



ScaleRichView

ScaleRichView © trichview.com

Table of Contents

Part I ScaleRichView	18
1..Version history	19
New after v12.0	20
New in v12.0	20
New in v11.0	21
New in v10.0	22
New in v9.0	23
New in v8.0	25
New in v7.0	26
New in v6.0	27
New in v5.0	28
New in v4.0	30
New in v3.0	32
New in v2.0	34
2..Overview	36
Document parts in TSRichViewEdit	36
Notes and text boxes	38
3..Components	39
TSRichViewEdit	40
Properties	44
TSRichViewEdit.AcceptDragDropFormats, AcceptPasteFormats	47
TSRichViewEdit.ActiveEditor	47
TSRichViewEdit.AnimationMode	47
TSRichViewEdit.BackgroundProperty	48
TSRichViewEdit.BiDiMode	48
TSRichViewEdit.BottomMarginPix	48
TSRichViewEdit.CanUpdate, CanScroll, CanUpdatePosition	48
TSRichViewEdit.CanUpdateMargin	49
TSRichViewEdit.CaretBlinkTime	49
TSRichViewEdit.CaretPos	49
TSRichViewEdit.CaretVisible	49
TSRichViewEdit.CPEventKind	49
TSRichViewEdit.CurControl	50
TSRichViewEdit.CurrentNote, CurrentNoteParentEditor	50
TSRichViewEdit.CurrentPage	50
TSRichViewEdit.Document	51
TSRichViewEdit.ExternalRVStyle, -Header, -Footer	51
TSRichViewEdit.FirstPageNo	51
TSRichViewEdit.FixMarginsMode	52
TSRichViewEdit.HMaxScrollPos	52
TSRichViewEdit.HScrollBar	52
TSRichViewEdit.HScrollBarSchemeIndex	53
TSRichViewEdit.HScrollPos	53
TSRichViewEdit.HTMLReadProperties	53
TSRichViewEdit.HTMLSaveProperties	53
TSRichViewEdit.LastPageNo	53
TSRichViewEdit.LeftMarginPix	54
TSRichViewEdit.LineNumberProperty	54
TSRichViewEdit.MarkdownProperties	54
TSRichViewEdit.MaxPrintedItemNo, MaxPrintedOffsInItem	54

TSRichViewEdit.MenuHButtons	55
TSRichViewEdit.MenuHorizontal	55
TSRichViewEdit.MenuVButtons	55
TSRichViewEdit.MenuVertical	56
TSRichViewEdit.MinPrintedItemNo, MinPrintedOffsInItem	56
TSRichViewEdit.OffsetX	56
TSRichViewEdit.OffsetY	57
TSRichViewEdit.PageCount	57
TSRichViewEdit.PageHeightPix, PageHeight100Pix	57
TSRichViewEdit.PageNoMouseDown	57
TSRichViewEdit.PageNoMouseMove	57
TSRichViewEdit.PagePosProperty	58
TSRichViewEdit.PageProperty	58
TSRichViewEdit.PageWidthPix, PageWidth100Pix	58
TSRichViewEdit.RangeSearch	58
TSRichViewEdit.RangeSearchFirstY	59
TSRichViewEdit.RangeSearchLastY	59
TSRichViewEdit.ReadModeProperty	59
TSRichViewEdit.ReadOnly	59
TSRichViewEdit.RichViewEdit	59
TSRichViewEdit.RightMarginPix	60
TSRichViewEdit.RTFOptions	60
TSRichViewEdit.RTFReadProperties	60
TSRichViewEdit.RVBackgroundBitmap	61
TSRichViewEdit.RVBackgroundStyle	61
TSRichViewEdit.RVColor	61
TSRichViewEdit.RVEditorOptions	61
TSRichViewEdit.RVFooter	61
TSRichViewEdit.RVFOptions	62
TSRichViewEdit.RVFPaStylesReadMode	63
TSRichViewEdit.RVFTextStylesReadMode	63
TSRichViewEdit.RVHeader	63
TSRichViewEdit.RVNote	64
TSRichViewEdit.RVOptions	64
TSRichViewEdit.ScrolledPage	65
TSRichViewEdit.SkinManager	65
TSRichViewEdit.SmartPopupProperties	65
TSRichViewEdit.SmartPopupVisible	65
TSRichViewEdit.StyleTemplateInsertMode	65
TSRichViewEdit.Subdocuments	66
TSRichViewEdit.TabNavigation	67
TSRichViewEdit.TextEngine	67
TSRichViewEdit.TopMarginPix	67
TSRichViewEdit.UnitsProgram	68
TSRichViewEdit.UseDrawHyperlinksEvent	68
TSRichViewEdit.UseStyleTemplates	68
TSRichViewEdit.Version	68
TSRichViewEdit.ViewProperty	68
TSRichViewEdit.VMaxScrollPos	68
TSRichViewEdit.VScrollBar	69
TSRichViewEdit.ScrollBarControlStyle	69
TSRichViewEdit.VScrollBarSchemeIndex	69
TSRichViewEdit.VScrollPos	69
Methods	70
TSRichViewEdit.BeginComponentsUpdate	72
TSRichViewEdit.CalculateAllPagePositions	72
TSRichViewEdit.CalculatePageCount	72
TSRichViewEdit.CalculatePagePosition	72

TSRichViewEdit.Clear	73
TSRichViewEdit.ConvertRVtoSRV	73
TSRichViewEdit.ConvertSRVtoRV	73
TSRichViewEdit.ConvertTo	74
TSRichViewEdit.Copy	75
TSRichViewEdit.Create	75
TSRichViewEdit.Cut	75
TSRichViewEdit.DeletePage	75
TSRichViewEdit.Destroy	76
TSRichViewEdit.DrawMetafile	76
TSRichViewEdit.DrawPage	76
TSRichViewEdit.EndComponentsUpdate	77
TSRichViewEdit.FindNextCheckpoint	77
TSRichViewEdit.FindNextHeading	78
TSRichViewEdit.FindNextHyperlink	78
TSRichViewEdit.FindNextItem	79
TSRichViewEdit.FindPriorCheckpoint	80
TSRichViewEdit.FindPriorHeading	80
TSRichViewEdit.FindPriorHyperlink	81
TSRichViewEdit.FindPriorItem	81
TSRichViewEdit.FirstCurPage	82
TSRichViewEdit.Format, Reformat	82
TSRichViewEdit.GetCaretPosInUnits	83
TSRichViewEdit.GetCurrentLineCol	83
TSRichViewEdit.GetFirstItemVisibleRV	83
TSRichViewEdit.GetFirstVisiblePage	83
TSRichViewEdit.GetItemAt	84
TSRichViewEdit.GetItemBounds, GetItemBounds100	84
TSRichViewEdit.GetItemCoords100	85
TSRichViewEdit.GetItemPages	86
TSRichViewEdit.GetLastItemVisibleRV	86
TSRichViewEdit.GetLastVisiblePage	86
TSRichViewEdit.GetPageAt	86
TSRichViewEdit.GetPageClientRect	87
TSRichViewEdit.GetPageLastItemNo	87
TSRichViewEdit.GetPageNo	87
TSRichViewEdit.GetPageStartItemNo	88
TSRichViewEdit.GetPageStartTableItem	88
TSRichViewEdit.GetPageStartTableRow	88
TSRichViewEdit.GetTableIconItem	89
TSRichViewEdit.GetTableIconRVData	89
TSRichViewEdit.InsertPageBreak	89
TSRichViewEdit.InZoomRect	89
TSRichViewEdit.LastCurPage	90
TSRichViewEdit.LoadRVF	90
TSRichViewEdit.LoadRVFFromStream	90
TSRichViewEdit.NextCurCheckpoint	90
TSRichViewEdit.NextCurHeading	90
TSRichViewEdit.NextCurHyperlink	91
TSRichViewEdit.NextCurItem	91
TSRichViewEdit.NextCurPage	91
TSRichViewEdit.OutZoomRect	92
TSRichViewEdit.Paste	92
TSRichViewEdit.PrintAll	92
TSRichViewEdit.PrintCurrent	93
TSRichViewEdit.PrintRange	93
TSRichViewEdit.PriorCurCheckpoint	94
TSRichViewEdit.PriorCurHeading	94

TSRichViewEdit.PriorCurHyperlink.....	94
TSRichViewEdit.PriorCurItem.....	94
TSRichViewEdit.PriorCurPage.....	95
TSRichViewEdit.ProcessControls	95
TSRichViewEdit.ReturnToNote.....	95
TSRichViewEdit.ScrollToCaret, ScrollCaretToCenter	95
TSRichViewEdit.ScrollToItem.....	96
TSRichViewEdit.SelectAll	96
TSRichViewEdit.Set*PropertyEd.....	96
TSRichViewEdit.SetMargin.....	97
TSRichViewEdit.SetMarginMM.....	97
TSRichViewEdit.SetMarginUnit.....	98
TSRichViewEdit.SetRVMargins.....	98
TSRichViewEdit.StartEditing.....	98
TSRichViewEdit.StartEditNote.....	99
TSRichViewEdit.UnitsPerInchH.....	99
TSRichViewEdit.UnitsPerInchV.....	100
TSRichViewEdit.UpdateBuffer, UpdateBufferAt.....	100
TSRichViewEdit.ZoomCanvas, RestoreCanvasZoom.....	100
TSRichViewEdit.ZoomRectIn, ZoomRectOut.....	100
TSRichViewEdit.ZoomToFullPages	101
Events	101
TSRichViewEdit.OnAssignImageFileName.....	103
TSRichViewEdit.OnBeforeOleDrop, OnAfterOleDrop.....	103
TSRichViewEdit.OnCaretGetOut.....	104
TSRichViewEdit.OnCaretMove, OnCaretMoving.....	104
TSRichViewEdit.OnChange.....	104
TSRichViewEdit.OnChangeActiveEditor.....	104
TSRichViewEdit.OnChangeCurrentControl	104
TSRichViewEdit.OnChangeViewModeAfter.....	104
TSRichViewEdit.OnChangeViewModeBefore.....	105
TSRichViewEdit.OnChanging.....	105
TSRichViewEdit.OnCheckControl.....	105
TSRichViewEdit.OnCheckpointVisible.....	105
TSRichViewEdit.OnCheckStickingItems.....	106
TSRichViewEdit.OnClickPage.....	106
TSRichViewEdit.OnContextPopup.....	106
TSRichViewEdit.OnControlAction.....	106
TSRichViewEdit.OnCopy.....	106
TSRichViewEdit.OnCurParaStyleChanged.....	106
TSRichViewEdit.OnCurrentPageChange.....	107
TSRichViewEdit.OnCurTextStyleChanged.....	107
TSRichViewEdit.OnDrawHyperlink.....	107
TSRichViewEdit.OnDropFile, OnDropFiles	107
TSRichViewEdit.OnFullRedraw.....	108
TSRichViewEdit.OnGetPagePos.....	108
TSRichViewEdit.OnHMenuClickButton.....	109
TSRichViewEdit.OnHMenuEnterButton.....	109
TSRichViewEdit.OnHScrolled	109
TSRichViewEdit.OnHTMLSaveImage.....	109
TSRichViewEdit.OnImportPicture.....	110
TSRichViewEdit.OnItemAction	110
TSRichViewEdit.OnItemHint.....	110
TSRichViewEdit.OnItemResize.....	110
TSRichViewEdit.OnItemTextEdit.....	110
TSRichViewEdit.OnJump.....	110
TSRichViewEdit.OnLoadCustomFormat.....	111
TSRichViewEdit.OnLoadDocument.....	111

TSRichViewEdit.OnMarginsChanged.....	111
TSRichViewEdit.OnMessageControl	112
TSRichViewEdit.OnNewDocument.....	112
TSRichViewEdit.OnOleDragEnter.....	113
TSRichViewEdit.OnOleDragLeave.....	113
TSRichViewEdit.OnOleDragOver	113
TSRichViewEdit.OnOleDrop.....	113
TSRichViewEdit.OnPageCountChanged.....	113
TSRichViewEdit.OnPageFormatChanged.....	114
TSRichViewEdit.OnPageScrolled	114
TSRichViewEdit.OnPaint.....	114
TSRichViewEdit.OnPaintComponent.....	114
TSRichViewEdit.OnPaintPage.....	115
TSRichViewEdit.OnParaStyleConversion.....	116
TSRichViewEdit.OnPaste.....	116
TSRichViewEdit.OnPrinting.....	116
TSRichViewEdit.OnProgress.....	116
TSRichViewEdit.OnReadField.....	116
TSRichViewEdit.OnReadHyperlink.....	117
TSRichViewEdit.OnReadMergeField.....	117
TSRichViewEdit.OnResize, OnResizing.....	117
TSRichViewEdit.OnResizeDoc.....	117
TSRichViewEdit.OnRVDbClick.....	117
TSRichViewEdit.OnRVFControlNeeded	117
TSRichViewEdit.OnRVFImageListNeeded.....	118
TSRichViewEdit.OnRVFPictureNeeded.....	118
TSRichViewEdit.OnRVMouseDown.....	118
TSRichViewEdit.OnRVMouseMove.....	118
TSRichViewEdit.OnRVMouseUp.....	118
TSRichViewEdit.OnRVRightClick.....	119
TSRichViewEdit.OnSaveComponentToFile.....	119
TSRichViewEdit.OnSaveCustomFormat.....	119
TSRichViewEdit.OnSaveDocXExtra.....	119
TSRichViewEdit.OnSaveHTMLExtra	119
TSRichViewEdit.OnSaveImage2	119
TSRichViewEdit.OnSaveItemToFile.....	120
TSRichViewEdit.OnSaveRTFExtra.....	120
TSRichViewEdit.OnSelect.....	120
TSRichViewEdit.OnSmartPopupClick.....	120
TSRichViewEdit.OnSpellingCheck.....	120
TSRichViewEdit.OnStyleConversion.....	120
TSRichViewEdit.OnTableIconClick.....	121
TSRichViewEdit.OnTextFound	121
TSRichViewEdit.OnURLNeeded.....	121
TSRichViewEdit.OnVMenuClickButton	121
TSRichViewEdit.OnVMenuEnterButton.....	121
TSRichViewEdit.OnVScrolled.....	122
TSRichViewEdit.OnWriteHyperlink.....	122
TSRichViewEdit.OnWriteObjectProperties.....	122
TSRichViewEdit.OnZoomChanged	122
TSRichViewEdit.OnZoomViewChanged.....	122
Classes of properties.....	123
TSRVBackgroundProperty.....	123
Properties	123
TSRVBackgroundProperty.ColorBegin.....	124
TSRVBackgroundProperty.ColorEnd.....	124
TSRVBackgroundProperty.ColorMiddle.....	124
TSRVBackgroundProperty.FillType.....	124

TSRVBackgroundProperty.GlobalPageBackgroundColor	125
TSRVBackgroundProperty.PercentMiddle	125
TSRVBackgroundProperty.Picture	125
TSRVBackgroundProperty.PicturePosition	126
TSRVBackgroundProperty.TransparentColor	126
TSRVBackgroundProperty.Visible	126
TSRVCustomPropertyTB	126
Properties	127
TSRVCustomPropertyTB.AlignButton	128
TSRVCustomPropertyTB.BorderWidth	128
TSRVCustomPropertyTB.ButtonHeight	128
TSRVCustomPropertyTB.Buttons	128
TSRVCustomPropertyTB.ButtonWidth	128
TSRVCustomPropertyTB.Color	129
TSRVCustomPropertyTB.ColorDisable	129
TSRVCustomPropertyTB.ColorDown	129
TSRVCustomPropertyTB.ColorSelect	129
TSRVCustomPropertyTB.ImageList	130
TSRVCustomPropertyTB.Indent	130
TSRVCustomPropertyTB.PenFrame	130
TSRVCustomPropertyTB.ScaleImagesForDPI	130
TSRVCustomPropertyTB.Spacer	130
TSRVCustomPropertyTB.SRVToolBar	131
TSRVLineNumberProperty	131
Properties	131
TSRVLineNumberProperty.DistanceFromText	131
TSRVLineNumberProperty.Font	132
TSRVLineNumberProperty.StartFrom	132
TSRVLineNumberProperty.Step	132
TSRVLineNumberProperty.TablesAsLines	132
TSRVLineNumberProperty.Ticks	132
TSRVLineNumberProperty.Visible	133
TSRVPageProperty	133
Properties	134
TSRVPageProperty.AutoWidth	135
TSRVPageProperty.BorderPen	135
TSRVPageProperty.BottomMargin	135
TSRVPageProperty.CaretPen	135
TSRVPageProperty.FacingPages	136
TSRVPageProperty.FooterVisible	136
TSRVPageProperty.FooterY	136
TSRVPageProperty.HeaderVisible	137
TSRVPageProperty.HeaderY	137
TSRVPageProperty.LeftMargin	137
TSRVPageProperty.MirrorMargins	138
TSRVPageProperty.Orientation	138
TSRVPageProperty.PageFormat	138
TSRVPageProperty.PageHeight	139
TSRVPageProperty.PageNoFirst	139
TSRVPageProperty.PageNoFont	139
TSRVPageProperty.PageNoFromNumber	139
TSRVPageProperty.PageNoHAlign	139
TSRVPageProperty.PageNoVAlign	140
TSRVPageProperty.PageNoVisible	140
TSRVPageProperty.PageViewMode	140
TSRVPageProperty.PageWidth	141
TSRVPageProperty.PrintableAreaPen	141
TSRVPageProperty.RightMargin	141

TSRVPageProperty.TitlePage.....	141
TSRVPageProperty.TopMargin.....	142
TSRVPageProperty.UsePageOrders.....	142
Methods	142
TSRVPageProperty.ConvertPageFormatToPageSize.....	142
TSRVPageProperty.ConvertPageSizeToPageFormat.....	142
TSRVPagePositionProperty.....	143
Properties	143
TSRVPagePositionProperty.AlignPageH.....	144
TSRVPagePositionProperty.AlignPageV.....	144
TSRVPagePositionProperty.HiddenDistances	145
TSRVPagePositionProperty.HPadding, VPadding.....	145
TSRVPagePositionProperty.MaxPageColCount.....	146
TSRVPagePositionProperty.PageHSpacing, PageVSpacing.....	146
TSRVPropertyTBH	147
Properties	148
TSRVPropertyTBH.Align.....	148
TSRVPropertyTBH.Visible.....	148
TSRVPropertyTBV	148
Properties	149
TSRVPropertyTBV.Align.....	149
TSRVPropertyTBV.Visible.....	150
TSRVReadModeProperty	150
Properties	151
TSRVReadModeProperty.Active.....	151
TSRVReadModeProperty.HidePageBackgroundImage.....	151
TSRVReadModeProperty.MinPageColCount, MaxPageColCount.....	151
TSRVReadModeProperty.PageColor	152
TSRVReadModeProperty.PageFlipEffect.....	152
TSRVViewProperty	153
Properties	154
TSRVViewProperty.AutoVScrollBarPosition.....	155
TSRVViewProperty.DarkMode.....	156
TSRVViewProperty.DarkModeUI.....	156
TSRVViewProperty.FooterPen	156
TSRVViewProperty.FooterTitleColor.....	156
TSRVViewProperty.FooterTitleFont.....	157
TSRVViewProperty.FreePosPage.....	157
TSRVViewProperty.HeaderFooterShade.....	157
TSRVViewProperty.HeaderPen.....	158
TSRVViewProperty.HeaderTitleColor.....	158
TSRVViewProperty.HeaderTitleFont.....	158
TSRVViewProperty.HintFont.....	159
TSRVViewProperty.HintPrefixText.....	159
TSRVViewProperty.IconStyle.....	159
TSRVViewProperty.MainDocumentShade.....	159
TSRVViewProperty.MainPen	160
TSRVViewProperty.MainTitleColor.....	160
TSRVViewProperty.MainTitleFont.....	160
TSRVViewProperty.MarginsPen.....	161
TSRVViewProperty.MarginsRectVisible.....	161
TSRVViewProperty.MouseWheelZoom.....	161
TSRVViewProperty.PageHeight.....	161
TSRVViewProperty.PageWidth.....	161
TSRVViewProperty.ShowScrollHint.....	162
TSRVViewProperty.TableIconDelay.....	162
TSRVViewProperty.TableIconPopupMenu.....	162
TSRVViewProperty.Texts	162

TSRVViewProperty.UseTableIcons	163
TSRVViewProperty.UseVCLThemes	164
TSRVViewProperty.ViewMode	164
TSRVViewProperty.WheelStep	164
TSRVViewProperty.WorkArea	164
TSRVViewProperty.ZoomFont	165
TSRVViewProperty.ZoomMenu	165
TSRVViewProperty.ZoomMin, ZoomMax	165
TSRVViewProperty.ZoomMode	165
TSRVViewProperty.ZoomModeEdit	166
TSRVViewProperty.ZoomModeIN	166
TSRVViewProperty.ZoomModeOUT	166
TSRVViewProperty.ZoomPanelFont	166
TSRVViewProperty.ZoomPanelUsesGradient	167
TSRVViewProperty.ZoomPanelVisible	167
TSRVViewProperty.ZoomPercent	167
TSRVViewProperty.ZoomPercentEdit	168
TSRVViewProperty.ZoomPercentIN	168
TSRVViewProperty.ZoomPercentOUT	168
TDBSRichViewEdit	168
Properties	169
TDBSRichViewEdit.AutoDisplay	172
TDBSRichViewEdit.DataField	172
TDBSRichViewEdit.DataSource	172
TDBSRichViewEdit.ExternalRV	173
TDBSRichViewEdit.FieldFormat	173
TDBSRichViewEdit.IgnoreEscape	173
TDBSRichViewEdit.RichViewEdit	173
TSRVPageScroll	173
Properties	174
TSRVPageScroll.AlignText	175
TSRVPageScroll.AutoScrolled	176
TSRVPageScroll.AutoUpdate	176
TSRVPageScroll.BackgroundColor	176
TSRVPageScroll.CachedPagesCount	177
TSRVPageScroll.CacheScrollLimit	177
TSRVPageScroll.ChangeDelay	177
TSRVPageScroll.Columns	177
TSRVPageScroll.MarginHorizontal	178
TSRVPageScroll.MarginVertical	178
TSRVPageScroll.PageBorderColor	178
TSRVPageScroll.PageBreakHeight	178
TSRVPageScroll.PageBreakWidth	179
TSRVPageScroll.PageHeight	179
TSRVPageScroll.PageNoFont	179
TSRVPageScroll.PageNoVisible	180
TSRVPageScroll.PageProportions	180
TSRVPageScroll.PageWidth	180
TSRVPageScroll.ScrollType	180
TSRVPageScroll.SelectColor	181
TSRVPageScroll.ShadowColor	181
TSRVPageScroll.ShadowWidth	181
TSRVPageScroll.SmoothScroll	181
TSRVPageScroll.SmoothThumbnails	182
TSRVPageScroll.SRichViewEdit	182
Methods	182
TSRVPageScroll.BeginUpdate	182
TSRVPageScroll.EndUpdate	182

TSRVPageScroll.First	183
TSRVPageScroll.GetPageAt.....	183
TSRVPageScroll.Last	183
TSRVPageScroll.Next	183
TSRVPageScroll.Prev	184
TSRVPageScroll.ScrollToPage.....	184
TSRVPageScroll.UpdateCache.....	184
Events	184
TSRVPageScroll.OnDrawPageBorder.....	185
TSRVPageScroll.OnDrawPageNumber.....	185
TSRVPageScroll.OnDrawSelectedPage.....	186
TSclRVRuleR	186
Properties	186
TSclRVRuleR.SRichViewEdit.....	187
Methods	187
TSclRVRuleR.CheckMargins	187
TSclRVRuleR.GetPageProperties.....	187
TSclRVRuleR.Scrolled	188
TSclRVRuleR.SetPageProperties	188
Events	188
TSclRVRuleR.OnAfterMarginChanged	188
TSRVToolBar	188
Properties	189
TSRVToolBar.AlignButton.....	190
TSRVToolBar.AlignText	190
TSRVToolBar.AutoSize	191
TSRVToolBar.BorderWidth.....	191
TSRVToolBar.ButtonCol	191
TSRVToolBar.ButtonHeight.....	191
TSRVToolBar.Buttons	191
TSRVToolBar.ButtonWidth.....	192
TSRVToolBar.Color	192
TSRVToolBar.ColorDisable.....	192
TSRVToolBar.ColorDown.....	192
TSRVToolBar.ColorSelect.....	193
TSRVToolBar.HintCancel.....	193
TSRVToolBar.HintFont	193
TSRVToolBar.HintHeight.....	193
TSRVToolBar.ImageList	194
TSRVToolBar.Indent	194
TSRVToolBar.PenFrame	194
TSRVToolBar.ScaleImagesForDPI.....	194
TSRVToolBar.Spacer	195
Events	195
TSRVToolBar.OnClickButton.....	195
TSRVToolBar.OnDrawButtonBackground.....	195
TSRVToolBar.OnEnter	195
TSRVToolBar.OnEnterButton.....	196
TSRVToolBar.OnLeave	196
TSRVToolWindow	196
Properties	197
TSRVToolWindow.AlignText.....	197
TSRVToolWindow.BorderWidth.....	198
TSRVToolWindow.ButtonCol.....	198
TSRVToolWindow.ButtonHeight.....	198
TSRVToolWindow.Buttons.....	198
TSRVToolWindow.ButtonWidth.....	198
TSRVToolWindow.Color.....	199

TSRVToolWindow.ColorDisable.....	199
TSRVToolWindow.ColorDown.....	199
TSRVToolWindow.ColorSelect.....	200
TSRVToolWindow.HintCancel.....	200
TSRVToolWindow.HintFont.....	200
TSRVToolWindow.HintHeight.....	200
TSRVToolWindow.ImageList.....	201
TSRVToolWindow.PenFrame.....	201
TSRVToolWindow.ScaleImagesForDPI.....	201
TSRVToolWindow.Spacer.....	201
Methods	201
TSRVToolWindow.Execute.....	202
Events	202
TSRVToolWindow.OnClickButton.....	202
TSRVToolWindow.OnEnter	202
TSRVToolWindow.OnEnterButton.....	202
TSRVToolWindow.OnLeave.....	203
TSRVToolWindow.OnSelectEvent.....	203
TSRVPrint	203
Properties	205
TSRVPrint.AutoUpdate	206
TSRVPrint.Center	206
TSRVPrint.ClipMargins	207
TSRVPrint.FrameCountX.....	207
TSRVPrint.FrameCountY.....	208
TSRVPrint.FrameHeightPix.....	208
TSRVPrint.FrameWidthPix.....	209
TSRVPrint.IntegerCount	209
TSRVPrint.MetafileCompatibility.....	210
TSRVPrint.NoMetafiles	210
TSRVPrint.OffXPix	210
TSRVPrint.OffYPix	211
TSRVPrint.Orientation	211
TSRVPrint.PageColCount.....	212
TSRVPrint.PageFormat	212
TSRVPrint.PageHeightPix.....	212
TSRVPrint.PageRowCount.....	213
TSRVPrint.PageWidthPix.....	213
TSRVPrint.PrintMode	213
TSRVPrint.RealHeight	214
TSRVPrint.RealWidth	215
TSRVPrint.ScaleImage	215
TSRVPrint.SRichViewEdit.....	216
TSRVPrint.TotalFrameHPix.....	216
TSRVPrint.TotalFrameWPix.....	216
TSRVPrint.UsePhysicalOffsets.....	216
Methods	217
TSRVPrint.BeginUpdate	217
TSRVPrint.CheckMargin.....	217
TSRVPrint.EndUpdate	218
TSRVPrint.OptimalOrientation.....	218
TSRVPrint.PaperCodeToPageFormat.....	218
TSRVPrint.Print	218
TSRVPrint.PrintFrame	219
TSRVPrint.PrintFrames	219
TSRVPrint.PrintFramesEx.....	220
TSRVPrint.PrintPages	220
TSRVPrint.Update	221

Events	221
TSRVPrint.OnSendingToPrinter.....	221
TSRVPrint.OnUpdateData	221
TSRVPreview	222
Properties	222
TSRVPreview.BorderColor.....	224
TSRVPreview.BorderWidth.....	224
TSRVPreview.FrameBorderColor	224
TSRVPreview.FrameBorderWidth.....	224
TSRVPreview.PageBorderColor.....	225
TSRVPreview.PageBorderStyle.....	225
TSRVPreview.PageBorderWidth.....	225
TSRVPreview.PageNo	225
TSRVPreview.SRVPrint	226
Methods	226
TSRVPreview.First	226
TSRVPreview.Last	226
TSRVPreview.Next	226
TSRVPreview.Prev	226
Events	226
TSRVPreview.OnAfterPaint.....	227
TSRVPreview.OnBeforePaint.....	227
TSRVSkinManager	227
Properties	229
TSRVSkinManager.CurrentSkin.....	229
TSRVSkinManager.SkinIndex.....	229
TSRVSkinManager.Skins.....	229
Classes of properties	229
TSRVSkin	230
Properties	231
TSRVSkin.BoxSchemes.....	231
TSRVSkin.Elements.....	231
TSRVSkin.Fonts	231
TSRVSkin.HorizontalScrollBarSchemes	231
TSRVSkin.HorizontalTabSetSchemes	231
TSRVSkin.Name.....	232
TSRVSkin.VerticalScrollBarSchemes	232
TSRVSkin.VerticalTabSetSchemes	232
TSRVBoxScheme	232
TSRVBoxSchemeCollection	233
TSRVSkinCollection	234
TSRVSkinFont	234
TSRVSkinFontCollection.....	235
TSRVSkinElement	236
TSRVSkinElementCollection.....	237
TSRVScrollBarSkinScheme.....	237
TSRVScrollBarSkinSchemeCollection	239
TSRVTabSetSkinScheme	240
TSRVTabSetSkinSchemeCollection	241
TSRVScrollBar	242
Properties	243
TSRVScrollBar.DecButtonWidth.....	244
TSRVScrollBar.Flat	244
TSRVScrollBar.FocusBlinkDelay.....	244
TSRVScrollBar.IncButtonWidth.....	245
TSRVScrollBar.Kind	245
TSRVScrollBar.PageSize	245
TSRVScrollBar.Position	245

TSRVScrollBar.ScrollingDelay.....	246
TSRVScrollBar.SkinManager	246
TSRVScrollBar.SkinSchemeIndex.....	246
TSRVScrollBar.SmallChange, LargeChange.....	246
TSRVScrollBar.SRVControlStyle.....	247
TSRVScrollBar.UseXPThemes	247
Events	247
TSRVScrollBar.OnDraw.....	248
TSRVTabSet	248
Properties	250
TSRVTabSet.AutoTabWidth.....	251
TSRVTabSet.CanMoveTabs.....	252
TSRVTabSet.CloseButton	252
TSRVTabSet.DarkMode, AutoDarkMode.....	252
TSRVTabSet.FirstTabIndex	252
TSRVTabSet.ImageList	252
TSRVTabSet.Indent	253
TSRVTabSet.Kind	253
TSRVTabSet.LastTabIndex	253
TSRVTabSet.Margins	253
TSRVTabSet.MaxTabSize.....	254
TSRVTabSet.MinTabSize.....	254
TSRVTabSet.MirrorText	254
TSRVTabSet.OppositeTabPosition.....	255
TSRVTabSet.ScrollButtonWidth.....	255
TSRVTabSet.ScrrollerMenu.....	255
TSRVTabSet.ScrrollerPosition.....	255
TSRVTabSet.ScrollingDelay.....	256
TSRVTabSet.SkinManager	256
TSRVTabSet.SkinSchemeIndex.....	256
TSRVTabSet.Spacings	256
TSRVTabSet.SRVControlStyle.....	257
TSRVTabSet.TabHeight	257
TSRVTabSet.TabIndex	257
TSRVTabSet.Tabs	258
TSRVTabSet.TextAlignH	258
TSRVTabSet.TextAlignV	258
Methods	258
TSRVTabSet.ScrollToTab.....	258
Events	259
TSRVTabSet.OnChange	259
TSRVTabSet.OnCloseTab.....	259
TSRVTabSet.OnDraw... ..	259
TSRVTabSet.OnMoveTab.....	260
Classes of properties.....	260
TSRVCloseButton	261
Properties	261
TSRVCloseButton.AllowCloseLastTab.....	262
TSRVCloseButton.Height.....	262
TSRVCloseButton.HorizontalAlign.....	262
TSRVCloseButton.IndentX.....	262
TSRVCloseButton.IndentY.....	262
TSRVCloseButton.VerticalAlign.....	263
TSRVCloseButton.Visibility.....	263
TSRVCloseButton.Visible.....	263
TSRVCloseButton.Width.....	263
TSRVTabItem	263
Properties	264

TSRVTabItem.Enabled.....	264
TSRVTabItem.ImageIndex.....	264
TSRVTabItem.SkinFontIndex.....	264
TSRVTabItem.SkinSchemeIndex.....	265
TSRVTabItem.Tag.....	265
TSRVTabItem.Text.....	265
TSRVTabItem.Visible.....	265
TSRVTabCollection	266
4..Types	266
Custom drawing types	267
TSRVDrawScrollBarAreaEvent type.....	267
TSRVDrawScrollBarEvent type.....	267
TSRVDrawState, TSRVDrawStates types.....	268
TSRVDrawTabEvent type.....	268
TSRVDrawTabSetEvent type	269
TSRVOnPaint type.....	269
ToolBar types	270
TSRVAlignButton type.....	270
TSRVAlignText type.....	270
TSRVButtonCollection.....	270
TSRVClickButtonEvent type.....	271
TSRVTool Button.....	271
Properties	271
TSRVTool Button.AllowAllUp.....	272
TSRVTool Button.Caption.....	272
TSRVTool Button.Enabled.....	272
TSRVTool Button.Down.....	272
TSRVTool Button.Hint.....	272
TSRVTool Button.GroupIndex.....	273
TSRVTool Button.ImageIndexDisable.....	273
TSRVTool Button.ImageIndexDown.....	273
TSRVTool Button.ImageIndexMove.....	273
TSRVTool Button.ImageIndexNormal	274
TSRVTool Button.ShowHint.....	274
TSRVTool Button.Tag.....	274
TSRVTool Button.Visible.....	274
TSRVControlStyle	275
TSRVFloatProperty type	275
TSRVIntProperty type	276
TSRVMouseEvent type	276
TSRVPageFormat type	277
TSRVTHAlign, TSRVTVAlign types	280
5..Global variables	280
Part II ScaleRichView Actions	281
1..Localization	282
2..Basic actions	283
TsrvAction	283
Properties	284
TsrvAction.Control	284
TsrvCustomActionThatNeedsDocProps	284
Properties	285
TsrvCustomActionThatNeedsDocProps.ActionSave.....	285
3..Print, preview, page setup	285
TsrvActionCustomOrientation	285

