreviewOptions 1513
TFastReport:oPrintOptions 1513
TFastReport:oReportOptions 1513
TFastReport:PrepareReport 1524
TFastReport:PreviewClose 1524
TFastReport:Print 1524
TFastReport:Properties 1508
TFastReport:RestoreVariables 1524
TFastReport:SaveToFile 1524
TFastReport:SaveToFPFile 1525
TFastReport:SaveToString 1525
TFastReport:SaveVariables 1525
TFastReport:SetADODConnectStr 1525
TFastReport:SetFormatSettings 1526
TFastReport:SetIcon 1528
TFastReport:SetLicense 1528
TFastReport:SetProperty 1529
TFastReport:SetVariable 1530
TFastReport:ShowPreparedReport 1530
TFastReport:ShowReport 1530
TFastReport:StopReport 1530
TFastReport:VariablesList 1531
TFastReportAddArray 1515
TFileName 1431
TFileName:cFileName 1432
TFileName:Create 1434
TFileName:Drive 1432
TFileName:Extension 1433
TFileName:FullName 1433
TFileName:HasDrive 1433
TFileName:HasExtension 1433
TFileName:HasPath 1434
TFileName:Methods 1432
TFileName:New 1434
TFileName:Path 1434
TFileName:Properties 1431
TFileName:SetExtension 1434
TFileName:SetName 1435
TFileName:SetPath 1435
TFileName:ShortName 1435
TFileName:Unique 1435
TFileName:WithExtension 1436
TFileName:WithName 1436
TFileName:WithPath 1436
TFileOpenDlg 1098
TFileOpenDlg:cDefaultExt 1099
TFileOpenDlg:cFilename 1099
TFileOpenDlg:cFilenameOnly 1100
TFileOpenDlg:cFilePath 1100
TFileOpenDlg:cFilter 1100
TFileOpenDlg:cFullFilename 1101
TFileOpenDlg:cInitialDir 1101
TFileOpenDlg:cTitle 1101
TFileOpenDlg:Events 1106
TFileOpenDlg:Execute 1106
TFileOpenDlg:HelpID 1101
TFileOpenDlg:IAllowMultiSelect 1102
TFileOpenDlg:ICenter 1102
TFileOpenDlg:ICreatePrompt 1102
TFileOpenDlg:IDontAddToRecent 1102
TFileOpenDlg:IEnableSizing 1103
TFileOpenDlg:IFileMustExist 1103
TFileOpenDlg:IForceShowHidden 1103
TFileOpenDlg:IHideReadOnly 1103
TFileOpenDlg:INoChangeDir 1104
TFileOpenDlg:INoDereferenceLinks 1104
TFileOpenDlg:INoNetworkButton 1104
TFileOpenDlg:INoValidate 1104
TFileOpenDlg:IPathMustExist 1105
TFileOpenDlg:IReadOnly 1105
TFileOpenDlg:IShareAware 1105
TFileOpenDlg:Methods 1106
TFileOpenDlg:nFilterIndex 1105
TFileOpenDlg:OnClose 1106
TFileOpenDlg:OnHelp 1106
TFileOpenDlg:OnShow 1107
TFileOpenDlg:Properties 1098
TFileOpenDlg:Run 1106
TFileOpenImageDlg 1098
TFileOperationDlg 1108
TFileOperationDlg:aFrom 1109
TFileOperationDlg:aRenamed 1109
TFileOperationDlg:aTo 1109
TFileOperationDlg:cText 1110
TFileOperationDlg:Execute 1113
TFileOperationDlg:IAborted 1110

TFileOperationDlg:IAllowUndo 1110
 TFileOperationDlg:IAutoRename 1110
 TFileOperationDlg:IFilesOnly 1111
 TFileOperationDlg:INoConfirmation 1111
 TFileOperationDlg:INoConfirmMkDir 1111
 TFileOperationDlg:INoCopySecurityAttrib 1111
 TFileOperationDlg:INoErrorUI 1112
 TFileOperationDlg:ISilent 1112
 TFileOperationDlg:ISimpleProgress 1112
 TFileOperationDlg:IWantMapping 1112
 TFileOperationDlg:Methods 1113
 TFileOperationDlg:nAction 1113
 TFileOperationDlg:Properties 1108
 TFileOperationDlg:Run 1113
 TFileSaveDlg 1114
 TFileSaveDlg:cDefaultExt 1115
 TFileSaveDlg:cTitle 1115
 TFileSaveDlg:INoTestFileCreate 1115
 TFileSaveDlg:IOverWritePrompt 1115
 TFileSaveDlg:Properties 1114
 TFileSaveImageDlg 1114
 TFindDlg 1117
 TFindDlg:cText 1117
 TFindDlg:Events 1120
 TFindDlg:Execute 1120
 TFindDlg:HelpID 1118
 TFindDlg:ICenter 1118
 TFindDlg:IDown 1118
 TFindDlg:IMatchCase 1118
 TFindDlg:INoCase 1119
 TFindDlg:INoUpDown 1119
 TFindDlg:INoWholeWord 1119
 TFindDlg:IShowHelp 1119
 TFindDlg:IWholeWord 1119
 TFindDlg:Methods 1120
 TFindDlg:OnClose 1121
 TFindDlg:OnFind 1120
 TFindDlg:OnHelp 1121
 TFindDlg:Properties 1117
 TFindDlg:Run 1120
 TFindFile 1121
 TFindFile:cCreationTime 1122
 TFindFile:cFileMask 1123
 TFindFile:cFileName 1123
 TFindFile:cFullPath 1123
 TFindFile:cLastAccessTime 1123
 TFindFile:cLastWriteTime 1123
 TFindFile:Create 1127
 TFindFile:cShortFileName 1124
 TFindFile:cShortPath 1124
 TFindFile:dCreationDate 1124
 TFindFile:Destroy 1128
 TFindFile:dLastAccessDate 1124
 TFindFile:dLastWriteDate 1125
 TFindFile:Events 1128
 TFindFile:Find 1128
 TFindFile:FindNext 1128
 TFindFile:IArchive 1125
 TFindFile:ICompressed 1125
 TFindFile:IDirectory 1125
 TFindFile:IEncrypted 1125
 TFindFile:IHidden 1126
 TFindFile:INormal 1126
 TFindFile:IOffline 1126
 TFindFile:IReadOnly 1126
 TFindFile:ISystem 1127
 TFindFile:ITemporary 1127
 TFindFile:Methods 1127
 TFindFile:nFileSize 1127
 TFindFile:OnFind 1129
 TFindFile:OnStart 1129
 TFindFile:OnStop 1129
 TFindFile:Properties 1122
 TFolder 489
 TFolder:altems 489
 TFolder:Properties 489
 TFolderPage 490
 TFolderPage:cText 491
 TFolderPage>Delete 491
 TFolderPage:Disable 492
 TFolderPage:Enable 492
 TFolderPage:ITransparent 490
 TFolderPage:Methods 491
 TFolderPage:nClrText 491
 TFolderPage:nImage 491
 TFolderPage:Properties 490
 TFolderPage>Select 492
 TFont 194
 TFont:Clone 198
 TFont:Compare 199

- TFont:Create 199
- TFont:Destroy 200
- TFont:GetTextHeight 200
- TFont:GetTextSize 200
- TFont:GetTextWidth 200
- TFont:IBold 195
- TFont:Italic 195
- TFont:IStrikeOut 195
- TFont:IUnderline 195
- TFont:Methods 198
- TFont:nCharSet 196
- TFont:nHeight 196
- TFont:nOrientation 196
- TFont:nSize 197
- TFont:nStyle 197
- TFont:nWeight 197
- TFont:nWidth 198
- TFont:oDevice 198
- TFont:Properties 194
- TFont:RestoreFromText 201
- TFont:SaveToText 201
- TFontComboBox 330
- TFontComboBox:cSampleText 331
- TFontComboBox:IComboFont 331
- TFontComboBox:IEditBmp 331
- TFontComboBox:IFixedPitch 332
- TFontComboBox:nCharSet 332
- TFontComboBox:nFontHeight 332
- TFontComboBox:nFontType 332
- TFontComboBox:Properties 331
- TForm 88
- TForm:aComponents 89
- TForm:Close 99
- TForm:Events 102
- TForm:IActive 89
- TForm:IAppWindow 90
- TForm:ICentered 90
- TForm:ICloseBox 90
- TForm:IDirectShortCuts 90
- TForm:IHelpIcon 91
- TForm:IHideOnClose 91
- TForm:IMaximizeBox 91
- TForm:IMinimizeBox 92
- TForm:IModal 92
- TForm:IMsgAuto 92
- TForm:IParentIcon 92
- TForm:ISysMenu 93
- TForm:Maximize 100
- TForm:Methods 99
- TForm:Minimize 100
- TForm:nBorderStyle 93
- TForm:nFormType 94
- TForm:nMaxHeight 94
- TForm:nMaxPosX 94
- TForm:nMaxPosY 94
- TForm:nMaxWidth 95
- TForm:nMinHeight 95
- TForm:nMinWidth 95
- TForm:nModalResult 95
- TForm:nShowMode 96
- TForm:oActiveControl 96
- TForm:oCancelButton 96
- TForm:oDefaultButton 96
- TForm:olcon 97
- TForm:oMDIClient 97
- TForm:oMDIMenu 97
- TForm:oMsgBar 98
- TForm:OnActivate 102
- TForm:OnChangeFocus 102
- TForm:OnClose 103
- TForm:OnCopyData 103
- TForm:OnCreate 103
- TForm:OnDeactivate 104
- TForm:OnHelp 104
- TForm:OnHelpClick 104
- TForm:OnInitialize 105, 1056
- TForm:OnMenuSelect 105
- TForm:OnMove 106
- TForm:OnSize 106
- TForm:oPopup 98
- TForm:oPrevWindow 98
- TForm:Properties 88
- TForm:Restore 100
- TForm:RestoreState 100
- TForm:SaveState 101
- TForm:SetFocus 101
- TForm:Show 101
- TForm:ShowModal 101
- TForm:XHtmlHelp 98
- TFormatDriveDlg 1130

TFormatDriveDlg:cDrive 1130
TFormatDriveDlg:cLabel 1131
TFormatDriveDlg:Execute 1131
TFormatDriveDlg:Methods 1131
TFormatDriveDlg:nLastError 1131
TFormatDriveDlg:nType 1131
TFormatDriveDlg:Properties 1130
TFormatDriveDlg:Run 1131
TFrArrayDataset 1545
TFrArrayDataset:aFields 1545
TFrArrayDataset:Properties 1545
TFrDataset 1538
TFrDataset:aRelationFields 1539
TFrDataset:cName 1539
TFrDataset:Events 1542
TFrDataset:IsActive 1541
TFrDataset:IsLoaded 1541
TFrDataset:ILoadOnDemand 1539
TFrDataset:Methods 1541
TFrDataset:nLoaded 1540
TFrDataset:nMaxRecsOnDesign 1540
TFrDataset:oDsMaster 1540
TFrDataset:OnAfterLoad 1542
TFrDataset:OnClose 1542
TFrDataset:OnFirst 1542
TFrDataset:OnNext 1543
TFrDataset:OnOpen 1543
TFrDataset:OnPrior 1543
TFrDataset:oReport 1540
TFrDataset:Properties 1538
TFrDataset:Refresh 1541
TFrDataset:SetMaster 1541
TFrDbfDataset 1544
TFrDbfDataset:aFields 1544
TFrDbfDataset:Properties 1544
TFrEngineOptions 1546
TFrEngineOptions:cTempDir 1546
TFrEngineOptions:IDoublePass 1547
TFrEngineOptions:IIgnoreDevByZero 1547
TFrEngineOptions:IPrintIfEmpty 1547
TFrEngineOptions:ISilentMode 1547
TFrEngineOptions:IUseFileCache 1548
TFrEngineOptions:nMaxMemSize 1548
TFrEngineOptions:nNewSilentMode 1548
TFrEngineOptions:Properties 1546
TFrEngineOptions:ConvertNulls 1547
TFrPreview 1557
TFrPreview:Close 1564
TFrPreview:Edit 1564
TFrPreview:Email 1564
TFrPreview:Export 1565
TFrPreview:ExportBIF 1565
TFrPreview:ExportBMP 1565
TFrPreview:ExportCSV 1566
TFrPreview:ExportEMF 1566
TFrPreview:ExportExcel 1566
TFrPreview:ExportGIF 1566
TFrPreview:ExportHTML 1566
TFrPreview:ExportJPEG 1567
TFrPreview:ExportMail 1567
TFrPreview:ExportODS 1567
TFrPreview:ExportODT 1567
TFrPreview:ExportPDF 1568
TFrPreview:ExportRTF 1568
TFrPreview:ExportTIFF 1568
TFrPreview:ExportTXT 1568
TFrPreview:Find 1568
TFrPreview:IBtnClose 1558
TFrPreview:IBtnEdit 1558
TFrPreview:IBtnEmail 1558
TFrPreview:IBtnExport 1558
TFrPreview:IBtnFind 1559
TFrPreview:IBtnNav 1559
TFrPreview:IBtnOpen 1559
TFrPreview:IBtnOutline 1559
TFrPreview:IBtnPage 1560
TFrPreview:IBtnPDF 1560
TFrPreview:IBtnPrint 1560
TFrPreview:IBtnSave 1560
TFrPreview:IBtnThumb 1560
TFrPreview:IBtnZoom 1561
TFrPreview:IStatusBar 1561
TFrPreview:IToolbar 1561
TFrPreview:Methods 1563
TFrPreview:NavFirst 1569
TFrPreview:NavLast 1569
TFrPreview:NavNext 1569
TFrPreview:NavPrior 1569
TFrPreview:nBorderStyle 1561
TFrPreview:nHeight 1562

- TFrPreview:nPage 1562
- TFrPreview:nPageCount 1562
- TFrPreview:nWidth 1562
- TFrPreview:nZoomMode 1562
- TFrPreview:Open 1570
- TFrPreview:oReport 1563
- TFrPreview:Outline 1570
- TFrPreview:Page 1570
- TFrPreview:Print 1570
- TFrPreview:Properties 1557
- TFrPreview:Save 1570
- TFrPreview:Thumb 1571
- TFrPreview:ZoomMinus 1571
- TFrPreview:ZoomPlus 1571
- TFrPreviewOptions 1549
- TFrPreviewOptions:lAllowEdit 1549
- TFrPreviewOptions:lDoubleBuffered 1549
- TFrPreviewOptions:lMaximized 1550
- TFrPreviewOptions:lModal 1550
- TFrPreviewOptions:lOutlineExpand 1550
- TFrPreviewOptions:lOutlineVisible 1550
- TFrPreviewOptions:lPictureCacheInFile 1550
- TFrPreviewOptions:Methods 1552
- TFrPreviewOptions:nButtons 1551
- TFrPreviewOptions:nOutlineWidth 1551
- TFrPreviewOptions:nZoom 1552
- TFrPreviewOptions:Properties 1549
- TFrPreviewOptions:SetBounds 1552
- TFrPrintOptions 1552
- TFrPrintOptions:ClearOptions 1555
- TFrPrintOptions:cPageNumbers 1553
- TFrPrintOptions:cPrinter 1553
- TFrPrintOptions:lCollate 1553
- TFrPrintOptions:lReverse 1554
- TFrPrintOptions:lShowDialog 1554
- TFrPrintOptions:Methods 1554
- TFrPrintOptions:nCopies 1554
- TFrPrintOptions:nPrintPages 1554
- TFrPrintOptions:Properties 1553
- TFrReportOptions 1555
- TFrReportOptions:aDescription 1555
- TFrReportOptions:cAuthor 1556
- TFrReportOptions:cInitString 1556
- TFrReportOptions:cPassword 1556
- TFrReportOptions:lCompressed 1556
- TFrReportOptions:Properties 1555
- TFrXailerDataset 1543
- TFrXailerDataset:aFields 1544
- TFrXailerDataset:oDataset 1544
- TFrXailerDataset:Properties 1543
- TFTP 771
- TFTP:CloseFile 773
- TFTP:Command 774
- TFTP:Create 773
- TFTP:CreateDirectory 773
- TFTP>DeleteFile 774
- TFTP:Directory 774
- TFTP:DownloadFile 775
- TFTP:Events 778
- TFTP:GetCurrentDirectory 775
- TFTP:GetFile 775
- TFTP:GetFileSize 776
- TFTP:lPassive 772
- TFTP:Methods 772
- TFTP:nBuffer 772
- TFTP:nTransferType 772
- TFTP:OnCommand 778
- TFTP:OnComplete 779
- TFTP:OnDirectory 779
- TFTP:OnError 779
- TFTP:OnProgress 779
- TFTP:OnStart 780
- TFTP:OpenFile 776
- TFTP:OpenFileRead 776
- TFTP:OpenFileWrite 776
- TFTP:Properties 772
- TFTP:PutFile 777
- TFTP:RemoveDirectory 777
- TFTP:RenameFile 777
- TFTP:SetCurrentDirectory 777
- TFTP:UploadFile 778
- TFuture 1606
- TFuture:AddThreadTask 1608
- TFuture:aThreadTasks 1607
- TFuture:CreateFrom 1609
- TFuture:CurrentTask 1609
- TFuture:Eventos 1610
- TFuture:LastAwaitTask 1607
- TFuture:Metodos 1608
- TFuture:nState 1607

TFuture:OnComplete 1610
TFuture:OnCompleteTask 1610
TFuture:OnError 1610
TFuture:Properties 1607
TFuture:RunSynchroTask 1608
TFuture:Stop 1609
TFuture:TaskCount 1609
TFutureTask 1611
TFutureTask:nState 1611
TFutureTask:oError 1611
TFutureTask:oParent 1611
TFutureTask:Properties 1611
TFutureTask:ReturnValue 1612
TGlassBtn 333
TGlassBtn:nClrGlow 333
TGlassBtn:nClrInnerBorder 334
TGlassBtn:nClrOuterBorder 334
TGlassBtn:nClrPane 334
TGlassBtn:nClrShine 334
TGlassBtn:nClrText 335
TGlassBtn:nHeight 335
TGlassBtn:nWidth 335
TGlassBtn:Properties 333
TGMailSend 745
TGMailSend:Authorize 750
TGMailSend:cBCC 747
TGMailSend:cCC 747
TGMailSend:cClientId 747
TGMailSend:cLanguage 748
TGMailSend:cMessageId 748
TGMailSend:cSecretId 748
TGMailSend:cSubject 748
TGMailSend:cText 748
TGMailSend:cTo 749
TGMailSend:cToken 749
TGMailSend:IHtml 749
TGMailSend:nError 749
TGMailSend:Properties 746
TGMailSend:Send 750
TGridEdit 479
TGridEdit:altems 479
TGridEdit:altemsBound 480
TGridEdit:IAutoSelect 480
TGridEdit:IEditable 480
TGridEdit:oDataField 481
TGridEdit:oDataSet 481
TGridEdit:Properties 479
TGroupBox 274
TGroupBox:Alignment 274
TGroupBox:nHeight 274
TGroupBox:nWidth 275
TGroupBox:Properties 274
TGroupBoxMod 919
TGroupBoxMod:Alignment 920
TGroupBoxMod:nFrameClrPane 920
TGroupBoxMod:nFrameClrPen 920
TGroupBoxMod:nFrameElipseX 920
TGroupBoxMod:nFrameElipseY 921
TGroupBoxMod:nFrameRect 921
TGroupBoxMod:nFrameSize 921
TGroupBoxMod:nHeight 922
TGroupBoxMod:nLineGap 921
TGroupBoxMod:nTextGap 922
TGroupBoxMod:nTextVAlignment 922
TGroupBoxMod:nWidth 922
TGroupBoxMod:Properties 919
THeader 492
THeader:AddItem 498
THeader:altems 493
THeader:DeleteItem 498
THeader:DeleteItems 499
THeader>EditFilter 499
THeader:Events 500
THeader:HitTest 499
THeader:InsertItem 499
THeader:IBorder 493
THeader:IButtons 494
THeader:ICancelDrag 494
THeader:ICancelTrack 494
THeader:IDragDrop 494
THeader:IFilterBar 495
THeader:IFlat 495
THeader:IFullDrag 495
THeader:IHotTrack 495
THeader:IResize 495
THeader:ITabStop 496
THeader:IWordBreak 496
THeader:Methods 498
THeader:nHeight 497
THeader:nWidth 497

- THeader:oEditFilter 497
- THeader:olmageList 497
- THeader:OnBeginDrag 500
- THeader:OnBeginTrack 501
- THeader:OnClick 501
- THeader:OnDbIcIck 501
- THeader:OnDividerDbIcIck 502
- THeader:OnEndDrag 502
- THeader:OnEndTrack 502
- THeader:OnFilterBtnClick 502
- THeader:OnFilterChange 503
- THeader:OnFilterEdit 503
- THeader:OnItemChanged 503
- THeader:OnItemChanging 504
- THeader:OnRClick 504
- THeader:OnTrack 504
- THeader:Properties 493
- THeaderItem 504
- THeaderItem:cFilter 505
- THeaderItem:cText 505
- THeaderItem:cToolTip 505
- THeaderItem:Events 507
- THeaderItem:nAlignment 506
- THeaderItem:nImage 506
- THeaderItem:nIndex 506
- THeaderItem:nOrder 506
- THeaderItem:nSort 507
- THeaderItem:nWidth 507
- THeaderItem:OnClick 507
- THeaderItem:OnDbIcIck 508
- THeaderItem:OnDividerDbIcIck 508
- THeaderItem:OnRClick 508
- THeaderItem:Properties 505
- THeaderItemMod 936
- THeaderItemMod:cFilter 937
- THeaderItemMod:cHeader 937
- THeaderItemMod:cTooltip 937
- THeaderItemMod:ICanDrag 937
- THeaderItemMod:ICanFilter 938
- THeaderItemMod:ICanHide 938
- THeaderItemMod:ICanHot 938
- THeaderItemMod:ICanResize 938
- THeaderItemMod:ICanSort 938
- THeaderItemMod:IMultiline 938
- THeaderItemMod:IVisible 939
- THeaderItemMod:nAlignment 939
- THeaderItemMod:nFilterEditStyle 939
- THeaderItemMod:nImage 939
- THeaderItemMod:nSort 940
- THeaderItemMod:nVAlignment 940
- THeaderItemMod:nWidth 940
- THeaderItemMod:oEditFilter 940
- THeaderItemMod:oGridCol 940
- THeaderItemMod:Properties 936
- THeaderMod 922
- THeaderMod:AddItem 928
- THeaderMod:altems 923
- THeaderMod:ClearAllFilters 929
- THeaderMod:ClearFilter 929
- THeaderMod:DeleteItem 929
- THeaderMod:DeleteItems 930
- THeaderMod:Events 932
- THeaderMod:HitTest 930
- THeaderMod:InsertItem 930
- THeaderMod:IsEmpty 931
- THeaderMod:IsFilter 931
- THeaderMod:ICanDrag 924
- THeaderMod:ICanHide 924
- THeaderMod:IFilterBar 924
- THeaderMod:IHotTrack 924
- THeaderMod:IResize 925
- THeaderMod:Methods 928
- THeaderMod:nBmpMargin 925
- THeaderMod:nClrDivider 925
- THeaderMod:nClrEditPane 925
- THeaderMod:nClrHotPane 926
- THeaderMod:nClrHotText 926
- THeaderMod:nClrPane 926
- THeaderMod:nClrSelPane 926
- THeaderMod:nClrText 927
- THeaderMod:nFilterTimeOut 927
- THeaderMod:nMarginHor 927
- THeaderMod:nMarginVer 927
- THeaderMod:nSelect 927
- THeaderMod:olmageList 928
- THeaderMod:OnDividerDbIcIck 932
- THeaderMod:OnDrawItem 932
- THeaderMod:OnFilterBtnClick 933
- THeaderMod:OnFilterChange 933
- THeaderMod:OnFilterEdit 934

THeaderMod:OnFilterTimeOut 934
THeaderMod:OnItemBeginDrag 934
THeaderMod:OnItemChanged 934
THeaderMod:OnItemChanging 935
THeaderMod:OnItemClick 935
THeaderMod:OnItemEndDrag 935
THeaderMod:OnItemRClick 936
THeaderMod:OnRebuild 936
THeaderMod:Properties 923
THeaderMod:RestoreState 931
THeaderMod:SaveState 931
THeaderMod:SwapItems 932
THelp 201
THelp:cFilename 204
THelp:Close 206
THelp:Create 206
THelp:IModal 204
THelp:IShowTopics 205
THelp:Methods 205
THelp:New 206
THelp:nType 205
THelp:PopUp 206
THelp:Properties 204
THelp:SetFile 207
THelp:ShowContext 207
THelp:ShowFinder 207
THelp:ShowIndex 207
THelp:ShowKeyword 208
THelp:ShowSearch 208
THelp:ShowTOC 208
THelp:ShowTopic 208
THotKey 509
THotKey:Events 511
THotKey:IAlt 509
THotKey:IBorder 509
THotKey:ICtrl 510
THotKey:IShift 510
THotKey:IWinKey 510
THotKey:nClrPane 510
THotKey:nHeight 511
THotKey:nKey 511
THotKey:nWidth 511
THotKey:OnChange 511
THotKey:Properties 509
Thumb 1571
THyperLink 335
THyperLink:cURL 336
THyperLink:Events 337
THyperLink:IUnderline 336
THyperLink:nClrOver 336
THyperLink:nClrText 336
THyperLink:OnClick 337
THyperLink:OnRClick 337
THyperLink:Properties 336
TIcon 209
TIcon:Clone 210
TIcon:cName 195, 209
TIcon:Create 210
TIcon:Destroy 211
TIcon:Load 211
TIcon:LoadFromFile 211
TIcon:LoadFromResource 211
TIcon:Methods 210
TIcon:nHeight 209
TIcon:nWidth 209
TIcon:Paint 212
TIcon:Properties 209
TImage 338
TImage:IAutoFit 339
TImage:ISmooth 339
TImage:ITabStop 339
TImage:ITransparent 339
TImage:nBorderStyle 340
TImage:nClrPane 340
TImage:nHeight 340
TImage:nWidth 340
TImage:nZoom 341
TImage:oPicture 341
TImage:Properties 338
TImageEditor 341
TImageEditor:ApplyLevels 344
TImageEditor:BlackWhite 344
TImageEditor:Crop 344
TImageEditor:Events 347
TImageEditor:FlipH 344
TImageEditor:FlipV 345
TImageEditor:GrayScale 345
TImageEditor:IShowHandlers 342
TImageEditor:Methods 343
TImageEditor:nBrightness 342

- TImageEditor:nClrPane 342
- TImageEditor:nContrast 342
- TImageEditor:Negative 345
- TImageEditor:nGamma 343
- TImageEditor:nSaturation 343
- TImageEditor:nTransparency 343
- TImageEditor:OnChangeRect 347
- TImageEditor:Polaroid 345
- TImageEditor:Properties 341
- TImageEditor:RGBToBGR 346
- TImageEditor:Rotate 346
- TImageEditor:Save 346
- TImageEditor:SaveToStream 346
- TImageEditor:Sepia 347
- TImageEditor:SetColorMatrix 347
- TImageList 1027
- TImageList:Add 1029
- TImageList:AddIcon 1029
- TImageList:Create 1029
- TImageList:Destroy 1030
- TImageList:GetBitmap 1030
- TImageList:GetIcon 1030
- TImageList:GetIconSize 1031
- TImageList:GetImageCount 1031
- TImageList:Methods 1028
- TImageList:New 1029
- TImageList:nHeight 1027
- TImageList:nWidth 1028
- TImageList:oParent 1028
- TImageList:Properties 1027
- TImageList:Remove 1031
- TImageList:RemoveAll 1031
- TImageList:Replacelcon 1032
- TImageList:SetIconSize 1032
- Timers & Threads functions 1703
- TIni 212
- TIni:AddSection 213
- TIni:cFilename 212
- TIni:Commit 214
- TIni:Create 214
- TIni>DeleteEntry 214
- TIni>DeleteSection 215
- TIni>DeleteSections 215
- TIni:EnumSectionEntries 215
- TIni:EnumSectionValues 215
- TIni:GetArray 216
- TIni:GetDate 216
- TIni:GetEntry 216
- TIni:GetLogical 217
- TIni:GetNumeric 217
- TIni:GetSection 217
- TIni:GetSectionEx 217
- TIni:GetSections 218
- TIni:GetSectionsEx 218
- TIni:Methods 213
- TIni:nBuffer 213
- TIni:New 214
- TIni:Properties 212
- TIni:RenameSection 218
- TIni:SetEntry 219
- TIni:SetSection 219
- TIni:SortSection 219
- TIni:SystemChanged 219
- TInplaceEdit 1174
- TInplaceEdit:Create 1174
- TInplaceEdit:Methods 1174
- TInternet 759
- TInternet:AddRequestHeader 762
- TInternet:AttemptConnect 762
- TInternet:cAgent 759
- TInternet:CanonicalizeUrl 763
- TInternet:CheckConnection 763
- TInternet:Close 763
- TInternet:CloseFile 763
- TInternet:CloseRequest 764
- TInternet:CloseURL 764
- TInternet:ConfirmZoneCrossing 764
- TInternet:Connect 764
- TInternet:cPassword 760
- TInternet:cUser 760
- TInternet:Destroy 765
- TInternet:ErrorDialog 765
- TInternet:Events 770
- TInternet:GetConnectedState 765
- TInternet:GetErrorDescription 766
- TInternet:GetFileSize 766
- TInternet:GetRequest 766
- TInternet:GetVersion 766
- TInternet:hSession 760
- TInternet:Methods 761

TInternet:nLastError 760
TInternet:nPort 760
TInternet:nService 761
TInternet:nTimeOut 761
TInternet:OnClose 770
TInternet:OnConnect 771
TInternet:OnGetData 771
TInternet:OnOpen 771
TInternet:Open 767
TInternet:OpenRequest 767
TInternet:OpenURL 768
TInternet:PostRequest 768
TInternet:Properties 759
TInternet:QueryInfo 768
TInternet:ReadFile 769
TInternet:SendRequest 769
TInternet:SetProxyData 769
TInternet:SetTimeOut 770
TInternet:WriteFile 770
TLabel 275
TLabel:Adjust 277
TLabel:IAutoSize 275
TLabel:IMultiLine 276
TLabel:IPathStyle 276
TLabel:ITransparent 276
TLabel:Methods 277
TLabel:nAlignment 276
TLabel:nBorderStyle 276
TLabel:nVAlignment 277
TLabel:Properties 275
TLabelBuddy 348
TLabelBuddy:nBuddyAlign 348
TLabelBuddy:nBuddyOffset 349
TLabelBuddy:nVAlignment 349
TLabelBuddy:oBuddy 349
TLabelBuddy:Properties 348
TLabelEx 349
TLabelEx:Events 353
TLabelEx:IAutoSize 350
TLabelEx:IMultiLine 351
TLabelEx:ITransparent 351
TLabelEx:nAlignment 351
TLabelEx:nBorderStyle 351
TLabelEx:nClrLink 352
TLabelEx:nLineSpacing 352
TLabelEx:nMargins 352
TLabelEx:nOpacity 352
TLabelEx:nScrollIncrement 352
TLabelEx:nVAlignment 353
TLabelEx:OnLinkClick 353
TLabelEx:OnScrollDown 353
TLabelEx:OnScrollPageDown 354
TLabelEx:OnScrollPageUp 354
TLabelEx:OnScrollUp 354
TLabelEx:OnThumbPosV 355
TLabelEx:OnThumbTrackV 355
TLabelEx:Properties 350
TLine 277
TLine:IVertical 278
TLine:nPenColor 278
TLine:nPenSize 278
TLine:nPenStyle 278
TLine:Properties 278
TListBox 279
TListBox:AddItem 284
TListBox:alItems 280
TListBox:cText 280
TListBox:DeleteItem 284
TListBox:DeleteItems 284
TListBox:Events 287
TListBox:GetCurSel 284
TListBox:GetHorzExtent 285
TListBox:GetSelItems 285
TListBox:InsertItem 285
TListBox:IAutoSort 280
TListBox:IBorder 280
TListBox:IHScroll 280
TListBox:IIntegralHeight 281
TListBox:IMultiColumn 281
TListBox:IMultipleSel 281
TListBox:IScrollBarAlways 281
TListBox:IUseTabStops 282
TListBox:Methods 283
TListBox:ModifyItem 285
TListBox:nClrPane 282
TListBox:nColsWidth 282
TListBox:nHeight 282
TListBox:nHorzExtent 282
TListBox:nIndex 283
TListBox:nTopIndex 283