TsrvActionLineNumbers	286
TsrvActionOrientationLandscape	286
TsrvActionOrientationPortrait	287
TsrvActionPageFormat	288
Properties	288
TsrvActionPageFormat.PageFormat.....	289
TsrvActionPageFormat.PageHeight.....	289
TsrvActionPageFormat.PageWidth.....	289
TsrvActionPageFormat.Units	289
TsrvActionPageSetup	290
Properties	290
TsrvActionPageSetup.PageFormats*	291
Events	292
TsrvActionPageSetup.OnChange.....	292
TsrvActionPreview	292
TsrvActionPrint	292
Properties	293
TsrvActionPrint.Title	293
TsrvActionQuickPrint	293
Properties	294
TsrvActionQuickPrint.Title.....	294
TsrvCustomPrintAction	295
4 View	295
TsrvActionCustomLayout	295
Events	297
TsrvActionCustomLayout.OnExecuted.....	297
TsrvActionCustomZoom	298
TsrvActionLayoutDraft	298
Properties	300
TsrvActionLayoutDraft.HPadding.....	300
TsrvActionLayoutPrint	300
Properties	302
TsrvActionLayoutPrint.Color.....	302
TsrvActionLayoutPrint.FooterVisible.....	302
TsrvActionLayoutPrint.HeaderVisible.....	302
TsrvActionLayoutPrint.HPadding.....	303
TsrvActionLayoutRead	303
Properties	304
TsrvActionLayoutRead.Color.....	305
TsrvActionLayoutRead.HidePageBackgroundImage.....	305
TsrvActionLayoutRead.HPadding.....	305
TsrvActionLayoutRead.PageColor.....	305
TsrvActionLayoutRead.PageFlipEffect.....	305
TsrvActionLayoutSideToSide	306
Properties	307
TsrvActionLayoutSideToSide.Color.....	308
TsrvActionLayoutSideToSide.FooterVisible.....	308
TsrvActionLayoutSideToSide.HeaderVisible.....	308
TsrvActionLayoutSideToSide.HPadding.....	308
TsrvActionLayoutSideToSide.NoPageSpacing.....	308
TsrvActionLayoutSideToSide.PageFlipEffect.....	309
TsrvActionLayoutWeb	309
TsrvActionThumbnails	310
Properties	311
TsrvActionThumbnails.PageScroll.....	311
Events	311
TsrvActionThumbnails.OnExecuted.....	311

TsrvActionZoom	312
Properties	312
TsrvActionZoom.ZoomPercent.....	312
TsrvActionZoomFullPage	313
TsrvActionZoomPageWidth	313
5..Editing	314
TsrvActionEditHeader	314
TsrvActionEditFooter	314
TsrvActionEditMain	315
6..Notes	315
TsrvActionCustomInsertSidenote	316
Properties	316
TsrvActionCustomInsertSidenote.BoxPosition.....	317
TsrvActionCustomInsertSidenote.BoxProperties.....	317
TsrvActionCustomInsertSidenote.StyleTemplateName.....	317
TsrvActionEditNote	318
TsrvActionInsertEndnote	318
TsrvActionInsertFootnote	319
TsrvActionInsertNote	320
Properties	321
TsrvActionInsertNote.Font.....	321
TsrvActionInsertSidenote	321
Properties	322
TsrvActionInsertSidenote.RefStyleTemplateName.....	323
TsrvActionInsertTextBox	323
TsrvActionReturnToNote	324

Part III SRVControls

324

1..Ancestor classes	326
TCustomSRVEdit	326
TCustomSRVGraphicControl	328
TCustomSRVPanel	329
TSRVCanvasControl	330
TSRVCustomComboBox	331
TSRVCustomControl	334
TSRVCustomListBox	335
TSRVEditControl	338
TSRVGraphicControl	338
TSRVWinControl	339
2..Main Controls	341
TSRVButton	342
TSRVCheckBox	344
TSRVComboBox	345
TSRVEdit	348
TSRVGroupBox	351
TSRVImagesScroll	353
TSRVLabel	357
TSRVListBox	358
TSRVMemo	361
TSRVPaintBox	363
TSRVPanel	364
TSRVRadioButton	366
3..Data-Aware Controls	367
TSRVDBCheckBox	368
TSRVDBComboBox	369
TSRVDBEdit	370

TSRVDBListBox	371
TSRVDBMemo	372
TSRVDBText	373
4. Classes of properties	374
TSRVBoxItemCollection, TSRVLBoxItemCollection	374
TSRVImageScrollCollection	374
TSRVBoxItem, TSRVLBoxItem	374
TSRVImageScrollItem	378

Index

380

1 ScaleRichView

Version 11+ (What's new? ²⁰)

Introduction

ScaleRichView is a package of Delphi/C++Builder/Lazarus components.

It uses:

- TRichView Package (www.trichview.com)
- RichViewActions (www.trichview.com/resources/actions/).

TRichView and RichViewActions must be installed in order to use ScaleRichView.

Components



TSVRichViewEdit ⁴⁰ allows editing rich text documents in WYSIWYG mode. It supports scaling and different view modes, including a page-view mode.



TDBSRichViewEdit ¹⁶⁸ is a data-aware version of WYSIWYG editing component.



TSVRuler ¹⁸⁶ is a horizontal or vertical ruler designed to work with TSVRichViewEdit. It is inherited from TRVRuler component included in RichViewActions.



TSVRToolBar ¹⁸⁸ is a simplified analog of TToolBar. Instances of TSVRToolbar are used in TSVRichViewEdit: they allow adding buttons in scrollbar area.



TSVRToolWindow ¹⁹⁶ uses the TSVRToolBar and displays it in a popup window.



TSVRPageScroll ¹⁷³ displays thumbnails of pages for documents prepared in TSVRichViewEdit component.



TSVRPrint ²⁰³ and TSVRPreview ²²² are components for advanced printing of documents prepared in TSVRichViewEdit. For example, they allow printing posters (e.g. A3 page on A4 printer).



TSVRSkinManager ²²⁷ allows applying skins to scrollbars of TSVRichViewEdit ⁴⁰ /TDBSRichViewEdit ¹⁶⁸, to TSVRScrollBar ²⁴², to TSVRTabSet ²⁴⁸.

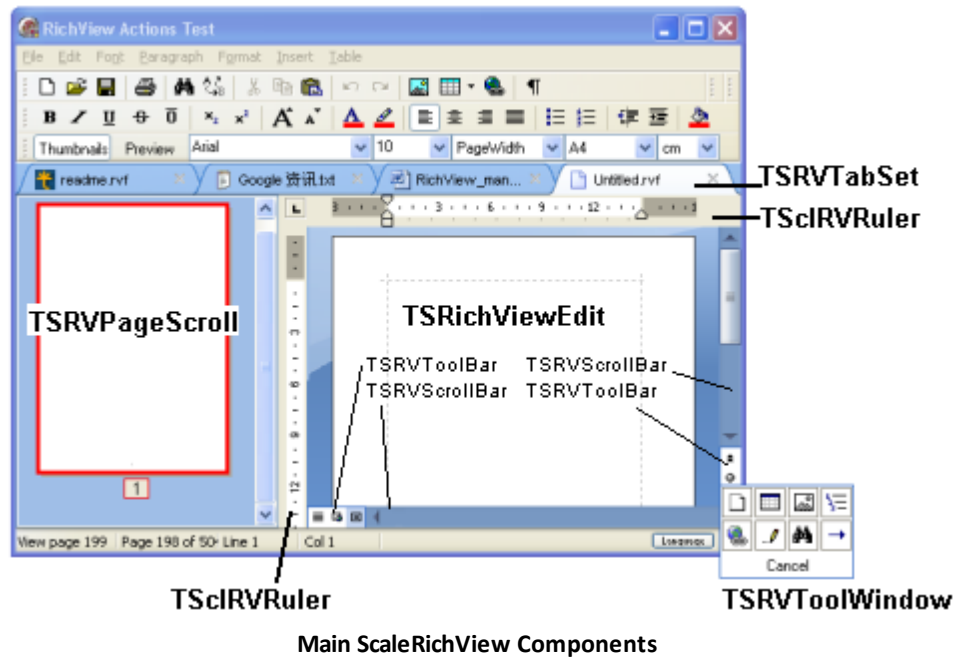


TSVRTabSet ²⁴⁸ – advanced skinnable tabs, can be used to implement a tabbed multiple document interface.



TSVRScrollBar ²⁴² – skinnable scrollbar. This component is used for scrollbars in TSVRichViewEdit, but may be useful as a separate component as well.

Example



Main ScaleRichView Components

In the screenshot above, you can see:

- `TSRichViewEdit`⁽⁴⁰⁾ component;
- `TSRVPPageScroll`⁽¹⁷³⁾ component;
- two `TScIRVRuler`⁽¹⁸⁶⁾ components;
- a window displayed by `TSRVToolWindow`⁽¹⁹⁶⁾.

You can also see three instances of `TSRVToolBar`⁽¹⁸⁸⁾: two toolbars embedded in a scrollbar area of `TSRichViewEdit`⁽⁴⁰⁾ component, and a toolbar inside a `TSRVToolWindow`⁽¹⁹⁶⁾'s window. These three toolbars are not separate components, they are owned by `TSRichViewEdit`⁽⁴⁰⁾ and `TSRVToolWindow`⁽¹⁹⁶⁾, and their properties are defined by their owners.

1.1 Version history

Version History

- New after v12.0⁽²⁰⁾ (dark mode, GDI+ support in Lazarus)
- New in v12.0⁽²⁰⁾ (VCL themes, LiveBindings, HTML)
- New in v11.0⁽²¹⁾ (improved zooming, share properties, end-user help)
- New in v10.0⁽²²⁾ (smooth scrolling, reading mode, improved bi-di, data-aware SRVControls)
- New in v9.0⁽²³⁾ (Lazarus, better zooming and page positioning, high DPI, major rewrite of SRVControls)
- New in v8.0⁽²⁵⁾ (printing options, system caret)
- New in v7.0⁽²⁶⁾ (Uniscribe, transparency, line numbers, setup)
- New in v6.0⁽²⁷⁾ (text boxes and sidenotes, smooth image scaling, touch screen support)
- New in v5.0⁽²⁸⁾ (style templates, zooming panel, animations, "smart popups")
- New in v4.0⁽³⁰⁾ (footnotes and endnotes, new actions, new printing modes)
- New in v3.0⁽³²⁾ (header and footer, printing actions, new SRVControls)
- New in v2.0⁽³⁴⁾ (new printing modes)

1.1.1 New after v12.0

Compatibility issues

Changes in packages for Lazarus: designtime and db- packages are removed (their units are moved to main packages). Only two packages left: srvpkg laz.lpk and srvcontrols laz.lpk ("runtime + designtime" packages)

Delphi and C++Builder

Delphi and C++Builder 12 are supported

Support of "Windows 64-bit (Modern)" platform in C++Builder 12+.

Lazarus

SRVControls can use GDI+ drawing in Lazarus. See GDI+ notes ⁽³²⁶⁾.

OnUTF8KeyPress event is supported.

Appearance

New property to implement a "dark mode": `TSRichViewEdit.ViewProperty(68).DarkMode(156)`, `DarkModeUI(156)`.

You can define appearance of scrollbars in `TSRichViewEdit` using `ScrollBarControlStyle(69)` property.

All `SRVControls` have `DarkMode` and `AutoDarkMode` properties too. By default, their `AutoDarkMode` = `True`, so, when inserted in an editor (`TRichView` or `TSRichViewEdit`), they are drawn according to the editor's `DarkMode`. Otherwise, their own `DarkMode` property is used.

1.1.2 New in v12.0

Compatibility issues

The default value of `TSRichViewEdit.Color` is changed from `$0099A8AC` to `clBtnShadow`.

The type of `Sender` parameter in `OnSaveCustomFormat(119)` and `OnLoadCustomFormat(111)` events is changed from `TDBSRichViewEdit(168)` to `TSRichViewEdit(40)`.

Printing metafiles as bitmaps

New optional parameter `NoMetafiles` in `TSRichViewEdit.DrawPage(76)`, `DrawMetafile(76)`, `PrintAll(92)`, `PrintRange(93)`, `PrintCurrent(93)`.

New property: `TSRVPrint.NoMetafiles(210)`.

VCL themes

`TSRVScrollBar(242)` supports VCL themes (in Delphi XE2 and newer). Improved VCL themes support in `TSRichViewEdit`.

DataBindings

A new `TSRichViewEdit.Document` ⁽⁵¹⁾ property can be linked to a database field (`TBlobField`) using `LiveBindings` (in Delphi XE2 and newer). The following events are moved from `TDBSRichViewEdit` ⁽¹⁶⁸⁾ to `TSRichViewEdit` ⁽⁴⁰⁾ and work for `LiveBindings` as well: `OnSaveCustomFormat` ⁽¹¹⁹⁾, `OnLoadCustomFormat` ⁽¹¹¹⁾, `OnNewDocument` ⁽¹¹²⁾, `OnLoadDocument` ⁽¹¹¹⁾.

`SRVControls` support `DataBindings` as well: `TSRVEdit` ⁽³⁴⁸⁾ and `TSRVCheckBox` ⁽³⁴⁴⁾ as live binding editors; `TSRVLabel` ⁽³⁵⁷⁾, `TSRVButton` ⁽³⁴²⁾, `TSRVPanel` ⁽³⁶⁴⁾ as sources.

HTML

`ScaleRichView` supports HTML loading, including storing documents as HTML in databases by `TDBSRichViewEdit` ⁽¹⁶⁸⁾.

1.1.3 New in v11.0

Compatibility issues

`TSRichViewEdit.SelectAll` ⁽⁹⁶⁾ now selects in `ActiveEditor` ⁽⁴⁷⁾ instead of `RichViewEdit` ⁽⁵⁹⁾.

The following properties of `TSRichViewEdit` were removed: `ExternalRV`, `ExternalRVH`, `ExternalRVF`, `ExternalRVN`. `External TRVStyle` components still can be used.

RAD Studio 11 Alexandria

Delphi and C++Builder 11 Alexandria are supported.

Improved zooming

If `TSRichViewEdit.PageProperty` ⁽⁵⁸⁾.`AutoWidth` ⁽¹³⁵⁾ = `True`, `TSRichViewEdit.ViewProperty` ⁽⁶⁸⁾.`ZoomPercent` ⁽¹⁶⁷⁾ is applied to document content (not to pages). In this mode, to apply changed `ZoomPercent`, the editor needs to reformat the whole document. Because of this, for large documents, fast multiple changes of `ZoomPercent` may be a problem.

Since this version, when zooming is initiated by a mouse wheel ⁽¹⁶¹⁾ or a track bar on a zooming panel ⁽¹⁶⁷⁾, the editor enters in a special zooming mode. The zooming percent is displayed using `TSRichViewEdit.ViewProperty` ⁽⁶⁸⁾.`ZoomFont` ⁽¹⁶⁵⁾. `ZoomPercent` is assigned when the user finishes using a mouse wheel / trackbar.

Shared TRVStyle properties

`TRVStyle` of internal `TRichViewEdit` objects in `TSRichViewEdit` component now share all properties (except for collections of text, paragraph, and list styles). If you change a property of one of these `TRVStyle` objects, you change this property in all of them.

For example, if you assign `SRichViewEdit1.RichViewEdit` ⁽⁵⁹⁾.`Style.SelOpacity`, this change affects not only the main document editor, but also editors of headers, footers, and notes.

End-user help

`HelpContext` and `HelpKeyword` are assigned to a page setup dialog ⁽²⁹⁰⁾.

1.1.4 New in v10.0

This topic includes changes made in v10.1.

UI

New UI translation: Slovenian.

Smooth scrolling

A new optional parameter `SmoothScroll` is added to the following `TSRichViewEdit` methods:

- `ScrollToItem`⁽⁹⁶⁾, `ScrollToCaret`, `ScrollCaretToCenter`⁽⁹⁵⁾,
- `StartEditNote`⁽⁹⁹⁾, `ReturnToNote`⁽⁹⁵⁾,
- `FirstCurPage`⁽⁸²⁾, `NextCurPage`⁽⁹¹⁾, `PriorCurPage`⁽⁹⁵⁾, `LastCurPage`⁽⁹⁰⁾.

It is *False* by default.

Since this version, the actions use smooth scrolling:

- when scrolling to a checkpoint,
- when editing a note or a text box (switching between a note document and a note itself)
- when activating headers/footers.

New property of `TSRVPageScroll`⁽¹⁷³⁾: `SmoothScroll`⁽¹⁸¹⁾.

Read mode


New property of `TSRichViewEdit`⁽⁴⁰⁾: `ReadModeProperty`⁽⁵⁹⁾. It allows displaying document in a way convenient for reading.

Features:

- always displaying full pages
- auto page size or auto zooming
- page flipping effect
- options to overriding page color and for hiding page background image

New actions for switching to a read mode:

 `TsrvActionLayoutSideToSide`⁽³⁰⁶⁾ (auto page zooming)

 `TsrvActionLayoutRead`⁽³⁰³⁾ (auto page size)

Improved BiDiMode support


If `TSRichViewEdit`⁽⁴⁰⁾.`BiDiMode`⁽⁴⁸⁾ = *rvbdRightToLeft*, the following changes are applied (in addition to placing a vertical scrollbar at the left side):






- when pages are arranged in columns, pages in each row are ordered from right to left
- line selection with the mouse works on the right page margin

SRVControls

`SRVControls` are ported to Lazarus, Windows platform (except for `TSRVMediaPlayer`).

New data-aware versions of `SRVControls`⁽³²⁴⁾:

 `TSRVDBCheckBox`⁽³⁶⁸⁾ – data-aware check box;

-  TSRVDBComboBox⁽³⁴⁵⁾ – data-aware combo box;
-  TSRVDBEdit⁽³⁷⁰⁾ – data-aware plain-text one-line editor;
-  TSRVDBListBox⁽³⁷¹⁾ – data-aware scrollable list of items;
-  TSRVDBMemo⁽³⁷²⁾ – data-aware plain-text multi-line editor;
-  TSRVDBText⁽³⁷³⁾ – data-aware text label.

New property for TSRVListBox⁽³⁵⁸⁾ and TSRVComboBox⁽³⁴⁵⁾: UseSelectedImagesInHotState. It specifies which image indexes are used for highlighted items.

Other changes

New events and properties

In TSRichViewEdit⁽⁴⁰⁾:

- OnTextFound⁽¹²¹⁾ event
- MarkdownProperties⁽⁵⁴⁾ property
- MinPrintedOffInItem⁽⁵⁶⁾ and MaxPrintedOffInItem⁽⁵⁴⁾ properties allow defining a range of items to print with higher precision

In TSRVToolbar:

- OnDrawButtonBackground⁽¹⁹⁵⁾ event

1.1.5 New in v9.0

Compatibility issues

Changes in ScaleRichView

The following properties were moved:

- AlignPageH⁽¹⁴⁴⁾, AlignPageV⁽¹⁴⁴⁾ from TSRichViewEdit.ViewProperty⁽⁶⁸⁾ to PagePosProperty⁽⁵⁸⁾
- MaxPageColCount⁽¹⁴⁶⁾ from TSRichViewEdit.ViewProperty⁽⁶⁸⁾ to PagePosProperty⁽⁵⁸⁾

The following properties were removed:

- from TSRichViewEdit.PageProperty⁽⁵⁸⁾: BoundLeftRight (replaced by PagePosProperty⁽⁵⁸⁾.HPadding⁽¹⁴⁵⁾), PageBreakHeight, MinPageBreakHeight, MaxPageBreakHeight (replaced by PagePosProperty⁽⁵⁸⁾.PageVSpacing⁽¹⁴⁶⁾)
- from TsrvActionLayoutDraft⁽²⁹⁸⁾: BoundLeftRight (replaced by HPadding⁽³⁰⁰⁾)
- from TsrvActionLayoutPrint⁽³⁰⁰⁾: BoundLeftRight (replaced by HPadding⁽³⁰³⁾)

Changes in SRVControls

SRVControls⁽³²⁴⁾ and TSRVTabSet⁽²⁴⁸⁾ have a new default appearance (see SRVControlStyle properties)

In TSRVTabSet⁽²⁴⁸⁾, TextAlignH⁽²⁵⁸⁾ and TextAlignV⁽²⁵⁸⁾ do not affect icons positions anymore; they depend only on BiDiMode and MirrorText⁽²⁵⁴⁾.

In TSVREdit⁽³⁴⁸⁾ and TSRVComboBox⁽³⁴⁵⁾, types of parameters are changed in OnEditHints (Text and EditHints parameters) and OnCloseHints (NewText parameter).

New development environments

- RAD Studio 10.3 compatibility
- Lazarus compatibility (Windows 32 and Windows 64 platforms)

Zooming, high DPI

ScaleRichView does not assign the global variable `RichViewPixelsPerInch = 96` anymore. Moreover, it's recommended to leave `RichViewPixelsPerInch` at the default state, to allow zooming relative to the actual screen DPI.

Zooming is changed: previously, pages were scaled relative to 96 DPI. Since this version, pages are scaled relative to the screen DPI. Since this version, `*100Pix` properties do not necessary correspond to 100% zoom: now they correspond to 100% zoom in 96 DPI.

Zooming panel appearance is changed in the mode `TSRichViewEdit(40).ViewProperty(68).IconStyle(159) = srvisFlat.`

`TSRichViewEdit` supports high-DPI display modes, "per monitor" and "per monitor v2" modes (if they are supported by an application).

You can specify whether toolbars should scale image according to the screen DPI: new `ScaleImagesForDPI` property is added to `TSRVToolbar(188)`, `TSRVToolWindow(196)`, `TSRichViewEdit(40).MenuHorizontal(55)` and `TSRichViewEdit.MenuVertical(56)`.

New demo projects for RAD Studio 10.3 are added: `Delphi\ActionTestTabs_MultiRes` and `CBuilderUnicode\ActionTestTabs_MultiRes`. They use virtual image lists (containing 16x16, 32x32 and selected 64x64 images) and support "per monitor v2" mode.

Page positioning

New `TSRichViewEdit` property is added: `PagePosProperty(58)`: `TSRVPagePositionProperty(143)`. It contains sub-properties controlling arrangement of pages in the editor.

SRVControls

All additional components (that can be inserted in `TSRichViewEdit` or used on a form) are updated: `TSRVTabSet(248)`, `TSRVScrollBar(242)`, and all `SRVControls(324)`.

Main improvements:

- the components support high-dpi screen modes;
- the components support bi-directional texts (`BiDiMode` property is published and completely supported);
- all text properties are Unicode (`TRVUnicodeString`) in controls in all versions of Delphi;
- new default appearance: modern flat look, smooth antialiased lines (antialiasing requires Delphi XE2+);
- old appearance can be returned using `SRVControlStyle` property (the only control that has the old appearance by default is `TSRVScrollBar(242)`); but the old appearance is improved as well;
- `TSRVMemo(361)` and `TSRVEdit(348)` are rewritten completely; since this version, they are not based on standard memo and edit controls, they are implemented from scratch. `TSRVMemo(361)` supports scroll bars;

- SkinFontIndex property is added to TSRVEdit⁽³⁴⁸⁾, TSRVMemo⁽³⁶¹⁾, TSRVButton⁽³⁴²⁾, TSRVCheckBox⁽³⁴⁴⁾, TSRVRadioButton⁽³⁶⁶⁾, TSRVPanel⁽³⁶⁴⁾, TSRVGroupBox⁽³⁵¹⁾ to allow using fonts⁽²³¹⁾ defined in TSRVSkinManager⁽²²⁷⁾ component.

New property for TSRVTabSet⁽²⁴⁸⁾: OppositeTabPosition⁽²⁵⁵⁾

Other changes

- Type of string properties is changed to TRVUnicodeString.
- TSRVPrint⁽²⁰³⁾.PrintMode⁽²¹³⁾ = *srvpTiles* works differently if ScaleRichView document does not fit the paper sheet. Now it scales down the page to fit paper, and tries to print multiple copies of this scaled down page.
- new method TSRichViewEdit⁽⁴⁰⁾.ConvertToEMU⁽⁷⁴⁾

1.1.6 New in v8.0

This topic includes changes added in v8.1.

Compatibility issues

SRVItemResizeMode global variable is removed.

Since this version, colors of VCL themes are not used for page backgrounds only if TSRichViewEdit⁽⁴⁰⁾.ViewProperty⁽⁶⁸⁾.UseVCLThemes⁽¹⁶⁴⁾.

Themes of Windows and VCL do not affect printing any more.

The default value TSRichViewEdit⁽⁴⁰⁾.PageProperty⁽⁵⁸⁾.CaretPen⁽¹³⁵⁾.Width is changed from 2 to 0.

Printing

New parameters are added in TSRVPrint⁽²⁰³⁾.Print⁽²¹⁸⁾ and PrintPages⁽²²⁰⁾ methods. They allow printing odd/even pages, and printing in reverse order.

Themes of Windows and VCL do not affect printing any more (*clWindowText* is always printed black, *clWindow* is always printed white, etc.)

Editor appearance

New properties of TSRichViewEdit⁽⁴⁰⁾.ViewProperty⁽⁶⁸⁾:

- ZoomPanelUsesGradient⁽¹⁶⁷⁾.
- UseVCLThemes⁽¹⁶⁴⁾

Drag and drop

New properties of TSRichViewEdit: AcceptDragDropFormats, AcceptPasteFormats⁽⁴⁷⁾.

New events of TSRichViewEdit: OnBeforeOleDrop, OnAfterOleDrop⁽¹⁰³⁾.

TSRVPageScroll⁽¹⁷³⁾

More efficient repainting: new ChangeDelay⁽¹⁷⁷⁾ property. Ability to display smoothed thumbnails of page images, SmoothThumbnails⁽¹⁸²⁾.

Caret

Since this version, the editor uses the system caret width by default. You can change the caret thickness by assigning a positive value to `TSRichViewEdit(40).PageProperty(58).CaretPen(135).Width`.

A wide caret is not centered at the insertion position any more, it is positioned according to the global `RichViewEditCaretPosition` variable (RVERVData unit); by default, the caret is positioned at the right side of the insertion position for left-to-right items, and to the left side if right-to-left items.

Corners of a wide caret are not rounded any more.

Other changes

Packages are separated into runtime and designtime packages.

A new parameter `AllowFloating` is added to `TSRichViewEdit`'s methods `GetItemCoords100(85)`, `GetItemBounds`, `GetItemBounds100(84)`.

1.1.7 New in v7.0

Compatibility issues

- A zooming panel⁽¹⁶⁷⁾ is always drawn using the parent `TSRichViewEdit`'s `BackgroundProperty(48)`. It does not use `TSRichViewEdit.Color` any more.
- Previously, the default values of `TSRichViewEdit`'s `MenuVertical(56).ButtonHeight(128)` and `MenuHorizontal(55).ButtonWidth(128)` were ignored, a size of buttons was autocalculated. In this update, autocalculation is removed.
- A new parameters for `TSRichViewEdit.DrawPage(76)`: `UseWordPainters`, `ForMetafile`.
- The default value of `TSRichViewEdit`'s `MenuVertical(56).ButtonHeight(128)` is changed from 26 to 20.

Uniscribe

Since this version, `TSRichViewEdit` draws and prints text using Uniscribe. You can switch back to Windows API using `TextEngine(67)` property.

Printing output may be incompatible with metafiles (so it may cause problems to some virtual printers). A metafile-compatible mode can be set by:

- new optional parameter of `TSRichViewEdit.PrintAll(92)`, `PrintCurrent(93)`, `PrintRange(93)`, `DrawPage(76)` methods;
- `TSRVPrint.MetafileCompatibility(210)` property
- `TRVAControlPanel.MetafileCompatibility` property (used in `TsrvActionPrint(292)` and `TsrvActionQuickPrint(293)`).

Transparency

Since this version, `TSRichViewEdit` supports transparent objects, including pictures and semitransparent backgrounds of paragraphs, tables and text boxes.

Transparency is supported both on the screen and when printing. The most of printers do not support transparency, so TSRichViewEdit draws semitransparent areas in bitmaps and prints this bitmap.

Previously, TSRichViewEdit only were able to draw transparent pictures on the screen.

Line numbers

TSRichViewEdit can show line numbers. New property: LineNumberProperty⁽⁵⁴⁾.

New action:

 TsrvActionLineNumbers⁽²⁸⁶⁾ shows or hides line numbers.

Installation and directory structure

Starting from this update, ScaleRichView is installed automatically in Delphi and C++Builder IDE.

The new installer installs the components in Delphi and C++Builder, both for 32-bit and 64-bit platforms (if available). The installer adds all necessary paths to RAD Studio library.

Source code is moved to "Source" folder, inc-file is moved to "Source\Include" folder. Demo projects that use SRVControls are moved to "Demos" folder of SRVControls.

This help file is integrated in RAD Studio IDE (for XE8+)

1.1.8 New in v6.0

Compatibility issues

- Since this version, TSRichViewEdit must be formatted by its method Format⁽⁸²⁾. You cannot format it by formatting RichViewEdit⁽⁵⁹⁾, RVHeader⁽⁶³⁾ and RVFooter⁽⁶¹⁾, or by calling SetRVMargins⁽⁹⁸⁾.
- Headers and footers are stored in Subdocuments⁽⁶⁶⁾. RVHeader⁽⁶³⁾ and RVFooter⁽⁶¹⁾ are used only for editing Subdocuments.
- It's not necessary to call UpdateNote or UpdateNoteEx after inserting a note or a text box.
- The following properties of TSRVViewProperty⁽¹⁵³⁾ are removed: HeaderTitle, FooterTitle, MainTitle. They are replaced with Texts⁽¹⁶²⁾.
- When adding a page break between table rows, TSRichViewEdit⁽⁴⁰⁾ reserves space for table borders (previously, they might be truncated); so documents containing tables may be paginated differently.
- A different font is used in a zooming panel inside TSRichViewEdit:
TSRichViewEdit.ViewProperty⁽⁶⁸⁾.ZoomPanelFont⁽¹⁶⁶⁾ (TSRichViewEdit.Font was used in previous versions)

Changes

TSRichViewEdit allows editing sidenotes (notes in a floating box) and text box items.

Different headers and footers for the first page, for odd and even pages. Headers and footers are stored in Subdocuments⁽⁶⁶⁾. RVHeader⁽⁶³⁾ and RVFooter⁽⁶¹⁾ are used only for editing Subdocuments.

Smooth image scaling. In addition to images inside documents, TSVRichViewEdit.BackgroundImage⁽⁴⁸⁾.Picture⁽¹²⁵⁾ is scaled when displayed stretched.

rvoHideReadOnlyCaret option in TSVRichViewEdit.RVEditorOptions⁽⁶¹⁾ is supported.

Accurate drawing of table borders in places where a table is split across pages.

New properties of TSVRichViewEdit.ViewProperty⁽⁶⁸⁾: ZoomPanelFont⁽¹⁶⁶⁾ and IconStyle⁽¹⁵⁹⁾.

New methods of TSVRichViewEdit: SetIntPropertyEd, SetFloatPropertyEd⁽⁹⁶⁾, Reformat⁽⁸²⁾.

New properties of TSVRichViewEdit: CanUpdatePosition⁽⁴⁸⁾, CurrentNoteParentEditor⁽⁵⁰⁾.

New events of TSVRichViewEdit: OnCaretMoving⁽¹⁰⁴⁾, OnResizing⁽¹¹⁷⁾

Touch screen support: panning in all visual components, two finger zooming and selection handles in TSVRichViewEdit.

Changes in actions

TsvrActionPageSetup⁽²⁹⁰⁾, TsvrActionPageFormat⁽²⁸⁸⁾, TsvrActionCustomOrientation⁽²⁸⁵⁾ now applies changes as editing operations (undoable).

New actions:

- TsvrActionInsertSidenote⁽³²¹⁾
- TsvrActionInsertTextBox⁽³²³⁾

TsvrActionEditNote⁽³¹⁸⁾ and TsvrActionReturnToNote⁽³²⁴⁾ support sidenotes and text box items.

1.1.9 New in v5.0

Compatibility issues

TSVRichViewEdit⁽⁴⁰⁾.CurControl⁽⁵⁰⁾ now works according to the documentation: this is a focused control inside TSVRichViewEdit⁽⁴⁰⁾. Previously, this property may be changed to the control at the position of the caret. Use the new event OnChangeCurrentControl⁽¹⁰⁴⁾ to detect when this property is changed.

TSVRichViewEdit.ScrollToItem⁽⁹⁶⁾ has new parameters. They are optional, but for old versions of C++Builder they must be specified.