- TListBox:nWidth 283
- TListBox:OnChange 287
- TListBox:OnDbClick 288
- TListBox:Properties 279
- TListBox>SelectString 286
- TListBox:SetCurSel 286
- TListBox:SetHorzExtent 286
- TListBox:SetSel 287
- TListBox:SetSelItems 287
- TListBoxMod 941
- TListBoxMod:aChecked 942
- TListBoxMod:AddImage 947
- TListBoxMod:AddItem 948
- TListBoxMod:altems 942
- TListBoxMod:aSelected 942
- TListBoxMod:CheckAll 948
- TListBoxMod:cText 942
- TListBoxMod:DeleteItem 948
- TListBoxMod:DeleteItems 948
- TListBoxMod:Events 952
- TListBoxMod:FirstVisible 949
- TListBoxMod:ForceItemVisible 949
- TListBoxMod:InsertItem 949
- TListBoxMod:IsItemVisible 949
- TListBoxMod:ItemChecked 950
- TListBoxMod:ItemHot 950
- TListBoxMod:ItemPushed 950
- TListBoxMod:ItemSelected 950
- TListBoxMod>LastVisible 951
- TListBoxMod:ICheckBoxes 942
- TListBoxMod:IHideScrollbars 943
- TListBoxMod:IHotTrack 943
- TListBoxMod:IHtmlText 943
- TListBoxMod:IMultiLine 943
- TListBoxMod:IMultipleSel 944
- TListBoxMod:IShortCuts 944
- TListBoxMod:IShowActive 944
- TListBoxMod:Methods 947
- TListBoxMod:ModifyItem 951
- TListBoxMod:nClrHotPane 944
- TListBoxMod:nClrHotText 945
- TListBoxMod:nClrSelPane 945
- TListBoxMod:nClrSelText 945
- TListBoxMod:nIndex 945
- TListBoxMod:nItemHeight 946
- TListBoxMod:nMargin 946
- TListBoxMod:nOrientation 946
- TListBoxMod:nVAlignment 946
- TListBoxMod:olmageList 947
- TListBoxMod:OnCalcVirtualSize 952
- TListBoxMod:OnChange 952
- TListBoxMod:OnChanged 953
- TListBoxMod:OnChangeSelected 953
- TListBoxMod:OnCheckStateChanged 953
- TListBoxMod:OnClick 953
- TListBoxMod:OnDbClick 954
- TListBoxMod:OnDrawItem 954
- TListBoxMod:OnSelect 955
- TListBoxMod:Properties 941
- TListBoxMod:RecCount 951
- TListBoxMod:SetFirstVisible 951
- TListBoxMod:UnCheckAll 952
- TListView 512
- TListView:aColumns 513
- TListView:AddColumn 520
- TListView:AddGroup 521
- TListView:AddItem 521
- TListView:aGroups 513
- TListView:altems 513
- TListView:Arrange 522
- TListView:CancelEdit 522
- TListView:DelColumn 522
- TListView>DeleteAllItems 522
- TListView:DelGroup 523
- TListView:Events 525
- TListView:GetFocused 523
- TListView:GetSelection 523
- TListView:GetVisibleCount 523
- TListView:HitTest 524
- TListView:InsertItem 524
- TListView:IsEmpty 525
- TListView:IAlignLeft 514
- TListView:IAutoArrange 514
- TListView:IBorder 514
- TListView:ICheckBoxes 514
- TListView>IDoubleBuffer 515
- TListView:IEditLabels 515
- TListView:IFullRowSelect 515
- TListView:IGridLines 515
- TListView:IGroupView 515

TListView:IHeaderDragDrop 516
TListView:IHotTrackSelect 516
TListView:ILabelTip 516
TListView:ILabelWrap 516
TListView:INoColumnHeader 517
TListView:INoSortHeader 517
TListView:IShowSelAlways 517
TListView:ISingleSel 517
TListView:ITooltips 517
TListView:Methods 520
TListView:nClrPane 518
TListView:nClrText 518
TListView:nHeight 518
TListView:nView 518
TListView:nWidth 519
TListView:olmageListHeader 519
TListView:olmageListLarge 519
TListView:olmageListSmall 519
TListView:OnBeginLabelEdit 526
TListView:OnCheckStateChanged 526
TListView:OnClick 526
TListView:OnColumnClick 527
TListView:OnDeleteItem 527
TListView:OnDrawCell 527
TListView:OnEndLabelEdit 528
TListView:OnItemChanged 528
TListView:OnKeyDown 528
TListView:OnRClick 529
TListView:Properties 512
TListView:SetBkImage 525
TListView:Sort 525
TListViewColumn 529
TListViewColumn:Create 531
TListViewColumn:cText 529
TListViewColumn:Methods 531
TListViewColumn:nAlignment 530
TListViewColumn:nImage 530
TListViewColumn:nIndex 530
TListViewColumn:nWidth 530
TListViewColumn:oParent 531
TListViewColumn:Properties 529
TListViewGroup 531
TListViewGroup:cFooter 532
TListViewGroup:cHeader 532
TListViewGroup:Create 533
TListViewGroup:Metthods 533
TListViewGroup:nFooterAlign 532
TListViewGroup:nHeaderAlign 532
TListViewGroup:nIndex 533
TListViewGroup:nState 533
TListViewGroup:oParent 533
TListViewGroup:Properties 531
TListViewItem 534
TListViewItem:altems 534
TListViewItem:Create 536
TListViewItem:cText 534
TListViewItem:cTooltip 535
TListViewItem>Delete 537
TListViewItem>Edit 537
TListViewItem:EnsureVisible 537
TListViewItem:IChecked 535
TListViewItem:Methods 536
TListViewItem:nGroup 535
TListViewItem:nImage 535
TListViewItem:nIndent 536
TListViewItem:nIndex 536
TListViewItem:oParent 536
TListViewItem:Properties 534
TListViewItem>Select 537
TMailSlotClient 787
TMailSlotClient:Close 788
TMailSlotClient:cName 788
TMailSlotClient:cServer 788
TMailSlotClient:Events 789
TMailSlotClient:IsOpen 788
TMailSlotClient:Methods 788
TMailSlotClient:OnClose 789
TMailSlotClient:OnOpen 790
TMailSlotClient:OnWrite 790
TMailSlotClient:Open 789
TMailSlotClient:Properties 787
TMailSlotClient:Write 789
TMailSlotServer 790
TMailSlotServer:Close 791
TMailSlotServer:cName 791
TMailSlotServer:cServer 791
TMailSlotServer:Events 793
TMailSlotServer:GetMessageCount 792
TMailSlotServer:HasMessages 792
TMailSlotServer:IsOpen 792

- TMailSlotServer:Methods 791
- TMailSlotServer:OnClose 793
- TMailSlotServer:OnOpen 793
- TMailSlotServer:OnRead 794
- TMailSlotServer:Open 792
- TMailSlotServer:Properties 790
- TMailSlotServer:Read 793
- TMailSlotServer:SetReadTimeout 793
- TMapi 780
- TMapi:aAttachments 781
- TMapi:aReceipts 781
- TMapi:cAccount 782
- TMapi:cFromAddress 782
- TMapi:cFromName 782
- TMapi:cMessage 782
- TMapi:cName 782
- TMapi:cPassword 783
- TMapi:Create 785
- TMapi:cSubject 783
- TMapi:cTime 783
- TMapi:dDate 783
- TMapi:Destroy 786
- TMapi:GetAccountNames 786
- TMapi:Installed 786
- TMapi:IAskPassword 784
- TMapi:IAskRecipients 784
- TMapi:ILoggedOn 784
- TMapi:INewSession 784
- TMapi:Logoff 786
- TMapi:Logon 787
- TMapi:IReceipt 784
- TMapi:IUseUI 785
- TMapi:Methods 749, 785
- TMapi:nLastError 785
- TMapi:Properties 780
- TMapi:Send 787
- TMariaDBDataSource 1228
- TMariaDBDataSource:BeginTrans 1231
- TMariaDBDataSource:cDatabase 1229
- TMariaDBDataSource:cHost 1229
- TMariaDBDataSource:CommitTrans 1231
- TMariaDBDataSource:Connect 1232
- TMariaDBDataSource:cPassword 1229
- TMariaDBDataSource:CreateTable 1232
- TMariaDBDataSource:cUser 1230
- TMariaDBDataSource:DelTable 1232
- TMariaDBDataSource:Disconnect 1233
- TMariaDBDataSource:Execute 1233
- TMariaDBDataSource:File 1233
- TMariaDBDataSource:GetCatalogs 1234
- TMariaDBDataSource:GetTables 1234
- TMariaDBDataSource:IAutoReconnect 1230
- TMariaDBDataSource:Methods 1231
- TMariaDBDataSource:nAffectedRows 1230
- TMariaDBDataSource:nPort 1230
- TMariaDBDataSource:nTimeOut 1230
- TMariaDBDataSource:Properties 1229
- TMariaDBDataSource:Query 1234
- TMariaDBDataSource:QueryArray 1234
- TMariaDBDataSource:QueryArrayHash 1235
- TMariaDBDataSource:QueryReport 1235
- TMariaDBDataSource:QueryRow 1235
- TMariaDBDataSource:QueryRowHash 1236
- TMariaDBDataSource:QueryValue 1236
- TMariaDBDataSource:RollBackTrans 1237
- TMariaDBDataSource:Table 1237
- TMaskEdit 355
- TMaskEdit:cPicture 356
- TMaskEdit:ITrimSpaces 357
- TMaskEdit:IZeroFill 357
- TMaskEdit:Properties 356
- TMemDataset 1342, 1355
- TMemDataSet:aData 1342
- TMemDataSet:AddField 1343
- TMemDataSet:aWork 1342
- TMemDataSet:Clone 1344
- TMemDataSet:Filter 1345
- TMemDataSet:GetRow 1345
- TMemDataSet:GetStructFrom 1345
- TMemDataSet:IAbortOnErrors 1343
- TMemDataSet:Methods 1343
- TMemDataSet:Open 1346
- TMemDataset:OrgRecno 1346
- TMemDataSet:Properties 1342
- TMemDataSet:ReleaseState 1346
- TMemDataSet:RestoreState 1347
- TMemDataSet:SaveState 1347
- TMemDataSet:Search 1348
- TMemDataSet:SetFilter 1345
- TMemDataSet:Sort 1348

TMemDataset:SqlInsert 1349
TMemDataSet:SqlUpdate 1349
TMemDataSet:Zap 1349
TMemo 288
TMemo:Append 291
TMemo:Clear 291
TMemo:IAutoHScroll 289
TMemo:IAutoVScroll 289
TMemo:IHScroll 289
TMemo:IVScroll 289
TMemo:IWantReturn 289
TMemo:IWantTab 290
TMemo:Methods 291
TMemo:nColumn 290
TMemo:nHeight 290
TMemo:nLine 290
TMemo:nLineCount 291
TMemo:nWidth 291
TMemo:Properties 288
TMemo:Replace 292
TMemoMod 955
TMemoMod:IModBorder 956
TMemoMod:Properties 956
TMemotMod:nClrColorization 956
TMenu 292
TMenu:AddItem 295
TMenu:AddSeparator 295
TMenu:altems 293
TMenu:cMessage 293
TMenu:Create 296
TMenu:Destroy 296
TMenu:DoDefault 296
TMenu:Methods 294
TMenu:nStyle 293
TMenu:oFont 294
TMenu:olmageList 293
TMenu:oParent 294
TMenu:Properties 292
TMenu:Refresh 296
TMenu:SetMenu 297
TMenuItem 297
TMenuItem:AddMenu 300
TMenuItem:cMessage 297
TMenuItem:Create 300
TMenuItem:cText 298
TMenuItem:Destroy 301
TMenuItem:Events 301
TMenuItem:GetMainMenu 301
TMenuItem:IChecked 298
TMenuItem:IDefault 298
TMenuItem:IEnabled 298
TMenuItem:IHilited 299
TMenuItem:IRight 299
TMenuItem:ISeparator 299
TMenuItem:Methods 300
TMenuItem:nImage 299
TMenuItem:olmageList 301
TMenuItem:OnCheckState 301
TMenuItem:OnClick 302
TMenuItem:osubMenu 300
TMenuItem:Properties 297
TMessageBox 1132
TMessageBox:aButtons 1133
TMessageBox:Beep 1137
TMessageBox:cSoundFile 1133
TMessageBox:cText 1133
TMessageBox:cTitle 1134
TMessageBox:Events 1138
TMessageBox:Execute 1138
TMessageBox:HelpId 1133
TMessageBox:IAppModal 1134
TMessageBox:ICenter 1134
TMessageBox:IHelpButton 1134
TMessageBox:IRightText 1135
TMessageBox:ISetForeground 1135
TMessageBox:ITaskModal 1135
TMessageBox:ITopMost 1135
TMessageBox:Methods 1137
TMessageBox:nDefaultButton 1136
TMessageBox:nResult 1136
TMessageBox:nStyle 1136
TMessageBox:nTimeout 1136
TMessageBox:nTimeoutResult 1137
TMessageBox:olcon 1137
TMessageBox:OnShow 1138
TMessageBox:Properties 1132
TMessageBox:Run 1138
TMonthCalendar 538
TMonthCalendar:aHolidays 539
TMonthCalendar:dMaxDate 539

- TMonthCalendar:dMinDate 539
- TMonthCalendar:dToday 539
- TMonthCalendar:dValue 539
- TMonthCalendar:Events 543
- TMonthCalendar:IBorder 540
- TMonthCalendar:IMultiSelect 540
- TMonthCalendar:INoToday 540
- TMonthCalendar:INoTodayCircle 540
- TMonthCalendar:IWeekNumbers 541
- TMonthCalendar:nClrPane 541
- TMonthCalendar:nClrText 541
- TMonthCalendar:nClrTitlePane 542
- TMonthCalendar:nClrTitleText 542
- TMonthCalendar:nClrTrailingText 542
- TMonthCalendar:nHeight 542
- TMonthCalendar:nMaxSelCount 543
- TMonthCalendar:nWidth 543
- TMonthCalendar:OnChange 543
- TMonthCalendar:OnSelect 543
- TMonthCalendar:properties 538
- TMutex 220
- TMutex:cName 220
- TMutex:Create 221
- TMutex:Destroy 221
- TMutex:Events 221
- TMutex:Methods 220
- TMutex:nTimeOut 220
- TMutex:OnCreate 221
- TMutex:OnFail 222
- TMutex:Properties 220
- TMySQLDataSource 1219
- TMySQLDataSource:BeginTrans 1222
- TMySQLDataSource:cDatabase 1220
- TMySQLDataSource:cHost 1220
- TMySQLDataSource:CommitTrans 1222
- TMySQLDataSource:Connect 1222
- TMySQLDataSource:cPassword 1220
- TMySQLDataSource:CreateTable 1223
- TMySQLDataSource:cUser 1220
- TMySQLDataSource:DelTable 1223
- TMySQLDataSource:Disconnect 1223
- TMySQLDataSource:Execute 1224
- TMySQLDataSource:File 1224
- TMySQLDataSource:GetCatalogs 1224
- TMySQLDataSource:GetTables 1224
- TMySQLDataSource:IAutoReconnect 1220
- TMySQLDataSource:Methods 1221
- TMySQLDataSource:nAffectedRows 1221
- TMySQLDataSource:nPort 1221
- TMySQLDataSource:nTimeOut 1221
- TMySQLDataSource:Properties 1219
- TMySQLDataSource:Query 1225
- TMySQLDataSource:QueryArray 1225
- TMySQLDataSource:QueryArrayHash 1225
- TMySQLDataSource:QueryReport 1226
- TMySQLDataSource:QueryRow 1226
- TMySQLDataSource:QueryRowHash 1226
- TMySQLDataSource:QueryValue 1227
- TMySQLDataSource:RollBackTrans 1227
- TMySQLDataSource:SqlNextResult 1227,1237
- TMySQLDataSource:Table 1228
- TNoEdit 357
- TNoEdit:cPicture 358
- TNoEdit:nClrPane 358
- TNoEdit:nClrText 358
- TNoEdit:nHeight 359
- TNoEdit:nWidth 359
- TNoEdit:Properties 358
- TNoEdit:Value 359
- TNotifyIcon 1032
- TNotifyIcon:AddIcon 1034
- TNotifyIcon:Create 1035
- TNotifyIcon:cToolTip 1033
- TNotifyIcon:Destroy 1035
- TNotifyIcon:Events 1038
- TNotifyIcon:Hide 1035
- TNotifyIcon:HideBalloon 1035
- TNotifyIcon:IVisible 1033
- TNotifyIcon:Methods 1034
- TNotifyIcon:New 1036
- TNotifyIcon:nImage 1033
- TNotifyIcon:oImageList 1033
- TNotifyIcon:oMenu 1034
- TNotifyIcon:OnBalloonClick 1038
- TNotifyIcon:OnBalloonHide 1038
- TNotifyIcon:OnBalloonShow 1038
- TNotifyIcon:OnBalloonTimeout 1039
- TNotifyIcon:OnDbClick 1039
- TNotifyIcon:OnLButtonDown 1039
- TNotifyIcon:OnLButtonUp 1039

TNotifyIcon:OnMouseMove 1040
TNotifyIcon:OnRButtonDbClick 1040
TNotifyIcon:OnRButtonDown 1041
TNotifyIcon:OnRButtonUp 1040
TNotifyIcon:oParent 1034
TNotifyIcon:Properties 1032
TNotifyIcon:Refresh 1036
TNotifyIcon:SetIcon 1036
TNotifyIcon:SetMenu 1036
TNotifyIcon:SetToolTip 1037
TNotifyIcon:Show 1037
TNotifyIcon:ShowBalloon 1037
TNtxDataSource 1196
TNtxDataSource:cDriver 1196
TNtxDataSource:DefIdxExtension 1197
TNtxDataSource:Methods 1196
TNtxDataSource:Properties 1196
TOdbcDataSet 1350
TOdbcDataSet:CanEdit 1352
TOdbcDataSet:Complete 1352
TOdbcDataSet:cSelect 1350
TOdbcDataSet:Events 1354
TOdbcDataSet:FieldIsNull 1352
TOdbcDataSet:GetPrimaryKeys 1353
TOdbcDataSet:IsComplete 1353
TOdbcDataSet:IAsynFetch 1351
TOdbcDataSet:INoCompare 1351
TOdbcDataSet:Methods 1351
TOdbcDataSet:OnComplete 1354
TOdbcDataSet:Properties 1350
TOdbcDataSet:RecordHasChanged 1353
TOdbcDataSet:SetPrimaryKeys 1353
TOdbcDataSource 1261
TOdbcDataSource:cConnect 1262
TOdbcDataSource:cDSN 1262
TOdbcDataSource:CommitAll 1264
TOdbcDataSource:cPassword 1263
TOdbcDataSource:cUser 1263
TOdbcDataSource:DelTable 1264
TOdbcDataSource:Execute 1265
TOdbcDataSource:File 1265
TOdbcDataSource:GetTables 1265
TOdbcDataSource:IAutoCommit 1263
TOdbcDataSource:IDateAsString 1263
TOdbcDataSource:Methods 1264
TOdbcDataSource:nAffectedRecords 1264
TOdbcDataSource:Properties 1262
TOdbcDataSource:Query 1266
TOdbcDataSource:QueryArray 1266
TOdbcDataSource:QueryRow 1266
TOdbcDataSource:QueryValue 1267
TOdbcDataSource:RollBackAll 1267
ToExcel 431, 837, 1485
Toggle 255, 978
ToJson 1431
TOptionItem 359
TOptionItem:AddItem 363
TOptionItem:AddSeparator 364
TOptionItem:alters 360
TOptionItem:Create 364
TOptionItem:cText 360
TOptionItem>Delete 365
TOptionItem:Events 366
TOptionItem:InsertItem 365
TOptionItem:IChecked 360
TOptionItem:ICollapsible 361
TOptionItem:IEnabled 361
TOptionItem:IExpanded 361
TOptionItem:IMultiLine 361
TOptionItem:Methods 363
TOptionItem:nImage 362
TOptionItem:nImageDisabled 362
TOptionItem:nType 362
TOptionItem:OnClick 366
TOptionItem:oOptionList 362
TOptionItem:Properties 360
TOptionList 366
TOptionList:AddItem 371
TOptionList:AddSeparator 372
TOptionList:alters 367
TOptionList:Events 374
TOptionList:GetItemByPos 372
TOptionList:GetItemCount 373
TOptionList:GetItemPos 373
TOptionList:HitTest 373
TOptionList:InsertItem 373
TOptionList:IAutoSize 367
TOptionList:ITabStop 367
TOptionList:ITransparent 368
TOptionList:Methods 371

- TOptionList:nBorderStyle 368
- TOptionList:nClrPane 368
- TOptionList:nClrText 368
- TOptionList:nClrTextHot 369
- TOptionList:nHeight 369
- TOptionList:nHotStyle 369
- TOptionList:nItemHeight 369
- TOptionList:nLeftMargin 370
- TOptionList:nMargin 370
- TOptionList:nScrollIncrement 370
- TOptionList:nTopMargin 370
- TOptionList:nWidth 371
- TOptionList:olmageList 371
- TOptionList:OnScrollDown 374
- TOptionList:OnScrollPageDown 375
- TOptionList:OnScrollPageUp 375
- TOptionList:OnScrollUp 375
- TOptionList:OnThumbPosV 376
- TOptionList:OnThumbTrackV 376
- TOptionList:Properties 366
- ToString 1689
- TPage 376
- TPage:Delete 378
- TPage:ITransparent 377
- TPage:Methods 378
- TPage:nItem 377
- TPage:Properties 377
- TPage>Select 378
- TPager 544
- TPager:IAutoScroll 544
- TPager:nHeight 545
- TPager:nOrientation 545
- TPager:nPosition 545
- TPager:nScrollIncrement 545
- TPager:nWidth 545
- TPager:Properties 544
- TPages 378
- TPages:Additem 381
- TPages:altems 379
- TPages:DeleteItem 381
- TPages:DeleteItems 382
- TPages:InsertItem 382
- TPages:ITabStop 380
- TPages:ITransparent 380
- TPages:Methods 381
- TPages:nBorderStyle 380
- TPages:Next 382
- TPages:nHeight 380
- TPages:nIndex 380
- TPages:nWidth 381
- TPages:Previous 382
- TPages:Properties 379
- TPageSetupDlg 1139
- TPageSetupDlg:aMargins 1140
- TPageSetupDlg:aMinMargins 1140
- TPageSetupDlg:aPaperSize 1140
- TPageSetupDlg:Events 1145
- TPageSetupDlg:Execute 1145
- TPageSetupDlg:HelpID 1141
- TPageSetupDlg:ICenter 1141
- TPageSetupDlg:IDefMinMargins 1141
- TPageSetupDlg:InInches 1142
- TPageSetupDlg:IMargins 1142
- TPageSetupDlg:IMinMargins 1142
- TPageSetupDlg:INoMargins 1142
- TPageSetupDlg:INoNetwork 1143
- TPageSetupDlg:INoOrientation 1143
- TPageSetupDlg:INoPageDraw 1143
- TPageSetupDlg:INoPaper 1143
- TPageSetupDlg:INoPrinter 1144
- TPageSetupDlg:INoWarning 1144
- TPageSetupDlg:IShowHelp 1144
- TPageSetupDlg:Methods 1145
- TPageSetupDlg:nLastError 1144
- TPageSetupDlg:OnClose 1145
- TPageSetupDlg:OnHelp 1146
- TPageSetupDlg:OnShow 1146
- TPageSetupDlg:oParent 1145
- TPageSetupDlg:Properties 1139
- TPageSetupDlg:Run 1145
- TPanel 383
- TPanel:IXPLook 383
- TPanel:nBevelInner 384
- TPanel:nBevelOuter 384
- TPanel:nHeight 383
- TPanel:nWidth 384
- TPanel:Properties 383
- TPdfSignTool 1582
- TPdfSignTool:cContact 1583
- TPdfSignTool:cError 1584

TPdfSignTool:cInputFile 1584
TPdfSignTool:cLocation 1584
TPdfSignTool:cOutputFile 1584
TPdfSignTool:cPassword 1585
TPdfSignTool:cReason 1585
TPdfSignTool:cTimeStampAccount 1585
TPdfSignTool:cTimeStampPassword 1585
TPdfSignTool:cTimeStampURL 1585
TPdfSignTool:cToolPath 1586
TPdfSignTool:IIMultiSign 1586
TPdfSignTool:IShowProcess 1586
TPdfSignTool:IShowResult 1586
TPdfSignTool:IShowSignature 1587
TPdfSignTool:Methods 1588
TPdfSignTool:nErrorCode 1587
TPdfSignTool:nSignColumns 1587
TPdfSignTool:nSignHeight 1587
TPdfSignTool:nSignHorzOffset 1587
TPdfSignTool:nSignHorzSep 1588
TPdfSignTool:nSignVertOffset 1588
TPdfSignTool:nSignVertSep` 1588
TPdfSignTool:nSignWidth 1588
TPdfSignTool:Properties 1583
TPdfSignTool:Run 1589
TPdfSignTool:cCertFile 1583
TPen 222
TPen:Clone 224
TPen:Create 224
TPen:Destroy 224
TPen:Methods 223
TPen:nColor 223
TPen:New 224
TPen:nStyle 222
TPen:nWidth 223
TPen:oDevice 223
TPen:Properties 222
TPickIconDlg 1147
TPickIconDlg:cFilename 1147
TPickIconDlg:Methods 1148
TPickIconDlg:nCount 1148
TPickIconDlg:nIndex 1148
TPickIconDlg:oParent 1148
TPickIconDlg:Properties 1147
TPickIconDlg:Run 1148
TPicture 224
TPicture:aTypes 225
TPicture:Clone 229
TPicture:cName 226
TPicture:Create 230
TPicture:CreateIndirect 230
TPicture:Destroy 230
TPicture:GetCaptureDateTime 231
TPicture:GetPictureHandle 231
TPicture:GetThumbnailSize 231
TPicture:hPalette 226
TPicture:IsBitmap 233
TPicture:IsEnhMetafile 233
TPicture:IsIcon 233
TPicture:IsMetafile 234
TPicture:IsPicture 234
TPicture:Load 231
TPicture:LoadFromFile 232
TPicture:LoadFromResource 232
TPicture:LoadFromStream 233
TPicture:IsScalable 226
TPicture:IsTransparent 226
TPicture:MakeThumbNail 234
TPicture:Methods 229
TPicture:nHeight 226
TPicture:nImageCount 227
TPicture:nImageIndex 227
TPicture:nImageType 227
TPicture:nLogicalHeight 227
TPicture:nLogicalWidth 228
TPicture:nTimeFrame 228
TPicture:nType 228
TPicture:nWidth 228
TPicture:nZoomH 228
TPicture:nZoomV 229
TPicture:Paint 236
TPicture:Properties 225
TPicture:Save 237
TPicture:SaveToStream 237
TPicture:SetTypes 238
TPop3Client 750
TPop3Client:aMails 751
TPop3Client:cLastMail 751
TPop3Client:Close 754
TPop3Client:Count 754
TPop3Client:cPassword 752

- TPop3Client:cServer 752
- TPop3Client:cUsername 752
- TPop3Client>Delete 755
- TPop3Client:Events 758
- TPop3Client:Functions 758
- TPop3Client:lConnected 753
- TPop3Client:lDelOnRead 753
- TPop3Client:lDigest 753
- TPop3Client:List 755
- TPop3Client:MailFiles 755
- TPop3Client:MailFrom 755
- TPop3Client:MailHtml 756
- TPop3Client:MailSubject 756
- TPop3Client:MailText 756
- TPop3Client:MailTo 756
- TPop3Client:Methods 754
- TPop3Client:New 757
- TPop3Client:nPort 752
- TPop3Client:nRcvBufSize 753
- TPop3Client:nTimeout 753
- TPop3Client:OnNewMail 758
- TPop3Client:OnTimeout 758
- TPop3Client:Open 757
- TPop3Client:Properties 751
- TPop3Client:Retrieve 757
- TPop3Client:RetrieveAll 757
- TPopup:nStyle 293
- TPopupMenu 292
- TPopupMenu:AddItem 295
- TPopupMenu:AddSeparator 295
- TPopupMenu:altems 293
- TPopupMenu:cMessage 293
- TPopupMenu:Create 296
- TPopupMenu:Destroy 296
- TPopupMenu:DoDefault 296
- TPopupMenu:Methods 294
- TPopupMenu:olmageList 293
- TPopupMenu:oParent 294
- TPopupMenu:Properties 292
- TPreviewControl 385
- TPreviewControl:nPage 386
- TPreviewControl:nZoomIndex 386
- TPreviewControl:Properties 385
- TPreviewPage 387
- TPreviewPage:Create 389
- TPreviewPage:Events 389
- TPreviewPage:Methods 388
- TPreviewPage:New 389
- TPreviewPage:nFitInPage 387
- TPreviewPage:nZoomFactor 388
- TPreviewPage:oEmf 388
- TPreviewPage:OnFirstPage 389
- TPreviewPage:OnLastPage 389
- TPreviewPage:OnNextPage 390
- TPreviewPage:OnPreviousPage 390
- TPreviewPage:OnZoomChanged 390
- TPreviewPage:OnZoomIn 390
- TPreviewPage:OnZoomOut 391
- TPreviewPage:oPrinter 388
- TPreviewPage:Properties 387
- TPrintDlg:hDC 1150
- TPrintDlg 1149
- TPrintDlg:Events 1156
- TPrintDlg:Execute 1155
- TPrintDlg:HelpID 1150
- TPrintDlg:lAllPages 1150
- TPrintDlg:lCenter 1151
- TPrintDlg:lCollate 1151
- TPrintDlg:lHidePrintToFile 1151
- TPrintDlg:lNoPageNums 1151
- TPrintDlg:lNoSelection 1152
- TPrintDlg:lNoWarning 1152
- TPrintDlg:lPageNums 1152
- TPrintDlg:lPrintSetup 1152
- TPrintDlg:lPrintToFile 1152
- TPrintDlg:lReturnDC 1153
- TPrintDlg:lReturnIC 1153
- TPrintDlg:lSelection 1153
- TPrintDlg:lShowHelp 1153
- TPrintDlg:Methods 1155
- TPrintDlg:nCopies 1154
- TPrintDlg:nFromPage 1154
- TPrintDlg:nLastError 1154
- TPrintDlg:nMaxPage 1154
- TPrintDlg:nMinPage 1154
- TPrintDlg:nToPage 1155
- TPrintDlg:OnClose 1156
- TPrintDlg:OnHelp 1156
- TPrintDlg:OnShow 1157
- TPrintDlg:oParent 1155

TPrintDlg:Properties 1149
TPrintDlg:Run 1155
TPrinter 123
TPrinter.hDC 125
TPrinter::IPreview 125
TPrinter:AbortDoc 136
TPrinter:aPrinterNames 124
TPrinter:Attributes 136
TPrinter:CanPrintToPdf 136
TPrinter:Comment 136
TPrinter:CountBinSources 137
TPrinter:CountPapers 137
TPrinter:cPort 124
TPrinter:cPrinterName 124
TPrinter:Create 137
TPrinter:Destroy 137
TPrinter:DocumentSetup 138
TPrinter:DriverName 138
TPrinter:EndDoc 138
TPrinter:EndPage 138
TPrinter:Events 147
TPrinter:GetBinSourceNames 139
TPrinter:GetBinSourceTypes 139
TPrinter:GetPaperNames 139
TPrinter:GetPaperSizes 139
TPrinter:GetPaperTypes 140
TPrinter:GetPrinterDefault 140
TPrinter:GetPrinterFonts 140
TPrinter:GetPrinterNames 140
TPrinter:GetResolutions 141
TPrinter:Initialize 137
TPrinter:IsDuplex 141
TPrinter:IsPrinting 141
TPrinter:IAborted 125
TPrinter:ICollate 125
TPrinter:Location 141
TPrinter:IPreviewModal 125
TPrinter:IPrintWithNext 126
TPrinter:Methods 135
TPrinter:nColor 126
TPrinter:nCopies 126
TPrinter:nDuplex 127
TPrinter:nOrientation 127
TPrinter:nPageNumber 127
TPrinter:nPaperLength 128
TPrinter:nPaperSizeType 128
TPrinter:nPaperWidth 130
TPrinter:nPDFPrinter 130
TPrinter:nPreviewShowMode 131
TPrinter:nPrinterIndex 131
TPrinter:nPrinttQuality 131
TPrinter:nStatus 131
TPrinter:oCanvas 132
TPrinter:oExportInfo 132
TPrinter:OnPreview 147
TPrinter:oPageSetupDlg 134
TPrinter:oPreviewDC 134
TPrinter:oPrnDlg 134
TPrinter:oWndOwner 134
TPrinter:PageSetup 142
TPrinter:PaperPhysicalSize 142
TPrinter:PaperRes 142
TPrinter:PaperSize 142
TPrinter:PhysicalOffset 143
TPrinter:Port 143
TPrinter:Preview 143
TPrinter:PreviewControl 143
TPrinter:PrintProcessor 144
TPrinter:Properties 123
TPrinter:Resolution 144
TPrinter:Run 144
TPrinter:SepFile 145
TPrinter:ServerName 145
TPrinter:Setup 145
TPrinter:ShareName 145
TPrinter:StartDoc 146
TPrinter:StartDocPdf 146
TPrinter:StartPage 146
TPrinter:WriteData 147
TPrinter:c:JobTitle 124
TPrinter:n:DefaultBin 126
TProcess 1041
TProcess:cName 1041
TProcess:IsRunning 1042
TProcess:IWait 1041
TProcess:Methods 1042
TProcess:Properties 1041
TProcess:Run 1042
TProcess:Stop 1042
TProgress:Methods 549