A default value for TSVRVToolBar.Color⁽¹⁹²⁾ and TSVRVToolWindow.Color⁽¹⁹⁹⁾ is changed (from *clWhite* to *clWindow*)

In the new version, pages can be arranged in columns. To return the old layout (one page in a row), you can assign TSVRichViewEdit.ViewProperty⁽⁶⁸⁾.MaxPageColCount⁽¹⁴⁶⁾ = 1.

TSVRichViewEdit.ViewProperty.ZoomPercent⁽¹⁶⁷⁾ can be changed only in the range TSVRichViewEdit.ViewProperty.ZoomMin..ZoomMax⁽¹⁶⁵⁾.

The following code is not needed and must be removed from existing projects:

```
RVA_GetRichViewEditFromPopupComponent :=
```



```
SRVGetRichViewEditFromPopupComponent;  
RVA_GetRichViewEdit := SRVGetRichViewEdit
```

Style templates

New properties of T^{SRichViewEdit}: UseStyleTemplates⁶⁸ and StyleTemplateInsertMode⁶⁵.

If UseStyleTemplates⁶⁸=True, T^{srvActionInsertFootnote}³¹⁹ formats note characters using 'footnote reference' (or, if it does not exist, make it superscript like before), footnote text using 'footnote text' (or, if it does not exist, uses T^{srvActionInsertFootnote}.Font³²¹ like before).

T^{srvActionInsertEndnote}³¹⁸ uses 'endnote reference' and 'endnote text' styles.

Simplification

You can assign T^{SRichViewEdit} component to Control property of actions inherited from TrvAction (previously, only T^{RichViewEdit} could be assigned). This assignment can be made at a design time.

Font combo-boxes (TRVFontComboBox, TRVFontSizeComboBox, TRVFontCharsetComboBox) do not require additional code any more. They work automatically, if T^{SRichViewEdit} control is assigned to their Editor property (TrvActionFontEx action must be assigned to their ActionFont property as well).

You do not need to call the following code any more:

- RVA_GetRichViewEditFromPopupComponent := SRVGetRichViewEditFromPopupComponent;
- RVA_GetRichViewEdit := SRVGetRichViewEdit;
- S^{RichViewEdit1}.SetRVMargins in TRVAControlPanel.OnMarginsChanged

Animation

T^{SRichViewEdit}⁴⁰ can play animated (gif) images. Animation is played in ActiveEditor⁴⁷. New AnimationMode⁴⁷ property.

Zooming and page arrangement

T^{SRichViewEdit}⁴⁰ can display a zooming panel, if ViewProperty⁶⁸.ZoomPanelVisible¹⁶⁷=True.

In the new version, pages can be arranged in columns. T^{SRichViewEdit}.ViewProperty⁶⁸.MaxPageColCount¹⁴⁶ specifies the maximal count of columns.

Page layout

T^{SRichViewEdit}.PageProperty⁵⁸.MirrorMargins¹³⁸ allow exchanging left and right margins on even pages.

"Smart popups"

"Smart popups" are buttons that can be displayed for the current (at the position of the caret) item. New SmartPopupProperties⁶⁵, SmartPopupVisible⁶⁵ property and OnSmartPopupClick¹²⁰ event.

Other changes

New optional parameters for T^{SRichViewEdit}.ScrollToItem⁹⁶. New ScrollCaretToCenter⁹⁵ method.

New AlignButton¹⁹⁰ and Indent¹⁹⁴ properties of T^{SRVToolBar}¹⁸⁸.

New `TSRichViewEdit.OnPrinting` ⁽¹¹⁶⁾ event.

1.1.10 New in v4.0

Compatibility issues

- The type of Sender parameter was changed from `TSRichViewEdit` to `TCustomRichViewEdit` for `TSRichViewEdit.OnStyleConversion` ⁽¹²⁰⁾ and `OnParaStyleConversion` ⁽¹¹⁶⁾ events.
- Two new parameters are added to `TSRichViewEdit.OnPaintPage` ⁽¹¹⁵⁾: `Prepaint` and `Printing`.
- A new parameter is added to `TSRichViewEdit.OnPaint` ⁽¹¹⁴⁾: `PaintRect`.
- The user can zoom the editor in/out using **Ctrl** + mouse wheel. So, if you assume that the editor has the fixed values of `TSRichViewEdit.ViewProperty` ⁽⁶⁸⁾.`ZoomPercent` ⁽¹⁶⁷⁾ and `ViewProperty` ⁽⁶⁸⁾.`ZoomMode` ⁽¹⁶⁵⁾, assign `ViewProperty` ⁽⁶⁸⁾.`MouseWheelZoom` ⁽¹⁶¹⁾ = `False`.
- `TSRichViewEdit.ViewProperty` ⁽⁶⁸⁾.`HeaderFooterShade` is renamed to `ViewProperty` ⁽⁶⁸⁾.`MainDocumentShade` ⁽¹⁵⁹⁾. `ViewProperty` ⁽⁶⁸⁾.`HeaderFooterShade` ⁽¹⁵⁷⁾ now has a different meaning.
- `TSRichViewEdit.PageProperty` ⁽⁵⁸⁾.`PageWidth` ⁽¹⁴¹⁾ and `PageHeight` ⁽¹³⁹⁾ do not depend on the document display modes any more. Properties of the same names were added in `TSRichViewEdit.ViewProperty` ⁽⁶⁸⁾.

Units of measurement

All properties and parameters measured in `TSRichViewEdit` ⁽⁴⁰⁾.`UnitsProgram` ⁽⁶⁸⁾ have changed their types from `Extended` to `TRVLength`. At design time, you can see units of measurement for these properties in the Object Inspector.

By default, properties of all `RVStyles` in `TSRichViewEdit` ⁽⁴⁰⁾ are measured in pixels. To convert them to twips, call `TSRichViewEdit` ⁽⁴⁰⁾.`ConvertToTwips` ⁽⁷⁴⁾.

`TsrvActionPageSetup` ⁽²⁹⁰⁾ displays values in `TRVAControlPanel.UnitsDisplay`, not in `TSRichViewEdit` ⁽⁴⁰⁾.`UnitsProgram` ⁽⁶⁸⁾.

To do: a precision of drawing and printing is still up to one screen pixel. A higher precision is planned in newer version.

Incremental printing

`TSRichViewEdit` ⁽⁴⁰⁾ has new properties: `MinPrintedItemNo` ⁽⁵⁶⁾ and `MaxPrintedItemNo` ⁽⁵⁴⁾. If these properties define a subrange of items, only this subrange (and only the pages containing it) is printed.

These properties allow implementing an incremental printing: when the next portion of the document is ready, it can be printed below the previously printed fragment. This feature can be used for printing accounting journals, logs, etc.

New properties and methods

New properties:

- `TSRichViewEdit` ⁽⁴⁰⁾.`BackgroundProperty` ⁽⁴⁸⁾.`GlobalPageBackgroundColor` ⁽¹²⁵⁾ allows to define document background color without saving it in RVF files.

- `TSRichViewEdit`⁽⁴⁰⁾.`ViewProperty`⁽⁶⁸⁾.`MouseWheelZoom`⁽¹⁶¹⁾ allows to turn off a mouse wheel zooming
- `TSRVSkinManager`⁽²²⁷⁾.`Skins`⁽²²⁹⁾[].`BoxSchemes`⁽²³¹⁾ defines schemes for applying skin elements to rectangles (such as items of a list box)
- `TSRichViewEdit`⁽⁴⁰⁾.`ViewProperty`⁽⁶⁸⁾: `MainPen`⁽¹⁶⁰⁾, `MainTitle`, `MainTitleColor`⁽¹⁶⁰⁾, `MainTitleFont`⁽¹⁶⁰⁾ allow drawing borders and label for the main document (like it is done for a header and a footer).
- `TSRichViewEdit`⁽⁴⁰⁾.`ViewProperty`⁽⁶⁸⁾: `PageWidth`⁽¹⁶¹⁾, `PageHeight`⁽¹⁶¹⁾ return page size as it is shown on the screen.

New methods of `TSRichViewEdit`⁽⁴⁰⁾:

- `GetPageNo`⁽⁸⁷⁾ returns the page index for the specified position in the document.
- `GetItemAt`⁽⁸⁴⁾ returns the item at the specified coordinates.
- `Clear`⁽⁷³⁾, `Format`⁽⁸²⁾ (not new but improved) clears and format the editor completely, including header and footer
- `GetFirstVisiblePage`⁽⁸³⁾, `GetLastVisiblePage`⁽⁸⁶⁾ allow to get a range of visible pages (if a free page positioning mode is not used)

New properties of `TSRVScrollBar`:

- `UseXPThemes`⁽²⁴⁷⁾ allows/disallows using Windows themes (visual styles)
- `Flat`⁽²⁴⁴⁾ makes the scrollbar look flat.

Modified methods of `TSRichViewEdit`⁽⁴⁰⁾:

`GetItemPages`⁽⁸⁶⁾, `GetItemBounds`, `GetItemBounds100`⁽⁸⁴⁾, `GetItemCoords100`⁽⁸⁵⁾ support item parts, so they can return more precise information.

Advanced printing in `TSRVPrint`⁽²⁰³⁾

New `PrintMode`⁽²¹³⁾: *srvpGrid*. It allows printing several pages on one paper sheet.

Footnotes and endnotes

`TSRichViewEdit` can display and edit footnotes and endnotes.

New properties related to footnotes and endnotes:

- `RVNote`⁽⁶⁴⁾
- `CurrentNote`⁽⁵⁰⁾

New methods:

- `StartEditNote`⁽⁹⁹⁾,
- `ReturnToNote`⁽⁹⁵⁾,
- `UpdateNote` (is obsolete in version 6.0).

Additional information can be found in overview⁽³⁸⁾.



















Other features

Support of "keep with next" and "keep lines together" properties of paragraph style is restored (it was disabled in v3.0)


New `TSRichViewEdit`⁽⁴⁰⁾.`ViewProperty`⁽⁶⁸⁾.`AutoVScrollBarPosition`⁽¹⁵⁵⁾ property controls the position of vertical scrollbar for right-to-left documents.

Actions

New actions ⁽²⁸¹⁾:

-  TsrvActionPreview ⁽²⁹²⁾ switches editing and preview modes.
-  TsrvActionThumbnails ⁽³¹⁰⁾ shows/hides page thumbnails.
-  TsrvActionLayoutDraft ⁽²⁹⁸⁾ switches the editor to a draft layout.
-  TsrvActionLayoutWeb ⁽³⁰⁹⁾ switches the editor to a web layout.
-  TsrvActionLayoutPrint ⁽³⁰⁰⁾ switches the editor to a print layout.
-  TsrvActionZoom ⁽³¹²⁾ applies the specified zooming percent to the editor.
-  TsrvActionZoomPageWidth ⁽³¹³⁾ zooms the editor to fit the page width in the editor width.
-  TsrvActionZoomFullPage ⁽³¹³⁾ zooms the editor to fit the full page in the editor.
-  TsrvActionOrientationPortrait ⁽²⁸⁷⁾ changes page orientation to "portrait".
-  TsrvActionOrientationLandscape ⁽²⁸⁶⁾ changes page orientation to "landscape".
-  TsrvActionPageFormat ⁽²⁸⁸⁾ changes page size.
-  TsrvActionEditHeader ⁽³¹⁴⁾ starts editing a page header.
-  TsrvActionEditFooter ⁽³¹⁴⁾ starts editing a page footer.
-  TsrvActionEditMain ⁽³¹⁵⁾ starts editing the main document.
-  TsrvActionInsertFootnote ⁽³¹⁹⁾ inserts a footnote.
-  TsrvActionInsertEndnote ⁽³¹⁸⁾ inserts an endnote.
-  TsrvActionEditNote ⁽³¹⁸⁾ starts editing a footnote/endnote.
-  TsrvActionReturnToNote ⁽³²⁴⁾ moves the caret to the parent footnote/endnote.

Changes in actions:

-  TsrvActionPageSetup ⁽²⁹⁰⁾ can change HeaderY ⁽¹³⁷⁾ and FooterY ⁽¹³⁶⁾ properties.
- Localization procedures ⁽²⁸²⁾ for the ScaleRichView actions.

SRVControls

New controls: TSRVListBox ⁽³⁵⁸⁾, TSRVComboBox ⁽³⁴⁵⁾, TSRVImagesScroll ⁽³⁵³⁾.

TSRVEdit ⁽³⁴⁸⁾ can show suggestions on editing.

New help topics about SRVControls ⁽³²⁴⁾.

1.1.11 New in v3.0

Compatibility issues

Related to header and footer

Previously, TSRichViewEdit control contained only one RichViewEdit control (RichViewEdit ⁽⁵⁹⁾). Now, there are three of them: RichViewEdit ⁽⁵⁹⁾, RVHeader ⁽⁶³⁾ and RVFooter ⁽⁶¹⁾. All of them can be edited. All editing operations must be applied to ActiveEditor ⁽⁴⁷⁾, not RichViewEdit ⁽⁵⁹⁾. However, file saving and loading operations still must be called for RichViewEdit ⁽⁵⁹⁾.

Before loading a file, you need to clear not only RichViewEdit, but also header and footer (loading RVF or RTF does it automatically, but loading text does not).

After loading a file, you need to format all editors:

```
SRichViewEdit1.RVHeader (63).Clear;
```

```

SRichViewEdit1.RVFooter(61).Clear;
SRichViewEdit1.RichViewEdit(59).Clear;
SRichViewEdit1.RichViewEdit.LoadRTF(FileName);
SRichViewEdit1.RVHeader.Format;
SRichViewEdit1.RVFooter.Format;
SRichViewEdit1.SetRVMargins(98);

```

(update: in newer version, you can use a simpler code:

```

SRichViewEdit1.Clear(73);
SRichViewEdit1.RichViewEdit.LoadRTF(FileName);
SRichViewEdit1.Format(82);

```

update 2: since version 6, you must use TSRichViewEdit.Clear and TSRichViewEdit.Format)

)

Alternatively, you can hide header and footer using TSRVPageProperty⁽¹³³⁾'s HeaderVisible⁽¹³⁷⁾ and FooterVisible⁽¹³⁶⁾ properties.

The following events have TRichViewEdit control in Sender parameter: OnCurTextStyleChanged⁽¹⁰⁷⁾, OnCurParaStyleChanged⁽¹⁰⁶⁾, OnCaretMove⁽¹⁰⁴⁾ (previously, it was TSRichViewEdit). Now all events of TNotifyEvent type have RichViewEdit in Sender parameter, so you can distinguish if it was called from RichViewEdit⁽⁵⁹⁾, RVHeader⁽⁶³⁾ or RVFooter⁽⁶¹⁾.

In the new version, ConvertRVToSRV⁽⁷³⁾, ConvertSRVToRV⁽⁷³⁾, GetCurrentLineCol⁽⁸³⁾ methods work for ActiveEditor⁽⁴⁷⁾ instead of RichViewEdit⁽⁵⁹⁾.

Formatting

A new text formatting procedure is used, so existing documents may look not exactly like before.

The following properties of TSRVViewProperty⁽¹⁵³⁾ are renamed:

- BoundTextPen → MarginsPen⁽¹⁶¹⁾
- BoundTextVisible → MarginsRectVisible⁽¹⁶¹⁾

New components:



TSRVSkinManager⁽²²⁷⁾



TSRVTabSet⁽²⁴⁸⁾



TSRVScrollBar⁽²⁴²⁾

Header and footer

Two new properties of TSRichViewEdit control return TRichViewEdit controls used for editing/displaying header and footer: RVHeader⁽⁶³⁾, RVFooter⁽⁶¹⁾.

ActiveEditor⁽⁴⁷⁾ returns the currently edited control (see the compatibility issues above). OnChangeActiveEditor⁽¹⁰⁴⁾ event occurs when it changed.




New properties of TSRVPageProperty⁽¹³³⁾:

- HeaderY⁽¹³⁷⁾, FooterY⁽¹³⁶⁾
- HeaderVisible⁽¹³⁷⁾, FooterVisible⁽¹³⁶⁾

New properties of TSRVViewProperty⁽¹⁵³⁾:

- HeaderTitle, FooterTitle
- HeaderPen⁽¹⁵⁸⁾, FooterPen⁽¹⁵⁶⁾
- HeaderTitleFont⁽¹⁵⁸⁾, FooterTitleFont⁽¹⁵⁷⁾
- HeaderTitleColor⁽¹⁵⁸⁾, FooterTitleColor⁽¹⁵⁶⁾

New actions

-  TsrvActionQuickPrint⁽²⁹³⁾
-  TsrvActionPrint⁽²⁹²⁾
-  TsrvActionPageSetup⁽²⁹⁰⁾

Other changes

New properties and methods of TSRichViewEdit⁽⁴⁰⁾:

- GetPageStartItemNo⁽⁸⁸⁾
- SkinManager⁽⁶⁵⁾, HScrollBarSchemeIndex⁽⁵³⁾, VScrollBarSchemeIndex⁽⁶⁹⁾
- new optional parameter in SetRVMargins⁽⁹⁸⁾

New properties of TSRVBackgroundProperty⁽¹²³⁾:

- TransparentColor⁽¹²⁶⁾

Headings

- FindNextHeading⁽⁷⁸⁾/FindPriorHeading⁽⁸⁰⁾ search for headings.
- NextCurHeading⁽⁹⁰⁾/PriorCurHeading⁽⁹⁴⁾ move the caret to the next/previous heading.

1.1.12 New in v2.0

Compatibility issues

In several methods, the first letter is capitalized.

The properties related to margins are renamed. The pattern is "PageLeft → LeftMargin". This renaming includes both properties of TSRichViewEdit⁽⁴⁰⁾ (rarely used outside the component code) and properties of TSRichViewEdit.PageProperty⁽⁵⁸⁾:TSRVPageProperty⁽¹³³⁾:

- PageLeft → LeftMargin⁽¹³⁷⁾
- PageTop → TopMargin⁽¹⁴²⁾
- PageRight → RightMargin⁽¹⁴¹⁾
- PageBottom → BottomMargin⁽¹³⁵⁾

The properties related to page size are renamed. The pattern is "FormatWidth → PageWidth". This renaming includes both properties of TSRichViewEdit (rarely used outside the component code) and properties of TSRichViewEdit.PageProperty:

- FormatWidth → PageWidth⁽¹⁴¹⁾
- FormatHeight → PageHeight⁽¹³⁹⁾

Other TSRichViewEdit.PageProperty properties and methods are renamed:

- BreakPage → PageViewMode⁽¹⁴⁰⁾
- FormatMode → PageFormat⁽¹³⁸⁾
- ConvertPageSizeToFormatMode → ConvertPageSizeToPageFormat⁽¹⁴²⁾

- getPaperSize → ConvertPageFormatToPageSize⁽¹⁴²⁾

All properties, methods and events related to table icons (rectangles displayed at the top left corners of tables) are renamed. The pattern is "HintTable → TableIcon":

properties of TSRichViewEdit:

- getHintTableItem → GetTableIconItem⁽⁸⁹⁾
- getHintTableRVData → GetTableIconRVData⁽⁸⁹⁾
- OnHintTableClick → OnTableIconClick⁽¹²¹⁾

properties of TSRVViewProperty⁽¹⁵³⁾:

- HintTableTimeDelay → TableIconDelay⁽¹⁶²⁾
- HintTablePopupMenu → TableIconPopupMenu⁽¹⁶²⁾
- HintTableVisible → UseTableIcons⁽¹⁶³⁾

In addition, the following properties of TSRVViewProperty are renamed:

- HintPage → ShowScrollHint⁽¹⁶²⁾

The following methods and events of TSRichViewEdit are renamed:

- calculatePosPages → CalculateAllPagePositions⁽⁷²⁾
- calculatePosPagesFor → CalculatePagePosition⁽⁷²⁾
- RectPage → GetPageClientRect⁽⁸⁷⁾
- getPageMousePos → GetPageAt⁽⁸⁶⁾
- OnCurParaStyleChange → OnCurParaStyleChanged⁽¹⁰⁶⁾

The following properties of TSRVBackgroundProperty⁽¹²³⁾ are renamed:

- PictureType → PicturePosition⁽¹²⁶⁾

The following changes were made in TSRVToolButton⁽²⁷¹⁾:

- AlwaysUp property is removed
- AllowAllUp⁽²⁷²⁾ property is added (now the behavior of these buttons is identical to TSpeedButtons)

The following properties and methods of TSRVPrint⁽²⁰³⁾ are renamed:

- FormatMode → PageFormat⁽²¹²⁾
- PaperCodeToFormatPaper → PaperCodeToPageFormat⁽²¹⁸⁾
- UpdatePage → Update⁽²²¹⁾

The following methods of TSclRVRule⁽¹⁸⁶⁾ are renamed:

- getPageProperty → GetPageProperties⁽¹⁸⁷⁾
- setPageProperty → SetPageProperties⁽¹⁸⁸⁾

The following properties of TSRVPreview⁽²²²⁾ are changed:

- PageBorderColor → FrameBorderColor⁽²²⁴⁾
- PageBorderWidth → FrameBorderWidth⁽²²⁴⁾
- PageShadowColor, PageShadowWidth are removed (as well as "shadows" between frames)
- PageBoundsColor → PageBorderColor⁽²²⁵⁾
- PageBoundsWidth → PageBorderWidth⁽²²⁵⁾
- PageBoundsStyle → PageBorderStyle⁽²²⁵⁾
- BorderColor⁽²²⁴⁾, BorderWidth⁽²²⁴⁾, ShadowColor, ShadowWidth are added.

The following properties, methods and events of TSRVPageScroll⁽¹⁷³⁾ are changed:

- WidthBreakPage → PageBreakWidth⁽¹⁷⁹⁾

- HeightBreakPage → PageBreakHeight⁽¹⁷⁸⁾
- ShadowOff → ShadowWidth⁽¹⁸¹⁾
- CountPagesCache → CachedPagesCount⁽¹⁷⁷⁾
- LimitCacheUpdate → CacheScrollLimit⁽¹⁷⁷⁾
- GetPageNoToXY → GetPageAt⁽¹⁸³⁾
- OnDrawSelect → OnDrawSelectedPage⁽¹⁸⁶⁾
- OnDrawPageBorder⁽¹⁸⁵⁾'s parameters are changed
- UpdateCache⁽¹⁸⁴⁾ is made public
- the component highlights the page containing the caret, instead of the first page visible in TSRichViewEdit.

Other changes

Unnecessary and non working methods and events were removed (or made private).

TSRVFormatMode type is renamed to TSRVPageFormat⁽²⁷⁷⁾. TSRVPicture type is renamed to TSRVPicturePosition⁽¹²⁶⁾.

Parameters of TSRichViewEdit.OnGetPagePos⁽¹⁰⁸⁾ event are changed.

The way how TSRVPrint.UsePhysicalOffsets⁽²¹⁶⁾ is processed is changed.

New

New printing modes in TSRVPrint, see PrintMode⁽²¹³⁾ property.

New method TSRichViewEdit.GetItemPages⁽⁸⁶⁾ returns pages containing the specified item.

New method TSRichViewEdit.DeletePage⁽⁷⁵⁾ deletes a page as an editing operation.

New version is TSRVToolWindow.Execute⁽²⁰²⁾ simplifies calling window which is invoked with a button.

Many small tweaks and fixes.

1.2 Overview

- Document parts in TSRichViewEdit⁽³⁶⁾
- Footnotes and endnotes⁽³⁸⁾

1.2.1 Document parts in TSRichViewEdit

Editors

TSRichViewEdit⁽⁴⁰⁾ control contains several TRichViewEdit controls inside. These controls are invisible, they are used to render end edit different document parts. These controls are available as properties of TSRichViewEdit:

- RichViewEdit⁽⁵⁹⁾ for the main document;
- RVHeader⁽⁶³⁾ for page headers;
- RVFooter⁽⁶¹⁾ for page footers;
- RVNote⁽⁶⁴⁾ for notes and text box items.

RichViewEdit⁽⁵⁹⁾ stores the main document. Other editors are used to edit documents stored in different places:

- RVHeader⁽⁶³⁾ and RVFooter⁽⁶¹⁾ edit Subdocuments⁽⁶⁶⁾
- RVNote⁽⁶⁴⁾ edits Document of footnotes, endnotes, sidenotes, text boxes.

RichViewEdit⁽⁵⁹⁾ always contain the main document. Other editors contain valid documents only when they are active (equal to ActiveEditor⁽⁴⁷⁾).

Loading

When you load a new document, call TSRichViewEdit.Format⁽⁸²⁾.

This method copies page properties from TSRichViewEdit.RichViewEdit⁽⁵⁹⁾.DocParameters to TSRichViewEdit.PageProperty⁽⁵⁸⁾ (when loading files, page properties are loaded in RichViewEdit.DocParameters), formats RVHeader⁽⁶³⁾, RVFooter⁽⁶¹⁾, RichViewEdit⁽⁵⁹⁾, notes and text boxes, paginates the document.

Example of loading RTF file:

```
SRichViewEdit1.Clear(73);
SRichViewEdit1.RichViewEdit(59).LoadRTF(FileName);
SRichViewEdit1.Format(82);
```

The same sequence must be called for other loading methods of TSRichViewEdit.RichViewEdit⁽⁵⁹⁾: LoadRTFFromStream, LoadRVF, LoadRVFFromStream, LoadText, etc.

For RVF, you can use the methods of TSRichViewEdit: LoadRVF⁽⁹⁰⁾ and LoadRVFFromStream⁽⁹⁰⁾. These methods load, format and assign all necessary data themselves.

If you do not want to display a header and a footer, assign:

```
SRichViewEdit1.PageProperty(58).HeaderVisible(137) := False;
SRichViewEdit1.PageProperty(58).FooterVisible(136) := False;
```

Switching between document parts


The active TRichViewEdit control (the control corresponding to the document part containing the caret) is returned in the TSRichViewEdit.ActiveEditor⁽⁴⁷⁾ property.


To activate editing of the main document, a header, or a footer, call StartEditing⁽⁹⁸⁾.


To activate editing of a footnote or an endnote, call StartEditNote⁽⁹⁹⁾.

The user can activate editing of the page header/footer by double clicking.

If you use RichViewActions, you can use the following actions to switch between document parts:

 TsrvActionEditHeader⁽³¹⁴⁾ starts editing a page header;

 TsrvActionEditFooter⁽³¹⁴⁾ starts editing a page footer;

 TsrvActionEditMain⁽³¹⁵⁾ to return to the main document.

Then ActiveEditor⁽⁴⁷⁾ is changed, OnChangeActiveEditor⁽¹⁰⁴⁾ event occurs.

Editing

All editing operations must be applied to ActiveEditor⁽⁴⁷⁾, for example:

```
SRichViewEdit1.ActiveEditor.Undo;
```

ChangeStyleTemplates must not be called for ActiveEditor. It must not be called if ActiveEditor=RVNote⁽⁶⁴⁾.

Operations applied to the whole document (such as saving or loading) must be called for the main editor:

```
SRichViewEdit1.RichViewEdit(59).SaveRTF(...);
```

In the events of T^{SRichViewEdit} (such as OnStyleConversion⁽¹²⁰⁾, OnParaStyleConversion⁽¹¹⁶⁾, OnCurTextStyleChanged⁽¹⁰⁷⁾, OnCurParaStyleChanged⁽¹⁰⁶⁾) you can distinguish the T^{RichViewEdit} control using the Sender parameter.

1.2.2 Notes and text boxes

T^{RichViewEdit} control can edit documents with footnotes, endnotes, sidenotes (notes in a floating box) and floating boxes.







When a note or a text box is being edited, its Document is loaded in T^{RichViewEdit}.RVNote⁽⁶⁴⁾, ActiveEditor⁽⁴⁷⁾ becomes equal to RVNote⁽⁶⁴⁾, CurrentNote⁽⁵⁰⁾ returns this note/text box item, CurrentNoteParentEditor⁽⁵⁰⁾ returns the editor containing this note.

Footnotes and endnotes can be inserted only in the main document. Sidenotes and text box items can be inserted in the main document, headers and footers.

Limitations:

- document of footnote/endnote must not be higher than one page;
- ChangeStyleTemplates must not be called when a note editor is active.

Notes and text boxes in RichViewActions

-  TsvActionInsertFootnote⁽³¹⁹⁾ inserts a footnote.
-  TsvActionInsertEndnote⁽³¹⁸⁾ inserts an endnote.
-  TsvActionInsertSidenote⁽³²¹⁾ inserts a sidenote.
-  TsvActionInsertTextBox⁽³²³⁾ inserts a text box item.
-  TsvActionEditNote⁽³¹⁸⁾ starts editing a note or a text box.
-  TsvActionReturnToNote⁽³²⁴⁾ moves the caret to the parent footnote/endnote.

Use the actions above instead of their analogs included in RichViewActions for T^{RichViewEdit} controls.

Editing notes and text boxes without RichViewActions

Inserting a new footnote in SRichViewEdit1 (this example does not use StyleTemplates):

```
var Note: TRVFootnoteItemInfo;
    NoteRef: TRVNoteReferenceItemInfo;
    rve: TCustomRichViewEdit;

// we can insert a footnote only in the main document
rve := SRichViewEdit1.ActiveEditor(47);
if rve<>SRichViewEdit1.RichViewEdit(59) then
    exit;
// we will use item indices, so we require the top level editor
rve := rve.TopLevelEditor;
// creating a footnote
```

```

Note := TRVFootnoteItemInfo.CreateEx(rve.RVData,
    RVGetNoteTextStyleNo(rve.Style, rve.CurTextStyleNo), 1, False);
// filling Note.Document: adding a footnote number and a space character
NoteRef := TRVNoteReferenceItemInfo.CreateEx(Note.Document,
    RVGetNoteTextStyleNo(rve.Style, 0));
Note.Document.AddItem(' ', NoteRef);
Note.Document.AddNL(' ', 0, -1);
// inserting
if rve.InsertItem(' ', Note) then begin
    // starting editing the inserted footnote
    SRichViewEdit1.StartEditNote(99)(rve.RVData, rve.CurItemNo);
    // moving the caret to the end of the Note.Document
    rve := SRichViewEdit1.ActiveEditor(47);
    // now rv = SRichViewEdit.RVNote(64)
    rve.SetSelectionBounds(
        rve.ItemCount-1, rve.GetOffsAfterItem(rve.ItemCount-1),
        rve.ItemCount-1, rve.GetOffsAfterItem(rve.ItemCount-1));
end;

```

Returning from RVNote⁽⁶⁴⁾ to the parent footnote/endnote:

```
SRichViewEdit1.ReturnToNote(95);
```

1.3 Components

Components



TSRichViewEdit⁽⁴⁰⁾ allows editing rich text documents in WYSIWYG mode. It supports scaling and different view modes, including a page-view mode.



TDBSRichViewEdit⁽¹⁶⁸⁾ is a data-aware version of WYSIWYG editing component.



TScIRVRuler⁽¹⁸⁶⁾ is a horizontal or vertical ruler designed to work with TSRichViewEdit. It is inherited from TRVRuler component included in RichViewActions.



TSRVToolBar⁽¹⁸⁸⁾ is a simplified analog of TToolBar. Instances of TSRVToolbar are used in TSRichViewEdit: they allow adding buttons in scrollbar area.




TSRVToolWindow⁽¹⁹⁶⁾ uses the TSRVToolBar and displays it in a popup window.



TSRVPageScroll⁽¹⁷³⁾ displays thumbnails of pages for documents prepared in TSRichViewEdit component.




TSRVPrint⁽²⁰³⁾ and  TSRVPreview⁽²²²⁾ are components for advanced printing of documents prepared in TSRichViewEdit. For example, they allow printing posters (e.g. A3 page on A4 printer).



TSRVSkinManager⁽²²⁷⁾ allows applying skins to scrollbars of TSRichViewEdit⁽⁴⁰⁾ / TDBSRichViewEdit⁽¹⁶⁸⁾, to TSRVScrollBar⁽²⁴²⁾, to TSRVTabSet⁽²⁴⁸⁾.



TSRVTabSet⁽²⁴⁸⁾ – advanced skinnable tabs, can be used to implement a tabbed multiple document interface.

 **TSRVScrollBar**⁽²⁴²⁾ – skinnable scrollbar. This component is used for scrollbars in **TSRichViewEdit**, but may be useful as a separate component as well.

1.3.1 TSRichViewEdit

TSRichViewEdit is a word processing component.

Unit SclRView.

Syntax

```
TSRichViewEdit = class (TCustomControl);
```

Hierarchy

TObject

TPersistent

TComponent

TControl

TWinControl

TCustomControl

Description

The main features of the component are:

- zooming;
- WYSIWYG mode;
- real-time repagination;
- displaying and printing in different formats (e.g. the document in A5 format can be printed in A3 format);
- free positioning of page on the screen (optional);
- variety of document view modes (web mode, draft mode, page view mode, etc.).
- hint on scrolling containing page number and a short text;
- vertical and horizontal lines of buttons in the scroll bar area.

Properties

Main properties

RichViewEdit⁽⁵⁹⁾ returns a **TRichViewEdit** control used to render a document. Use properties and methods of this control to modify document or get information about it. By default, **RichViewEdit**⁽⁵⁹⁾ uses auto-created **TRVStyle** object. If you want to use an external **TRVStyle** component, use **ExternalRVStyle**⁽⁵¹⁾ property.

Other editors are used to render different parts of document:

- **RVHeader**⁽⁶³⁾ for page header;
- **RVFooter**⁽⁶¹⁾ for page footer;
- **RVNote**⁽⁶⁴⁾ for notes and text boxes.

The active editor is returned in **ActiveEditor**⁽⁴⁷⁾ property. Additional information about different part of documents is available in **overview**⁽³⁶⁾.

Properties related to page attributes are available in **PageProperty**⁽⁵⁸⁾.

Properties related to viewing modes are available in `ViewProperty`⁶⁸.

Properties related to background are available in `BackgroundProperty`⁴⁸.

Properties related to page arrangement inside the component window are available in `PagePosProperty`⁵⁸.