- TProgressBar 546
- TProgressBar:Events 550
- TProgressBar:IMarquee 546
- TProgressBar:ISmooth 547
- TProgressBar:nHeight 547
- TProgressBar:nMarqueeSpeed 547
- TProgressBar:nMax 547
- TProgressBar:nMin 548
- TProgressBar:nOrientation 548
- TProgressBar:nStep 548
- TProgressBar:nValue 548
- TProgressBar:nWidth 549
- TProgressBar:OnChange 550
- TProgressBar:Properties 546
- TProgressBar:SetBarColor 549
- TProgressBar:SetBkColor 549
- TProgressBar:StepIt 550
- TProgressBarMod 956
- TProgressBarMod:Events 959
- TProgressBarMod:IMarquee 957
- TProgressBarMod:nHeight 957
- TProgressBarMod:nMax 958
- TProgressBarMod:nMin 958
- TProgressBarMod:nOrientation 958
- TProgressBarMod:nStep 959
- TProgressBarMod:nValue 959
- TProgressBarMod:nWidth 959
- TProgressBarMod:OnChange 959
- TProgressBarMod:Properties 957
- TProgressBarMod:nMarqueeSpeed 958
- TProgressCircle 960
- TProgressCircle:Events 963
- TProgressCircle:IMarquee 961
- TProgressCircle:IShowPercent 961
- TProgressCircle:nCircleGap 961
- TProgressCircle:nCircleSize 961
- TProgressCircle:nClrCircle 962
- TProgressCircle:nHeight 962
- TProgressCircle:nMarqueeSpeed 962
- TProgressCircle:nMax 962
- TProgressCircle:nMin 963
- TProgressCircle:nValue 963
- TProgressCircle:nWidth 963
- TProgressCircle:OnChange 963
- TProgressCircle:Properties 960
- TProgressDlg 1157
- TProgressDlg:cMessage 1158
- TProgressDlg:cMessageCancel 1158
- TProgressDlg:Create 1162
- TProgressDlg:cText 1158
- TProgressDlg:cTitle 1158
- TProgressDlg:Destroy 1162
- TProgressDlg:Events 1163
- TProgressDlg:Execute 1162
- TProgressDlg:IsCancel 1162
- TProgressDlg:IAutoTime 1159
- TProgressDlg:ICenter 1159
- TProgressDlg:ICompactPath 1159
- TProgressDlg:IModal 1159
- TProgressDlg:INoCancel 1159
- TProgressDlg:INoMinimize 1160
- TProgressDlg:INoProgress 1160
- TProgressDlg:INoTime 1160
- TProgressDlg:Methods 1161
- TProgressDlg:nCommonAVI 1160
- TProgressDlg:New 1163
- TProgressDlg:nMax 1161
- TProgressDlg:nResourceAVI 1161
- TProgressDlg:nValue 1161
- TProgressDlg:OnCancel 1164
- TProgressDlg:OnClose 1164
- TProgressDlg:OnReset 1164
- TProgressDlg:OnShow 1164
- TProgressDlg:Properties 1157
- TProgressDlg:Reset 1163
- TProgressDlg:Run 1162
- TProgressDlg:Stop 1163
- TRadio 302
- TRadio:Click 305
- TRadio:Events 305
- TRadio:IChecked 303
- TRadio:IMultiLine 303
- TRadio:IPushLike 303
- TRadio:ITransparent 303
- TRadio:Methods 305
- TRadio:nAlignment 304
- TRadio:nHeight 304
- TRadio:nVAlignment 304
- TRadio:nWidth 304
- TRadio:OnChange 305

TRadio:OnClick 305
TRadio:Properties 302
TRadioMenu 391
TRadioMenu:alters 392
TRadioMenu:Events 394
TRadioMenu:ITabStop 392
TRadioMenu:ITransparent 392
TRadioMenu:nBorderStyle 392
TRadioMenu:nClrPane 392
TRadioMenu:nColumns 393
TRadioMenu:nHeight 393
TRadioMenu:nIndex 393
TRadioMenu:nMargin 393
TRadioMenu:nOrientation 394
TRadioMenu:nSpacing 394
TRadioMenu:nWidth 394
TRadioMenu:OnChange 394
TRadioMenu:OnClick 395
TRadioMenu:Properties 391
TRadioMod 964
TRadioMod:cGroup 964
TRadioMod:Events 966
TRadioMod:IChecked 964
TRadioMod:ITransparent 965
TRadioMod:nClrColorization 965
TRadioMod:nClrTextDisabled 965
TRadioMod:nLineSpacing 965
TRadioMod:nVAlignment 966
TRadioMod:OnChange 966
TRadioMod:OnClick 966
TRadioMod:Properties 964
TRebar 550
TRebar:Additem 554
TRebar:alters 551
TRebar:IBandBorders 551
TRebar:IBorder 551
TRebar:IDbIClkToggle 552
TRebar:IFixedOrder 552
TRebar:IVarHeight 553
TRebar:Methods 554
TRebar:nAlign 552
TRebar:nWidth 553
TRebar:olmageList 553
TRebar:Properties 551
TRebarBand 555
TRebarBand:Create 558
TRebarBand:cText 555
TRebarBand>Delete 559
TRebarBand:IBreak 556
TRebarBand:IChildEdge 556
TRebarBand:IFixedBmp 556
TRebarBand:IFixedSize 556
TRebarBand:IGripper 556
TRebarBand:IVisible 557
TRebarBand:Methods 558
TRebarBand:nImage 557
TRebarBand:nIndex 557
TRebarBand:oControl 558
TRebarBand:Properties 555
TRecentList 1437
TRecentList:Add 1439
TRecentList:alters 1437
TRecentList:clniFile 1437
TRecentList:clniSection 1438
TRecentList>Delete 1439
TRecentList:Deleteltems 1439
TRecentList:IAutoRemove 1438
TRecentList:IAutoSave 1438
TRecentList:Methods 1438
TRecentList:nMaxItems 1438
TRecentList:Properties 1437
TRecentList:Read 1440
TRecentList:Refresh 1440
TRecentList:SetIniFile 1440
TRecentList:SetItems 1440
TRecentList:Write 1441
TRecentListMenu 1441
TRecentListMenu:cMessage 1442
TRecentListMenu:Create 1444
TRecentListMenu:cTextEmpty 1443
TRecentListMenu:Events 1445
TRecentListMenu:IFullPath 1442
TRecentListMenu:INumerate 1442
TRecentListMenu:IPrefix 1442
TRecentListMenu:ISeparator 1442
TRecentListMenu:ISubMenu 1443
TRecentListMenu:IText 1443
TRecentListMenu:Methods 1444
TRecentListMenu:nMaxWidth 1443
TRecentListMenu:nMenu 1444

TRecentListMenu:nMenuPos 1444
TRecentListMenu:oMenu 1444
TRecentListMenu:OnClick 1445
TRecentListMenu:Properties 1441
TRecentListMenu:Refresh 1445
TRegistry 1043
TRegistry:CloseKey 1045
TRegistry:Connect 1045
TRegistry:CreateKey 1045
TRegistry>DeleteKey 1045
TRegistry>DeleteValue 1046
TRegistry:Destroy 1046
TRegistry:Disconnect 1046
TRegistry:EnumKeys 1046
TRegistry:EnumValues 1047
TRegistry:Flush 1047
TRegistry:GetBinary 1047
TRegistry:GetDate 1048
TRegistry:GetLogical 1048
TRegistry:GetNumeric 1048
TRegistry:GetString 1048
TRegistry:hCurrentKey 1043
TRegistry:Methods 1044
TRegistry:nLastError 1044
TRegistry:OpenKey 1049
TRegistry:Properties 1043
TRegistry:Root 1043
TRegistry:SetDate 1049
TRegistry:SetLogical 1049
TRegistry:SetNumeric 1050
TRegistry:SetString 1050
TRegistry:SetValue 1050
TReplaceDlg 1165
TReplaceDlg:cReplace 1165
TReplaceDlg:cText 1166
TReplaceDlg:Events 1168
TReplaceDlg:Execute 1167
TReplaceDlg:HelpID 1166
TReplaceDlg:ICenter 1166
TReplaceDlg:IMatchCase 1166
TReplaceDlg:INoCase 1166
TReplaceDlg:INoWholeWord 1167
TReplaceDlg:IShowHelp 1167
TReplaceDlg:IWholeWord 1167
TReplaceDlg:Methods 1167
TReplaceDlg:OnClose 1168
TReplaceDlg:OnFind 1169
TReplaceDlg:OnHelp 1168
TReplaceDlg:OnReplace 1169
TReplaceDlg:OnReplaceAll 1169
TReplaceDlg:Properties 1165
TReplaceDlg:Run 1167
TReport 1471
TReport:aColumns 1472
TReport:AddColumn 1481
TReport:AddGroup 1482
TReport:Adjust 1482
TReport:aFonts 1472
TReport:aGroups 1473
TReport:AreGroups 1482
TReport:BackLine 1482
TReport:bFor 1473
TReport:bSkip 1473
TReport:bWhile 1473
TReport:cGrandTotal 1473
TReport:cJobName 1474
TReport:cPageTotal 1474
TReport:cPdfFilename 1474
TReport:cXLSDData 1474
TReport:DelColumn 1482
TReport:DelGroup 1483
TReport:EndLine 1483
TReport:Events 1486
TReport:IAutoLand 1475
TReport:ICancel 1475
TReport:LinesLeft 1483
TReport:IPageTotal 1475
TReport:IPreview 1475
TReport:IPreviewModal 1475
TReport:IPrintWithNext 1476
TReport:ISummary 1476
TReport:ITotal 1476
TReport:IXLSNumeric 1476
TReport:Methods 1481
TReport:nBottomMargin 1476
TReport:nClrLined 1477
TReport:nCounter 1477
TReport:NeedNewPage 1484
TReport:NewLine 1484
TReport:NewPage 1484

TReport:nGridStyle 1477
TReport:nGroupLine 1477
TReport:nLeftMargin 1477
TReport:nLineHeight 1478
TReport:nPage 1478
TReport:nRow 1478
TReport:nSeparator 1478
TReport:nTitleDnLine 1478
TReport:nTitleUpLine 1479
TReport:nTopMargin 1479
TReport:nTotalLine 1479
TReport:oDataset 1479
TReport:oDevice 1480
TReport:oFooter 1480
TReport:oHeader 1480
TReport:OnChange 1486
TReport:OnChanged 1486
TReport:OnEnd 1486
TReport:OnEndGroup 1487
TReport:OnEndLine 1487
TReport:OnEndPage 1487
TReport:OnPostEnd 1488
TReport:OnPostGroup 1488
TReport:OnPostPage 1488
TReport:OnPreChange 1488
TReport:OnStartGroup 1488
TReport:OnStartLine 1489
TReport:OnStartPage 1489
TReport:oParent 1480
TReport:oPen 1480
TReport:oTitle 1481
TReport:PreviewNow 1485
TReport:PrintAtCol 1484
TReport:Properties 1471
TReport:Run 1485
TReport:StartLine 1485
TReportToExcel 1485
TRichEdit 559
TRichEdit:aMargins 561
TRichEdit:CanClear 578
TRichEdit:CanCopy 579
TRichEdit:CanCut 579
TRichEdit:CanPaste 579
TRichEdit:CanRedo 579
TRichEdit:CanUndo 580
TRichEdit:cFileName 561
TRichEdit:cFindText 561
TRichEdit:cFontName 562
TRichEdit:CharFromPos 580
TRichEdit:Clear 580
TRichEdit:ColorDlg 580
TRichEdit:Copy 581
TRichEdit:cReplace 562
TRichEdit:cText 562
TRichEdit:Cut 581
TRichEdit:EmptyUndoBuffer 581
TRichEdit:Events 607
TRichEdit:FindDlg 581
TRichEdit:FindText 582
TRichEdit:FontDlg 582
TRichEdit:GetBullet 582
TRichEdit:GetCol 582
TRichEdit:GetEventMask 583
TRichEdit:GetFirstVisibleLine 583
TRichEdit:GetIndent 584
TRichEdit:GetLine 583
TRichEdit:GetLineCount 583
TRichEdit:GetLineFromChar 584
TRichEdit:GetLineIndex 584
TRichEdit:GetModify 584
TRichEdit:GetNumbering 585
TRichEdit:GetOptions 585
TRichEdit:GetParagraphSpacing 585
TRichEdit:GetPos 585
TRichEdit:GetPrintHeight 586
TRichEdit:GetRedoName 586
TRichEdit:GetRow 586
TRichEdit:GetRTF 586
TRichEdit:GetSel 587
TRichEdit:GetSelection 587
TRichEdit:GetSelectionType 588
TRichEdit:GetText 588
TRichEdit:GetTextMode 588
TRichEdit:GetTextRange 588
TRichEdit:GetTypographicOptions 589
TRichEdit:GetUndoName 589
TRichEdit:GetZoom 589
TRichEdit:GotoLine 589
TRichEdit:HasText 590
TRichEdit:HTMLToRTF 590

TRichEdit:InsertBitmap 590
TRichEdit:InsertFile 591
TRichEdit:InsertObject 591
TRichEdit:InsertObjectFromFile 591
TRichEdit:InsertPageBreak 591
TRichEdit:InsertPicture 592
TRichEdit:InsertRTF 592
TRichEdit:InsertTable 592
TRichEdit:InsertText 593
TRichEdit:IsRTF 593
TRichEdit:IsSelection 593
TRichEdit:IBold 562
TRichEdit:IBorder 563
TRichEdit:IDisableNoScroll 563
TRichEdit:Len 593
TRichEdit:IFindCase 563
TRichEdit:IFindDown 563
TRichEdit:IFindWord 564
TRichEdit:IHot 564
TRichEdit:IHScroll 564
TRichEdit:IHTML 564
TRichEdit:Italic 564
TRichEdit:LoadFile 594
TRichEdit:LoadRTF 594
TRichEdit:IPDF 565
TRichEdit:IProtected 565
TRichEdit:IRE30 565
TRichEdit:IRE40 565
TRichEdit:IRE50 566
TRichEdit:IRE60 566
TRichEdit:IRE70 566
TRichEdit:IRE80 566
TRichEdit:IReadOnly 567
TRichEdit:IRule 567
TRichEdit:IShowMessages 567
TRichEdit:ISmallCaps 567
TRichEdit:IStrikeOut 568
TRichEdit:ISubScript 568
TRichEdit:ISupScript 568
TRichEdit:ITransparent 568
TRichEdit:IUnderline 569
TRichEdit:IURLDetect 569
TRichEdit:IVScroll 569
TRichEdit:IWord 570
TRichEdit:IWordEx 570
TRichEdit:IWordSel 571
TRichEdit:IWordWrap 571
TRichEdit:IWordX 570
TRichEdit:IWrite 570
TRichEdit:Methods 576
TRichEdit:MovePos 594
TRichEdit:nAlignment 571
TRichEdit:nClrPane 571
TRichEdit:nClrText 572
TRichEdit:nConverter 572
TRichEdit:nFontColor 572
TRichEdit:nFontSize 572
TRichEdit:nFormat 573
TRichEdit:nHighLight 573
TRichEdit:nLineSpacing 573
TRichEdit:nMaxLength 574
TRichEdit:nMaxUndo 574
TRichEdit:nNumbering 574
TRichEdit:nNumStyle 574
TRichEdit:nPageWidth 575
TRichEdit:nUlineColor 575
TRichEdit:nUlineType 575
TRichEdit:nUnits 575
TRichEdit:nWidth 576
TRichEdit:OnChange 607
TRichEdit:OnContextMenu 607
TRichEdit:OnHScroll 608
TRichEdit:OnInsertPicture 608
TRichEdit:OnPrinting 608
TRichEdit:OnRequestResize 608
TRichEdit:oPopup 576
TRichEdit:PageDlg 595
TRichEdit:Paste 595
TRichEdit:PasteSpecial 595
TRichEdit:PictureDlg 595
TRichEdit:PosFromChar 596
TRichEdit:Preview 596
TRichEdit:Print 596
TRichEdit:PrintBox 597
TRichEdit:PrintDlg 597
TRichEdit:PrintSetup 597
TRichEdit:Properties 559
TRichEdit:Redo 598
TRichEdit:Replace 598
TRichEdit:ReplaceAll 598

TRichEdit:ReplaceDlg 599
TRichEdit:RTFToHTML 599
TRichEdit:RTFToPDF 599
TRichEdit:RTFToWord 599
TRichEdit:RTFToWordX 600
TRichEdit:RTFToWrite 600
TRichEdit:SaveFile 600
TRichEdit:SelectAll 600
TRichEdit:SelectNone 601
TRichEdit:SetBullet 601
TRichEdit:SetEventMask 601
TRichEdit:SetHideSelection 601
TRichEdit:SetHyperLinkTooltips 602
TRichEdit:SetIndent 602
TRichEdit:SetModify 602
TRichEdit:SetNumbering 603
TRichEdit:SetOptions 603
TRichEdit:SetParagraphSpacing 603
TRichEdit:SetPos 604
TRichEdit:SetSel 604
TRichEdit:SetText 604
TRichEdit:SetTextMode 604
TRichEdit:SetTypographyOptions 605
TRichEdit:SetZoom 605
TRichEdit:StopGroupTyping 605
TRichEdit:Undo 606
TRichEdit:WordToRtf 606
TRichEdit:WordXToRtf 606
TRichEdit:WriteToRtf 606
TRichEditGetRTFRange 587
TRptColumn:aData 1490
TRptColumn:aPicture 1490
TRptColumn:aTitle 1490
TRptColumn:cTotalPict 1491
TRptColumn:Events 1494
TRptColumn:ITotal 1491
TRptColumn:nAlignment 1491
TRptColumn:nCharSize 1491
TRptColumn:nClrPane 1492
TRptColumn:nDataFont 1492
TRptColumn:nTitleAlignment 1492
TRptColumn:nTitleClrPane 1493
TRptColumn:nTitleClrText 1493
TRptColumn:nTitleFont 1493
TRptColumn:nTitleHeight 1493
TRptColumn:nTotal 1493
TRptColumn:nTotalFont 1494
TRptColumn:nWidth 1494
TRptColumn:OnAddToTotal 1494
TRptColumn:OnPrintData 1495
TRptColumn:OnPrintTitle 1495
TRptColumn:OnPrintTotal 1496
TRptColumn:oReport 1494
TRptColumn:Properties 1489
TRptGroup:aTotal 1500
TRptGroup:bFooter 1500
TRptGroup:bGroup 1500
TRptGroup:bHeader 1500
TRptGroup:cOldValue 1500
TRptGroup:cValue 1501
TRptGroup:Events 1503
TRptGroup:IEject 1501
TRptGroup:nClrText 1501
TRptGroup:nCounter 1501
TRptGroup:nFont 1502
TRptGroup:nFooterHeight 1502
TRptGroup:nHeaderHeight 1502
TRptGroup:nOrder 1502
TRptGroup:OnPrintFooter 1503
TRptGroup:OnPrintHeader 1503
TRptGroup:OnPrintTotal 1503
TRptGroup:oReport 1502
TRptGroup:Properties 1499
TRptLine:aAlignment 1496
TRptLine:aFonts 1497
TRptLine:aLine 1497
TRptLine:Events 1498
TRptLine:nColor 1498
TRptLine:nHeight 1498
TRptLine:nWidth 1498
TRptLine:OnPrintLine 1498
TRptLine:oReport 1498
TRptLine:Properties 1496
TRunFileDialog 1170
TRunFileDialog:cDirectory 1171
TRunFileDialog:cMessage 1171
TRunFileDialog:cName 156
TRunFileDialog:cTitle 1171
TRunFileDialog:Events 1173
TRunFileDialog:Execute 1173

- TRunFileDialog:I CalcDirectory 1171
- TRunFileDialog:I NoBrowse 1171
- TRunFileDialog:I NoDefault 1172
- TRunFileDialog:I NoLabel 1172
- TRunFileDialog:I NoSeparateMem 1172
- TRunFileDialog:Methods 1173
- TRunFileDialog:olcon 1172
- TRunFileDialog:OnExecute 1173
- TRunFileDialog:Properties 1170
- TRunFileDialog:Run 1173
- TRunProcess 1051
- TRunProcess:c CommandLine 1051
- TRunProcess:c ErrorMessage 1051
- TRunProcess:c WorkingDir 1052
- TRunProcess:End 1053
- TRunProcess:Eventos 1054
- TRunProcess:h Process 1052
- TRunProcess:h Thread 1052
- TRunProcess:IsRunning 1054
- TRunProcess:Kill 1054
- TRunProcess:Methods 1053
- TRunProcess:n ShowMode 1052
- TRunProcess:OnEnd 1054
- TRunProcess:OnRead 1055
- TRunProcess:OnStart 1055
- TRunProcess:Properties 1051
- TRunProcess:Run 1053
- TRunProcess:Write 1054
- TScreen 147
- TScreen:a Monitors 148
- TScreen:CreateDC 153
- TScreen>DeleteDC 153
- TScreen:Handle 149
- TScreen:hDC 149
- TScreen:Initialize 153
- TScreen:ITouch 149
- TScreen:Methods 153
- TScreen:n BitsPerPixel 149
- TScreen:n CaptionHeight 150
- TScreen:n ClientHeight 150
- TScreen:n ClientLeft 150
- TScreen:n ClientTop 150
- TScreen:n ClientWidth 151
- TScreen:n ColorPlanes 151
- TScreen:n Height 151
- TScreen:n Monitors 151
- TScreen:n PixelsPerInch 151
- TScreen:n Width 152
- TScreen:o CursorArrow 152
- TScreen:o CursorDropNo 152
- TScreen:o CursorDropYes 152
- TScreen:PaperPhysicalSize 154
- TScreen:PaperRes 154
- TScreen:PaperSize 154
- TScreen:Properties 148
- TScreen:Resolution 154
- TScreenKeyboard 967
- TScreenKeyboard:a CClrKeys 967
- TScreenKeyboard:a Keys 968
- TScreenKeyboard:Events 970
- TScreenKeyboard:I HotTrack 968
- TScreenKeyboard:I Transparent 968
- TScreenKeyboard:n BorderStyle 968
- TScreenKeyboard:n CClrKeys 969
- TScreenKeyboard:n Height 969
- TScreenKeyboard:n Layout 969
- TScreenKeyboard:n Style 969
- TScreenKeyboard:n Width 970
- TScreenKeyboard:o FontShortCut 970
- TScreenKeyboard:OnKeyClick 970
- TScreenKeyboard:Properties 967
- TScrollBar 306
- TScrollBar:Dec 308
- TScrollBar:Events 309
- TScrollBar:Inc 309
- TScrollBar:IDisableNoScroll 306
- TScrollBar:Methods 308
- TScrollBar:n CClrPane 306
- TScrollBar:n Max 307
- TScrollBar:n Min 307
- TScrollBar:n Orientation 307
- TScrollBar:n Page 308
- TScrollBar:n Pos 308
- TScrollBar:n Width 308
- TScrollBar:OnChange 309
- TScrollBar:OnDown 309
- TScrollBar:OnPageDown 310
- TScrollBar:OnPageUp 310
- TScrollBar:OnThumbPos 310
- TScrollBar:OnThumbTrack 311

TScrollBar:OnUp 311
TScrollBar:Properties 306
TScrollBar 399
TScrollBar:ITransparent 399
TScrollBar:nBorderStyle 400
TScrollBar:nHeight 400
TScrollBar:nWidth 400
TScrollBar:Properties 399
TScrollingWinControl 81
TScrollingWinControl:Events 84
TScrollingWinControl:IAutoScroll 82
TScrollingWinControl:IHideScrollBars 82
TScrollingWinControl:nClientLeft 83
TScrollingWinControl:nClientTop 83
TScrollingWinControl:nScrollIncrement 82
TScrollingWinControl:nScrollShowDelay 82
TScrollingWinControl:nVirtualHeight 83
TScrollingWinControl:nVirtualWidth 83
TScrollingWinControl:OnScrollDown 84
TScrollingWinControl:OnScrollLeft 84
TScrollingWinControl:OnScrollPageDown 85
TScrollingWinControl:OnScrollPageLeft 85
TScrollingWinControl:OnScrollPageRight 85
TScrollingWinControl:OnScrollPageUp 86
TScrollingWinControl:OnScrollRight 86
TScrollingWinControl:OnScrollUp 86
TScrollingWinControl:OnThumbPosH 87
TScrollingWinControl:OnThumbPosV 87
TScrollingWinControl:OnThumbTrackH 87
TScrollingWinControl:OnThumbTrackV 87
TScrollingWinControl:Properties 81
TSearchComboBox 400
TSearchComboBox:IExact 401
TSearchComboBox:INoCase 401
TSearchComboBox:Properties 400
TShape 971
TShape:IFillShape 971
TShape:nHeight 972
TShape:nLineWidth 972
TShape:nOpacity 972
TShape:nRadius 973
TShape:nShapeType 973
TShape:nVertices 973
TShape:nWidth 973
TShape:Properties 971
TSharedModule 1055
TSharedModule:cClassname 1056
TSharedModule:Events 1056
TSharedModule:OnCreate 1056
TSharedModule:Properties 1056
TShiftPanel 395
TShiftPanel:Events 398
TShiftPanel:IOpened 396
TShiftPanel:nCaptionSize 396
TShiftPanel:nClrTitleEnd 396
TShiftPanel:nClrTitleStart 396
TShiftPanel:nClrTitleText 397
TShiftPanel:nIconSize 397
TShiftPanel:nOrientation 397
TShiftPanel:nVAlignment 397
TShiftPanel:oFontTitle 398
TShiftPanel:OnClose 398
TShiftPanel:OnClosed 398
TShiftPanel:OnOpen 398
TShiftPanel:OnOpened 399
TShiftPanel:Properties 395
TSplitButton 401
TSplitButton:IMenuOnRight 401
TSplitButton:oMenu 402
TSplitButton:Properties 401
TSplitter 402
TSplitter:Events 405
TSplitter:IGripper 403
TSplitter:nBorderStyle 403
TSplitter:nClrPane 403
TSplitter:nHeight 403
TSplitter:nMaxPos 404
TSplitter:nMinPos 404
TSplitter:nMinSize 404
TSplitter:nWidth 404
TSplitter:OnMove 405
TSplitter:Properties 402
TSQLiteDataSource 1237
TSQLiteDataSource:BeginTrans 1242
TSQLiteDataSource:cConnect 1239
TSQLiteDataSource:CommitTrans 1242
TSQLiteDataSource:Connect 1242
TSQLiteDataSource:cPassword 1239
TSQLiteDataSource:DelTable 1243
TSQLiteDataSource:Disconnect 1243

- TSQliteDataSource:Encrypt 1243
- TSQliteDataSource:Execute 1243
- TSQliteDataSource:File 1244
- TSQliteDataSource:GetTables 1244
- TSQliteDataSource:IDateAsString 1239
- TSQliteDataSource:IDoubleQuotes 1241
- TSQliteDataSource:IReadToCache 1241
- TSQliteDataSource:Methods 1241
- TSQliteDataSource:nAffectedRows 1241
- TSQliteDataSource:Properties 1239
- TSQliteDataSource:Query 1244
- TSQliteDataSource:QueryArray 1245
- TSQliteDataSource:QueryArrayHash 1245
- TSQliteDataSource:QueryReport 1245
- TSQliteDataSource:QueryRow 1246
- TSQliteDataSource:QueryRowHash 1246
- TSQliteDataSource:QueryValue 1246
- TSQliteDataSource:RollBackTrans 1247
- TSQliteDataSource:Table 1247
- TSqlParam 1360
- TSqlParam:cName 1360
- TSqlParam:Create 1363
- TSqlParam:Initial 1361
- TSqlParam:IsInputParam 1363
- TSqlParam:IsOutputParam 1363
- TSqlParam:nDirection 1361
- TSqlParam:nSize 1361
- TSqlParam:nType 1361
- TSqlParam:oSqlProc 1362
- TSqlParam:Properties 1360
- TSqlParam:Value 1362
- TSqlParams:Methods 1363
- TSqlProc 1355
- TSqlProc:AddParam 1357
- TSqlProc:aResults 1355
- TSqlProc:aSqlParams 1355
- TSqlProc:Close 1357
- TSqlProc:cName 1356
- TSqlProc:Create 1358
- TSqlProc:Events 1359
- TSqlProc:IDisplayErrors 1356
- TSqlProc:IOpen 1356
- TSqlProc:oDataSource 1356
- TSqlProc:OnClose 1359
- TSqlProc:OnCreate 1359
- TSqlProc:OnOpen 1359
- TSqlProc:OnPostClose 1360
- TSqlProc:OnPostOpen 1360
- TSqlProc:Open 1358
- TSqlProc:OutputParams 1358
- TSqlProc:ParamByName 1358
- TSqlProc:Properties 1355
- TSqlProc:ReturnValue 1359
- TSqlProc:Run 1358
- TSqlProcs 1364
- TSqlProct:Methods 1357
- TSQLQuery 1365
- TSQLQuery:aSQLParams 1366
- TSQLQuery:cSelect 1366
- TSQLQuery:Properties 1366
- TSQLTable 1366
- TSQLTable:cTableName 1367
- TSQLTable:Properties 1367
- TStatusBar 609
- TStatusBar:AddBitmap 612
- TStatusBar:AddIcon 613
- TStatusBar:AddItem 613
- TStatusBar:altems 609
- TStatusBar:cText 609
- TStatusBar:DeleteItem 614
- TStatusBar:DeleteItems 614
- TStatusBar:Events 614
- TStatusBar:IFullSize 610
- TStatusBar:ISimple 610
- TStatusBar:ISizeGrip 610
- TStatusBar:Methods 612
- TStatusBar:nAlign 610
- TStatusBar:nBorderStyle 611
- TStatusBar:nClrPane 611
- TStatusBar:nHeight 611
- TStatusBar:nMinHeight 611
- TStatusBar:olcon 612
- TStatusBar:olmImageList 612
- TStatusBar:OnClick 614
- TStatusBar:OnDrawItem 615
- TStatusBar:OnRClick 615
- TStatusBar:OnSimpleMode 615
- TStatusBar:Properties 609
- TStatusPanel 615
- TStatusPanel:Create 619