Properties related to line numbering are available in `LineNumberProperty`⁵⁴.

Properties related to a read-mode are available in `ReadModelProperty`⁵⁹.

Quick access to properties of `RichViewEdit`⁵⁹

The following properties provide access to properties of `RichViewEdit`⁵⁹ of the same name (some of these properties are applied to `ActiveEditor`⁴⁷ or to all editors, see more information in the topics about these properties):

- `AnimationMode`⁴⁷
- `BiDiMode`⁴⁸
- `CPEventKind`⁴⁹
- `ReadOnly`⁵⁹
- `RTFOptions`⁶⁰
- `RTFReadProperties`⁶⁰
- `RVBackgroundBitmap`⁶¹ (access to `BackgrounBitmap`)
- `RVBackgroundStyle`⁶¹ (access to `BackgrounStyle`)
- `RVColor`⁶¹ (access to `Color`)
- `RVEditorOptions`⁶¹ (access to `EditorOptions`)
- `RVFOptions`⁶²
- `RVFParaStylesReadMode`⁶³
- `RVFTextStylesReadMode`⁶³
- `RVOptions`⁶⁴ (access to `Options`)
- `SmartPopupProperties`⁶⁵, `SmartPopupVisible`⁶⁵
- `TabNavigation`⁶⁷
- `UseStyleTemplates`⁶⁸, `StyleTemplateInsertMode`⁶⁵.

These properties (except for a public property `SmartPopupVisible`⁶⁵) can be used to assign values at design time.

Scrolling

`HScrollPos`⁵³ (`0..HMaxScrollPos`⁵²) and `VScrollPos`⁶⁹ (`0..VMaxScrollPos`⁶⁸) allow getting/setting the scrolling positions.

`HScrollBar`⁵² and `VScrollBar`⁶⁹ hide/show scrollbars.

`OnHScrolled`¹⁰⁹ and `OnVScrolled`¹²² events occur on scrolling.

`ScrolledPage`⁶⁵ returns the index of the first visible page.

Menus (toolbars) in the scrollbar area

`MenuHButtons`⁵⁵ and `MenuHorizontal`⁵⁵ define properties of horizontal menu.

`MenuVButtons`⁵⁵ and `MenuVertical`⁵⁶ define properties of vertical menu.

Skins

Skins can be applied to scrollbars. Skins are linked using `SkinManager`⁽⁶⁵⁾ property. `HScrollBarSkinSchemeIndex`⁽⁵³⁾ and `HScrollBarSkinSchemeIndex`⁽⁶⁹⁾ are used to specify skin schemes for scrollbars.

Converted properties of PageProperty⁽⁵⁸⁾

Some properties return layout-related properties of `PageProperty`⁽⁵⁸⁾ converted to screen pixels and scaled according to `ViewProperty`⁽⁶⁸⁾. `ZoomPercent`⁽¹⁶⁷⁾:

- `LeftMarginPix`⁽⁵⁴⁾, `TopMarginPix`⁽⁶⁷⁾, `RightMarginPix`⁽⁶⁰⁾, `BottomMarginPix`⁽⁴⁸⁾
- `PageHeightPix`⁽⁵⁷⁾, `PageWidthPix`⁽⁵⁸⁾

Menus (toolbars) in scrollbars areas

Buttons are defined in `MenuHButtons`⁽⁵⁵⁾ and `MenuVButtons`⁽⁵⁵⁾. Properties of menus are defined in `MenuHorizontal`⁽⁵⁵⁾ and `MenuVertical`⁽⁵⁶⁾.

See also events below.

Methods

Coordinates

`ConvertRVToSRV`⁽⁷³⁾ and `ConvertSRVToRV`⁽⁷³⁾ allow translating coordinates between this `TSRichViewEdit` control and `RichViewEdit`⁽⁵⁹⁾.

`GetPageClientRect`⁽⁸⁷⁾ returns the position of page, `GetItemBounds`⁽⁸⁴⁾ returns the position of item in the page. The page is returned by `GetItemPages`⁽⁸⁶⁾.

`GetCaretPosInUnits`⁽⁸³⁾ returns the caret position on the page.

`GetPageAt`⁽⁸⁶⁾ returns the page at the specified coordinates, `GetItemAt`⁽⁸⁴⁾ returns the item at the specified coordinates.

`UnitsPerInchH`⁽⁹⁹⁾ and `UnitsPerInchV`⁽¹⁰⁰⁾ allow converting coordinates from units to units.

Finding items

`FindNextHyperlink`⁽⁷⁸⁾/`FindPriorHyperlink`⁽⁸¹⁾, `FindNextCheckpoint`⁽⁷⁷⁾/`FindPriorCheckpoint`⁽⁸⁰⁾, `FindNextItem`⁽⁷⁹⁾/`FindPriorItem`⁽⁸¹⁾, `FindNextHeading`⁽⁷⁸⁾/`FindPriorHeading`⁽⁸⁰⁾ search for hyperlinks, *checkpoints*, items, headings.

`NextCurHyperlink`⁽⁹¹⁾/`PriorCurHyperlink`⁽⁹⁴⁾, `NextCurCheckpoint`⁽⁹⁰⁾/`PriorCurCheckpoint`⁽⁹⁴⁾, `NextCurItem`⁽⁹¹⁾/`PriorCurItem`⁽⁹⁴⁾ select the next/previous hyperlink, item with *checkpoint*, item.

`NextCurHeading`⁽⁹⁰⁾/`PriorCurHeading`⁽⁹⁴⁾ moves the caret to the next/previous heading.

By default, the search is performed in the whole document. But you can search in the specified range of vertical coordinates, see `RangeSearch`⁽⁵⁸⁾ property.

`FirstCurPage`⁽⁸²⁾, `PriorCurPage`⁽⁹⁵⁾, `NextCurPage`⁽⁹¹⁾, `LastCurPage`⁽⁹⁰⁾ moves the caret to the beginning of the first/previous/next/last page.

Printing

You can print the whole document (`PrintAll`⁽⁹²⁾), the current page (`PrintCurrent`⁽⁹³⁾) or a range of pages (`PrintRange`⁽⁹³⁾).

With `MinPrintedItemNo`⁽⁵⁶⁾ and `MaxPrintedItemNo`⁽⁵⁴⁾ you can define a subrange of items for printing. Items outside of this range are not printed (but places are still reserved for them). Using

these properties, you can implement an incremental printing: when the next portion of the document is ready, you can print it below the previously printed fragment.

Drawing on external canvas

DrawPage⁽⁷⁶⁾/DrawMetafile⁽⁷⁶⁾ draws page onto the specified canvas/metafile.

Scrolling

ScrollToCaret⁽⁹⁵⁾/ScrollToItem⁽⁹⁶⁾ makes the caret/item visible.

GetFirstItemVisibleRV⁽⁸³⁾ returns the first visible item.

Footnotes and endnotes

Call StartEditNote⁽⁹⁹⁾ to edit a footnote/endnote, call ReturnToNote⁽⁹⁵⁾ to finish editing.

Additional information can be found in overview⁽³⁸⁾.

Events

The most of TSRichViewEdit events provides access to RichViewEdit⁽⁵⁹⁾'s events of the same names.

Events on changing properties

OnPageFormatChanged⁽¹¹⁴⁾ occurs when PageProperty⁽⁵⁸⁾.PageFormat⁽¹³⁸⁾ is changed.

OnChangeViewModeAfter⁽¹⁰⁴⁾ and OnChangeViewModeBefore⁽¹⁰⁵⁾ occur when ViewProperty⁽⁶⁸⁾.ViewMode⁽¹⁶⁴⁾ is changed.

OnZoomChanged⁽¹²²⁾ occurs when ViewProperty⁽⁶⁸⁾.ZoomPercent⁽¹⁶⁷⁾ or .ZoomMode⁽¹⁶⁵⁾ are changed.

OnZoomViewChanged⁽¹²²⁾ occurs when other properties of ViewProperty⁽⁶⁸⁾ related to zooming are changed.

Events on scrolling, clicking and caret moving

OnHScrolled⁽¹⁰⁹⁾, OnVScrolled⁽¹²²⁾, OnPageScrolled⁽¹¹⁴⁾ occur on scrolling.

OnCurrentPageChange⁽¹⁰⁷⁾ occurs when the caret is moved to another page.

OnPageCountChanged⁽¹¹³⁾ occurs when the count of pages is changed.

OnClickPage⁽¹⁰⁶⁾ occurs when the user clicked on the page.

Database

The following items occurs if Document⁽⁵¹⁾ property is linked to a database field using LiveBindings, or in inherited TDBSRichViewEdit⁽¹⁶⁸⁾ component.

OnNewDocument⁽¹¹²⁾ occurs when creating or before loading

OnLoadDocument⁽¹¹¹⁾ occurs after loading

OnSaveCustomFormat⁽¹¹⁹⁾, OnLoadCustomFormat⁽¹¹¹⁾ allow storing data in DB in custom format.

Custom drawing

In addition to custom drawing events of TRVStyle, you can use OnPaintPage⁽¹¹⁵⁾ for drawing on page, OnPaintComponent⁽¹¹⁴⁾ for drawing inserted controls, and OnPaint⁽¹¹⁴⁾ for drawing something else.

Inserted controls

OnPaintComponent⁽¹¹⁴⁾ allows drawing inserted controls (useful if the default procedure cannot draw them properly).

OnCheckControl⁽¹⁰⁵⁾ allows enumerating all inserted controls.

OnMessageControl⁽¹¹²⁾ allows responding on messages from inserted controls.

Free page position and ordering

OnGetPagePos⁽¹⁰⁸⁾ allows defining position, zooming and Z-order of pages

Menus (toolbars) in scrollbars areas

OnHMenuClickButton⁽¹⁰⁹⁾ and OnVMenuClickButton⁽¹²¹⁾ occur when the user clicked a button.

OnHMenuEnterButton⁽¹⁰⁹⁾ and OnVMenuEnterButton⁽¹²¹⁾ occur when the mouse pointer is above a button.

Table "icons"

When the user moves the mouse pointer above some table, a special icon is displayed at the top left corner of this table (if ViewProperty⁽⁶⁸⁾.UseTableIcons⁽¹⁶³⁾=True).

This icon is displayed for some time (specified in ViewProperty⁽⁶⁸⁾.TableIconDelay⁽¹⁶²⁾) after the mouse pointer leaves this table.

When the user clicks on this icon, OnTableIconClick⁽¹²¹⁾ occurs.

1.3.1.1 Properties

In TSRichViewEdit

- AcceptDragDropFormats⁽⁴⁷⁾
- AcceptPasteFormats⁽⁴⁷⁾
- ▶ ActiveEditor⁽⁴⁷⁾
- AnimationMode⁽⁴⁷⁾
- BackgroundProperty⁽⁴⁸⁾
- BiDiMode⁽⁴⁸⁾
- ▶ BottomMarginPix⁽⁴⁸⁾
- CanScroll⁽⁴⁸⁾
- CanUpdate⁽⁴⁸⁾
- CanUpdatePosition⁽⁴⁸⁾
- CanUpdateMargin⁽⁴⁹⁾
- CaretBlinkTime⁽⁴⁹⁾
- ▶ CaretPos⁽⁴⁹⁾
- CaretVisible⁽⁴⁹⁾
- CPEventKind⁽⁴⁹⁾
- ▶ CurControl⁽⁵⁰⁾
- ▶ CurrentNote⁽⁵⁰⁾
- ▶ CurrentNoteParentEditor⁽⁵⁰⁾
- CurrentPage⁽⁵⁰⁾
- ExternalRVStyle⁽⁵¹⁾
- Document⁽⁵¹⁾
- ExternalRVStyleFooter⁽⁵¹⁾

- ExternalRVStyleHeader⁵¹
- ▶ FirstPageNo⁵¹
- FixMarginsMode⁵²
- ▶ HMaxScrollPos⁵²
- HScrollBar⁵²
- HScrollBarSchemeIndex⁵³
- HScrollPos⁵³
- HTMLReadProperties⁵³
- HTMLSaveProperties⁵³
- ▶ LastPageNo⁵³
- ▶ LeftMarginPix⁵⁴
- MarkdownProperties⁵⁴
- MaxPrintedItemNo⁵⁴
- MaxPrintedOffsInItem⁵⁴
- MenuHButtons⁵⁵
- MenuHorizontal⁵⁵
- MenuVButtons⁵⁵
- MenuVertical⁵⁶
- MinPrintedItemNo⁵⁶
- MinPrintedOffsInItem⁵⁶
- ▶ OffsetX⁵⁶
- ▶ OffsetY⁵⁷
- ▶ PageCount⁵⁷
- ▶ PageHeightPix, PageHeight100Pix⁵⁷
- ▶ PageNoMouseDown⁵⁷
- ▶ PageNoMouseMove⁵⁷
- PageProperty⁵⁸
- ▶ PageWidthPix, PageWidth100Pix⁵⁸
- RangeSearch⁵⁸
- RangeSearchFirstY⁵⁹
- RangeSearchLastY⁵⁹
- ReadModeProperty⁵⁹
- ReadOnly⁵⁹
- RichViewEdit⁵⁹
- ▶ RightMarginPix⁶⁰
- RTFOptions⁶⁰
- RTFReadProperties⁶⁰
- RVBackgroundBitmap⁶¹
- RVBackgroundStyle⁶¹
- RVColor⁶¹
- RVEditorOptions⁶¹
- RVFooter⁶¹
- RVFOptions⁶²
- RVFParaStylesReadMode⁶³
- RVFTTextStylesReadMode⁶³
- RVHeader⁶³
- RVNote⁶⁴
- RVOptions⁶⁴

- ▶ ScrolledPage ⁶⁵
- SkinManager ⁶⁵
- SmartPopupProperties ⁶⁵
 - SmartPopupVisible ⁶⁵
- StyleTemplateInsertMode ⁶⁵
- TabNavigation ⁶⁷
- TextEngine ⁶⁷
- ▶ TopMarginPix ⁶⁷
- UnitsProgram ⁶⁸
- UseDrawHyperlinksEvent ⁶⁸
- UseStyleTemplates ⁶⁸
- Version ⁶⁸
- ViewProperty ⁶⁸
- ▶ VMaxScrollPos ⁶⁸
- VScrollBar ⁶⁹
- VScrollBarSchemeIndex ⁶⁹
 - VScrollPos ⁶⁹

Derived from TCustomControl

- Align
- Anchors
- Constraints
- Ctl3D
- DragMode
- Enabled
- Height
- HelpContext
- HelpKeyword (D6+)
- HelpType (D6+)
- Hint
- Left
- Name
- ParentCtl3D
- ParentShowHint
- PopupMenu
- ShowHint
- TabOrder
- TabStop
- Tag
- Top
- Touch (D2010+)
- Visible
- Width

1.3.1.1.1 TSVRichViewEdit.AcceptDragDropFormats, AcceptPasteFormats

AcceptDragDropFormats lists formats that can be accepted when dropping data in TSVRichViewEdit.

AcceptPasteFormats lists formats that can be accepted when pasting in TSVRichViewEdit.

property AcceptDragDropFormats: TRVDragDropFormats;

property AcceptPasteFormats: TRVDragDropFormats;

These properties provide access to RichViewEdit⁽⁵⁹⁾.AcceptDragDropFormats and AcceptPasteFormats properties.

Default values

- AcceptDragDropFormats: [rvddRVF, rvddRTF, rvddText, rvddUnicodeText, rvddBitmap, rvddMetafile, rvddFiles]
- AcceptPasteFormats: [rvddRVF, rvddRTF, rvddText, rvddUnicodeText, rvddBitmap, rvddMetafile, rvddFiles, rvddHTML]

1.3.1.1.2 TSVRichViewEdit.ActiveEditor

Returns the active editor belonging to this TSVRichViewEdit control.

property ActiveEditor: TRichViewEdit;

This property returns the editor containing the caret. All editing operations must be applied to this editor.

This property can return one of the following values:

- RichViewEdit⁽⁵⁹⁾
- RVHeader⁽⁶³⁾
- RVFooter⁽⁶¹⁾
- RVNote⁽⁶⁴⁾

When the user starts editing another RichViewEdit, OnChangeActiveEditor⁽¹⁰⁴⁾ event occurs.

See also methods:

- StartEditing⁽⁹⁸⁾
- StartEditNote⁽⁹⁹⁾
- Document parts in TSVRichViewEdit⁽³⁶⁾

1.3.1.1.3 TSVRichViewEdit.AnimationMode

Specifies when an animation of images starts.

property AnimationMode: TRVAnimationMode;

Animation is played in ActiveEditor⁽⁴⁷⁾. Animation in other editors is disabled.

To enable gif animation, include in your project:

- RVGifAnimate.pas (animation using the Anders Melander's TGifImage, for Delphi 4-2006);
- RVGifAnimate2007.pas (animation using VCL's TGifImage, for Delphi 2007 or newer);
- RVJvGifAnimate.pas (animation using the JVCL's TJvGifImage).

Default value:

rvaniOnFormat (animation is started automatically)

1.3.1.1.4 TSRichViewEdit.BackgroundProperty

Contains background settings.

```
property BackgroundProperty: TSRVBackgroundProperty(123);
```

1.3.1.1.5 TSRichViewEdit.BiDiMode

Specifies the default bidirectional mode for the control.

```
property BiDiMode: TRVBiDiMode;
```

This property provides access to RichViewEdit⁽⁵⁹⁾'s property of the same name.

A value of this property changes layout of the whole editor. If it is equal to *rvbdRightToLeft*, the following changes are applied:

- the vertical scrollbar is placed at the left side (unless you assign *False* to ViewProperty⁽⁶⁸⁾.AutoVScrollBarPosition)⁽¹⁵⁵⁾
- if pages are arranged in columns, pages in each row are ordered from right to left
- PagePosProperty⁽⁵⁸⁾.AlignPageH⁽¹⁴⁴⁾ = *srvaphLeft* aligns pages to the right; *srvaphRight* aligns pages to the left.

1.3.1.1.6 TSRichViewEdit.BottomMarginPix

Returns the bottom margin, in pixels.

```
property BottomMarginPix: Integer;
```

```
property BottomMargin100Pix: Integer;
```

The bottom margin is defined in PageProperty⁽⁵⁸⁾.BottomMargin⁽¹³⁵⁾. These properties return this margin converted to pixels.

BottomMarginPix is scaled according to ViewProperty⁽⁶⁸⁾.ZoomPercent⁽¹⁶⁷⁾ (so this property returns the margin as it is displayed on the screen).

BottomMargin100Pix is not scaled, it is the margin in 96 DPI and 100% zooming:

BottomMarginPix := **BottomMargin100Pix** * (ZoomPercent / 100) * (<screen DPI> / 96).

See also:

- LeftMarginPix⁽⁵⁴⁾
- TopMarginPix⁽⁶⁷⁾
- RightMarginPix⁽⁶⁰⁾

1.3.1.1.7 TSRichViewEdit.CanUpdate, CanScroll, CanUpdatePosition

Enables/disables repainting.

```
property CanUpdate: Boolean;
```

```
property CanScroll: Boolean;
```

```
property CanUpdatePosition: Boolean;
```

If CanUpdate=*False*, the component does not update its image on the screen and does not move the caret.

If `CanScroll=False`, the component does not scroll itself.

If `CanUpdatePosition=False`, the component does not scroll to make the caret visible.

See also:

- `CalculatePageCount`⁽⁷²⁾

1.3.1.1.8 TSRichViewEdit.CanUpdateMargin

Enables/disables margins changing.

property `CanUpdateMargin: Boolean;`

If this property is *False*:

- the component does not allow changing `FormatWidth`⁽¹⁴¹⁾, `FormatHeight`⁽¹³⁹⁾, `PageLeft`⁽¹³⁷⁾, `PageTop`⁽¹⁴²⁾, `PageRight`⁽¹⁴¹⁾, `PageBottom`⁽¹³⁵⁾ properties;
- `SetMargin`⁽⁹⁷⁾ and `SetMarginMM`⁽⁹⁷⁾ functions do not set the margin values.

1.3.1.1.9 TSRichViewEdit.CaretBlinkTime

Caret blinking interval.

property `CaretBlinkTime: Cardinal;`

This property defines the period when the caret remains in one of the states: displayed or not.

Default value:

a system default value returned by `GetCaretBlinkTime` function.

1.3.1.1.10 TSRichViewEdit.CaretPos

Client coordinates of the caret.

property `CaretPos: TPoint;`

The caret coordinates relative to the top left corner of the component.

See also:

- `GetCaretPosInUnit`⁽⁸³⁾

1.3.1.1.11 TSRichViewEdit.CaretVisible

Enable/disable the caret display.

property `CaretVisible: Boolean;`

`CursorVisible` indicates whether to display caret or not, for example for implementing a read-only mode or a print preview.

By default the caret is visible.

1.3.1.1.12 TSRichViewEdit.CPEventKind

Specifies the mode of generating `OnCheckpointVisible`⁽¹⁰⁵⁾ event.

property `CPEventKind: TCPEventKind;`

This property provides access to `RichViewEdit`⁽⁵⁹⁾'s property of the same name.

TSRichViewEdit supports this property&event, despite of the fact that they are not supported by TRichViewEdit.

1.3.1.1.13 TSRichViewEdit.CurControl

Returns the control (inserted in the editor) having the input focus.

property CurControl: TControl;

Nil, if no control has focus.

See also:

- OnChangeCurrentControl⁽¹⁰⁴⁾

1.3.1.1.14 TSRichViewEdit.CurrentNote, CurrentNoteParentEditor

The properties return the note (or a text box) item edited in RVNote⁽⁶⁴⁾, and the editor containing this note.

property CurrentNote: TCustomRVNoteItemInfo;

property CurrentNoteParentEditor: TCustomRichViewEdit;

These properties return *nil* if a note is not being edited.

Use StartEditNote⁽⁹⁹⁾ to start editing a note, use ReturnToNote⁽⁹⁵⁾ to finish editing.

CurrentNote can be of one of the following types:

- footnotes (TRVFootnoteItemInfo)
- endnotes (TRVEndnoteItemInfo)
- sidenotes (TRVSidenoteItemInfo)
- text box items (TRVTextBoxItemInfo)

Footnotes and endnotes can be inserted only in RichViewEdit⁽⁵⁹⁾. Sidenotes and text boxes can be inserted in RichViewEdit⁽⁵⁹⁾, RVHeader⁽⁶³⁾ and RVFooter⁽⁶¹⁾.

CurrentNoteParentEditor always returns a root editor (RichViewEdit⁽⁵⁹⁾, RVHeader⁽⁶³⁾ or RVFooter⁽⁶¹⁾), even if a note is inserted in a table cell).

See also:

- Footnotes and endnotes⁽³⁸⁾

1.3.1.1.15 TSRichViewEdit.CurrentPage

Index of the current page (from 1)

property CurrentPage: Integer;

This page contains the caret.

If you assign a value to this property, the caret is moved to the beginning of the specified page.

See also:

- ScrolledPage⁽⁶⁵⁾
- OnCurrentPageChange⁽¹⁰⁷⁾
- FirstCurPage⁽⁸²⁾, PriorCurPage⁽⁹⁵⁾, NextCurPage⁽⁹¹⁾, LastCurPage⁽⁹⁰⁾

1.3.1.1.16 TSRichViewEdit.Document

Allows linking this TSRichViewEdit component to a database field (Delphi XE2+)

property Document: TRVDocumentProperty;

You can use LiveBindings to link this property to a dataset field of TBlobField type.

This is an alternative way of using ScaleRichView components to view and edit database fields. Another way is using TDBSRichViewEdit⁽¹⁶⁸⁾ components.

This property contains sub-properties controlling this linking.

Important note: in order to link **Document** to TBlobField, DBRV unit must be included in the project (add it in "uses").

Note: while this property is available in TDBSRichViewEdit⁽¹⁶⁸⁾ as well, it's highly recommended to use LiveBindings only for TSRichViewEdit.

1.3.1.1.17 TSRichViewEdit.ExternalRVStyle, -Header, -Footer

These properties allow using external TRVStyle components.

property ExternalRVStyle: TRVStyle;

property ExternalRVStyleHeader: TRVStyle;

property ExternalRVStyleFooter: TRVStyle;

Assigning these properties is optional. You can assign links to TRVStyle components to these properties.

If assigned (not *nil*):

- ExternalRVStyle is assigned to RichViewEdit⁽⁵⁹⁾.Style
- ExternalRVStyleHeader is assigned to RVHeader⁽⁶³⁾.Style
- ExternalRVStyleFooter is assigned to RVFooter⁽⁶¹⁾.Style

By default, these properties are not assigned (*nil*), and internal TRVStyle components are used.

All these components must be different, you cannot assign the same component to two or more of these properties.

These properties are useful if you want to define properties of TRVStyle at design time.

If UseStyleTemplates⁽⁶⁸⁾=*True*, the component assigns:

- RichViewEdit⁽⁵⁹⁾.Style.MainRVStyle = *nil*;
- RVHeader⁽⁶³⁾.Style.MainRVStyle = RichViewEdit⁽⁵⁹⁾.Style
- RVFooter⁽⁶¹⁾.Style.MainRVStyle = RichViewEdit⁽⁵⁹⁾.Style

If some of these TRVStyle components are external, these assignments are made to their properties too.

1.3.1.1.18 TSRichViewEdit.FirstPageNo

Returns the index of the first page that will be printed, from 1.

property FirstPageNo: Integer;

This property takes the properties MinPrintedItemNo⁽⁵⁶⁾ and MaxPrintedItemNo⁽⁵⁴⁾ (and the corresponding offsets in items) into account. By default, when these properties define the range

including the whole document, `FirstPageNo` returns 1. If `MinPrintedItemNo`⁽⁵⁶⁾ > 0, this property returns the index of the page containing the `MinPrintedItemNo`⁽⁵⁶⁾-th item (see `GetPageNo`⁽⁸⁷⁾). If `MinPrintedItemNo`⁽⁵⁶⁾ >= `RichViewEdit`⁽⁵⁹⁾.`ItemCount`, `FirstPageNo` returns `PageCount`⁽⁵⁷⁾ + 1 (and nothing will be printed).

1.3.1.1.19 TSRichViewEdit.FixMarginsMode

Specifies how to work with too small margins (less than the printer supports).

type

```
TSRVFixMarginsMode = (srvfmmAutoCorrect, srvfmmIgnore);
```

property `FixMarginsMode` : `TSRVFixMarginsMode`;

This property is similar to `TRVPrint`'s property of the same name.

Mode	Meaning
<i>srvfmmAutoCorrect</i>	Minimal supported margins are used (if the left and/or the top margin were too small, the output is moved to the right and/or down)
<i>srvfmmIgnore</i>	The specified values of margins are used (document may be cropped)

Default value:

srvfmmAutoCorrect

See also

- `PageProperty`⁽⁵⁸⁾.`PrintableAreaPen`⁽¹⁴¹⁾

1.3.1.1.20 TSRichViewEdit.HMaxScrollPos

Defines the maximum position of horizontal scroll bar.

property `HMaxScrollPos`: `Integer`;

This value is used to calculate the maximum possible value for `HScrollPos`⁽⁵³⁾.

This is a read-only property.

See also:

- `VMaxScrollPos`⁽⁶⁸⁾

1.3.1.1.21 TSRichViewEdit.HScrollBar

Shows/hides the horizontal scrollbar.

property `HScrollBar`: `Boolean`;

Default value:

True (visible)

1.3.1.1.22 TSRichViewEdit.HScrollBarSchemeIndex

Determines which skin scheme is used for the horizontal scrollbar.

property HScrollBarSchemeIndex: Integer;

This is an index in SkinManager⁽⁶⁵⁾.CurrentSkin⁽²²⁹⁾.HorizontalTabSetSchemes⁽²³¹⁾ collection.

If SkinManager⁽⁶⁵⁾ = *nil*, or SkinManager⁽⁶⁵⁾.CurrentSkin⁽²²⁹⁾ = *nil*, or this index is not valid for this collection (negative or too large), then this scrollbar has a default (no-skin) appearance.

Default value

0

See also

- VScrollBarSchemeIndex⁽⁶⁹⁾
- ScrollBarControlStyle⁽⁶⁹⁾

1.3.1.1.23 TSRichViewEdit.HScrollPos

Defines the position of horizontal scroll bar, measured in screen pixels.

property HScrollPos: Integer;

You can use this property for scrolling to the specified position.

Value of this property is in the range from 0 to HMaxScrollPos⁽⁵²⁾ - ClientWidth + 1.

See also:

- VScrollPos⁽⁶⁹⁾

1.3.1.1.24 TSRichViewEdit.HTMLReadProperties

A set of properties controlling HTML import.

property HTMLReadProperties: TRVHTMLReaderProperties;

This property provides access to RichViewEdit⁽⁵⁹⁾'s property of the same name.

1.3.1.1.25 TSRichViewEdit.HTMLSaveProperties

A set of properties controlling HTML export.

property HTMLSaveProperties: TRVHTMLSaveProperties;

This property provides access to RichViewEdit⁽⁵⁹⁾'s property of the same name.

Note: the default value for **HTMLSaveProperty**.ImageOptions is different for control types:

- for TSRichViewEdit⁽⁴⁰⁾: []
- for TDBSRichViewEdit⁽¹⁶⁸⁾: [*rvhtmlsiolnlineImages*]

1.3.1.1.26 TSRichViewEdit.LastPageNo

Returns the index of the last page that will be printed, from 1.

property LastPageNo: Integer;

This property takes the properties MinPrintedItemNo⁽⁵⁶⁾ and MaxPrintedItemNo⁽⁵⁴⁾ into account. By default, when these properties define the range including the whole document, LastPageNo returns PageCount⁽⁵⁷⁾. If MaxPrintedItemNo⁽⁵⁴⁾ is inside the items range (0..RichViewEdit⁽⁵⁹⁾.ItemCount-1),

this property returns the index of the page containing the `MaxPrintedItemNo`⁽⁵⁴⁾-th item (see `GetPageNo`⁽⁸⁷⁾).

1.3.1.1.27 TSRichViewEdit.LeftMarginPix

Returns the left margin, in pixels.

```
property LeftMarginPix: Integer;
property LeftMargin100Pix: Integer;
function GetLeftMarginPix(PageNo : Integer) : Integer;
function GetLeftMargin100Pix(PageNo : Integer) : Integer;
```

The left margin is defined in `PageProperty`⁽⁵⁸⁾.`LeftMargin`⁽¹³⁷⁾.

Properties

These properties return this margin converted to pixels.

LeftMarginPix is scaled according to `ViewProperty`⁽⁶⁸⁾.`ZoomPercent`⁽¹⁶⁷⁾ (so this property returns the margin as it is displayed on the screen).

LeftMargin100Pix is not scaled, it is the margin in 96 DPI and 100% zooming:

$$\text{LeftMarginPix} := \text{LeftMargin100Pix} * (\text{ZoomPercent} / 100) * (<\text{screen DPI}> / 96).$$

Methods

The methods return the left margin for the specified page, taking `PageProperty`⁽⁵⁸⁾.`MirrorMargins`⁽¹³⁸⁾ into account.

See also:

- `TopMarginPix`⁽⁶⁷⁾
- `RightMarginPix`⁽⁶⁰⁾
- `BottomMarginPix`⁽⁴⁸⁾

1.3.1.1.28 TSRichViewEdit.LineNumberProperty

Contains settings for displaying line numbers

```
property LineNumberProperty: TSRVLineNumberProperty(131);
```

1.3.1.1.29 TSRichViewEdit.MarkdownProperties

A set of properties controlling Markdown export.

```
property MarkdownProperties: TRVMarkdownProperties;
```

This property provides access to `RichViewEdit`⁽⁵⁹⁾'s property of the same name.

1.3.1.1.30 TSRichViewEdit.MaxPrintedItemNo, MaxPrintedOffsInItem

The properties define the ending position to print.

```
property MaxPrintedItemNo: Integer;
property MaxPrintedOffsInItem: Integer;
```

The component prints items from `(MinPrintedItemNo, MinPrintedOffsInItem)`⁽⁵⁶⁾ to `(MaxPrintedItemNo, MaxPrintedOffsInItem)`. By default, this range includes all items.

Items outside of this range are not printed (but they still occupy their spaces). The pages that do not contain items in this range are skipped (the component prints pages from FirstPageNo⁽⁵¹⁾ to LastPageNo⁽⁵³⁾).

MaxPrintedItemNo is the index of item in RichViewEdit⁽⁵⁹⁾, the last item to print. The value -1 is treated like RichViewEdit⁽⁵⁹⁾.ItemCount-1, i.e. the document is printed to the end.

MaxPrintedOffsInItem is the offset of the last position in the **MaxPrintedItemNo**-th item to print. The value -1 is treated like RichViewEdit⁽⁵⁹⁾.GetOffsAfterItem(**MaxPrintedItemNo**), i.e. the item is printed completely. If this is a text item, this property allows printing a part of it. The position is defined with up-to-line, not up-to-character precision (i.e., if the item is displayed as several lines, and the specified position is in the middle of a line, the part of item on this line is printed).

These properties allow implementing an incremental printing: when the next portion of the document is ready, it can be printed below the previously printed fragment. This feature can be used for printing accounting journals, logs, etc.

Default value:

-1

1.3.1.1.31 TSRichViewEdit.MenuHButtons

A collection of buttons for the horizontal menu.

property MenuHButtons: TSRVButtonCollection⁽²⁷⁰⁾;

A horizontal menu (horizontal toolbar) is placed in the horizontal scrollbar area.

Properties of horizontal menu are defined in MenuHorizontal⁽⁵⁵⁾ property.