TStatusPanel:cText 616
TStatusPanel:cToolTip 616
TStatusPanel:Destroy 620
TStatusPanel:End 620
TStatusPanel:Events 621
TStatusPanel:IOwnerDraw 617
TStatusPanel:Methods 619
TStatusPanel:nAlignment 617
TStatusPanel:nBorderStyle 617
TStatusPanel:New 620
TStatusPanel:nImage 617
TStatusPanel:nMinWidth 618
TStatusPanel:nResizeStyle 618
TStatusPanel:nType 618
TStatusPanel:nWidth 618
TStatusPanel:oControl 619
TStatusPanel:OnClick 621
TStatusPanel:OnRClick 621
TStatusPanel:Properties 616
TStatusPanel:Update 620
TStdControl 71
TStdControl:aDropFiles 71
TStdControl:cMessage 71
TStdControl:Eventos 73
TStdControl:IBorder 72
TStdControl:IDragAcceptFiles 72
TStdControl:ITabStop 72
TStdControl:Metodos 72
TStdControl:OnChar 73
TStdControl:OnCheckState 74
TStdControl:OnDropFiles 75
TStdControl:OnEnter 74
TStdControl:OnExit 74
TStdControl:OnKeyDown 75
TStdControl:OnKeyUp 76
TStdControl:Properties 71
TStdControl:RequestState 73
TSumatraPrinter 1576
TSumatraPrinter:cBin 1577
TSumatraPrinter:cExeFile 1577
TSumatraPrinter:cPrinter 1577
TSumatraPrinter:cRange 1578
TSumatraPrinter:IExitWhenDone 1578
TSumatraPrinter:IShowDialog 1578
TSumatraPrinter:ISilent 1578
TSumatraPrinter:nColor 1579
TSumatraPrinter:nCopies 1579
TSumatraPrinter:nDuplex 1579
TSumatraPrinter:nPageSize 1580
TSumatraPrinter:nPageType 1580
TSumatraPrinter:nScale 1580
TSumatraPrinter:Print 1581
TSumatraPrinter:Properties 1576
TSumatraPrinterr:Metodos 1581
TSumatraViewer 1573
TSumatraViewer:cExeFile 1573
TSumatraViewer:cExePath 1573,1577
TSumatraViewer:cFileName 1574
TSumatraViewer:cLastError 1574
TSumatraViewer:FileOpen 1576
TSumatraViewer:IHideToolBar 1574
TSumatraViewer:IRestrictMode 1574
TSumatraViewer:Methods 1576
TSumatraViewer:nInitialPage 1575
TSumatraViewer:nInitialView 1575
TSumatraViewer:nInitialZoom 1575
TSumatraViewer:Properties 1573
TSummerNote 974
TSummerNote:cHtml 974
TSummerNote:cLanguage 974
TSummerNote:cToolBar 975
TSummerNote:GetHtml 975
TSummerNote:Methods 975
TSummerNote:Properties 974
TSummerNote:SetHtml 976
TSwitch 976
TSwitch:Click 978
TSwitch:Events 978
TSwitch:IChecked 977
TSwitch:ITransparent 977
TSwitch:Methods 978
TSwitch:nClrColorization 977
TSwitch:nHeight 977
TSwitch:nWidth 977
TSwitch:OnChange 978
TSwitch:OnClick 979
TSwitch:Properties 976
TSwitch:Toggle 978
TSysAnimate 622
TSysAnimate:Close 625

- TSysAnimate:cName 623
- TSysAnimate:Create 625
- TSysAnimate:Destroy 626
- TSysAnimate:Events 628
- TSysAnimate:IActive 623
- TSysAnimate:IAutoPlay 623
- TSysAnimate:ICentered 623
- TSysAnimate:Load 626
- TSysAnimate:LoadCommon 626
- TSysAnimate:LoadFromFile 626
- TSysAnimate:LoadFromResource 627
- TSysAnimate:ITransparent 624
- TSysAnimate:IUseTimer 624
- TSysAnimate:Methods 625
- TSysAnimate:nCommonAvi 622
- TSysAnimate:nFirstFrame 624
- TSysAnimate:nLastFrame 624
- TSysAnimate:nRepeat 624
- TSysAnimate:OnClose 628
- TSysAnimate:OnOpen 628
- TSysAnimate:OnStart 628
- TSysAnimate:OnStop 628
- TSysAnimate:Play 627
- TSysAnimate:Properties 622
- TSysAnimate:Seek 627
- TSysAnimate:Stop 627
- TSysImageList 1057
- TSysImageList:Create 1059
- TSysImageList:GetBitmap 1059
- TSysImageList:GetIcon 1059
- TSysImageList:GetIconIndex 1059
- TSysImageList:GetIconName 1060
- TSysImageList:GetIconType 1060
- TSysImageList:ILargeIcons 1057
- TSysImageList:IOverlay 1058
- TSysImageList:ISelected 1058
- TSysImageList:IShellSize 1058
- TSysImageList:IUseFileAttributes 1057
- TSysImageList:Methods 1058
- TSysImageList:Properties 1057
- TSysVer 238, 1702
- TSysVer:cCSDVersion 239
- TSysVer:Create 247
- TSysVer:cSysName 239
- TSysVer:cSysVer 239
- TSysVer:cWinNTProduct 240
- TSysVer:llsWin10 240
- TSysVer:llsWin2000 241
- TSysVer:llsWin7 240
- TSysVer:llsWin98 240
- TSysVer:llsWinHomeServer 242
- TSysVer:llsWinMe 241
- TSysVer:llsWinServer2003 241
- TSysVer:llsWinServer2016 242
- TSysVer:llsWinServer2019 242
- TSysVer:llsWinVista 242
- TSysVer:llsWinXP 241
- TSysVer:llsWinXP64 241
- TSysVer:llsWow64 243
- TSysVer:Methods 247
- TSysVer:nBuildNumber 243
- TSysVer:nMajorVersion 243
- TSysVer:nMemoryRAM 244
- TSysVer:nMinorVersion 244
- TSysVer:nPlatformID 244
- TSysVer:nProcessorArch 245
- TSysVer:nProductType 245
- TSysVer:nReleaseID 245
- TSysVer:nServicePackMajor 246
- TSysVer:nServicePackMinor 246
- TSysVer:nSuiteMask 246
- TSysVer:Properties 238
- TTabCtrl 629
- TTabCtrl:AddImage 634
- TTabCtrl:AddItem 635
- TTabCtrl:altems 630
- TTabCtrl:DeleteItem 635
- TTabCtrl:DeleteItems 635
- TTabCtrl:Events 637
- TTabCtrl:InsertItem 636
- TTabCtrl:IButtonStyle 630
- TTabCtrl:IFixedWidth 630
- TTabCtrl:IFlatButtons 630
- TTabCtrl:IForceIconLeft 631
- TTabCtrl:IForceLabelLeft 631
- TTabCtrl:IHotTrack 631
- TTabCtrl:IMultiLine 631
- TTabCtrl:IRightJustify 631
- TTabCtrl:IScrollOpposite 632
- TTabCtrl:ITabCloseButtons 632

TTabCtrl:ITabStop 632
TTabCtrl:ITransparent 632
TTabCtrl:ITransparentBody 633
TTabCtrl:Methods 634
TTabCtrl:nHeight 633
TTabCtrl:nIndex 633
TTabCtrl:nMinTabWidth 633
TTabCtrl:nOrientation 633
TTabCtrl:nWidth 634
TTabCtrl:olmageList 634
TTabCtrl:OnChange 637
TTabCtrl:OnChanging 637
TTabCtrl:OnCloseTab 637
TTabCtrl:OnRClick 638
TTabCtrl:Properties 629
TTabCtrl:SetIndex 636
TTabFolderPage:Create 1005
TTabItem 638
TTabItem:Create 640, 999
TTabItem:cText 639
TTabItem:cToolTip 639
TTabItem>Delete 641, 901, 999
TTabItem:IsSelected 901
TTabItem:isEnabled 639
TTabItem:Methods 640, 998
TTabItem:nCrLfText 639
TTabItem:nCrLfTextHot 639
TTabItem:nImage 640
TTabItem:nItem 640
TTabItem:Properties 638
TTabItem:Select 641, 999
TTabItem:SetItem 641
TTabs 982
TTabs:AddItem 990
TTabs:altItems 983
TTabs:DeleteItem 990
TTabs:DeleteItems 991
TTabs:Events 992
TTabs:InsertItem 991
TTabs:IFixedWidth 983
TTabs:IPhantomTab 983
TTabs:ISameFrameSize 983
TTabs:IShowActive 984
TTabs:IShowTabsMenu 984
TTabs:ITabCloseButtons 984
TTabs:ITabsCanMove 985
TTabs:ITabStop 985
TTabs:ITransparent 985
TTabs:Methods 990
TTabs:Movetab 991
TTabs:nFrame 985
TTabs:nHeight 986
TTabs:nIndex 986
TTabs:nItemPadding 986
TTabs:nOrientation 986
TTabs:nPhantomImage 987
TTabs:nTabsClrPane 987
TTabs:nTabsClrPaneHot 987
TTabs:nTabsClrPaneSel 987
TTabs:nTabsClrPen 988
TTabs:nTabsClrText 988
TTabs:nTabsClrTextHot 988
TTabs:nTabsClrTextSel 988
TTabs:nTabsMargin 989
TTabs:nTabsSize 989
TTabs:nWidth 989
TTabs:olmageList 989
TTabs:OnCalcRect 992
TTabs:OnChange 993
TTabs:OnChanging 993
TTabs:OnClick 994
TTabs:OnCloseTab 993
TTabs:OnDrawTab 994
TTabs:OnDrawTabsBackground 995
TTabs:OnNewTab 995
TTabs:OnRClick 995
TTabs:Properties 982
TTabs:RestoreState 992
TTabs:SaveState 992
TTabsFolder 1001
TTabsFolder:altItems 1001
TTabsFolder:Properties 1001
TTabsFolderPage 1001
TTabsFolderPage:cText 1003
TTabsFolderPage:cToolTip 1003
TTabsFolderPage>Delete 1006
TTabsFolderPage:Events 1006
TTabsFolderPage:isEnabled 1002
TTabsFolderPage:ITransparent 1003
TTabsFolderPage:IVisible 1002

TTabsFolderPage:Methods 1005
TTabsFolderPage:nAlignment 1003
TTabsFolderPage:nBmpAlignment 1004
TTabsFolderPage:nClrText 1004
TTabsFolderPage:nImage 1004
TTabsFolderPage:nItem 1004
TTabsFolderPage:nSize 1005
TTabsFolderPage:OnClick 1007
TTabsFolderPage:OnClose 1007
TTabsFolderPage:OnRClick 1007
TTabsFolderPage:oParent 1005
TTabsFolderPage:Properties 1002
TTabsFolderPage:Selected 1006
TTabsItem 995
TTabsItem:cText 996
TTabsItem:cToolTip 996
TTabsItem:Eventss 1000
TTabsItem:IEnabled 996
TTabsItem:IVisible 997
TTabsItem:nAlignment 997
TTabsItem:nBmpAlignment 997
TTabsItem:nClrText 997
TTabsItem:nImage 998
TTabsItem:nItem 998
TTabsItem:nSize 998
TTabsItem:OnClick 1000
TTabsItem:OnClose 1000
TTabsItem:OnRClick 1000
TTabsItem:oParent 998
TTabsItem:Properties 996
TTabsItem:Selected 1000
TTabsTabsFolderPage:Select 1006
TThread 1600
TThread:Events 1605
TThread:Lock 1601
TThread:Methods 1601
TThread:Notify 1602
TThread:NotifyThread 1602
TThread:nStatus 1601
TThread:OnEnd 1605
TThread:OnNotify 1605
TThread:OnNotifyThread 1605
TThread:OnPause 1606
TThread:OnResume 1606
TThread:OnStart 1606
TThread:Pause 1602
TThread:Properties 1601
TThread:Quit 1603
TThread:Resume 1603
TThread:Run 1603
TThread:Signal 1604
TThread:Stop 1604
TTimePicker 642
TTimePicker:cCustomFormat 642
TTimePicker:cMaxTime 643
TTimePicker:cMinTime 643
TTimePicker:cTime 643
TTimePicker:Events 644
TTimePicker:nHeight 644
TTimePicker:nWidth 644
TTimePicker:OnChange 644
TTimePicker:Properties 642
TTimer 1060
TTimer:Create 1062
TTimer:Destroy 1062
TTimer:Disable 1062
TTimer:Enable 1062
TTimer:Events 1063
TTimer:IEnabled 1061
TTimer:Methods 1061
TTimer:New 1063
TTimer:nInterval 1061
TTimer:OnTimer 1063
TTimer:oParent 1061
TTimer:Properties 1060
TTimer:SetInterval 1063
TTimer:Stop 1063
TToolBar 644
TToolBar:AddButton 650
TToolBar:AddItem 650
TToolBar:AddSeparator 651
TToolBar:altems 645
TToolBar:DeleteItem 651
TToolBar:Events 653
TToolBar:InsertItem 652
TToolBar:InsertSeparator 652
TToolBar:IAlwaysTip 645
TToolBar:IAutoSize 646
TToolBar:IBallons 646
TToolBar:IFlat 646

TToolBar:IList 646
TToolBar:ITransparent 647
TToolBar:Methods 650
TToolBar:nAlign 647
TToolBar:nBorderStyle 647
TToolBar:nBtnHeight 645, 646, 647, 648, 649, 650, 651, 652, 653, 1410, 1411, 1412, 1413, 1414, 1415, 1416
TToolBar:nBtnWidth 648
TToolBar:nHeight 648
TToolBar:nIndent 648
TToolBar:nWidth 648
TToolBar:olmageList 649
TToolBar:olmageListDis 649
TToolBar:olmageListHot 649
TToolBar:OnDropDown 653
TToolBar:Properties 645
TToolButton 653
TToolButton:Click 657
TToolButton:Create 656
TToolButton:cText 654
TToolButton:cTooltip 654
TToolButton:Delete 657
TToolButton:Events 658
TToolButton:IAutoSize 654
TToolButton:IChecked 654
TToolButton:IEnabled 655
TToolButton:IGroup 655
TToolButton:ISeparator 655
TToolButton:IVisible 654, 655, 656, 657, 658
TToolButton:Methods 656
TToolButton:nImage 655
TToolButton:nIndex 656
TToolButton:oMenu 656
TToolButton:OnClick 658
TToolButton:Properties 653
TToolTip 247
TToolTip:AddTool 248
TToolTip:Create 248
TToolTip:DelTool 248
TToolTip:Metodos 247
TToolTip:SetToolInfo 249
TTrackBar 658
TTrackBar:ClearTicks 664
TTrackBar:Events 666
TTrackBar:GetNumTicks 664
TTrackBar:GetTick 665
TTrackBar:GetTickPos 665
TTrackBar:GetTicks 665
TTrackBar:IAutoTicks 659
TTrackBar:IDownIsLeft 659
TTrackBar:IFixedLength 659
TTrackBar:INoThumb 659
TTrackBar:INoTicks 660
TTrackBar:IReversed 660
TTrackBar:ISelRange 660
TTrackBar:ITooltips 660
TTrackBar:ITransparent 661
TTrackBar:Methods 664
TTrackBar:nLineSize 661
TTrackBar:nMax 661
TTrackBar:nMin 661
TTrackBar:nOrientation 662
TTrackBar:nPageSize 662
TTrackBar:nSelEnd 662
TTrackBar:nSelStart 662
TTrackBar:nTickFreq 663
TTrackBar:nTickStyle 663
TTrackBar:nTipSide 663
TTrackBar:nValue 663
TTrackBar:nWidth 664
TTrackBar:OnChange 666
TTrackBar:Properties 658
TTrackBar:SetTick 665
TTrackBarMod 979
TTrackBarMod:Events 981
TTrackBarMod:nClrBarBack 980
TTrackBarMod:nClrBarFore 980
TTrackBarMod:nHeight 979
TTrackBarMod:nMax 980
TTrackBarMod:nMin 980
TTrackBarMod:nOrientation 981
TTrackBarMod:nValue 981
TTrackBarMod:nWidth 981
TTrackBarMod:OnChange 981
TTrackBarMod:Properties 979
TTreeView 666
TTreeView:AddImage 674
TTreeView:AddItem 674
TTreeView:altems 667
TTreeView:CollapseAll 675

- TTreeView:DeleteItems 675
- TTreeView:EndEditLabelNow 675
- TTreeView:Events 679
- TTreeView:ExpandAll 675
- TTreeView:GetAllItems 676
- TTreeView:GetCount 676
- TTreeView:GetItemByHandle 676
- TTreeView:GetItemByName 676
- TTreeView:GetSelectedItem 677
- TTreeView:GetVisibleCount 677
- TTreeView:HitTest 677
- TTreeView:InsertItem 677
- TTreeView:IsEmpty 678
- TTreeView:IAutoHScroll 667
- TTreeView:IBorder 668
- TTreeView:ICheckBoxes 668
- TTreeView:IDoubleBuffer 668
- TTreeView:IDragDropItem 668
- TTreeView:IEditLabels 668
- TTreeView:IFadeInOutExp 669
- TTreeView:IFullRowSelect 669
- TTreeView:IHasButtons 669
- TTreeView:IHasLines 669
- TTreeView:IHotTrack 670
- TTreeView:ILinesAtRoot 670
- TTreeView:IRichTooltip 670
- TTreeView:IShowSelAlways 670
- TTreeView:ISingleExpand 670
- TTreeView:ITooltips 671
- TTreeView:Methods 673
- TTreeView:nClrLine 671
- TTreeView:nClrPane 671
- TTreeView:nClrSelFocus 672
- TTreeView:nClrSelItem 672
- TTreeView:nClrText 671
- TTreeView:nHeight 672
- TTreeView:nIndent 672
- TTreeView:nWidth 673
- TTreeView:olmageList 673
- TTreeView:OnBeginDragItem 680
- TTreeView:OnBeginLabelEdit 680
- TTreeView:OnCheckStateChanged 680
- TTreeView:OnClick 681
- TTreeView:OnDragOverItem 681
- TTreeView:OnEndDragItem 681
- TTreeView:OnEndLabelEdit 682
- TTreeView:OnItemExpanded 682
- TTreeView:OnItemExpanding 682
- TTreeView:OnKeyDown 683
- TTreeView:OnRClick 683
- TTreeView:OnSelChanged 683
- TTreeView:OnSelChanging 684
- TTreeView:OnSingleExpand 684
- TTreeView:Properties 666
- TTreeView:RestoreExpandState 678
- TTreeView:SaveExpandState 679
- TTreeView>SelectItem 679
- TTreeView>SelectRoot 679
- TTreeViewItem 685
- TTreeViewItem:AddItem 688
- TTreeViewItem:altItems 685
- TTreeViewItem:AltSortChildren 689
- TTreeViewItem:Collapse 689
- TTreeViewItem:Create 690
- TTreeViewItem:cText 685
- TTreeViewItem>Delete 690
- TTreeViewItem>DeleteChilds 690
- TTreeViewItem:DeleteItems 690
- TTreeViewItem>Edit 690
- TTreeViewItem:EnsureVisible 691
- TTreeViewItem:Expand 691
- TTreeViewItem:ExpandPartial 691
- TTreeViewItem:GetNextSibling 691
- TTreeViewItem:GetPrevSibling 692
- TTreeViewItem:Handle 686
- TTreeViewItem:InsertItem 692
- TTreeViewItem:IsExpanded 693
- TTreeViewItem:IBold 686
- TTreeViewItem:IChecked 686
- TTreeViewItem:IParam 686
- TTreeViewItem:Methods 688
- TTreeViewItem:MoveAfter 693
- TTreeViewItem:MoveDown 693
- TTreeViewItem:MoveInto 693
- TTreeViewItem:MoveUp 694
- TTreeViewItem:nImage 687
- TTreeViewItem:nIndex 687
- TTreeViewItem:nSelImage 687
- TTreeViewItem:oParent 687
- TTreeViewItem:oTreeView 688

TTreeViewItem:Properties 685
TTreeViewItem:Select 694
TTreeViewItem:SetExpandable 694
TTreeViewItem:SortChildren 694
TTreeViewItemMod 1015,1019
TTreeViewItemMod:AddItem 1023
TTreeViewItemMod:alters 1020
TTreeViewItemMod:Collapse 1023
TTreeViewItemMod:cText 1020
TTreeViewItemMod:cTooltip 1020
TTreeViewItemMod:Delete 1023
TTreeViewItemMod:DeleteChilds 1024
TTreeViewItemMod:DeleteItems 1024
TTreeViewItemMod:EnsureVisible 1024
TTreeViewItemMod:Expand 1024
TTreeViewItemMod:FirstChild 1024
TTreeViewItemMod:GetNextSibling 1024
TTreeViewItemMod:GetPrevSibling 1025
TTreeViewItemMod:HasChilds 1025
TTreeViewItemMod:InsertItem 1025
TTreeViewItemMod:LastChild 1025
TTreeViewItemMod:IChecked 1020
TTreeViewItemMod:IExpanded 1021
TTreeViewItemMod:ISelected 1021
TTreeViewItemMod:Methods 1022
TTreeViewItemMod:MoveAfter 1026
TTreeViewItemMod:MoveDown 1026
TTreeViewItemMod:MoveUp 1026
TTreeViewItemMod:nClrText 1021
TTreeViewItemMod:nID 1021
TTreeViewItemMod:nImage 1021
TTreeViewItemMod:nIndex 1022
TTreeViewItemMod:nSellImage 1022
TTreeViewItemMod:oParent 1022
TTreeViewItemMod:oTreeView 1022
TTreeViewItemMod:Properties 1019
TTreeViewItemMod:Select 1026
TTreeViewItemMod:SortChildren 1026
TTreeViewMod 1008
TTreeViewMod:AddItem 1011
TTreeViewMod:alters 1008
TTreeViewMod:BeginPaint 1011
TTreeViewMod:CollapseAll 1012
TTreeViewMod:DeleteItems 1012
TTreeViewMod:EditLabel 1012
TTreeViewMod:EndEditLabelNow 1012
TTreeViewMod:EndPaint 1012
TTreeViewMod:Events 1016
TTreeViewMod:ExpandAll 1013
TTreeViewMod:GetAllItems 1013
TTreeViewMod:GetCount 1013
TTreeViewMod:GetItemAtPos 1013
TTreeViewMod:GetItemByID 1014
TTreeViewMod:GetItemByName 1014
TTreeViewMod:GetItemPos 1014
TTreeViewMod:GetSelectedItem 1015
TTreeViewMod:InsertItem 1015
TTreeViewMod:IsEmpty 1015
TTreeViewMod:IEditLabels 1009
TTreeViewMod:IHasLines 1009
TTreeViewMod:IMultiLine 1009
TTreeViewMod:ISingleExpand 1009
TTreeViewMod:ITooltips 1010
TTreeViewMod:Methods 1010
TTreeViewMod:nGap 1010
TTreeViewMod:nMargin 1010
TTreeViewMod:oEdit 1010
TTreeViewMod:OnBeginLabelEdit 1016
TTreeViewMod:OnClick 1016
TTreeViewMod:OnDbClick 1017
TTreeViewMod:OnDrawItem 1017
TTreeViewMod:OnEndLabelEdit 1018
TTreeViewMod:OnGetTooltipText 1018
TTreeViewMod:OnItemExpanded 1019
TTreeViewMod:OnItemExpanding 1019
TTreeViewMod:Propiedades 1008
TTreeViewMod>SelectItem 1015
TTreeViewMod>SelectRoot 1016
TUnzipFile 1445
TUnzipFile:aFileMask 1446
TUnzipFile:aFiles 1447
TUnzipFile:cDirectory 1447
TUnzipFile:cFileName 1447
TUnZipFile:cLastError 1447
TUnzipFile:cPassword 1447
TUnZipFile:DeleteFiles 1450
TUnzipFile:Events 1452
TUnzipFile:Execute 1451
TUnZipFile:FilesCount 1450
TUnZipFile:FilesInfo 1450

- TUnzipFile:FileSize 1451
- TUnzipFile:HasPassword 1451
- TUnzipFile:IncludePath 1448
- TUnzipFile:Methods 1449
- TUnzipFile:nBufferSize 1448
- TUnzipFile:nCount 1448
- TUnzipFile:nExtractBufferSize 1448
- TUnzipFile:nLastError 1449
- TUnzipFile:nReadBufferSize 1449
- TUnzipFile:nWriteBufferSize 1449
- TUnzipFile:OnChangeDisk 1452
- TUnzipFile:OnChangeFile 1452
- TUnzipFile:OnProgress 1452
- TUnzipFile:Properties 1446
- TUnzipFile:Run 1451
- TUpDown 695
- TUpDown:aAccels 695
- TUpDown:Events 699
- TUpDown:IArrowKeys 696
- TUpDown:IAutoBuddy 696
- TUpDown:IHotTrack 696
- TUpDown:INoThousands 696
- TUpDown:ISyncBuddy 697
- TUpDown:IWrap 697
- TUpDown:nAccelTime 697
- TUpDown:nBase 697
- TUpDown:nBuddyAlign 697
- TUpDown:nIncrement 698
- TUpDown:nMax 698
- TUpDown:nMin 698
- TUpDown:nOrientation 698
- TUpDown:nValue 699
- TUpDown:oBuddy 699
- TUpDown:OnChange 699
- TUpDown:Properties 695
- TWebBrowser 794
- TWebBrowser:Events 796
- TWebBrowser:Methods 795
- TWebBrowser:Properties 794
- TWebDataSource 1247
- TWebDataSource:BeginTrans 1253
- TWebDataSource:BulkExecute 1253
- TWebDataSource:cConnect 1249
- TWebDataSource:cCryptKey 1249
- TWebDataSource:cDatabase 1250
- TWebDataSource:cLastError 1250
- TWebDataSource:CommitTrans 1253
- TWebDataSource:Connect 1254
- TWebDataSource:cPhpModule 1250
- TWebDataSource:CurrentDir 1254
- TWebDataSource>DeleteFile 1254
- TWebDataSource:Disconnect 1254
- TWebDataSource:Eventos 1261
- TWebDataSource:Execute 1255
- TWebDataSource:File 1255
- TWebDataSource:ForceDir 1255
- TWebDataSource:GetCatalogs 1256
- TWebDataSource:GetTables 1256
- TWebDataSource:IsDir 1256
- TWebDataSource:ITinyIntAsBoolean 1250
- TWebDataSource:Metodos 1252
- TWebDataSource:Mkdir 1259
- TWebDataSource:MyIP 1259
- TWebDataSource:MyLocation 1259
- TWebDataSource:nAffectedRows 1251
- TWebDataSource:nDbType 1251
- TWebDataSource:nInsertId 1251
- TWebDataSource:nPort 1251
- TWebDataSource:nSendTimeOut 1252
- TWebDataSource:OnValidate 1261
- TWebDataSource:Properties 1249
- TWebDataSource:Query 1256
- TWebDataSource:QueryArray 1257
- TWebDataSource:QueryArrayHash 1257
- TWebDataSource:QueryReport 1257
- TWebDataSource:QueryRow 1258
- TWebDataSource:QueryRowHash 1258
- TWebDataSource:QueryValue 1258
- TWebDataSource:RollBackTrans 1259
- TWebDataSource:SendMail 1260
- TWebDataSource:Table 1260
- TWebDataSource:UpdateFile 1260
- TWebView 699
- TWebView:CanGoBack 704
- TWebView:CanGoForward 704
- TWebView:CleanUserData 704
- TWebView:cRequestedFilter 701
- TWebView:cRuntimeDir 701
- TWebView:cUserDataDir 701
- TWebView>DeleteAllCookies 704

TWebView:Events 707
TWebView:ExecuteScript 705
TWebView:GetTitle 705
TWebView:GoBack 705
TWebView:GoForward 705
TWebView:IsInstalled 706
TWebView:AllowDevTools 701
TWebView:AllowScripts 701
TWebView:CleanUserData 702
TWebView:ContextMenu 702
TWebView:StatusBar 702
TWebView:Methods 703
TWebView:Navigate 706
TWebView:NavigateEx 706
TWebView:nBorderStyle 702
TWebView:nHeight 703
TWebView:nWidth 703
TWebView:nZoomFactor 703
TWebView:OnAcceleratorKeyPressed 708
TWebView:OnContentLoading 708
TWebView:OnDocumentTitleChanged 708
TWebView:OnExecuteScriptCompleted 709
TWebView:OnHistoryChanged 709
TWebView:OnNavigationCompleted 709
TWebView:OnNavigationStarting 709
TWebView:OnNewWindowRequested 710
TWebView:OnSourceChanged 710
TWebView:OnWebResourceRequested 710
TWebView:Properties 700
TWebView:Reload 706
TWebView:SetContent 707
TWebView:SetVirtualHostNameToFolder 707
TWebView:Stop 707
TWiaScan 1453
TWiaScan:cDeviceld 1453
TWiaScan:DeviceInfo 1454
TWiaScan:Devices 1455
TWiaScan:Eventos 1456
TWiaScan:ImageSize 1455
TWiaScan:Methods 1454
TWiaScan:nImageMode 1454
TWiaScan:nResolution 1454
TWiaScan:OnScan 1456
TWiaScan:Properties 1453
TWiaScan:Scan 1455
TWinContol:GoFirstControl 79
TWinContol:Redraw 80
TWinControl 76
TWinControl:aControls 77
TWinControl:GoNextControl 80
TWinControl:GoPrevControl 80
TWinControl:InsertControl 80
TWinControl:Methods 79
TWinControl:nBkGndMarginX 78
TWinControl:nBkGndMarginY 78
TWinControl:nBkGndMode 78
TWinControl:nClrPaneEnd 77
TWinControl:nGradient 77
TWinControl:oActiveControl 78
TWinControl:oBkGnd 79
TWinControl:oMenu 79
TWinControl:Properties 77
TWinControl:RemoveControl 81
TWinControl:RequestState 81
TWinObject 36
TWinObject:Handle 36
TWinObject:Properties 36
TWinService 1064
TWinService:cDescription 1065
TWinService:cDisplayName 1065
TWinService:cName 1065
TWinService:Events 1067
TWinService:Install 1066
TWinService:IsInstalled 1066
TWinService:IsRunning 1066
TWinService:Methods 1065
TWinService:nStatus 1065
TWinService:OnPowerEvent 1068
TWinService:OnRun 1068
TWinService:OnShutDown 1068
TWinService:Properties 1064
TWinService:Run 1066
TWinService:Start 1067
TWinService:Stop 1067
TWinService:Uninstall 1067
TZipFile 1456
TZipFile:AddFile 1462
TZipFile:aFileMask 1457
TZipFile:aFiles 1457
TZipFile:cComment 1457

TZipFile:cFileName 1458
 TZipFile:cLastError 1458
 TZipFile:cPassword 1458
 TZipFile:Events 1462
 TZipFile:Execute 1462
 TZipFile:FilesSize 1462
 TZipFile:IncludePath 1460
 TZipFile:IMasksComPath 1461
 TZipFile:IOverwrite 1461
 TZipFile:IRrecurse 1461
 TZipFile:IUnicode 1461
 TZipFile:Methods 1462
 TZipFile:nBufferSize 1458
 TZipFile:nCompressionLevel 1458
 TZipFile:nCount 1459
 TZipFile:nExtractBuferSize 1459
 TZipFile:nLastError 1459
 TZipFile:nReadBufferSize 1460
 TZipFile:nSpanMode 1460
 TZipFile:nWriteBufferSize 1460
 TZipFile:OnChangeDisk 1463
 TZipFile:OnChangeFile 1463
 TZipFile:OnProgress 1463
 TZipFile:Properties 1456
 TZipFile:Run 1462

- U -

uDefaultValue 461, 863
 UnCheckAll 952
 Undo 606
 Uninstall 1067
 Unique 1435
 UnSort 837
 Update 62, 620, 1310
 UpdateControls 1310
 UpdateFile 1260
 UploadFile 778

- V -

VAlignment 946
 Valtpe 1275
 Value 270, 321, 359, 463, 863, 1273, 1362, 1375,
 1377, 1380, 1383, 1389, 1401, 1418, 1420, 1422
 VarGet 1275, 1311

VariablesList 1531
 VarPut 1275, 1311
 Version 1219

- W -

W2Hex 1633
 WideToAnsi 1633
 Windows functions 1704
 WithExtension 1436
 WithName 1436
 WithPath 1436
 WordToRtf 606
 WordXToRtf 606
 Write 789, 1054, 1441
 WriteData 147
 WriteFile 770
 WriteToRtf 606

- X -

XA_Aes256Decrypt 1689
 XA_Aes256Encrypt 1689
 XA_CaptureBitmap 1675
 XA_DateTimeZ2Local 1633
 XA_DrawRoundedRect 1675
 XA_FillRoundedRect 1675
 XA_GetColorization 1676
 XA_GpfHandler 1663
 XA_GradientFillH 1676
 XA_GradientFillV 1676
 XA_isGDIPlus 1663
 XA_IsVistaOrHigher 1701
 XA_IsWin10 1700
 XA_IsWin11 1700
 XA_IsWin7 1699
 XA_IsWin8 1699
 XA_IsWin9x 1699
 XA_IsWinNT 1700
 XA_IsWinVista 1701
 XA_IsWow64 1700
 XA_IsXPOrHigher 1701
 XA_Lightness 1677
 XA_LineCount 1690
 XA_LoadImageAndResize 1677

XA_Luminance 1678
XA_PaintCross 1678
XA_RemoveTags 1690
XA_ResourceSearchOrder 1663
XA_RTFToTxt 1690
XA_SetBmpDisabled 1678
XA_SetKey 1679
XA_SLCreate 1693
XA_SLFree 1694
XA_SLSearch 1694
XA_UrDecode 1634
XA_UrDecodeToAnsi 1634
XA_UrlEncode 1633
XA_WLAdd 1692
XA_WLCreate 1691
XA_WLFree 1693
XA_WLSearch 1692
XailerVersion 1701
xHarbour 1716
XHtmlHelp 98

- Z -

Zap 445, 1340, 1349
ZoomMinus 1571
ZoomPlus 1571

Endnotes 2... (after index)

Back Cover