See also:

- MenuVButtons⁽⁵⁵⁾

1.3.1.1.32 TSRichViewEdit.MenuHorizontal

Properties of a horizontal menu (horizontal toolbar)

property MenuHorizontal: TSRVPropertyTBH⁽¹⁴⁷⁾;

See also:

- MenuHButtons⁽⁵⁵⁾
- OnHMenuClickButton⁽¹⁰⁹⁾
- OnHMenuEnterButton⁽¹⁰⁹⁾
- MenuVertical⁽⁵⁶⁾

1.3.1.1.33 TSRichViewEdit.MenuVButtons

A collection of buttons for the vertical menu.

property MenuVButtons: TSRVButtonCollection⁽²⁷⁰⁾;

A vertical menu (vertical toolbar) is placed in the vertical scrollbar area.

Properties of vertical menu are defined in MenuVertical⁽⁵⁶⁾ property.

See also:

- MenuHButtons⁽⁵⁵⁾

1.3.1.1.34 TSRichViewEdit.MenuVertical

Properties of a vertical menu (vertical toolbar)

property MenuVertical: TSRVPropertyTBV⁽¹⁴⁸⁾;

See also:

- MenuVButtons⁽⁵⁵⁾
- OnVMenuClickButton⁽¹²¹⁾
- OnVMenuEnterButton⁽¹²¹⁾
- MenuHorizontal⁽⁵⁵⁾

1.3.1.1.35 TSRichViewEdit.MinPrintedItemNo, MinPrintedOffsInItem

The properties define the starting position to print.

property MinPrintedItemNo: Integer;

property MinPrintedOffsInItem: Integer;

The component prints items from (**MinPrintedItemNo**, **MinPrintedOffsInItem**) to (**MaxPrintedItemNo**, **MaxPrintedOffsInItem**⁽⁵⁴⁾). By default, this range includes all items.

Items outside of this range are not printed (but they still occupy their spaces). The pages that do not contain items in this range are skipped (the component prints pages from **FirstPageNo**⁽⁵¹⁾ to **LastPageNo**⁽⁵³⁾).

MinPrintedItemNo is the index of item in RichViewEdit⁽⁵⁹⁾, the first item to print.

MinPrintedOffsInItem is the offset of the first position in the **MinPrintedItemNo**-th item to print. If this is a text item, this property allows printing a part of it. The position is defined with up-to-line, not up-to-character precision (i.e., if the item is displayed as several lines, and the specified position is in the middle of a line, the part of item on this line is printed).

These properties allow implementing an incremental printing: when the next portion of the document is ready, it can be printed below the previously printed fragment. This feature can be used for printing accounting journals, logs, etc.

Default value:

0

1.3.1.1.36 TSRichViewEdit.OffsetX

A horizontal offset of document (page) image in the editor window.

property OffsetX: Single;

This property is a maximal value of:

- $(\text{ClientWidth} - \text{PageWidthPix}^{(58)}) / 2$
- $\text{PagePosProperty}^{(58)}.HPadding^{(145)}$ (taking $\text{PagePosProperty}^{(58)}.HiddenDistances$ into account)

See also:

- OffsetY⁽⁵⁷⁾

1.3.1.1.37 TSRichViewEdit.OffsetY

A vertical offset of document (page) image in the editor window.

property OffsetY: Single;

This property is a maximal value of:

- (ClientHeight - PageHeightPix⁽⁵⁷⁾) div 2
- 0

See also:

- OffsetX⁽⁵⁶⁾

1.3.1.1.38 TSRichViewEdit.PageCount

Returns the count of pages in the document.

property PageCount: Integer;

Accessing this property does not invoke a repagination.

Use calculatePageCount⁽⁷²⁾ method for repagination.

1.3.1.1.39 TSRichViewEdit.PageHeightPix, PageHeight100Pix

The properties return a page height of document on the screen.

property PageHeightPix: Integer;

property PageHeight100Pix: Integer;

PageHeightPix returns the page height converted to screen pixels and scaled according to ViewProperty⁽⁶⁸⁾.ZoomPercent⁽¹⁶⁷⁾ (i.e. a page height as it can be seen in the control window).

PageHeight100Pix returns the page height converted to screen pixels in 96 DPI and ZoomPercent=100:

PageHeightPix := **PageHeight100Pix** * (ZoomPercent / 100) * (<screen DPI> / 96).

The same value measured in UnitsProgram⁽⁶⁸⁾ is returned in ViewProperty⁽⁶⁸⁾.PageHeight⁽¹⁶¹⁾.

The page height is defined in PageProperty⁽⁵⁸⁾.PageHeight⁽¹³⁹⁾.

See also:

- PageWidthPix⁽⁵⁸⁾

1.3.1.1.40 TSRichViewEdit.PageNoMouseDown

Returns the index of page (from 1) where a mouse button was pressed last time.

property PageNoMouseDown: Integer;

1.3.1.1.41 TSRichViewEdit.PageNoMouseMove

Returns the index of page (from 1) below the mouse pointer.

property PageNoMouseMove: Integer;

1.3.1.1.42 TSRichViewEdit.PagePosProperty

Pages positioning properties.

property PagePosProperty: TSRVPagePositionProperty⁽¹⁴³⁾;

See also:

- ViewProperty⁽⁶⁸⁾.FreePosPage⁽¹⁵⁷⁾

1.3.1.1.43 TSRichViewEdit.PageProperty

Page properties.

property PageProperty: TSRVPageProperty⁽¹³³⁾;

1.3.1.1.44 TSRichViewEdit.PageWidthPix, PageWidth100Pix

The properties return a page width of document on the screen.

property PageWidthPix: Integer;

property PageWidth100Pix: Integer;

PageWidthPix returns the page width converted to screen pixels and scaled according to ViewProperty⁽⁶⁸⁾.ZoomPercent⁽¹⁶⁷⁾ (i.e. a page width as it can be seen in the control window).

PageWidth100Pix returns the page height converted to screen pixels in 96 DPI and ZoomPercent=100:

PageWidthPix := **PageWidth100Pix** * (ZoomPercent / 100) * (<screen DPI> / 96).

The same value measured in UnitsProgram⁽⁶⁸⁾ is returned in ViewProperty⁽⁶⁸⁾.PageWidth⁽¹⁶¹⁾.

The page width is defined in PageProperty⁽⁵⁸⁾.PageWidth⁽¹⁴¹⁾.

See also:

- PageHeightPix⁽⁵⁷⁾

1.3.1.1.45 TSRichViewEdit.RangeSearch

Allows to narrow a search scope.

RangeSearch: Boolean;

If it is *True*, the search is performed only in items located between coordinates specified in RangeSearchFirstY⁽⁵⁹⁾..RangeSearchLastY⁽⁵⁹⁾ properties.

This range affects the following methods:

- FindPriorItem⁽⁸¹⁾, FindNextItem⁽⁷⁹⁾,
- FindPriorHyperlink⁽⁸¹⁾, FindNextHyperlink⁽⁷⁸⁾,
- FindPriorCheckpoint⁽⁸⁰⁾, FindNextCheckpoint⁽⁷⁷⁾,
- PriorCurlItem⁽⁹⁴⁾, NextCurlItem⁽⁹¹⁾,
- PriorCurHyperlink⁽⁹⁴⁾, NextCurHyperlink⁽⁹¹⁾,
- PriorCurCheckpoint⁽⁹⁴⁾, NextCurCheckpoint⁽⁹⁰⁾.

This feature can be used, for example, to search only on the some pages.

Example: searching for the first visible hyperlink (if any hyperlink is visible):

var

RVData: TCustomRVData;

```

RVData2: TCustomRVFormattedData;
ItemNo, OffsetInItem: Integer;
..
SRichViewEdit1.RangeSearch := True;
SRichViewEdit1.RangeSearchFirstY(59) := 0;
SRichViewEdit1.RangeSearchLastY(59) := SRichViewEdit1.ClientHeight;
SRichViewEdit1.GetFirstItemVisibleRV(83) (RVData2, ItemNo, OffsetInItem);
RVData := RVData2;
Dec (ItemNo);
if FindNextHyperlink(78) (RVData, ItemNo) then
    ...

```

1.3.1.1.46 TSRichViewEdit.RangeSearchFirstY

Allows to narrow a search scope.

RangeSearchFirstY: Integer;

This is a client coordinate (relative to the top of the component window).

See RangeSeach⁽⁵⁸⁾ property.

1.3.1.1.47 TSRichViewEdit.RangeSearchLastY

Allows to narrow a search scope.

RangeSearchLastY: Integer;

This is a client coordinate (relative to the top of the component window).

See RangeSeach⁽⁵⁸⁾ property.

1.3.1.1.48 TSRichViewEdit.ReadModeProperty

Contains settings for a read mode and side-to-side page displaying mode.

property ReadModeProperty: TSRVReadModeProperty⁽¹⁵⁰⁾;

1.3.1.1.49 TSRichViewEdit.ReadOnly

When set to *True*, this property prevents user's input in the editor; it also blocks execution of all editing-style methods of RichViewEdit⁽⁵⁹⁾.

property ReadOnly : Boolean;

This property provides access to RichViewEdit⁽⁵⁹⁾'s property of the same name.

1.3.1.1.50 TSRichViewEdit.RichViewEdit

Returns TRichViewEdit component used to render documents.

property RichViewEdit: TRichViewEdit;

This editor is used to render and edit the main part of document.

Other editors are used to render different parts of document:

- RVHeader⁽⁶³⁾ for page header;
- RVFooter⁽⁶¹⁾ for page footer;
- RVNote⁽⁶⁴⁾ for footnotes and endnotes.

The active editor is returned in `ActiveEditor`⁽⁴⁷⁾ property.

See also:

- Document parts in `TSRichViewEdit`⁽³⁶⁾

1.3.1.1.51 TSRichViewEdit.RightMarginPix

Returns the right margin, in pixels.

```
property RightMarginPix: Integer;
property RightMargin100Pix: Integer;
function GetRightMarginPix(PageNo : Integer) : Integer;
function GetRightMargin100Pix(PageNo : Integer) : Integer
```

The right margin is defined in `PageProperty`⁽⁵⁸⁾.`RightMargin`⁽¹⁴¹⁾.

Properties

These properties return this margin converted to pixels.

RightMarginPix is scaled according to `ViewProperty`⁽⁶⁸⁾.`ZoomPercent`⁽¹⁶⁷⁾ (so this property returns the margin as it is displayed on the screen).

RightMargin100Pix is not scaled, it is the margin in 96 DPI and 100% zooming:

$$\text{RightMarginPix} := \text{RightMargin100Pix} * (\text{ZoomPercent} / 100) * (<\text{screen DPI}> / 96).$$

Methods

The methods return the right margin for the specified page, taking `PageProperty`⁽⁵⁸⁾.`MirrorMargins`⁽¹³⁸⁾ into account.

See also:

- `LeftMarginPix`⁽⁵⁴⁾
- `TopMarginPix`⁽⁶⁷⁾
- `BottomMarginPix`⁽⁴⁸⁾

1.3.1.1.52 TSRichViewEdit.RTFOptions

Options for saving RTF (Rich Text Format).

```
property RTFOptions: TRVRTFOptions;
```

This property provides access to `RichViewEdit`⁽⁵⁹⁾'s property of the same name.

The initial value of this property for an internal `TRichViewEdit` includes *rvrtfSaveDocParameters* (in addition to the options included in the initial value of this property for `TRichView`)

1.3.1.1.53 TSRichViewEdit.RTFReadProperties

A set of properties controlling RTF (Rich Text Format) import.

```
property RTFReadProperties: TRVRTFReaderProperties;
```

This property provides access to `RichViewEdit`⁽⁵⁹⁾'s property of the same name.

The initial value of this property for an internal `TRichViewEdit` has:

- `RTFReadProperties.ReadDocParameters = True`.

1.3.1.1.54 TSRichViewEdit.RVBackgroundBitmap

Background image shown below the document

property RVBackgroundBitmap: TBitmap;

Provides access to RichViewEdit⁽⁵⁹⁾.BackgroundBitmap property.

See also:

- RVBackgroundStyle⁽⁶¹⁾
- RVColor⁽⁶¹⁾

1.3.1.1.55 TSRichViewEdit.RVBackgroundStyle

Defines how RVBackgroundBitmap⁽⁶¹⁾ is displayed

property RVBackgroundStyle: TBackgroundStyle;

Provides access to RichViewEdit⁽⁵⁹⁾.BackgroundStyle property.

See also:

- RVColor⁽⁶¹⁾

1.3.1.1.56 TSRichViewEdit.RVColor

Background color of document.

property RVColor: TColor;

Provides access to RichViewEdit⁽⁵⁹⁾.Color property.

See also:

- RVBackgroundStyle⁽⁶¹⁾
- RVBackgroundBitmap⁽⁶¹⁾

1.3.1.1.57 TSRichViewEdit.RVEditorOptions

Options affecting editing in TRichViewEdit.

property RVEditorOptions: TRVEditorOptions;

Provides access to RichViewEdit⁽⁵⁹⁾.EditorOptions property.

1.3.1.1.58 TSRichViewEdit.RVFooter

Specifies the TRichViewEdit control used for editing footers.

property RVFooter: TRichViewEdit;

Footers are visible if PageProperty⁽⁵⁸⁾.FooterVisible⁽¹³⁶⁾ = True.

When footer editing is activated, a document is loaded from Subdocuments⁽⁶⁶⁾ to RVFooter. Depending on the current page⁽⁵⁰⁾, PageProperty⁽⁵⁸⁾.TitlePage⁽¹⁴¹⁾ and PageProperty⁽⁵⁸⁾.

.FacingPages⁽¹³⁶⁾, a document is loaded from one of the following Subdocuments.Items:

srvhftFirstPageFooter, *srvhftEvenPagesFooter*, *srvhftNormalFooter*. An edited document is saved back when editing is finished, or when the editor needs to be saved or printed.

When this editor is active, it is returned in ActiveEditor⁽⁴⁷⁾ property. Only in this case (RVFooter=ActiveEditor) you can assume that the document in it is valid.

Distances from the page border to the footer are defined in PageProperty⁵⁸:

- LeftMargin¹³⁷ (from the left side)
- RightMargin¹⁴¹ (from the right side)
- FooterY¹³⁶ (from the bottom side)

Margins properties of this editor (RVFooter.LeftMargin, TopMargin, etc.) are ignored when calculating the footer position.

When the footer is being edited, it is visually indicated by the following properties of ViewProperty⁶⁸:

- Texts¹⁶²
- FooterPen¹⁵⁶
- FooterTitleFont¹⁵⁷
- FooterTitleColor¹⁵⁶

Using an external editor (not recommended!)

You can assign an external TRichViewEdit control to this property. Take a note that width and height of this RichViewEdit are set by TSRichViewEdit.

Assigning *nil* to this property makes TSRichViewEdit to use a hidden internal TRichViewEdit for the footer (this is a default behavior). Users cannot see it, it is responsible only for document rendering.

See also:

- RVHeader⁶³
- RVNote⁶⁴
- RichViewEdit⁵⁹
- Document parts in TSRichViewEdit³⁶

1.3.1.1.59 TSRichViewEdit.RVFOptions

Options for reading and saving RVF (RichView Format)

property RVFOptions: TRVFOptions;

This property provides access to RichViewEdit⁵⁹'s property of the same name.

Initial value (for internal RichViewEdit):

```
[rvfoLoadDocProperties, rvfoSaveDocProperties,
 rvfoSavePicturesBody, rvfoSaveControlsBody,
 rvfoIgnoreUnknownPicFmt, rvfoIgnoreUnknownCtrls,
 rvfoConvUnknownStylesToZero, rvfoConvLargeImageIdxToZero,
 rvfoSaveBinary, rvfoSaveBack, rvfoLoadBack,
 rvfoSaveTextStyles, rvfoSaveParaStyles, rvfoSaveLayout,
 rvfoLoadLayout]
```

See also:

- RVFParaStylesReadMode⁶³
- RVFTextStylesReadMode⁶³

1.3.1.1.60 TSRichViewEdit.RVFPaStylEsReadMode

Specifies how paragraph styles and list styles are loaded from RVF files or streams.

property RVFPaStylEsReadMode: TRVFReaderStyleMode;

This property provides access to RichViewEdit⁽⁵⁹⁾'s property of the same name.

See also:

- RVFTextStylEsReadMode⁽⁶³⁾
- RVFOptions⁽⁶²⁾

1.3.1.1.61 TSRichViewEdit.RVFTextStylEsReadMode

Specifies how text styles are loaded from RVF files or streams.

property RVFTextStylEsReadMode: TRVFReaderStyleMode;

This property provides access to RichViewEdit⁽⁵⁹⁾'s property of the same name.

See also:

- RVFPaStylEsReadMode⁽⁶³⁾
- RVFOptions⁽⁶²⁾

1.3.1.1.62 TSRichViewEdit.RVHeader

Specifies the TRichViewEdit control used for editing headers.

property RVHeader: TRichViewEdit;

Headers are visible if PageProperty⁽⁵⁸⁾.HeaderVisible⁽¹³⁷⁾ = True.

When header editing is activated, a document is loaded from Subdocuments⁽⁶⁶⁾ to RVHeader.

Depending on the current page⁽⁵⁰⁾, PageProperty⁽⁵⁸⁾.TitlePage⁽¹⁴¹⁾ and PageProperty⁽⁵⁸⁾.FacingPages⁽¹³⁶⁾, a document is loaded from one of the following Subdocuments.Items:

srvhftFirstPageHeader, *srvhftEvenPagesHeader*, *srvhftNormalHeader*. An edited document is saved back when editing is finished, or when the editor needs to be saved or printed.

When this editor is active, it is returned in ActiveEditor⁽⁴⁷⁾ property. Only in this case (RVHeader=ActiveEditor) you can assume that the document in it is valid.

Distances from the page border to the header are defined in PageProperty⁽⁵⁸⁾:

- LeftMargin⁽¹³⁷⁾ (from the left side)
- RightMargin⁽¹⁴¹⁾ (from the right side)
- HeaderY⁽¹³⁷⁾ (from the top side)

Margins properties of this editor (RVHeader.LeftMargin, TopMargin, etc.) are ignored when calculating the header position.

When the header is being edited, it is visually indicated by the following properties of ViewProperty⁽⁶⁸⁾:

- Texts⁽¹⁶²⁾
- HeaderPen⁽¹⁵⁸⁾
- HeaderTitleFont⁽¹⁵⁸⁾
- HeaderTitleColor⁽¹⁵⁸⁾

Using an external editor (not recommended!)

You can assign an external TRichViewEdit control to this property. Take a note that width and height of this RichViewEdit are set by TSRichViewEdit.

Assigning *nil* to this property makes TSRichViewEdit to use a hidden internal TRichViewEdit for the header (this is a default behavior). Users cannot see it, it is responsible only for document rendering.

See also:

- RVFooter⁶¹
- RVNote⁶⁴
- RichViewEdit⁵⁹
- Document parts in TSRichViewEdit³⁶

1.3.1.1.63 TSRichViewEdit.RVNote

Returns TRichViewEdit control used to edit notes and text boxes.

property RVNote: TRichViewEdit;

Unlike other similar properties (RVHeader⁶³, RVFooter⁶¹, RichViewEdit⁵⁹), the content of this editor is not persistent after document loading. This TRichViewEdit control is used to edit different footnotes/endnotes (see StartEditNote⁹⁹). The content of this editor is defined only if CurrentNote⁵⁰ <> *nil*.

When this editor is active, it is returned in ActiveEditor⁴⁷ property.

Documents of items of the following types can be edited:

- footnotes (TRVFootnoteItemInfo)
- endnotes (TRVEndnoteItemInfo)
- sidenotes (TRVSidenoteItemInfo)
- text box items (TRVTextBoxItemInfo)

See also:

- RVHeader⁶³
- RVFooter⁶¹
- RichViewEdit⁵⁹
- Document parts in TSRichViewEdit³⁶
- Footnotes and endnotes³⁸

1.3.1.1.64 TSRichViewEdit.RVOptions

Options for TRichViewEdit.

property RVOptions: TRVOptions;

Provides access to RichViewEdit⁵⁹.Options property.

Initial value (for internal RichViewEdit):

[*rvoAllowSelection*, *rvoScrollToEnd*, *rvoTagsArePChars*,
rvoAutoCopyText, *rvoAutoCopyRVF*, *rvoAutoCopyImage*,
rvoAutoCopyRTF, *rvoFormatInvalidate*, *rvoDbClickSelectsWord*,
rvoRClickDeselects]

1.3.1.1.65 TSRichViewEdit.ScrolledPage

Returns the index of the first visible page (from 1).

property ScrolledPage: Integer;

This value is not necessary the same as CurrentPage⁽⁵⁰⁾.

1.3.1.1.66 TSRichViewEdit.SkinManager

Determines which skin manager is associated with this editor.

property SkinManager: TSRVSkinManager⁽²²⁷⁾;

SkinManager defines a visual appearance of scrollbars. If it is not specified, scrollbars have a default (no-skin) view.

See also:

- HScrollBarSchemeIndex⁽⁵³⁾
- VScrollBarSchemeIndex⁽⁶⁹⁾

1.3.1.1.67 TSRichViewEdit.SmartPopupProperties

Properties for "smart popups"

property SmartPopupProperties: TRVSmartPopupProperties;

Provides access to RichViewEdit⁽⁵⁹⁾.SmartPopupProperties property.

TSRichViewEdit uses RichViewEdit⁽⁵⁹⁾.SmartPopupProperties not only for "smart popups" in RichViewEdit⁽⁵⁹⁾, but for "smart popups" in other editors as well (RVHeader⁽⁶³⁾, RVFooter⁽⁶¹⁾, RVNote⁽⁶⁴⁾). Values of SmartPopupProperties of these other editors are ignored.

You can show/hide a button using SmartPopupVisible⁽⁶⁵⁾ property.

See also:

- OnSmartPopupClick⁽¹²⁰⁾

1.3.1.1.68 TSRichViewEdit.SmartPopupVisible

Shows/hides a "smart popup" button for the current item in ActiveEditor⁽⁴⁷⁾.

property SmartPopupVisible: Boolean;

Properties of this button are defined in SmartPopupProperties⁽⁶⁵⁾.

See also:

- OnSmartPopupClick⁽¹²⁰⁾

1.3.1.1.69 TSRichViewEdit.StyleTemplateInsertMode

Controls how RichViewEdit⁽⁵⁹⁾.Style.StyleTemplates affect insertion of RTF and RVF files or streams.

property StyleTemplateInsertMode: TRVStyleTemplateInsertMode;

This property provides access to RichViewEdit⁽⁵⁹⁾'s property of the same name.

Default value

rvstimUseTargetFormatting

1.3.1.1.70 TSRichViewEdit.Subdocuments

A list of headers and footers.

type

```
TSRVSubDocuments = class (TPersistent)
```

property SubDocuments: TSRVSubDocuments;

The main property of TSRVSubDocuments is Items:

type

```
TSRVHeaderFooterType = (
    srvhftFirstPageHeader, srvhftEvenPagesHeader, srvhftNormalHeader,
    srvhftFirstPageFooter, srvhftEvenPagesFooter, srvhftNormalFooter);
TSRVSubDocumentRVData = class (TCustomRVData)
```

property Items[Index: TSRVHeaderFooterType]: TSRVSubDocumentRVData;

The Index specifies the header or footer type:

Value	Meaning	Used if...
<i>srvhftFirstPageHeader</i>	Header for the first page	PageProperty ⁽⁵⁸⁾ .TitlePage ⁽¹⁴¹⁾ =True
<i>srvhftEvenPagesHeader</i>	Header for even pages	PageProperty.FacingPages ⁽¹³⁶⁾ =True
<i>srvhftNormalHeader</i>	Normal header (for other pages)	Always
<i>srvhftFirstPageFooter</i>	Footer for the first page	PageProperty.TitlePage=True
<i>srvhftEvenPagesFooter</i>	Footer for even pages	PageProperty.FacingPages=True
<i>srvhftNormalFooter</i>	Normal footer (for other pages)	Always

Items is the default property, so you can access (for example) the first page header as `SRichViewEdit.Subdocuments[srvhftFirstPageHeader]`.

Headers are shown and used only if PageProperty⁽⁵⁸⁾.HeaderVisible⁽¹³⁷⁾=True. Footers are shown and used only if PageProperty.FooterVisible⁽¹³⁶⁾=True.

Each subdocument is linked to its own TRVStyle object. You can access it as `SRichViewEdit.Subdocuments[srvhftFirstPageHeader].GetRVStyle`. RichViewEdit⁽⁵⁹⁾.Style is assigned as MainRVStyle to these TRVStyle objects, so they all use a common collection of StyleTemplates.

When an event occurs in some subdocument, RichViewEdit⁽⁵⁹⁾ is passed as the Sender parameter of this event.

Editing

When a header is activated for editing (by double clicking or by calling StartEditing⁽⁹⁸⁾), a document is loaded in RVHeader⁽⁶³⁾. When a footer is activated for editing, a document is loaded in RVFooter⁽⁶¹⁾. An edited document is saved back to Subdocuments when the editing is finished, or when the main document needs to be saved or printed. The process of saving and loading data between Subdocuments and RVHeader/RVFooter is completely automatic and transparent for the user.

Saving to files

Headers and footers can be saved in RVF, RTF, DocX files, loaded from RVF, RTF files.

PageProperty.TitlePage and FacingPages only turn on/off displaying of the specific headers and footers. They do not affect saving and loading: all headers and footers are stored.

Assigning PageProperty.HeaderVisible=False prevents saving and loading headers.

Assigning PageProperty.FooterVisible=False prevents saving and loading footers.

1.3.1.1.71 TSRichViewEdit.TabNavigation

Mode of **Tab** and **Shift + Tab** navigation

property TabNavigation: TRVTabNavigationType;

This property provides access to RichViewEdit⁽⁵⁹⁾'s property of the same name.

This property does nothing in TSRichViewEdit. Reserved for future.

1.3.1.1.72 TSRichViewEdit.TextEngine

Specifies the text API which is used to draw text.

property TextEngine: TRVTextEngine;

Uniscribe provides reliable processing of bidirected text and complex scripts.

Windows API is faster.

Default value:

rvteUniscribe

1.3.1.1.73 TSRichViewEdit.TopMarginPix

Returns the top margin, in pixels.

property TopMarginPix: Integer;

property TopMargin100Pix: Integer;

The top margin is defined in PageProperty⁽⁵⁸⁾.TopMargin⁽¹⁴²⁾. These properties return this margin converted to pixels.

TopMarginPix is scaled according to ViewProperty⁽⁶⁸⁾.ZoomPercent⁽¹⁶⁷⁾ (so this property returns the margin as it is displayed on the screen).

TopMargin100Pix is not scaled, it is the margin in 96 DPI and 100% zooming:

TopMarginPix := TopMargin100Pix * (ZoomPercent / 100) * (<screen DPI> / 96).

See also:

- LeftMarginPix⁽⁵⁴⁾
- RightMarginPix⁽⁶⁰⁾
- BottomMarginPix⁽⁴⁸⁾

1.3.1.1.74 TSRichViewEdit.UnitsProgram

Measurement units for page margins and size.

property UnitsProgram: TRVUnits;

This property defines measurement units for:

- page size and margins properties of PageProperty⁽⁵⁸⁾,
- PageWidth⁽¹⁶¹⁾ and PageHeight⁽¹⁶¹⁾ properties of ViewProperty⁽⁶⁸⁾.

Note: "pixels" unit assumes 96 dpi (used internally in ScaleRichView).

Default value:

rvuMillimeters

1.3.1.1.75 TSRichViewEdit.UseDrawHyperlinksEvent

Specifies whether OnDrawHyperlink⁽¹⁰⁷⁾ event is used.

property UseDrawHyperlinksEvent: Boolean;

This property may be useful if you want to use this event only when drawing on external canvas (DrawMetafile⁽⁷⁶⁾, DrawPage⁽⁷⁶⁾).

Default value:

True

1.3.1.1.76 TSRichViewEdit.UseStyleTemplates

Allows or disallows using RichViewEdit⁽⁵⁹⁾.Style.StyleTemplates ("real styles").

property UseStyleTemplates: Boolean;

This property provides access to RichViewEdit⁽⁵⁹⁾'s property of the same name.

Default value:

False

1.3.1.1.77 TSRichViewEdit.Version

Returns the version of ScaleRichView.

property Version : TRVUnicodeString;

It is a string like 'v1.6.0'.

1.3.1.1.78 TSRichViewEdit.ViewProperty

Contains settings for document displaying.

property ViewProperty: TSRVViewProperty⁽¹⁵³⁾;

1.3.1.1.79 TSRichViewEdit.VMaxScrollPos

Defines the maximum position of vertical scroll bar.

property VMaxScrollPos: Integer;

This value is used to calculate the maximum possible value for VScrollPos⁽⁶⁹⁾.

This is a read-only property.

See also:

- HMaxScrollPos⁽⁵²⁾

1.3.1.1.80 TSVRichViewEdit.VScrollBar

Shows/hides the vertical scrollbar.

property VScrollBar: Boolean;

Default value:

True (visible)

1.3.1.1.81 TSVRichViewEdit.ScrollBarControlStyle

Defines appearance of scrollbars

property ScrollBarControlStyle: TSVControlStyle⁽²⁷⁵⁾;

Internally, scrollbars are represented by TSVScrollBar⁽²⁴²⁾ controls. Value of this property is assigned to SRVControlStyle⁽²⁴⁷⁾ property of these scrollbars.

Default value:

svcsClassic

See also

- ViewProperty⁽⁶⁸⁾.DarkModeUI⁽¹⁵⁶⁾
- HScrollBarSchemeIndex⁽⁵³⁾
- VScrollBarSchemeIndex⁽⁶⁹⁾

1.3.1.1.82 TSVRichViewEdit.VScrollBarSchemeIndex

Determines which skin scheme is used for the vertical scrollbar.

property VScrollBarSchemeIndex: Integer;

This is an index in SkinManager⁽⁶⁵⁾.CurrentSkin⁽²²⁹⁾.VerticalScrollBarSchemes⁽²³²⁾ collection.

If SkinManager⁽⁶⁵⁾ = nil, or SkinManager⁽⁶⁵⁾.CurrentSkin⁽²²⁹⁾ = nil, or this index is not valid for this collection (negative or too large), then this scrollbar has a default (no-skin) appearance.

Default value

0

See also

- HScrollBarSchemeIndex⁽⁵³⁾
- ScrollBarControlStyle⁽⁶⁹⁾

1.3.1.1.83 TSVRichViewEdit.VScrollPos

Defines the position of vertical scroll bar, measured in screen pixels.

property VScrollPos: Integer;

You can use this property for scrolling to the specified position.

Value of this property is in the range from 0 to VMaxScrollPos⁽⁶⁸⁾ - (ClientWidth div RichViewEdit⁽⁵⁹⁾.VSmallStep) + 1

See also:

- HScrollPos ⁽⁵³⁾

1.3.1.2 Methods**In TSRichViewEdit**

BeginComponentsUpdate ⁽⁷²⁾
 CalculateAllPagePositions ⁽⁷²⁾
 CalculatePageCount ⁽⁷²⁾
 CalculatePagePosition ⁽⁷²⁾
 Clear ⁽⁷³⁾
 ConvertRVtoSRV ⁽⁷³⁾
 ConvertSRVtoRV ⁽⁷³⁾
 ConvertToPixels, ConvertToTwips, ConvertToDifferentUnits ⁽⁷⁴⁾
 Copy ⁽⁷⁵⁾
 Create ⁽⁷⁵⁾
 Cut ⁽⁷⁵⁾
 DeletePage ⁽⁷⁵⁾
 Destroy ⁽⁷⁶⁾
 DrawMetafile ⁽⁷⁶⁾
 DrawPage ⁽⁷⁶⁾
 EndComponentsUpdate ⁽⁷⁷⁾
 FindNextCheckpoint ⁽⁷⁷⁾
 FindNextHeading ⁽⁷⁸⁾
 FindNextHyperlink ⁽⁷⁸⁾
 FindNextItem ⁽⁷⁹⁾
 FindPriorCheckpoint ⁽⁸⁰⁾
 FindPriorHeading ⁽⁸⁰⁾
 FindPriorHyperlink ⁽⁸¹⁾
 FindPriorItem ⁽⁸¹⁾
 FirstCurPage ⁽⁸²⁾
 Format ⁽⁸²⁾
 GetCaretPosInUnits ⁽⁸³⁾
 GetCurrentLineCol ⁽⁸³⁾
 GetFirstItemVisibleRV ⁽⁸³⁾
 GetFirstVisiblePage ⁽⁸³⁾
 GetItemAt ⁽⁸⁴⁾
 GetItemBounds, GetItemBounds100 ⁽⁸⁴⁾
 GetItemCoords ⁽⁸⁵⁾
 GetItemPages ⁽⁸⁶⁾
 GetLastItemVisibleRV ⁽⁸⁶⁾
 GetLastVisiblePage ⁽⁸⁶⁾
 GetLeftMarginPix, GetLeftMargin100Pix ⁽⁵⁴⁾
 GetPageAt ⁽⁸⁶⁾
 GetPageClientRect ⁽⁸⁷⁾
 GetPageLastItemNo ⁽⁸⁷⁾
 GetPageNo ⁽⁸⁷⁾

GetPageStartItemNo⁽⁸⁸⁾
GetPageStartTableItem⁽⁸⁸⁾
GetPageStartTableRow⁽⁸⁸⁾
GetRightMarginPix, GetRightMargin100Pix⁽⁶⁰⁾
GetTableIconItem⁽⁸⁹⁾
GetTableIconRVData⁽⁸⁹⁾
InsertPageBreak⁽⁸⁹⁾
InZoomRect⁽⁸⁹⁾
LastCurPage⁽⁹⁰⁾
LoadRVF⁽⁹⁰⁾
LoadRVFFromStream⁽⁹⁰⁾
NextCurCheckpoint⁽⁹⁰⁾
NextCurHeading⁽⁹⁰⁾
NextCurHyperlink⁽⁹¹⁾
NextCurItem⁽⁹¹⁾
NextCurPage⁽⁹¹⁾
OutZoomRect⁽⁹²⁾
Paste⁽⁹²⁾
PrintAll⁽⁹²⁾
PrintCurrent⁽⁹³⁾
PrintRange⁽⁹³⁾
PriorCurCheckpoint⁽⁹⁴⁾
PriorCurHeading⁽⁹⁴⁾
PriorCurHyperlink⁽⁹⁴⁾
PriorCurItem⁽⁹⁴⁾
PriorCurPage⁽⁹⁵⁾
ProcessControls⁽⁹⁵⁾
Reformat⁽⁸²⁾
ReturnToNote⁽⁹⁵⁾
ScrollToCaret⁽⁹⁵⁾
ScrollToItem⁽⁹⁶⁾
SelectAll⁽⁹⁶⁾
SetFloatPropertyEd⁽⁹⁶⁾
SetIntPropertyEd⁽⁹⁶⁾
SetMargin⁽⁹⁷⁾
SetMarginMM⁽⁹⁷⁾
SetMarginUnit⁽⁹⁸⁾
SetRVMargins⁽⁹⁸⁾
StartEditing⁽⁹⁸⁾
StartEditNote⁽⁹⁹⁾
UnitsPerInchH⁽⁹⁹⁾
UnitsPerInchV⁽¹⁰⁰⁾
UpdateBuffer, UpdateBufferAt⁽¹⁰⁰⁾
ZoomCanvas, RestoreCanvasZoom⁽¹⁰⁰⁾
ZoomRectIn, ZoomRectOut⁽¹⁰⁰⁾
ZoomToFullPages⁽¹⁰¹⁾

1.3.1.2.1 TSRichViewEdit.BeginComponentsUpdate

Locks redrawing of inserted controls.

procedure BeginComponentsUpdate;

Inserted components will not be repainted until EndComponentsUpdate⁽⁷⁷⁾ is called.

Calls to BeginComponentsUpdate and EndComponentsUpdate may be nested.

If redrawing of inserted controls is locked, repainting is faster.

1.3.1.2.2 TSRichViewEdit.CalculateAllPagePositions

Recalculates positions, zooming and Z-orders for all pages.

procedure CalculateAllPagePositions;

This method calls OnGetPagePos⁽¹⁰⁸⁾ event for all pages.

This event is used only if ViewProperty⁽⁶⁸⁾.FreePosPage⁽¹⁵⁷⁾ = *True*. Usually page positions are calculated automatically. Call this method if you want to recalculate them for existing pages.

Z-orders are recalculated only if PageProperty⁽⁵⁸⁾.UsePageOrders⁽¹⁴²⁾ = *True*.

See also:

- CalculatePagePosition⁽⁷²⁾

1.3.1.2.3 TSRichViewEdit.CalculatePageCount

Repaginates and returns the page count.

function CalculatePageCount: Integer;

Usually, if a document is changed by editing-style methods, it is repaginated automatically.

Call this method in the following cases:

- if you modified a document using viewer-style methods (methods that do not call OnChange⁽¹⁰⁴⁾)
- if you blocked updating with CanUpdate⁽⁴⁸⁾ property

Return value:

count of pages

See also:

- PageCount⁽⁵⁷⁾ property

1.3.1.2.4 TSRichViewEdit.CalculatePagePosition

Recalculates position, zooming and Z-order for the specified page.

function CalculatePagePosition(PageNo : Integer): Boolean;

PageNo is a page index, from 1.

This method calls OnGetPagePos⁽¹⁰⁸⁾ event for this page.

This event is used only if ViewProperty⁽⁶⁸⁾.FreePosPage⁽¹⁵⁷⁾ = *True*. Usually page positions are calculated automatically. Call this method if you want to recalculate them for the existing page.

Z-orders are recalculated only if PageProperty⁽⁵⁸⁾.UsePageOrders⁽¹⁴²⁾ = *True*.

See also:

- CalculateAllPagePositions ⁽⁷²⁾

1.3.1.2.5 TSRichViewEdit.Clear

Clears the whole editor content and deletes all unused styles.

```
procedure Clear;
```

This method calls Clear and DeleteUnusedStyles(True,True,True) for:

- RichViewEdit ⁽⁵⁹⁾
- Subdocuments ⁽⁶⁶⁾
- RVHeader ⁽⁶³⁾
- RVFooter ⁽⁶¹⁾

See also:

- Document parts in TSRichViewEdit ⁽³⁶⁾

1.3.1.2.6 TSRichViewEdit.ConvertRVtoSRV

Converts client coordinates in ActiveEditor ⁽⁴⁷⁾ to client coordinates of this TSRichViewEdit component.

```
function ConvertRVtoSRV(p: TPoint) : TPoint;
```

Input parameters:

p – coordinates relative to the top left corner of ActiveEditor ⁽⁴⁷⁾.

Return value:

coordinates relative to the top left corner of TSRichViewEdit's window.

See also

- ConvertSRVtoRV ⁽⁷³⁾

1.3.1.2.7 TSRichViewEdit.ConvertSRVtoRV

Converts client coordinates in this TSRichViewEdit component to client coordinates of ActiveEditor ⁽⁴⁷⁾.

```
function ConvertSRVtoRV(p: TPoint; var bInTextArea, bInLeft, bInPageArea: Boolean);
```

Input parameters:

p – coordinates relative to the top left corner of TSRichViewEdit's window.

Output parameters:

bInTextArea – *True* if **p** is inside of some page's content area.

bInLeft – *True* if **p** is to the left of some page (in the "line selection" area).

InPageArea – *True* if **p** is inside of some page's area (above content or above margins).

Return value:

coordinates relative to the top left corner of ActiveEditor ⁽⁴⁷⁾.

See also:

- ConvertRVtoSRV ⁽⁷³⁾

Example:

This code assigns text of the item at the position of the mouse pointer to Label1.Caption

// OnMouseMove event

```
procedure TForm3.SRichViewEdit1MouseMove(Sender: TObject;
  Shift: TShiftState; X, Y: Integer);
var pt: TPoint;
    RVData: TCustomRVFormattedData;
    InPageArea, InLeftArea, InTextArea: Boolean;
    ItemNo, Offs: Integer;
begin
  pt := SRichViewEdit1.ConvertSRVtoRV(Point(X, Y), InTextArea,
    InLeftArea, InPageArea);
  if InTextArea then begin
    pt := SRichViewEdit1.ActiveEditor(47).ClientToDocument(pt);
    SRichViewEdit1.ActiveEditor(47).GetItemAt(pt.X, pt.Y,
      RVData, ItemNo, Offs, True);
    if ItemNo >= 0 then
      Label1.Caption := RVData.GetItemText(ItemNo)
    else // in page, but not in item
      Label1.Caption := '';
    end
  else // not in page
    Label1.Caption := '';
  end;
```

1.3.1.2.8 TSRichViewEdit.ConvertTo...

The methods convert lengths in the documents and RVStyle components to twips, pixels or EMU.

```
procedure ConvertDocToTwips;
procedure ConvertDocToPixels;
procedure ConvertDocToEMU;
procedure ConvertDocToDifferentUnits(NewUnits: TRVStyleUnits);
```

The methods convert documents and RVStyles of the following properties to new units:

- RichViewEdit⁽⁵⁹⁾
- Subdocuments⁽⁶⁶⁾
- RVFooter⁽⁶¹⁾
- RVHeader⁽⁶³⁾
- RVNote⁽⁶⁴⁾ (if a note is being edited)

The methods convert documents in these editors (using ConvertDocTo* methods) and their RVStyles (using Style.ConvertTo* methods).

If you do not change RVStyles of these editors (and *rvfoCanChangeUnits* is not included in their RVFOptions), it is enough to call a conversion method one time, unit of measurement will be kept.

1.3.1.2.9 TSRichViewEdit.Copy

Copies the selected fragment to the Clipboard. Copies in all supported formats (such as text, RVF, RTF and images).

procedure Copy;

This procedure does nothing if the selection is empty.

This procedure empties the Clipboard before copying.

This method should be called only when document is formatted.

This method calls RichViewEdit⁽⁵⁹⁾.CopyDef.

See also:

- OnCopy⁽¹⁰⁶⁾

1.3.1.2.10 TSRichViewEdit.Create

A constructor, creates a new instance of TSRichViewEdit.

constructor Create(AOwner: TComponent);

AOwner is another component, typically the form, that is responsible for freeing the control. It becomes the value of the Owner property.

Controls placed on form at design time are created automatically.

1.3.1.2.11 TSRichViewEdit.Cut

Cuts the selected fragment to the Clipboard.

procedure Cut;

This procedure does nothing if the selection is empty.

This procedure empties the Clipboard before copying.

This method should be called only when document is formatted.

This method calls RichViewEdit⁽⁵⁹⁾.CutDef.

See also:

- OnCopy⁽¹⁰⁶⁾

1.3.1.2.12 TSRichViewEdit.DeletePage

Deletes content of the **PageNo**-th page, as an editing operation.

procedure DeletePage(PageNo : Integer);

It's not guaranteed that after calling this method other pages will not be changed. It's not even guaranteed that page count will be decreased exactly by 1.

1.3.1.2.13 TSRichViewEdit.Destroy

Disposes of the component.

destructor Destroy; **override**;

Do not call Destroy directly. Call Free instead. Free verifies that the component is not nil, and only then calls Destroy.

Never explicitly free a component in one of its own event handlers, nor free a component from the event handler of a component that it owns or contains.

Note: A form owns all the controls and non-visual components that are placed on it in design mode. When it is freed, all of these components are automatically freed as well.

1.3.1.2.14 TSRichViewEdit.DrawMetafile

Paints the specified page in the metafile.

```
procedure DrawMetafile(PageNo : Integer; PicturePage : TMetafile;
  PageNoVisible, ClipMargins, Printing : Boolean;
  NoMetafiles: Boolean = False);
```

PageNo – index of page, from 1.

PicturePage – a metafile to draw. Page image will be scaled to **PicturePage.Width** x **PicturePage.Height**.

PageNoVisible specifies whether to print page numbers (see the properties of PageProperty⁵⁸).

ClipMargins specifies whether all drawing outside the document area (i.e. drawing on margins) will be cropped.

Printing specifies whether this is a drawing for printing (a special method for drawing pictures is used in this case).

The parameter **NoMetafiles** is used only if **Printing = True**. If **NoMetafiles = False**, metafiles are drawn as they are. If **True**, metafiles are drawn as bitmaps. This option is useful when drawing on canvases of components (such as PDF generators) that do not handle embedded metafiles properly.

See also:

- DrawPage⁷⁶
- OnDrawHyperlink¹⁰⁷

1.3.1.2.15 TSRichViewEdit.DrawPage

Draws the **PageNo**-th page onto the specified canvas.

```
procedure DrawPage(PageNo, PageWd, PageHt, OffX, OffY: Integer;
  CanvasPage: TCanvas; PageNoVisible, ClipMargins, Printing: Boolean;
  UseWordPainters: Boolean; ForMetafile: Boolean = False;
  NoMetafiles: Boolean = False);
```

PageNo – index of page, from 1.

CanvasPage – a canvas where to draw.

Bounds(OffX, OffY, PageWd, PageHt) defines a rectangle on **CanvasPage** where the page will be drawn. Page image will be scaled to **PageWd** x **PageHt**.

PageNoVisible specifies whether to print page numbers (see the properties of **PageProperty**⁵⁸).

ClipMargins specifies whether all drawing outside the document area (i.e. drawing on margins) will be cropped.

Printing specifies whether this is a drawing for printing (a special method for drawing pictures is used in this case).

If **UseWordPainters**=*True*, live spelling underlines will be drawn.

If **ForMetafile**=*False*, printing output may be incompatible with metafiles; the component may use font glyph indexes instead of character codes when printing text. Such metafiles will be displayed incorrectly on computers without this font or with another version of this font.

If **ForMetafile**=*True*, the component always uses character codes when printing text, to create metafile-compatible output.

The parameter **NoMetafiles** is used only if **Printing** = *True*. If **NoMetafiles** = *False*, metafiles are drawn as they are. If *True*, metafiles are drawn as bitmaps. This option is useful when drawing on canvases of components (such as PDF generators) that do not handle embedded metafiles properly.

See also:

- DrawMetafile⁷⁶
- OnDrawHyperlink¹⁰⁷

1.3.1.2.16 TSRichViewEdit.EndComponentsUpdate

Unlocks redrawing of inserted controls.

procedure EndComponentsUpdate;

Unlocks redrawing of inserted controls, locked by BeginComponentsUpdate⁷².

Calls to BeginComponentsUpdate and EndComponentsUpdate may be nested.

1.3.1.2.17 TSRichViewEdit.FindNextCheckpoint

Finds the location of the next *checkpoint*.

function FindNextCheckpoint(**var** RVData : TCustomRVData;
var ItemNo : Integer; CheckRaiseEvent : Boolean) : Boolean;

Unlike TCustomRichView's methods for *checkpoints* enumeration, this method correctly processes *checkpoints* in table cells.

The search can be limited to the specified range of coordinates, see RangeSearch⁵⁸.

Input parameters:

RVData, **ItemNo** – starting position for the search. The search starts from the next item. **RVData** is a document (RichViewEdit⁵⁹).RVData, table cell, or cell inplace editor's RVData; if the cell is edited, it must be an inplace editor's RVData), **ItemNo** is an index of item in this document.

CheckRaiseEvent – if *True*, searches only for *checkpoints* with "raise event" flag.

Output parameters:

RVData, ItemNo – position of the found *checkpoint*. Valid only if the method returns *True*.

Return value:

True if found.

See also:

- FindPriorCheckpoint⁽⁸⁰⁾
- NextCurCheckpoint⁽⁹⁰⁾
- FindNextHyperlink⁽⁷⁸⁾
- FindNextItem⁽⁷⁹⁾

1.3.1.2.18 TSRichViewEdit.FindNextHeading

Finds the location of the next heading.

```
function FindNextHeading(var RVData: TCustomRVData;  
    var ItemNo: Integer): Boolean;
```

The search can be limited to the specified range of coordinates, see RangeSearch⁽⁵⁸⁾.

Heading is a paragraph with OutlineLevel property > 0.

Input parameters:

RVData, ItemNo – starting position for the search. The search starts from the next item. **RVData** is a document (RichViewEdit⁽⁵⁹⁾.RVData, table cell, or cell inplace editor's RVData; if the cell is edited, it must be an inplace editor's RVData), **ItemNo** is an index of item in this document.

Output parameters:

RVData, ItemNo – position of the found item. Valid only if the method returns *True*.

Return value:

True if found.

See also:

- FindPriorItem⁽⁸¹⁾
- NextCurItem⁽⁹¹⁾
- FindNextCheckpoint⁽⁷⁷⁾
- FindNextHyperlink⁽⁷⁸⁾

1.3.1.2.19 TSRichViewEdit.FindNextHyperlink

Finds the location of the next hyperlink.

```
function FindNextHyperlink(var RVData: TCustomRVData;  
    var ItemNo: Integer): Boolean;
```

The search can be limited to the specified range of coordinates, see RangeSearch⁽⁵⁸⁾.

Input parameters:

RVData, ItemNo – starting position for the search. The search starts from the next item. **RVData** is a document (RichViewEdit⁽⁵⁹⁾.RVData, table cell, or cell inplace editor's RVData; if the cell is edited, it must be an inplace editor's RVData), **ItemNo** is an index of item in this document.

Output parameters:

RVData, ItemNo – position of the found hyperlink. Valid only if the method returns *True*.

Return value:

True if found.

See also:

- FindPriorHyperlink⁽⁸¹⁾
- NextCurHyperlink⁽⁹¹⁾
- FindNextCheckpoint⁽⁷⁷⁾
- FindNextItem⁽⁷⁹⁾

1.3.1.2.20 TSRichViewEdit.FindNextItem

Finds the location of the next item of the specified styles (types).

```
function FindNextItem(StyleNo: array of Integer;  
    var RVData: TCustomRVData; var ItemNo: Integer) : Boolean;
```

The search can be limited to the specified range of coordinates, see RangeSearch⁽⁵⁸⁾.

Input parameters:

StyleNo – a list of styles (types) for searching.

RVData, ItemNo – starting position for the search. The search starts from the next item. **RVData** is a document (RichViewEdit⁽⁵⁹⁾.RVData, table cell, or cell inplace editor's RVData; if the cell is edited, it must be an inplace editor's RVData), **ItemNo** is an index of item in this document.

Output parameters:

RVData, ItemNo – position of the found item. Valid only if the method returns *True*.

Return value:

True if found.

Example:

```
var  
    RVData: TCustomRVData;  
    ItemNo: Integer;  
    ...  
    // searching for the first image in the document  
    RVData := SRichViewEdit1.RichViewEdit(59).RVData;  
    ItemNo := -1;  
    if SRichViewEdit1.FindNextItem(  
        [rvsPicture, rvsHotPicture, rvsBullet, rvsHotspot],  
        RVData, ItemNo) then  
        ...
```

See also:

- FindPriorItem⁽⁸¹⁾
- NextCurlItem⁽⁹¹⁾
- FindNextCheckpoint⁽⁷⁷⁾
- FindNextHyperlink⁽⁷⁸⁾

1.3.1.2.21 TSRichViewEdit.FindPriorCheckpoint

Finds the location of the previous *checkpoint*.

```
function FindNextCheckpoint(var RVData : TCustomRVData;  
    var ItemNo : Integer; CheckRaiseEvent : Boolean) : Boolean;
```

Unlike TCustomRichView's methods for *checkpoints* enumeration, this method correctly processes *checkpoints* in table cells.

The search can be limited to the specified range of coordinates, see RangeSearch⁽⁵⁸⁾.

Input parameters:

RVData, **ItemNo** – starting position for the search. The search starts from the previous item.

RVData is a document (RichViewEdit⁽⁵⁹⁾.RVData, table cell, or cell inplace editor's RVData; if the cell is edited, it must be an inplace editor's RVData), **ItemNo** is an index of item in this document.

CheckRaiseEvent – if *True*, searches only for *checkpoints* with "raise event" flag.

Output parameters:

RVData, **ItemNo** – position of the found *checkpoint*. Valid only if the method returns *True*.

Return value:

True if found.

See also:

- FindNextCheckpoint⁽⁷⁷⁾
- PriorCurCheckpoint⁽⁹⁴⁾
- FindPriorHyperlink⁽⁸¹⁾
- FindPriorItem⁽⁸¹⁾

1.3.1.2.22 TSRichViewEdit.FindPriorHeading

Finds the location of the next heading.

```
function FindPriorHeading(var RVData: TCustomRVData;  
    var ItemNo: Integer): Boolean;
```

The search can be limited to the specified range of coordinates, see RangeSearch⁽⁵⁸⁾.

Heading is a paragraph with OutlineLevel property > 0.

Input parameters:

RVData, **ItemNo** – starting position for the search. The search starts from the previous item.

RVData is a document (RichViewEdit⁽⁵⁹⁾.RVData, table cell, or cell inplace editor's RVData; if the cell is edited, it must be an inplace editor's RVData), **ItemNo** is an index of item in this document.

Output parameters:

RVData, **ItemNo** – position of the found item. Valid only if the method returns *True*.

Return value:

True if found.

See also:

- FindNextItem⁽⁷⁹⁾
- PriorCurlItem⁽⁹⁴⁾

- FindPriorCheckpoint⁽⁸⁰⁾
- FindPriorHyperlink⁽⁸¹⁾

1.3.1.2.23 TSRichViewEdit.FindPriorHyperlink

Finds the location of the previous hyperlink.

```
function FindPriorHyperlink(var RVData: TCustomRVData;  
    var ItemNo: Integer): Boolean;
```

The search can be limited to the specified range of coordinates, see RangeSearch⁽⁵⁸⁾.

Input parameters:

RVData, **ItemNo** – starting position for the search. The search starts from the previous item.

RVData is a document (RichViewEdit⁽⁵⁹⁾.RVData, table cell, or cell inplace editor's RVData; if the cell is edited, it must be an inplace editor's RVData), **ItemNo** is an index of item in this document.

Output parameters:

RVData, **ItemNo** – position of the found hyperlink. Valid only if the method returns *True*.

Return value:

True if found.

See also:

- FindNextHyperlink⁽⁷⁸⁾
- PriorCurHyperlink⁽⁹⁴⁾
- FindPriorCheckpoint⁽⁸⁰⁾
- FindPriorItem⁽⁸¹⁾

1.3.1.2.24 TSRichViewEdit.FindPriorItem

Finds the location of the next item of the specified styles (types).

```
function FindPriorItem(StyleNo: array of Integer;  
    var RVData: TCustomRVData; var ItemNo: Integer) : Boolean;
```

The search can be limited to the specified range of coordinates, see RangeSearch⁽⁵⁸⁾.

Input parameters:

StyleNo – a list of styles (types) for searching.

RVData, **ItemNo** – starting position for the search. The search starts from the previous item.

RVData is a document (RichViewEdit⁽⁵⁹⁾.RVData, table cell, or cell inplace editor's RVData; if the cell is edited, it must be an inplace editor's RVData), **ItemNo** is an index of item in this document.

Output parameters:

RVData, **ItemNo** – position of the found item. Valid only if the method returns *True*.

Return value:

True if found.

Example:

```
var  
    RVData: TCustomRVData;  
    ItemNo: Integer;
```

```

...
// searching for the last image in the document
RVData := SRichViewEdit1.RichViewEdit59.RVData;
ItemNo := SRichViewEdit1.RichViewEdit59.ItemCount;
if SRichViewEdit1.FindPriorItem(
    [rvsPicture, rvsHotPicture, rvsBullet, rvsHotspot],
    RVData, ItemNo) then
...

```

See also:

- FindNextItem⁷⁹
- PriorCurlItem⁹⁴
- FindPriorCheckpoint⁸⁰
- FindPriorHyperlink⁸¹

1.3.1.2.25 TSRichViewEdit.FirstCurPage

Moves the caret to the beginning of the first page.

procedure FirstCurPage(SmoothScroll: Boolean = False);

The same affect as CurrentPage⁵⁰ := 1, but may apply a smooth scrolling effect.

See also:

- PriorCurPage⁹⁵, NextCurPage⁹¹, LastCurPage⁹⁰

1.3.1.2.26 TSRichViewEdit.Format, Reformat

The methods format the document completely. Only formatted documents can be displayed or edited. In addition, this methods copy properties (document size, margins, etc.) from RichViewEdit⁵⁹.DocParameters to PageProperty⁵⁸.

procedure Format;

procedure Reformat;

The difference between these methods is the following. **Format** does not require document to be formatted before its call, and it moves the caret to the beginning. **Reformat** can be called only if the document is already formatted, and it keeps the selection and the caret position.

Format/Reformat do the following work:

- loads header and footer from Subdocuments⁶⁶ to RVHeader⁶³ and RVFooter⁶¹
- SetRVMargins⁹⁸(True/False).
- paginates the document

Call **Format** after loading files in RichViewEdit⁵⁹.

Call **Reformat** after calling SetIntPropertyEd⁹⁶(..., False)/SetFloatPropetyEd⁹⁶(...False).

See also:

- Document parts in TSRichViewEdit³⁶

1.3.1.2.27 TSRichViewEdit.GetCaretPosInUnits

Returns the caret coordinates relative to the top left corner of the page (CurrentPage⁵⁰).

```
function GetCaretPosInUnits(Units: TRVUnits;
    var CaretPosX, CaretPosY: TRVLength) : Boolean;
```

Units define the measurement units for the output coordinates. Note: "pixels" unit assumes 96 dpi (used internally in ScaleRichView).

The returned values are counted in zoom = 100%.

The coordinates are returned in **CaretPosX**, **CaretPosY**.

Return value:

True if the caret position is successfully returned.

See also:

- CaretPos⁴⁹
- OnCaretMove¹⁰⁴

1.3.1.2.28 TSRichViewEdit.GetCurrentLineCol

Returns a line and column at the caret position in ActiveEditor⁴⁷.

```
procedure GetCurrentLineCol(var Line, Column: Integer);
```

Number of the first line on the page is 1. Number of the leftmost caret position in the line is 1.

If the caret is in a table cell, the method returns values for this table.

See also:

- LineNumberProperty⁵⁴

1.3.1.2.29 TSRichViewEdit.GetFirstItemVisibleRV

Returns the first visible item.

```
procedure GetFirstItemVisibleRV(var RVData: TCustomRVFormattedData;
    var ItemNo, OffsetInItem: Integer);
```

This function returns the position where the caret would be placed if the user clicked at the top left corner of TSRichViewEdit control's window.

RVData is a document (RichViewEdit⁵⁹.RVData, table cell, or cell inplace editor's RVData), **ItemNo** is an index of item in this document, **OffsetInItem** is a position inside this item.

See also:

- GetLastItemVisibleRV⁸⁶

1.3.1.2.30 TSRichViewEdit.GetFirstVisiblePage

Returns the index of the first page visible on the screen (at least partially), from 1

```
function GetFirstVisiblePage: Integer;
```

See also:

- GetLastVisiblePage⁸⁶

1.3.1.2.31 TSRichViewEdit.GetItemAt

Returns the ActiveEditor⁽⁴⁷⁾'s item at the specified coordinates.

```
function GetItemAt(X, Y : Integer; var RVData: TCustomRVFormattedData;
var ItemNo, OffsetInItem: Integer; Strict: Boolean): Boolean;
```

Input parameters:

X, Y – coordinates relative to the client area of TSRichViewEdit control.

If **Strict=True**, the method returns the item only if it is located exactly at the specified position (if it exists). If **False**, the method returns the closest item for any position above a page.

Output parameters:

Output parameters are valid only if this method returns *True*.

RVData – object containing this item (ActiveEditor⁽⁴⁷⁾.RVData or table cell or RVData of cell inplace editor)

ItemNo – index of item (in the items list of **RVData**), or -1 if no item was found.

OffsetInItem – position of this item:

- for non-text: 0 (closer to the beginning of the item) or 1 (to the end);
- for text: position is before the **OffsetInItem**-th character of text; characters are indexed from 1, the last position is Length(text)+1.

Return value:

"item found?"

1.3.1.2.32 TSRichViewEdit.GetItemBounds, GetItemBounds100

The methods return the coordinates and size of the specified item or its part.

```
function GetItemBounds(RVData: TCustomRVData; ItemNo: Integer;
Part: Integer = -1; AllowFloating: Boolean = True) : TRect;
```

RVData is a document (RichViewEdit⁽⁵⁹⁾.RVData, table cell, or cell inplace editor's RVData), **ItemNo** is an index of item in this document.

If **Part=-1**, the result is returned for the whole item. Otherwise, it is returned for the specified item part, counted from 0. A text item may have more than one part if drawn on several lines. To enumerate all parts, start calling this method from **Part=0** and increase this value until the method returns Rect(-1, -1, -1, -1). You can use the function RichViewEdit.RVData.GetItemPart to get the item part for the position in the document.

The returned coordinates are relative to the top left corner of the item's (or part's) page (see GetItemPages⁽⁸⁶⁾). If the item spans across several pages (it is possible only if **Part=-1**), the top left corner is relative to the first item page, the bottom left corner is relative to the last item page.

Coordinates are scaled (see ViewProperty⁽⁶⁸⁾.ZoomPercent⁽¹⁶⁷⁾).

To get the client coordinates of the item, add top and left coordinates returned by GetPageClientRect⁽⁸⁷⁾.

AllowFloating defines which coordinates are returned for left- and right-aligned items (for all other items, this parameter is ignored). If *True*, the method returns coordinates of the object (such as an

image or an control). If *False*, the method returns coordinates of a placeholder, i.e. coordinates of the point of insertion of this object in a line.

```
function GetItemBounds100 (RVData: TCustomRVFormattedData;  
    ItemNo: Integer; var ARect: TRect; var PageNoB, PageNoE: Integer;  
    Part: Integer = -1; AllowFloating: Boolean = True) : Boolean;
```

This method is similar, but:

- the resulting coordinates are returned in **ARect** parameter; they are valid only if this method returns *True*;
- the coordinates are not scaled (they correspond to zooming = 100%)
- this method returns *False* if an incorrect part index is specified; all output parameters are valid only if this method returns *True*;
- this method returns the first and the last item (or part) pages in **PageNoB** and **PageNoE** parameters, so it is not necessary to call `GetItemPages`⁽⁸⁶⁾;

See also:

- `GetItemCoords100`⁽⁸⁵⁾

1.3.1.2.33 TSVRichViewEdit.GetItemCoords100

The methods return the coordinates of the specified item or its part.

```
function GetItemCoords100 (RVData: TCustomRVFormattedData;  
    ItemNo: Integer; var ALeft, ATop: Integer;  
    var PageNo: Integer; Part : Integer = -1;  
    AllowFloating: Boolean = True): Boolean;
```

Input parameters

RVData is a document (RichViewEdit⁽⁵⁹⁾.RVData, table cell, or cell inplace editor's RVData), **ItemNo** is an index of item in this document.

If **Part**=-1, the result is returned for the whole item. Otherwise, it is returned for the specified item part, counted from 0. A text item may have more than one part if drawn on several lines. To enumerate all parts, start calling this method from **Part**=0 and increase this value until the method returns *False*. You can use the function `RichViewEdit.RVData.GetItemPart` to get the item part for the position in the document.

The returned coordinates are relative to the top left corner of the item's (or part's) page (see `GetItemPages`⁽⁸⁶⁾). If the item spans across several pages (it is possible only if **Part**=-1), the top left corner is relative to the first item page, the bottom left corner is relative to the last item page.

Coordinates are not scaled (they correspond to zooming = 100%).

AllowFloating defines which coordinates are returned for left- and right-aligned items (for all other items, this parameter is ignored). If *True*, the method returns coordinates of an object (such as an image or an control). If *False*, the method returns coordinates of a placeholder, i.e. coordinates of the point of insertion of this object in a line.

See also:

- `GetItemBounds`, `GetItemBounds100`⁽⁸⁴⁾

1.3.1.2.34 TSRichViewEdit.GetItemPages

Returns pages containing the specified item.

```
function GetItemPages (RVData: TCustomRVFormattedData; ItemNo: Integer;
  var FirstPageNo, LastPageNo: Integer; Part : Integer = -1): Boolean;
```

Input parameters:

RVData is a document (RichViewEdit⁵⁹.RVData, table cell, or cell inplace editor's RVData), **ItemNo** is an index of item in this document.

If **Part**=-1, the result is returned for the whole item. Otherwise, it is returned for the specified item part, counted from 0. A text item may have more than one part if drawn on several lines. To enumerate all parts, start calling this method from **Part**=0 and increase this value until the method returns *False*. You can use the function RichViewEdit.RVData.GetItemPart to get the item part for the position in the document.

Output parameters:

FirstPageNo – the first page where this item (or its part) is located, from 1.

LastPageNo – the last page where this item (or its part) is located, from 1.

Return value:

True if the specified item part exists.

1.3.1.2.35 TSRichViewEdit.GetLastItemVisibleRV

Returns the last visible item.

```
procedure GetFirstItemVisibleRV (var RVData: TCustomRVFormattedData;
  var ItemNo, OffsetInItem: Integer);
```

This function returns the position where the caret would be placed if the user clicked at the bottom left corner of TSRichViewEdit control's window.

RVData is a document (RichViewEdit⁵⁹.RVData, table cell, or cell inplace editor's RVData), **ItemNo** is an index of item in this document, **OffsetInItem** is a position inside this item.

See also:

- GetFirstItemVisibleRV⁸³

1.3.1.2.36 TSRichViewEdit.GetLastVisiblePage

Returns the index of the last page visible on the screen (at least partially), from 1

```
function GetLastVisiblePage: Integer;
```

See also:

- GetFirstVisiblePage⁸³

1.3.1.2.37 TSRichViewEdit.GetPageAt

Returns the page at the specified coordinates.

```
function GetPageAt (X, Y: Integer; var PageNo: Integer) : Boolean;
```

Input parameters:

X, Y – client coordinates (relative to the top left corner of TScaleRichViewEdit component's window)

Output parameter:

PageNo – index of page, from 1. Valid only if the method returns *True*.

Return value:

True if the position is inside a page.

1.3.1.2.38 TScaleRichViewEdit.GetPageClientRect

Returns client coordinates of the specified page.

```
function GetPageClientRect(PageNo: Integer): TRect;
```

NoPage is the index of page (from 1).

Return value:

page area (TRect) relative to the top left corner of the component's client area.

1.3.1.2.39 TScaleRichViewEdit.GetPageLastItemNo

Returns the index of the last item at the end of the **PageNo**-th page.

```
procedure GetPageLastItemNo(PageNo: Integer; var ItemNo, Offs: Integer);
```

Input parameter

PageNo is a page index, from 1.

Output parameters

ItemNo – index of the item ending this page. This value is in range 0..RichViewEdit⁽⁵⁹⁾.ItemCount-1.

Offs – defines the position in the **ItemNo**-th item, where the last line on the page ends. For all items except for tables, this is an offset in item:

- for text items, this is the index of the character ending the page (from 1)+1,
- for non-text items (except for tables), this parameter equals to 1;
- for tables, this is the index of the last row on this page (from 0).

See also:

- GetPageStartItemNo⁽⁸⁸⁾

1.3.1.2.40 TScaleRichViewEdit.GetPageNo

Returns the index of page for the specified position in the document.

```
function GetPageNo(RVData: TCustomRVFormattedData;  
    ItemNo, Offs : Integer): Integer;
```

Parameters:

The parameters define a position in the source document,

RVData is a document/subdocument. It can be either RVData of the source TRichView control, or table cell, or RVData of a cell inplace editor.

ItemNo is the index of item in this **RVData**.

OffsetInItem is the position inside this item. For non-text items, 0 is a position before the item, 1 is a position after the item. For text items, offsets are counted from 1 (1 - before the first character, 2 - before the second character, ... Length+1 - after the last character).

Return value:

Index of the page for this position, from 1.

1.3.1.2.41 TSRichViewEdit.GetPageStartItemNo

Returns the index of the first item at the beginning of the **PageNo**-th page.

```
procedure GetPageStartItemNo(PageNo: Integer; var ItemNo, Offs: Integer);
```

Input parameter

PageNo is a page index, from 1.

Output parameters

ItemNo – index of the item starting this page. This value is in range 0..[RichViewEdit](#)⁵⁹.ItemCount-1.

Offs – defines the position in the **ItemNo**-th item, where the first line on the page starts. For all items except for tables, this is an offset in item:

- for text items, this is the index of the character starting the page (from 1);
- for non-text items (except for tables), this parameter equals to 0;
- for tables, this is the index of the first row on this page (from 0) (see [GetPageStartTableItem](#)⁸⁸ and [GetPageStartTableRow](#)⁸⁸).

See also:

- [GetPageLastItemNo](#)⁸⁷

1.3.1.2.42 TSRichViewEdit.GetPageStartTableItem

Returns the index of table item (TRVTableItemInfo) at the beginning of the **PageNo**-th page.

```
function GetPageStartTableItem(PageNo: Integer) : Integer;
```

PageNo is a page index, from 1.

Return value:

item index of table in [RichViewEdit](#)⁵⁹, or -1 if the page is not started from a table.

See also:

- [GetPageStartTableRow](#)⁸⁸

1.3.1.2.43 TSRichViewEdit.GetPageStartTableRow

Returns the index of table row at the beginning of the **PageNo**-th page.

```
function GetPageStartTableRow(PageNo : Integer) : Integer;
```

PageNo is a page index, from 1.

Return value:

index of row in the table, or -1 if the page is not started from a table.

The table is `TRVTableItemInfo(RichViewEdit59.GetItem(GetPageStartTableItem88)), if GetPageStartTableItem88 >= 0.`

1.3.1.2.44 TSRichViewEdit.GetTableIconItem

The function returns the table for which the "icon" (rectangle in the top left corner) is displayed.

function `GetTableIconItem: TRVTableItemInfo;`

This method can be used in `OnTableIconClick121` event.

If a table icon is not displayed, the function returns *nil*.

Table icon is displayed when the user moves the mouse pointer above the table, if `ViewProperty68.UseTableIcons163 = True`.

1.3.1.2.45 TSRichViewEdit.GetTableIconRVData

The function returns a document containing the table for which the "icon" (rectangle in the top left corner) is displayed.

function `GetTableIconRVData: TCustomRVFormattedData;`

The returned value may be `RichViewEdit59.RVData`, table cell, or cell inplace editor's `RVData`.

This method can be used in `OnTableIconClick121` event.

If a table icon is not displayed, the function returns *nil*.

Table icon is displayed when the user moves the mouse pointer above the table, if `ViewProperty68.UseTableIcons163 = True`.

1.3.1.2.46 TSRichViewEdit.InsertPageBreak

Inserts a page break in the position of the caret.

procedure `InsertPageBreak;`

The same as `RichViewEdit89.InsertPageBreak`, but keeps the selection.

1.3.1.2.47 TSRichViewEdit.InZoomRect

Zoom the rectangle in by the specified **Ratio**.

function `InZoomRect(r: TRect; Ratio: Single): TRect;`

This method multiplies all coordinates in **r** by **Ratio** and returns the modified rectangle.

This method may be useful for calculating coordinates in `TSRichViewEdit` component.

See also:

- `OutZoomRect92`
- `ViewProperty68.ZoomPercent167`

1.3.1.2.48 TSRichViewEdit.LastCurPage

Moves the caret to the beginning of the last page.

procedure LastCurPage (SmoothScroll: Boolean = False);

The same affect as CurrentPage⁽⁵⁰⁾ := PageCount⁽⁵⁷⁾, but may apply a smooth scrolling effect.

See also:

- FirstCurPage⁽⁸²⁾, PriorCurPage⁽⁹⁵⁾, NextCurPage⁽⁹¹⁾

1.3.1.2.49 TSRichViewEdit.LoadRVF

Load data in RVF (RichView Format) from the file **FileName**

function LoadRVF (FileName: TRVUnicodeString): Boolean;

This method calls RichViewEdit.LoadRVF, then formats a header and a footer and calls SetRVMargins⁽⁹⁸⁾.

Return value:

True if loading was successful

1.3.1.2.50 TSRichViewEdit.LoadRVFFromStream

Load data in RVF (RichView Format) from **Stream**.

function LoadRVFFromStream (Stream: TStream): Boolean;

This method calls RichViewEdit.LoadRVFFromStream, then formats a header and a footer and calls SetRVMargins⁽⁹⁸⁾.

Return value:

True if loading was successful

1.3.1.2.51 TSRichViewEdit.NextCurCheckpoint

Selects the next item having a *checkpoint*.

function NextCurCheckpoint: Boolean;

The search is started from the next item after the caret position (RichViewEdit⁽⁵⁹⁾.TopLevelEditor.RVData, RichViewEdit⁽⁵⁹⁾.TopLevelEditor.CurItemNo).

If a *checkpoint* is found, the method selects its item and returns *True*.

See also

- PriorCurCheckpoint⁽⁹⁴⁾

1.3.1.2.52 TSRichViewEdit.NextCurHeading

Moves the caret to the beginning of the next heading.

function NextCurHeading: Boolean;

The search is started from the next item after the caret position (RichViewEdit⁽⁵⁹⁾.TopLevelEditor.RVData, RichViewEdit⁽⁵⁹⁾.TopLevelEditor.CurItemNo).

If an item is found, the method moves the caret and returns *True*.

Heading is a paragraph with OutlineLevel property > 0.

See also:

- PriorCurHeading⁽⁹⁴⁾

1.3.1.2.53 TSRichViewEdit.NextCurHyperlink

Selects the next hyperlink.

function NextCurHyperlink : Boolean;

The search is started from the next item after the caret position (RichViewEdit⁽⁵⁹⁾.TopLevelEditor.RVData, RichViewEdit⁽⁵⁹⁾.TopLevelEditor.CurItemNo).

If a hyperlink is found, the method selects it and returns *True*.

See also

- PriorCurHyperlink⁽⁹⁴⁾

1.3.1.2.54 TSRichViewEdit.NextCurItem

Selects the next item of one of the specified styles (types).

function NextCurItem(StyleNo: **array of** Integer) : Boolean;

StyleNo – a list of styles (types) for searching. If this array is empty, this method searches for any item.

The search is started from the next item after the caret position (RichViewEdit⁽⁵⁹⁾.TopLevelEditor.RVData, RichViewEdit⁽⁵⁹⁾.TopLevelEditor.CurItemNo).

If an item is found, the method selects it and returns *True*.

This method can be used for the command "Find Picture" and "Find Table" in the horizontal toolbar⁽⁵⁵⁾.

See also:

- PriorCurItem⁽⁹⁴⁾

1.3.1.2.55 TSRichViewEdit.NextCurPage

Moves the caret to the beginning of the next page.

procedure NextCurPage(SmoothScroll: Boolean = False);

The same affect as

```
if CurrentPage(50) < PageCount(57) then
  CurrentPage(50) := CurrentPage(50) + 1;
```

The method may apply a smooth scrolling effect.

See also:

- FirstCurPage⁽⁸²⁾, PriorCurPage⁽⁹⁵⁾, LastCurPage⁽⁹⁰⁾
- NextCurHyperlink⁽⁹¹⁾
- NextCurCheckpoint⁽⁹⁰⁾
- NextCurItem⁽⁹¹⁾

1.3.1.2.56 TSVRichViewEdit.OutZoomRect

Zoom the rectangle out by the specified **Ratio**.

```
function OutZoomRect(r: TRect; Ratio: Single): TRect;
```

This method divides all coordinates in **r** by **Ratio** and returns the modified rectangle.

This method may be useful for calculating coordinates in TSVRichViewEdit component.

See also:

- OutZoomRect⁹²
- ViewProperty⁶⁸.ZoomPercent¹⁶⁷

1.3.1.2.57 TSVRichViewEdit.Paste

Pastes data from the Clipboard in the caret position.

```
procedure Paste;
```

This method calls RichViewEdit⁵⁹.Paste.

See also:

- OnPaste¹¹⁶

1.3.1.2.58 TSVRichViewEdit.PrintAll

Prints the document.

```
procedure PrintAll(MetafileCompatibility: Boolean = False;  
  NoMetafiles: Boolean = False);
```

This method takes MinPrintedItemNo⁵⁶ and MaxPrintedItemNo⁵⁴ properties into account.

The printer is defined in Printer.PrinterIndex.

If **MetafileCompatibility**=*False*, printing output may be incompatible with metafiles; the component may use font glyph indexes instead of character codes when printing text. Such metafiles will be displayed incorrectly on computers without this font or with another version of this font.

If **MetafileCompatibility**=*True*, the component always uses character codes when printing text, to create metafile-compatible output.

If **NoMetafiles** = *False*, metafiles are drawn as they are. If *True*, metafiles are drawn as bitmaps. This option is useful when drawing on canvases of components (such as PDF generators) that do not handle embedded metafiles properly.

See also:

- PrintCurrent⁹³
- PrintRange⁹³
- TSVRVPrint²⁰³ component
- TSVRVPrint.MetafileCompatibility²¹⁰
- TRVAControlPanel.MetafileCompatibility (in the RichViewActions help file)

1.3.1.2.59 TSRichViewEdit.PrintCurrent

Prints the current page.

```
procedure PrintCurrent (MetafileCompatibility: Boolean=False;  
    NoMetafiles: Boolean = False);
```

The printer is defined in Printer.PrinterIndex.

If **MetafileCompatibility=False**, printing output may be incompatible with metafiles; the component may use font glyph indexes instead of character codes when printing text. Such metafiles will be displayed incorrectly on computers without this font or with another version of this font.

If **MetafileCompatibility=True**, the component always uses character codes when printing text, to create metafile-compatible output.

If **NoMetafiles = False**, metafiles are drawn as they are. If **True**, metafiles are drawn as bitmaps. This option is useful when drawing on canvases of components (such as PDF generators) that do not handle embedded metafiles properly.

See also:

- CurrentPage⁽⁵⁰⁾
- PrintAll⁽⁹²⁾
- PrintRange⁽⁹³⁾
- TSRVPrint⁽²⁰³⁾ component
- TSRVPrint.MetafileCompatibility⁽²¹⁰⁾
- TRVAControlPanel.MetafileCompatibility (in the RichViewActions help file)

1.3.1.2.60 TSRichViewEdit.PrintRange

Prints all pages from **StartPageNo** to **EndPageNo**.

```
procedure PrintRange (StartPageNo, EndPageNo: Integer;  
    MetafileCompatibility: Boolean=False;  
    NoMetafiles: Boolean = False);
```

This method takes MinPrintedItemNo⁽⁵⁶⁾ and MaxPrintedItemNo⁽⁵⁴⁾ properties into account.

The printer is defined in Printer.PrinterIndex.

If **MetafileCompatibility=False**, printing output may be incompatible with metafiles; the component may use font glyph indexes instead of character codes when printing text. Such metafiles will be displayed incorrectly on computers without this font or with another version of this font.

If **MetafileCompatibility=True**, the component always uses character codes when printing text, to create metafile-compatible output.

If **NoMetafiles = False**, metafiles are drawn as they are. If **True**, metafiles are drawn as bitmaps. This option is useful when drawing on canvases of components (such as PDF generators) that do not handle embedded metafiles properly.

See also:

- PrintAll⁽⁹²⁾
- PrintCurrent⁽⁹³⁾
- TSRVPrint⁽²⁰³⁾ component
- TSRVPrint.MetafileCompatibility⁽²¹⁰⁾

- TRVAControlPanel.MetafileCompatibility (in the RichViewActions help file)

1.3.1.2.61 TSRichViewEdit.PriorCurCheckpoint

Selects the previous item having a *checkpoint*.

function PriorCurCheckpoint: Boolean;

The search is started from the item before the caret position (RichViewEdit⁵⁹.TopLevelEditor.RVData, RichViewEdit⁵⁹.TopLevelEditor.CurItemNo).

If a *checkpoint* is found, the method selects its item and returns *True*.

See also

- NextCurCheckpoint⁹⁰

1.3.1.2.62 TSRichViewEdit.PriorCurHeading

Moves the caret to the beginning of the previous heading.

function PriorCurHeading: Boolean;

The search is started from the item before the caret position (RichViewEdit⁵⁹.TopLevelEditor.RVData, RichViewEdit⁵⁹.TopLevelEditor.CurItemNo).

If an item is found, the method moves the caret and returns *True*.

Heading is a paragraph with OutlineLevel property > 0.

See also:

- NextCurHeading⁹⁰

1.3.1.2.63 TSRichViewEdit.PriorCurHyperlink

Selects the next hyperlink.

function PriorCurHyperlink : Boolean;

The search is started from the item before the caret position (RichViewEdit⁵⁹.TopLevelEditor.RVData, RichViewEdit⁵⁹.TopLevelEditor.CurItemNo).

If a hyperlink is found, the method selects it and returns *True*.

See also

- NextCurHyperlink⁹¹

1.3.1.2.64 TSRichViewEdit.PriorCurItem

Selects the previous item of one of the specified styles (types).

function NextCurItem(StyleNo: array of Integer) : Boolean;

StyleNo – a list of styles (types) for searching. If this array is empty, this method searches for any item.

The search is started from the item before the caret position (RichViewEdit⁵⁹.TopLevelEditor.RVData, RichViewEdit⁵⁹.TopLevelEditor.CurItemNo).

If an item is found, the method selects it and returns *True*.

This method can be used for the command "Find Picture" and "Find Table" in the horizontal toolbar⁽⁵⁵⁾.

See also:

- NextCurItem⁽⁹¹⁾

1.3.1.2.65 TSRichViewEdit.PriorCurPage

Moves the caret to the beginning of the previous page.

procedure PriorCurPage(SmoothScroll: Boolean = False);

The same affect as

```
if CurrentPage(50) > 1 then
  CurrentPage(50) := CurrentPage(50) - 1;
```

The method may apply a smooth scrolling effect.

See also:

- FirstCurPage⁽⁸²⁾, NextCurPage⁽⁹¹⁾, LastCurPage⁽⁹⁰⁾
- PriorCurHyperlink⁽⁹⁴⁾
- PriorCurCheckpoint⁽⁹⁴⁾
- PriorCurItem⁽⁹⁴⁾

1.3.1.2.66 TSRichViewEdit.ProcessControls

Calls OnCheckControl⁽¹⁰⁵⁾ event for all controls inserted in the document.

procedure ProcessControls;

1.3.1.2.67 TSRichViewEdit.ReturnToNote

Ends editing a note's Document and moves the caret to the location of this note in the main document.

function ReturnToNote(SmoothScroll: Boolean = False): Boolean;

This method does nothing if a note is not being edited (i.e. when ActiveEditor⁽⁴⁷⁾ <> RVNote⁽⁶⁴⁾).

When the caret is moved, the main document becomes active (ActiveEditor⁽⁴⁷⁾ becomes equal to RichViewEdit⁽⁵⁹⁾).

If **SmoothScroll** = *True*, a smooth scrolling effect is applied.

Return value:

True if the caret was moved successfully

See also:

- StartEditNote⁽⁹⁹⁾
- Footnotes and endnotes⁽³⁸⁾

1.3.1.2.68 TSRichViewEdit.ScrollToCaret, ScrollCaretToCenter

The methods scroll to make the caret visible.

function ScrollToCaret(SmoothScroll: Boolean = False): Boolean;

function ScrollCaretToCenter(SmoothScroll: Boolean = False): Boolean;

ScrollToCaret does nothing if the caret is already completely visible. **ScrollCaretToCenter** always scrolls.

ScrollToCaret scrolls to show the top of the caret in the middle of the editor. **ScrollCaretToCenter** scrolls to show the middle of the caret in the middle of the editor.

If **SmoothScroll** = *True*, a smooth scrolling effect is applied.

See also

- ScrollToItem⁽⁹⁶⁾

1.3.1.2.69 TSRichViewEdit.ScrollToItem

Scrolls to make the specified item visible. Optionally moves the caret to this item.

```
function ScrollToItem(RVData: TCustomRVFormattedData; ItemNo: Integer;
  SetCaret: Boolean = True; ToItemEnd: Boolean = False;
  ToItemCenter: Boolean = False; SmoothScroll: Boolean = False): Boolean;
```

The item location

RVData is a document (RichViewEdit⁽⁵⁹⁾.RVData, table cell, or cell inplace editor's RVData), **ItemNo** is an index of item in this document.

Moving the caret

If **SetCaret**=*True*, the method not only scrolls the window, but also moves the caret to the specified item. The caret is moved either to the beginning of this item (if **ToItemEnd**=*False*) or to the end of it (if **ToItemEnd**=*True*).

Scrolling

If **ToItemCenter**=*False*, the method scrolls so that the item top is shown at the top of the editor window,

If **ToItemCenter**=*True*, the method scrolls so that the item middle is shown at the middle of the editor window.

If **SmoothScroll** = *True*, a smooth scrolling effect is applied.

See also:

- ScrollToCaret, ScrollCaretToCenter⁽⁹⁵⁾

1.3.1.2.70 TSRichViewEdit.SelectAll

Selects the whole document in ActiveEditor⁽⁴⁷⁾.

```
procedure SelectAll;
```

1.3.1.2.71 TSRichViewEdit.Set*PropertyEd

Editing-style methods for changing properties of the editor.

```
procedure SetIntPropertyEd(Prop: TSRVIntProperty(276); Value: Integer;
  AutoReformat: Boolean);
```

```
procedure SetFloatPropertyEd(Prop: TSRVFloatProperty(275); Value: TRVLength;
  AutoReformat: Boolean);
```

Parameters:

Prop identifies the property to change.

Value – new property value. If the property is a property of enumerated type, use Ord(). If the property is a boolean property, any non-zero value means *True*.

If **AutoReformat**=*True*, the editor is reformatted (if necessary), and RichViewEdit⁽⁵⁹⁾.Change is called. Normally, this method is called with **AutoReformat**=*True* if you change a single property. If you call these methods multiple times for changing multiple properties, reformat⁽⁸²⁾ (if necessary) and call RichViewEdit⁽⁵⁹⁾.Change yourself.

1.3.1.2.72 TSRichViewEdit.SetMargin

Assigns margins.

```
procedure SetMargin(MarginLeft, MarginTop, MarginRight,
  MarginBottom: Integer);
```

MarginLeft, MarginTop, MarginRight, MarginBottom are new values of margins, measured in pixels.

Before assigning, OnMarginsChanged⁽¹¹¹⁾ occurs.

Only parameters <> -1 are assigned.

This method changes:

- PageProperty⁽⁵⁸⁾'s LeftMargin⁽¹³⁷⁾, TopMargin⁽¹⁴²⁾, RightMargin⁽¹⁴¹⁾, BottomMargin⁽¹³⁵⁾
- LeftMarginPix⁽⁵⁴⁾, TopMarginPix⁽⁶⁷⁾, RightMarginPix⁽⁶⁰⁾, BottomMarginPix⁽⁴⁸⁾

See also:

- SetMarginMM⁽⁹⁷⁾
- SetMarginUnit⁽⁹⁸⁾
- SetRVMargins⁽⁹⁸⁾

1.3.1.2.73 TSRichViewEdit.SetMarginMM

Assigns margins.

```
procedure SetMarginMM(MarginLeftMM, MarginTopMM, MarginRightMM,
  MarginBottomMM : Extended);
```

MarginLeftMM, MarginTopMM, MarginRightMM, MarginBottomMM are new values of margins, measured in millimeters.

Before assigning, OnMarginsChanged⁽¹¹¹⁾ occurs.

Only parameters <> -1 are assigned.

This method changes:

- PageProperty⁽⁵⁸⁾'s LeftMargin⁽¹³⁷⁾, TopMargin⁽¹⁴²⁾, RightMargin⁽¹⁴¹⁾, BottomMargin⁽¹³⁵⁾
- LeftMarginPix⁽⁵⁴⁾, TopMarginPix⁽⁶⁷⁾, RightMarginPix⁽⁶⁰⁾, BottomMarginPix⁽⁴⁸⁾

See also:

- SetMargin⁽⁹⁷⁾
- SetMarginUnit⁽⁹⁸⁾
- SetRVMargins⁽⁹⁸⁾

1.3.1.2.74 TSRichViewEdit.SetMarginUnit

Assigns margins.

```
procedure SetMarginUnit (MarginLeft, MarginTop, MarginRight,
    MarginBottom: TRVLength);
```

MarginLeft, MarginTop, MarginRight, MarginBottom are new values of margins, measured in UnitsProgram⁽⁶⁸⁾.

Before assigning, OnMarginsChanged⁽¹¹¹⁾ occurs.

Only parameters <> -1 are assigned.

This method changes:

- PageProperty⁽⁵⁸⁾'s LeftMargin⁽¹³⁷⁾, TopMargin⁽¹⁴²⁾, RightMargin⁽¹⁴¹⁾, BottomMargin⁽¹³⁵⁾
- LeftMarginPix⁽⁵⁴⁾, TopMarginPix⁽⁶⁷⁾, RightMarginPix⁽⁶⁰⁾, BottomMarginPix⁽⁴⁸⁾

See also:

- SetMarginMM⁽⁹⁷⁾
- SetMargin⁽⁹⁷⁾
- SetRVMargins⁽⁹⁸⁾

1.3.1.2.75 TSRichViewEdit.SetRVMargins

Assigns page properties from RichViewEdit⁽⁵⁹⁾.DocParameters to PageProperty⁽⁵⁸⁾, formats the editor.

```
procedure SetRVMargins (CompleteFormat: Boolean = True);
```

It's highly recommended to call call Format⁽⁸²⁾ or Reformat⁽⁸²⁾ instead of this method!

This method does the following work:

- copies page properties from RichViewEdit⁽⁵⁹⁾.DocParameters to PageProperty⁽⁵⁸⁾
- resets formatting for notes and text boxes (so that they will be reformatted on the next pagination)
- formats headers and footers (if CompleteFormat=False, the caret position is kept in RVHeader⁽⁶³⁾ and RVFooter⁽⁶¹⁾)
- formats RichViewEdit⁽⁵⁹⁾ (if CompleteFormat=False, the caret position is kept)
- (if RVNote⁽⁶⁴⁾ <> nil) formats RVNote⁽⁶⁴⁾ (if CompleteFormat=False, the caret position is kept)

1.3.1.2.76 TSRichViewEdit.StartEditing

Begins editing the specified subdocument.

type

```
TSRVRichViewEditType = (srvrveMain, srvrveHeader, srvrveFooter);
```

```
function StartEditing (Editor: TSRVRichViewEditType): Boolean;
```

Editor	Begins editing...
<i>srvrveMain</i>	RichViewEdit ⁽⁵⁹⁾ (main document)
<i>srvrveHeader</i>	RVHeader ⁽⁶³⁾ (page header for the current page ⁽⁵⁰⁾)

<i>rvrveFooter</i>	RVFooter ⁶¹ (page footer for the current page)
--------------------	---

This method changes value of ActiveEditor⁴⁷ property.

See also:

- StartEditNote⁹⁹

1.3.1.2.77 TRichViewEdit.StartEditNote

Begins editing the Document of the specified note or a text box item.

```
function StartEditNote(RVData: TCustomRVFormattedData;  
    ItemNo: Integer; ScrollToNote: Boolean = True;  
    SmoothScroll: Boolean = False): TRichViewEdit;
```

RVData and **ItemNo** must define the location of a note item. The following item types are supported:

- footnotes (TRVFootnoteItemInfo)
- endnotes (TRVEndnoteItemInfo)
- sidenotes (TRVSidenoteItemInfo)
- text box items (TRVTextBoxItemInfo)

This method changes value of ActiveEditor⁴⁷ property (it becomes equal to RVNote⁶⁴).

If **ScrollToNote**=*True*, the method scrolls the control to show the edited note. If **SmoothScroll** = *True*, this scrolling is smooth.

Return value:

ActiveEditor⁴⁷ if succeed. *Nil* if **RVData** and **ItemNo** do not define a location of a note or a text box item.

See also:

- StartEditing⁹⁸
- ReturnToNote⁹⁵
- Footnotes and endnotes³⁸

1.3.1.2.78 TRichViewEdit.UnitsPerInchH

Converts one inch to the specified units. Used for horizontal distances.

```
function UnitsPerInchH(Units: TRVUnits): TRVLength;
```

Since in ScaleRichView both vertical and horizontal resolution are assumed equal to 96 dpi, this function returns the same value as UnitsPerInchH¹⁰⁰.

This function can be used to convert one units to another, for example:

```
// converting horizontal size from millimeters to UnitsProgram68  
Size := SizeMM * UnitsPerInchH(UnitsProgram68) /  
    UnitsPerInchH(rvuMillimeters)
```

1.3.1.2.79 TSRichViewEdit.UnitsPerInchV

Converts one inch to the specified units. Used for vertical distances.

function UnitsPerInchV(Units: TRVUnits): TRVLength;

Since in ScaleRichView both vertical and horizontal resolution are assumed equal to 96 dpi, this function returns the same value as UnitsPerInchH⁽⁹⁹⁾.

This function can be used to convert one units to another, for example:

```
// converting vertical size from millimeters to UnitsProgram(68)
Size := SizeMM * UnitsPerInchV(UnitsProgram(68)) /
    UnitsPerInchV(rvuMillimeters)
```

1.3.1.2.80 TSRichViewEdit.UpdateBuffer, UpdateBufferAt

These methods redraw the internal buffer or its area.

procedure UpdateBuffer;

procedure UpdateBufferAt(PaintRect: TRect);

When drawing on the screen, TSRichViewEdit control uses an internal buffer (a bitmap). When you call Invalidate or Repaint method, the control simply copies from this buffer to the screen.

These methods update this buffer (or the area PaintRect specified in the client coordinates) and the corresponding area on the screen.

Normally, the buffer is updated when document is changed, so you do not need to call these methods. However, if you want to update some drawing made in OnPaintPage⁽¹¹⁵⁾, use these methods.

1.3.1.2.81 TSRichViewEdit.ZoomCanvas, RestoreCanvasZoom

Applies and restores zooming to the **Canvas**.

function ZoomCanvas(Canvas : TCanvas;
ScaleX, ScaleY : Single) : TSRVZoomInfo;

procedure RestoreCanvasZoom(Canvas : TCanvas;
OldInfo : TSRVZoomInfo);

type

TSRVZoomInfo = **record** ...

After calling ZoomCanvas, all subsequent drawing on **Canvas** will be scaled by **ScaleX** horizontally and by **ScaleY** vertically. Store the returned value and then call RestoreCanvasZoom to restore the original zooming.

These methods may be useful in OnPaint⁽¹¹⁴⁾ event, if you want to draw something in the same zooming ratio as pages (use ViewProperty⁽⁶⁸⁾.ZoomPercent⁽¹⁶⁷⁾ * 0.01 as **ScaleX** and **ScaleY** parameters).

1.3.1.2.82 TSRichViewEdit.ZoomRectIn, ZoomRectOut

Returns a scaled rectangle.

function ZoomRectIn(r: TRect; Ratio: Single): TRect;

function ZoomRectOut(r: TRect; Ratio: Single): TRect;

ZoomRectIn returns the rectangle **r** zoomed in by **Ratio**.

ZoomRectOut returns the rectangle **r** zoomed in by 1/**Ratio**.

1.3.1.2.83 TSRichViewEdit.ZoomToFullPages

Changes zooming (ViewProperty⁶⁸.ZoomPercent¹⁶⁷) to display pages in **ColCount** columns, if possible.

procedure ZoomToFullPages (ColCount: Integer);

The procedure changes ViewProperty⁶⁸.ZoomPercent¹⁶⁷ to display **ColCount** pages horizontally. It may not always be possible, because zooming must be in the range ViewProperty⁶⁸.ZoomMin..ZoomMax¹⁶⁵.

1.3.1.3 Events

In TSRichViewEdit

- OnAfterOleDrop¹⁰³
- OnAssignImageFileName¹⁰³
- OnBeforeOleDrop¹⁰³
- OnCaretGetOut¹⁰⁴
- OnCaretMove¹⁰⁴
- OnCaretMoving¹⁰⁴
- OnChange¹⁰⁴
- OnChangeActiveEditor¹⁰⁴
- OnChangeCurrentControl¹⁰⁴
- OnChangeViewModeAfter¹⁰⁴
- OnChangeViewModeBefore¹⁰⁵
- OnChanging¹⁰⁵
- OnCheckControl¹⁰⁵
- OnCheckpointVisible¹⁰⁵
- OnCheckStickingItems¹⁰⁶
- OnClickPage¹⁰⁶
- OnContextPopup¹⁰⁶
- OnControlAction¹⁰⁶
- OnCopy¹⁰⁶
- OnCurParaStyleChanged¹⁰⁶
- OnCurrentPageChange¹⁰⁷
- OnCurTextStyleChanged¹⁰⁷
- OnDrawHyperlink¹⁰⁷
- OnDropFiles¹⁰⁷
- OnFullRedraw¹⁰⁸
- OnGetPagePos¹⁰⁸
- OnHMenuClickButton¹⁰⁹
- OnHMenuEnterButton¹⁰⁹
- OnHScrolled¹⁰⁹
- OnHTMLSaveImage¹⁰⁹
- OnImportPicture¹¹⁰

- OnItemAction⁽¹¹⁰⁾
- OnItemHint⁽¹¹⁰⁾
- OnItemResize⁽¹¹⁰⁾
- OnItemTextEdit⁽¹¹⁰⁾
- OnLoadCustomFormat⁽¹¹¹⁾
- OnLoadDocument⁽¹¹¹⁾
- OnJump⁽¹¹⁰⁾
- OnMarginsChanged⁽¹¹¹⁾
- OnMessageControl⁽¹¹²⁾
- OnNewDocument⁽¹¹²⁾
- OnOleDragEnter⁽¹¹³⁾
- OnOleDragLeave⁽¹¹³⁾
- OnOleDragOver⁽¹¹³⁾
- OnOleDrop⁽¹¹³⁾
- OnPageCountChanged⁽¹¹³⁾
- OnPageFormatChanged⁽¹¹⁴⁾
- OnPageScrolled⁽¹¹⁴⁾
- OnPaint⁽¹¹⁴⁾
- OnPaintComponent⁽¹¹⁴⁾
- OnPaintPage⁽¹¹⁵⁾
- OnParaStyleConversion⁽¹¹⁶⁾
- OnPaste⁽¹¹⁶⁾
- OnPrinting⁽¹¹⁶⁾
- OnProgress⁽¹¹⁶⁾
- OnReadField⁽¹¹⁶⁾
- OnReadHyperlink⁽¹¹⁷⁾
- OnReadMergeField⁽¹¹⁷⁾
- OnResize⁽¹¹⁷⁾
- OnResizing⁽¹¹⁷⁾
- OnResizeDoc⁽¹¹⁷⁾
- OnRVDbClick⁽¹¹⁷⁾
- OnRVFControlNeeded⁽¹¹⁷⁾
- OnRVFImageListNeeded⁽¹¹⁸⁾
- OnRVFPictureNeeded⁽¹¹⁸⁾
- OnRVMouseDown⁽¹¹⁸⁾
- OnRVMouseMove⁽¹¹⁸⁾
- OnRVMouseUp⁽¹¹⁸⁾
- OnRVRightClick⁽¹¹⁹⁾
- OnSaveComponentToFile⁽¹¹⁹⁾
- OnSaveCustomFormat⁽¹¹⁹⁾
- OnSaveDocXExtra⁽¹¹⁹⁾
- OnSaveHTMLExtra⁽¹¹⁹⁾
- OnSaveImage2⁽¹¹⁹⁾
- OnSaveItemToFile⁽¹²⁰⁾
- OnSaveRTFExtra⁽¹²⁰⁾
- OnSelect⁽¹²⁰⁾
- OnSmartPopupClick⁽¹²⁰⁾
- OnSpellingCheck⁽¹²⁰⁾

- OnStyleConversion⁽¹²⁰⁾
- OnTableIconClick⁽¹²¹⁾
- OnTextFound⁽¹²¹⁾
- OnURLNeeded⁽¹²¹⁾
- OnVMenuClickButton⁽¹²¹⁾
- OnVMenuEnterButton⁽¹²¹⁾
- OnVScrolled⁽¹²²⁾
- OnWriteHyperlink⁽¹²²⁾
- OnWriteObjectProperties⁽¹²²⁾
- OnZoomChanged⁽¹²²⁾
- OnZoomViewChanged⁽¹²²⁾

Derived from TCustomControl

- OnClick
- OnDbClick
- OnDragDrop
- OnDragOver
- OnEndDrag
- OnEnter
- OnExit
- OnGesture (D2010+)
- OnKeyDown
- OnKeyPress
- OnKeyUp
- OnMouseMove
- OnMouseWheel
- OnStartDrag

1.3.1.3.1 TSVRichViewEdit.OnAssignImageFileName

Occurs when the component assigns background image file names; allows to modify file names before assigning.

property OnAssignImageFileName : TRVAssignImageFileNameEvent;

See the event of the same name in the TRichView help file (event of TCustomRichView).

See also:

- ActiveEditor⁽⁴⁷⁾

1.3.1.3.2 TSVRichViewEdit.OnBeforeOleDrop, OnAfterOleDrop

OLE drag&drop events. They occur when the user completes drag&drop operation into the editor. These events occur before and after insertion.

property OnBeforeOleDrop: TNotifyEvent;

property OnAfterOleDrop: TNotifyEvent;

See the events of the same name in the TRichView help file (event of TCustomRichViewEdit).

1.3.1.3.3 TSRichViewEdit.OnCaretGetOut

Occurs when user presses arrow keys at the beginning/end of document in "outside" direction.

property OnCaretGetOut : TRVOnCaretGetOutEvent;

See the event of the same name in the TRichView help file (event of TCustomRichViewEdit).

See also:

- ActiveEditor⁽⁴⁷⁾

1.3.1.3.4 TSRichViewEdit.OnCaretMove, OnCaretMoving

The events occur when the caret is moved to the another position within the document (TCustomRichViewEdit(Sender)).

property OnCaretMove: TNotifyEvent;

property OnCaretMoving: TNotifyEvent;

OnCaretMoving occurs before TSRichViewEdit processes a caret movement (and scrolls to move the caret visible).

OnCaretMove occurs after that.

See "OnCaretMove" in the TRichView help file (event of TCustomRichViewEdit).

1.3.1.3.5 TSRichViewEdit.OnChange

Occurs when document is changed.

property OnChange: TNotifyEvent;

See the event of the same name in the TRichView help file (event of TCustomRichViewEdit).

1.3.1.3.6 TSRichViewEdit.OnChangeActiveEditor

Occurs when value of ActiveEditor⁽⁴⁷⁾ property is changed.

type

```
TSRVChangeActiveEditorEvent = procedure (Sender: TSRichViewEdit(40);
  ActiveEditor: TRichViewEdit) of object;
```

property OnChangeActiveEditor : TSRVChangeActiveEditorEvent;

1.3.1.3.7 TSRichViewEdit.OnChangeCurrentControl

Occurs when value of CurControl⁽⁵⁰⁾ property is changed.

type

```
TSRVChangeCurrentControlEvents = procedure (Sender: TSRichViewEdit(40);
  AControl: TControl) of object;
```

property OnChangeCurrentControl : TSRVChangeCurrentControlEvents;

1.3.1.3.8 TSRichViewEdit.OnChangeViewModeAfter

The event occurs after changing a viewing mode (ViewProperty⁽⁶⁸⁾.ViewMode⁽¹⁶⁴⁾).

type

```
TSRVChangeViewModeEvent = procedure (Sender: TSRichViewEdit(40);
```

```
Value: TSRVViewMode(153)) of object;
```

property OnChangeViewModeAfter: TSRVChangeViewModeEvent;

Value is a new value of ViewMode⁽¹⁶⁴⁾.

See also:

- OnChangeViewModeBefore⁽¹⁰⁵⁾

1.3.1.3.9 TSRichViewEdit.OnChangeViewModeBefore

The event occurs before changing a viewing mode (ViewProperty⁽⁶⁸⁾.ViewMode⁽¹⁶⁴⁾).

type

```
TSRVChangeViewModeEvent = procedure (Sender: TSRichViewEdit(40);  
Value: TSRVViewMode(153)) of object;
```

property OnChangeViewModeBefore: TSRVChangeViewModeEvent;

Value is a new value of ViewMode⁽¹⁶⁴⁾.

See also:

- OnChangeViewModeAfter⁽¹⁰⁴⁾

1.3.1.3.10 TSRichViewEdit.OnChanging

Occurs before editing operations, allows preventing editing.

property OnChanging: TRVChangingEvent;

See the event of the same name in the TRichView help file (event of TCustomRichViewEdit).

1.3.1.3.11 TSRichViewEdit.OnCheckControl

This event is called by ProcessControls⁽⁹⁵⁾ method.

type

```
TSRVCheckControl = procedure (Sender: TSRichViewEdit;  
AControl : TControl) of object;
```

property OnCheckControl: TSRVCheckControl;

This event is called for each control (**AControl**) inserted in the current document.

1.3.1.3.12 TSRichViewEdit.OnCheckpointVisible

Occurs when *checkpoint* (with RaiseEvent flag = *True*) becomes visible as a result of vertical scrolling.

type

```
TSRVCheckpointVisibleEvent = procedure (Sender: TSRichViewEdit(40);  
RVData: TCustomRVData; ItemNo: Integer) of object;
```

property OnCheckpointVisible: TSRVCheckpointVisibleEvent;

See the event of the same name in the TRichView help file (event of TCustomRichView).

This event has different parameters comparing with the event of TCustomRichView: instead of CheckpointData: TCheckpointData, there are two parameters defining the location of the *checkpoint*: **RVData** and **ItemNo**. You can get the *checkpoint* as **RVData.GetItemCheckpoint(ItemNo)**.

1.3.1.3.13 TSRichViewEdit.OnCheckStickingItems

Allows preventing insertion between two items, or at the beginning of the document, or at the end of the document.

property OnCheckStickingItems: TRVStickingItemsEvent;

See the event of the same name in the TRichView help file (event of TCustomRichViewEdit).

1.3.1.3.14 TSRichViewEdit.OnClickPage

Occurs when the user clicked on a page.

property OnClickPage: TSRVMouseEvent²⁷⁶;

1.3.1.3.15 TSRichViewEdit.OnContextPopup

Occurs when the user right-clicks the control or otherwise invokes the popup menu (such as using the keyboard).

property OnContextPopup: TContextPopupEvent;

1.3.1.3.16 TSRichViewEdit.OnControlAction

Occurs when some important operation is performed on control inserted in Sender.

property OnControlAction: TRVControlActionEvent;

See the event of the same name in the TRichView help file (event of TCustomRichView).

1.3.1.3.17 TSRichViewEdit.OnCopy

Occurs when copying to the Clipboard

property OnCopy: TNotifyEvent;

This event allows copying to the Clipboard in your own formats (occurs between Clipboard.Open and Clipboard.Close).

See the event of the same name in the TRichView help file (event of TCustomRichView).

See also:

- ActiveEditor⁴⁷

1.3.1.3.18 TSRichViewEdit.OnCurParaStyleChanged

Occurs when the index of current paragraph style (value of TCustomRichViewEdit(Sender).CurParaStyleNo property) is changed.

property OnCurParaStyleChanged: TNotifyEvent;

See the event of the same name in the TRichView help file (event of TCustomRichViewEdit).

1.3.1.3.19 TSRichViewEdit.OnCurrentPageChange

Occurs when the caret is moved to another page

property OnCurrentPageChange: TNotifyEvent;

This event occurs after CurrentPage⁽⁵⁰⁾ property is changed.

See also:

- OnPageCountChanged⁽¹¹³⁾
- OnPageScrolled⁽¹¹⁴⁾

1.3.1.3.20 TSRichViewEdit.OnCurTextStyleChanged

Occurs when the index of current text style (value of TCustomRichViewEdit(Sender).CurTextStyleNo property) is changed.

property OnCurTextStyleChanged: TNotifyEvent;

See the event of the same name in the TRichView help file (event of TCustomRichViewEdit).

1.3.1.3.21 TSRichViewEdit.OnDrawHyperlink

Occurs when a hyperlink is drawn.

type

```
TSRVDDataDrawHyperlinkEvent = procedure (Sender: TSRichViewEdit(40);
  Canvas : TCanvas; RVData: TCustomRVData; PageNo, ItemNo: Integer;
  RepaintRect, R: TRect) of object;
```

property OnDrawHyperlink : TSRVDDataDrawHyperlinkEvent;

This event can be used for the following tasks:

- storing locations of hyperlinks when calling DrawPage⁽⁷⁶⁾ or DrawMetafile⁽⁷⁶⁾
- custom drawing (see the demo **Demos\Basic Demos\DrawHyperlink**)

Parameters:

Canvas – a canvas where the page is drawn.

PageNo – a page containing this hyperlink, from 1.

RVData, ItemNo – location of the hyperlink. **RVData** is a document (RichViewEdit⁽⁵⁹⁾.RVData, table cell, or cell inplace editor's RVData), **ItemNo** is an index of item in this document.

RepaintRect – page area (the same as returned by GetPageClientRect⁽⁸⁷⁾).

R – hyperlink area

1.3.1.3.22 TSRichViewEdit.OnDropFile, OnDropFiles

The events occur when dropping files in the editor (as a result of drag&drop operation, for example from Windows Explorer)

property OnDropFile: TRVDropFileEvent;

property OnDropFiles: TRVDropFilesEvent;

See the event of the same name in the TRichView help file (event of TCustomRichViewEdit).

1.3.1.3.23 TSVRichViewEdit.OnFullRedraw

Occurs when the component is completely repainted.

property OnFullRedraw: TNotifyEvent;

Mainly for internal use.

1.3.1.3.24 TSVRichViewEdit.OnGetPagePos

Allows to define page position, size and Z-order. Occurs if ViewProperty⁽⁶⁸⁾.FreePosPage⁽¹⁵⁷⁾ = True.

type

```
TSRVGetPagePosEvent = procedure (Sender: TSVRichViewEdit;
  PageNo : Integer;
  StartItemNo, EndItemNo, OffsetInStartItem,
  StartTableRow, EndTableRow: Integer;
  var Position : TPoint; var ZoomPercent : Single;
  var ZOrder : Integer) of object;
```

property OnGetPagePos: TSVRVGetPagePosEvent;

PageNo – page index (from 1).

StartItemNo – index of the first item on the page.

EndItemNo – index of the last item on the page.

OffsetInStartItem – offset in the first item (for non-text items: 0; for text items: index of the first character on this page, from 1).

StartTableRow – index of the first table row on the page, if the table is the first item on the page; -1 otherwise.

EndTableRow – index of the last table row on the page, if the table is the last item on the page; -1 otherwise.

Position – page position, coordinates of the top left page corner. By default, it is calculated by adding offset to the previous page. The coordinates are relative to the top left corner of a scrollable area in TSVRichViewEdit's window.

ZoomPercent – page zooming. By default, it is equal to zoom percent for the previous page. For the first page, it is equal to ViewProperty⁽⁶⁸⁾.ZoomPercent⁽¹⁶⁷⁾.

ZOrder – the page's Z-order. By default it is the same as for the previous page. For the first page, it is 0. Pages having the same Z-order value are drawn in the order of their numeration. Pages with larger Z-order value are drawn above the pages with smaller Z-order values. If PageProperty⁽⁵⁸⁾.UsePageOrders⁽¹⁴²⁾ = False, this parameter is ignored.

See also:

- CalculatePagePosition⁽⁷²⁾
- CalculateAllPagePositions⁽⁷²⁾

Example:

```
// Arranging pages in 4 columns, 10 pixels between rows,
// 30 pixels between columns
```



```

procedure TFColumns.SRichViewEdit1GetPagePos(Sender: TSCRichViewEdit;
  PageNo : Integer; StartItemNo, EndItemNo, OffsetInStartItem,
  StartTableRow, EndTableRow: Integer;
  var Position : TPoint; var ZoomPercent : Single;
  var ZOrder : Integer);
begin
  Position.X := 10 + ((PageNo - 1) mod 4) * (Sender.PageWidthPix + 30);
  Position.Y := 10 + ((PageNo - 1) div 4) * (Sender.PageHeightPix + 10);
end;

```

1.3.1.3.25 TSCRichViewEdit.OnHMenuClickButton

Occurs when the user clicks one of buttons of the horizontal menu (toolbar).

property OnHMenuClickButton: TSCRVClickButtonEvent⁽²⁷¹⁾;

ToolButton – the clicked button (one of MenuHButtons⁽⁵⁵⁾), or *nil*, if the user clicked outside any button.

See also:

- OnVMenuClickButton⁽¹²¹⁾
- OnHMenuEnterButton⁽¹⁰⁹⁾
- MenuHorizontal⁽⁵⁵⁾

1.3.1.3.26 TSCRichViewEdit.OnHMenuEnterButton

The event occurs when the user moves the mouse pointer to one of the buttons of the horizontal menu.

property OnHMenuEnterButton: TSCRVClickButtonEvent⁽²⁷¹⁾;

ToolButton – the button under the mouse pointer, one of MenuHButtons⁽⁵⁵⁾.

See also:

- OnVMenuEnterButton⁽¹²¹⁾
- OnHMenuClickButton⁽¹⁰⁹⁾
- MenuHorizontal⁽⁵⁵⁾

1.3.1.3.27 TSCRichViewEdit.OnHScrolled

Occurs on changing position of horizontal scrollbar.

property OnHScrolled: TNotifyEvent;

This event occurs when the user changes position of horizontal scrollbar, or when new value is assigned to HScrollPos⁽⁵³⁾ property.

See also:

- OnVScrolled⁽¹²²⁾

1.3.1.3.28 TSCRichViewEdit.OnHTMLSaveImage

Occurs when RichViewEdit⁽⁵⁹⁾ saves item containing image to HTML file or stream. This event allows to modify saving procedure for all (or some) images.

property OnHTMLSaveImage: TRVHTMLSaveImageEvent;

See the event of the same name in the TRichView help file (event of TCustomRichView).

See also:

- OnSaveImage2¹¹⁹

1.3.1.3.29 TSRichViewEdit.OnImportPicture

Occurs when loading RTF file or stream containing links to image files. This event also can be used by [TRVHtmlImporter](#).

property OnImportPicture: TRVImportPictureEvent;

See the event of the same name in the TRichView help file (event of TCustomRichView).

1.3.1.3.30 TSRichViewEdit.OnItemAction

Occurs on insertion, deletion and some other modifications of item in Sender's document.

property OnItemAction: TRVItemActionEvent;

See the event of the same name in the TRichView help file (event of TCustomRichView).

1.3.1.3.31 TSRichViewEdit.OnItemHint

Allows to define custom popup hints for items in Sender's document.

property OnItemHint: TRVItemHintEvent;

See the event of the same name in the TRichView help file (event of TCustomRichView).

1.3.1.3.32 TSRichViewEdit.OnItemResize

Occurs after resizing image, control or table with the mouse.

property OnItemResize : TRVItemResizeEvent;

See the event of the same name in the TRichView help file (event of TCustomRichViewEdit).

1.3.1.3.33 TSRichViewEdit.OnItemTextEdit

Occurs when text item's text is changed as a result of editing operation.

property OnItemTextEdit: TRVItemTextEditEvent;

See the event of the same name in the TRichView help file (event of TCustomRichViewEdit).

1.3.1.3.34 TSRichViewEdit.OnJump

Occurs when the user clicks on hypertext item.

property OnJump: TJumpEvent;

See the event of the same name in the TRichView help file (event of TCustomRichView).

See also:

- OnRVMouseMove¹¹⁸

1.3.1.3.35 TSVRichViewEdit.OnLoadCustomFormat

This event allows loading data from field in your own format.

type

```
TSRVCustomFormatEvent = procedure (Sender: TSVRichViewEdit40;  
    Stream: TStream; var DoDefault: Boolean) of object;
```

property OnLoadCustomFormat: TDBSRVCustomFormatEvent;

See the event of the same name in the TRichView help file (event of TRichView).

This event may be called:

- in TDBSRichViewEdit¹⁶⁸
- if Document⁵¹ is linked to a database field using LiveBindings

See also:

- OnSaveCustomFormat¹¹⁹

1.3.1.3.36 TSVRichViewEdit.OnLoadDocument

Occurs after loading document from the database field

property OnLoadDocument: TNotifyEvent;

See the event of the same name in the TRichView help file (event of TRichView).

This event allows to preprocess the loaded document before displaying.

This event may be called:

- in TDBSRichViewEdit¹⁶⁸
- if Document⁵¹ is linked to a database field using LiveBindings

See also:

- OnNewDocument¹¹²

1.3.1.3.37 TSVRichViewEdit.OnMarginsChanged

Occurs before margins are changed.

type

```
TSRVMarginsChangedEvent = procedure (Sender: TSVRichViewEdit;  
    var NewLeft, NewTop, NewRight, NewBottom : Integer;  
    OldLeft, OldTop, OldRight, OldBottom : Integer) of object;
```

property OnMarginsChanged : TSRVMarginsChanged;

NewLeft, NewTop, NewRight, NewBottom – new margins, pixels.

OldLeft, OldTop, OldRight, OldBottom – old margins, pixels.

Unlike values of LeftMarginPix⁵⁴, TopMarginPix⁶⁷, RightMarginPix⁶⁰, BottomMarginPix⁴⁸ properties, these parameters contain unscaled values (as if ViewProperty⁶⁸.ZoomPercent¹⁶⁷=100).

If one of **Old***** parameters is equal to -1, this margin was not changed; the corresponding **New***** parameter must be ignored.

If you assign -1 to one of **New***** parameters, this margin will not be changed.

This event occurs:

- when assigning new values to LeftMargin⁽¹³⁷⁾, RightMargin⁽¹⁴¹⁾, TopMargin⁽¹⁴²⁾, BottomMargin⁽¹³⁵⁾ properties of PageProperty⁽⁵⁸⁾, or
- when calling SetMargin⁽⁹⁷⁾, SetMarginMM⁽⁹⁷⁾ or SetMarginUnit⁽⁹⁸⁾ methods.

Example:

```
// SRVPrint: TSRVPrint(203) is placed on the form
procedure TForm1.SRichViewEdit1MarginsChanged(
  Sender: TSVRichViewEdit; var NewLeft, NewTop, NewRight,
  NewBottom: Integer; OldLeft, OldTop, OldRight, OldBottom: Integer);
var
  ValueLeft, ValueRight, ValueTop, ValueBottom : Integer;
begin
  if SRVPrint1.CheckMargin(217)(ValueLeft, ValueTop, ValueRight, ValueBottom,
    NewLeft, NewTop, NewRight, NewBottom) then begin
    NewLeft := ValueLeft;
    NewTop := ValueTop;
    NewRight := ValueRight;
    NewBottom := ValueBottom;
  end;
end;
```

1.3.1.3.38 TSVRichViewEdit.OnMessageControl

Occurs when a message from **AControl** is received.

type

```
TSRVMessageControl = procedure(Sender: TSVRichViewEdit(40);
  AControl : TControl; var Message: TMessage) of object;
```

property OnMessageControl : TSRVMessageControl;

AControl – the control that sent the message. This control was inserted in the editor.

Message – the message from the control.

You can use this method to process messages from controls inserted in the editor. This is an alternative way to using events of these controls.

Example:

```
procedure TForm1.SRichViewEdit1MessageControl(
  Sender: TSVRichViewEdit(40); AControl: TControl; var Message: TMessage);
begin
  // if 'Ready' button is pressed
  if (AControl.Name = 'Ready') and (Message.Msg = WM_LBUTTONDOWN) then
    ...
end;
```

1.3.1.3.39 TSVRichViewEdit.OnNewDocument

Occurs when creating or before loading document

property OnNewDocument: TNotifyEvent;

See the event of the same name in the TRichView help file (event of TRichView).

This event may be called:

- in TDBSRichViewEdit⁽¹⁶⁸⁾
- if Document⁽⁵¹⁾ is linked to a database field using LiveBindings

Occurs:

- on creating new document (before displaying field from newly added record);
- before loading document from the field.

Use this event to assign default values to properties (such as background, margins, styles).

See also:

- OnLoadDocument⁽¹¹¹⁾.

1.3.1.3.40 TSRichViewEdit.OnOleDragEnter

OLE drag&drop event. It occurs when the user drags the mouse pointer into the editor. Indicates whether a drop can be accepted, and, if so, the effect of the drop.

property OnOleDragEnter: TRVOleDragEnterEvent;

See the event of the same name in the TRichView help file (event of TCustomRichViewEdit).

1.3.1.3.41 TSRichViewEdit.OnOleDragLeave

OLE drag&drop event. It occurs when the user drags the mouse cursor out of the editor or cancels the current drag&drop operation.

property OnOleDragLeave: TNotifyEvent;

See the event of the same name in the TRichView help file (event of TCustomRichViewEdit).

1.3.1.3.42 TSRichViewEdit.OnOleDragOver

OLE drag&drop event. It occurs when the user moves the mouse pointer over the editor. Called only if the dragging was accepted in OnOleDragEnter⁽¹¹³⁾ (or automatically by the editor).

property OnOleDragOver: TRVOleDragOverEvent;

See the event of the same name in the TRichView help file (event of TCustomRichViewEdit).

1.3.1.3.43 TSRichViewEdit.OnOleDrop

OLE drag&drop event. It occurs when the user completes drag&drop operation into the editor. This event allows you to insert data in your own format.

property OnOleDrop: TRVOleDropEvent;

See the event of the same name in the TRichView help file (event of TCustomRichViewEdit).

1.3.1.3.44 TSRichViewEdit.OnPageCountChanged

Occurs after PageCount⁽⁵⁷⁾ is changed.

property OnPageCountChanged: TNotifyEvent;

See also:

- OnCurrentPageChange⁽¹⁰⁷⁾

1.3.1.3.45 TSRichViewEdit.OnPageFormatChanged

Occurs after PageProperty⁽⁵⁸⁾.PageFormat⁽¹³⁸⁾ is changed.

property OnPageFormatChanged: TNotifyEvent;

1.3.1.3.46 TSRichViewEdit.OnPageScrolled

Occurs when a different page becomes visible.

property OnPageScrolled: TNotifyEvent;

The event may occur on changing position of vertical scrollbar. If a new page comes into the user's field of view (the middle of the screen), the event occurs. Index of this page is assigned to ScrolledPage⁽⁶⁵⁾.

1.3.1.3.47 TSRichViewEdit.OnPaint

Occurs on repainting.

type

```
TSRVPaintEvent = procedure (Sender: TSRichViewEdit(40); Canvas: TCanvas;
  Prepaint: Boolean; PaintRect: TRect) of object;
```

property OnPaint: TSRVPaintEvent;

The event occurs after repainting. You can draw something on top of default drawing.

Canvas – canvas where to draw.

Prepaint reserved for future use. This event occurs when the document is already painted.

PaintRect specifies the area that needs to be repainted. You can use this parameter to optimize drawing.

See also:

OnPaintPage⁽¹¹⁵⁾

OnPaintComponent⁽¹¹⁴⁾

1.3.1.3.48 TSRichViewEdit.OnPaintComponent

Allows custom painting for controls inserted in the editor.

type

```
TSRVComponentEvent = procedure (Sender: TSRichViewEdit(40);
  PaintMe: TControl; Canvas: TCanvas; ComponentRect: TRect;
  var isPaint: Boolean) of object;
```

property OnPaintComponent: TSRVComponentEvent;

All controls are inserted in RichViewEdit⁽⁵⁹⁾. TSRichViewEdit component creates images of controls and uses them when it needs to display them.

Not all controls can be drawn by the default procedure. This event allows you to draw controls yourself.

PaintMe – control to paint.

Canvas – canvas where to paint.

ComponentRect – coordinates where to paint.

isPaint - set to *True* if you drew this control (otherwise it will be drawn by the default procedure).

1.3.1.3.49 TSVRichViewEdit.OnPaintPage

Occurs before and after the page is drawn.

type

```
TSRVPaintPage = procedure(Sender: TObject; PageNo: Integer;
    PageRect, R: TRect; Canvas: TCanvas; Prepaint, Printing: Boolean)
of object;
```

property OnPaintPage: TSVRVPaintPage;

You can draw something on below or above the default drawing.

PageNo – index of the page, from 1.

PageRect – rectangle where to draw the page. Its size is the page size in 96 DPI and 100% zooming.

R – client coordinates of area that needs to be repainted.

Canvas – canvas where to paint.

Prepaint is *True* if this event is called before the default drawing (to draw a background), and *False* otherwise.

Printing is *True* if this event is called for printing the page.

You should paint as if zooming is 100%. When drawing on the screen, all painting made in this event is automatically scaled according to ViewProperty⁽⁶⁸⁾.ZoomPercent⁽¹⁶⁷⁾.

Example: drawing page numbers

```
procedure TForm1.SRichVewEdit1PaintPage(Sender: TObject; PageNo: Integer;
    PageRect, R: TRect; Canvas: TCanvas; Prepaint, Printing: Boolean);
var
    H : Integer;
    Text: String;
begin
    if Prepaint then
        exit;
    Canvas.Brush.Style := bsClear;
    Canvas.Font.Assign(SRichViewEdit1.RichViewEdit(59).Style.TextStyles[0]);
    H := SRichViewEdit1.TopMargin100Pix(67);
    Text := 'Page ' + IntToStr(PageNo);
    Canvas.TextOut(
        (PageRect.Left + PageRect.Right - Canvas.TextWidth(Text)) div 2,
        PageRect.Top+H div 2, Text);
end;
```

See also:

OnPaint⁽¹¹⁴⁾

1.3.1.3.50 TSVRichViewEdit.OnParaStyleConversion

Occurs while executing ApplyParaStyleConversion method of RichViewEdit⁽⁵⁹⁾, RVHeader⁽⁶³⁾, RVFooter⁽⁶¹⁾, or RVNote⁽⁶⁴⁾ (these TSVRichViewEdit controls are specified in the Sender parameter)

property OnParaStyleConversion: TRVStyleConversionEvent;

This event allows you to create custom conversion procedure for styles of the selected paragraphs. This event is called for all selected items allowing to change their paragraph style.

It can be useful for implementing commands like "change paragraph alignment", "change paragraph indents", etc.

See the event of the same name in the TSVRichView help file (event of TCustomRichViewEdit).

See also:

- OnStyleConversion⁽¹²⁰⁾

1.3.1.3.51 TSVRichViewEdit.OnPaste

Occurs when pasting from the Clipboard. Allows pasting data in custom formats.

property OnPaste: TRVPasteEvent;

See the event of the same name in the TSVRichView help file (event of TCustomRichViewEdit).

See also:

- ActiveEditor⁽⁴⁷⁾

1.3.1.3.52 TSVRichViewEdit.OnPrinting

Occurs on printing.

type

```
TSRVPrintingEvent = procedure (Sender: TSVRichViewEdit(40);  
    PageCompleted: Integer; Step: TRVPrintingStep) of object;
```

property OnPrinting : TSRVPrintingProgressEvent;

This is the analog of TRVPrint.OnSendingToPrinter, see the TSVRichView help file.

See also:

- TSRVPrint.OnSendingToPrinter⁽²²¹⁾

1.3.1.3.53 TSVRichViewEdit.OnProgress

Occurs on long operations.

property OnProgress : TRVProgressEvent;

See the event of the same name in the TSVRichView help file (event of TCustomRichView).

1.3.1.3.54 TSVRichViewEdit.OnReadField

Occurs when reading a field from RTF or DocX, allows to insert custom content.

property OnReadField: TRVReadFieldEvent;

See the event of the same name in the TSVRichView help file (event of TCustomRichView).

1.3.1.3.55 TSVRichViewEdit.OnReadHyperlink

Occurs when reading hyperlinks from RTF, DocX, HTML, Markdown, or when dragging hyperlink into the editor.

property OnReadHyperlink: TRVReadHyperlink;

See the event of the same name in the TRichView help file (event of TCustomRichView).

See also:

- OnWriteHyperlink⁽¹²²⁾

1.3.1.3.56 TSVRichViewEdit.OnReadMergeField

Occurs when reading content of merge fields from RTF, allows to modify it.

property OnReadHyperlink: TRVReaMergeFieldEvent;

See the event of the same name in the TRichView help file (event of TCustomRichView).

1.3.1.3.57 TSVRichViewEdit.OnResize, OnResizing

The events occur when the component is resized.

property OnResize: TNotifyEvent;

property OnResizing: TNotifyEvent;

OnResizing occurs before TSVRichViewEdit processes resizing itself.

OnResize occurs after that.

1.3.1.3.58 TSVRichViewEdit.OnResizeDoc

Occurs when RichViewEdit⁽⁵⁹⁾ is resized.

property OnResizeDoc: TNotifyEvent;

1.3.1.3.59 TSVRichViewEdit.OnRVDbClick

Occurs when user double-clicks the primary mouse button when the mouse pointer is above the item in RichView (or on single left-click, if *rvsSingleClick* is in RichViewEdit⁽⁵⁹⁾.Options)

property OnRVDbClick: TRVDbClickEvent;

See the event of the same name in the TRichView help file (event of TCustomRichView).

1.3.1.3.60 TSVRichViewEdit.OnRVFControlNeeded

Occurs when reading control (inserted in Sender) from RVF file or stream, and this control was saved without its "body".

property OnRVFControlNeeded : TRVFControlNeededEvent;

See the event of the same name in the TRichView help file (event of TCustomRichView).

1.3.1.3.61 TSVRichViewEdit.OnRVFImageListNeeded

Occurs when reading *bullet*, *hotspot* or list level with imagelist picture from RVF.

property OnRVFImageListNeeded : TRVFImageListNeededEvent;

See the event of the same name in the TRichView help file (event of TCustomRichView).

1.3.1.3.62 TSVRichViewEdit.OnRVFPictureNeeded

Occurs when reading picture or *hot-picture* from RVF file or stream, and this picture was saved without its "body"

property OnRVFPictureNeeded : TRVFPictureNeededEvent;

See the event of the same name in the TRichView help file (event of TCustomRichView).

1.3.1.3.63 TSVRichViewEdit.OnRVMouseDown

Occurs when the user presses a mouse button with the mouse pointer over the control.

property OnRVMouseDown : TRVMouseEvent;

X and **Y** parameters in this events are coordinates relative to RichViewEdit⁵⁹. If you need coordinates of this TSVRichViewEdit control, use OnMouseDown event.

See the event of the same name in the TRichView help file (event of TCustomRichView).

See also:

- ConvertRVToSRV⁷³
- OnRVMouseUp¹¹⁸

1.3.1.3.64 TSVRichViewEdit.OnRVMouseMove

Occurs when mouse pointer is moved to the area of hypertext link.

property OnRVMouseMove : TRVMouseMoveEvent;

See the event of the same name in the TRichView help file (event of TCustomRichView).

See also:

- OnJump¹¹⁰

1.3.1.3.65 TSVRichViewEdit.OnRVMouseUp

Occurs when the user releases a mouse button that was pressed with the mouse pointer over the component.

property OnRVMouseUp : TRVMouseEvent;

X and **Y** parameters in this events are coordinates relative to RichViewEdit⁵⁹. If you need coordinates of this TSVRichViewEdit control, use OnMouseUp event.

See the event of the same name in the TRichView help file (event of TCustomRichView).

See also:

- ConvertRVToSRV⁷³
- OnRVMouseDown¹¹⁸

1.3.1.3.66 TSRichViewEdit.OnRVRightClick

Occurs when the user releases the right mouse button over the RichView component.

property OnRVRightClick: TRVRightClickEvent;

X and **Y** parameters in this events are coordinates relative to RichViewEdit⁽⁵⁹⁾. If you need coordinates of this TSRichViewEdit control, use OnMouseUp event.

See the event of the same name in the TRichView help file (event of TCustomRichView).

1.3.1.3.67 TSRichViewEdit.OnSaveComponentToFile

Occurs when RichViewEdit⁽⁵⁹⁾ wants to save inserted control in text, RTF or HTML file.

property OnSaveComponentToFile: TRVSaveComponentToFileEvent;

See the event of the same name in the TRichView help file (event of TCustomRichView).

1.3.1.3.68 TSRichViewEdit.OnSaveCustomFormat

This event allows saving data to field in your own format.

type

```
TSRVCustomFormatEvent = procedure (Sender: TSRichViewEdit(40);  
    Stream: TStream; var DoDefault: Boolean) of object;
```

property OnSaveCustomFormat: TDBSRVCustomFormatEvent;

This event may be called:

- in TDBSRichViewEdit⁽¹⁶⁸⁾
- if Document⁽⁵¹⁾ is linked to a database field using LiveBindings

See the event of the same name in the TRichView help file (event of TRichViewEdit).

See also:

- OnLoadCustomFormat⁽¹¹⁹⁾

1.3.1.3.69 TSRichViewEdit.OnSaveDocXExtra

Allows saving additional information in DocX files.

property OnSaveDocXExtra: TRVSaveDocXExtraEvent;

See the event of the same name in the TRichView help file (event of TCustomRichView).

1.3.1.3.70 TSRichViewEdit.OnSaveHTMLExtra

Allows saving additional information in HTML.

property OnSaveHTMLExtra: TRVSaveHTMLExtraEvent;

See the event of the same name in the TRichView help file (event of TCustomRichView).

1.3.1.3.71 TSRichViewEdit.OnSaveImage2

Occurs when RichViewEdit⁽⁵⁹⁾ saves image to HTML file or stream.

property OnSaveImage2 : TRVSaveImageEvent2;

See the event of the same name in the TRichView help file (event of TCustomRichView).

See also:

- OnHTMLSaveImage¹⁰⁹

1.3.1.3.72 TSVRichViewEdit.OnSaveItemToFile

Allows you to change how document items are saved in text, RTF or HTML file or stream.

property OnSaveItemToFile: TRVSaveItemToFileEvent;

See the event of the same name in the TRichView help file (event of TCustomRichView).

1.3.1.3.73 TSVRichViewEdit.OnSaveRTFExtra

Allows saving additional information in RTF.

property OnSaveRTFExtra: TRVSaveRTFExtraEvent;

See the event of the same name in the TRichView help file (event of TCustomRichView).

1.3.1.3.74 TSVRichViewEdit.OnSelect

Occurs when selection in the document was changed (not contents, but bounds)

property OnSelect: TNotifyEvent;

See the event of the same name in the TRichView help file (event of TCustomRichView).

1.3.1.3.75 TSVRichViewEdit.OnSmartPopupClick

Occurs when user clicks "smart popup" button.

property OnSmartPopupClick: TRVSmartPopupClickEvent;

This event allows implementing your own effect when clicking "smart popup" button (or on pressing its ShortCut). If you use menu, it's not necessary to process this event.

See also:

- SmartPopupProperties⁶⁵
- SmartPopupVisible⁶⁵

1.3.1.3.76 TSVRichViewEdit.OnSpellingCheck

Allows to mark misspelled words.

property OnSpellingCheck: TRVSpellingCheckEvent;

See the event of the same name in the TRichView help file (event of TCustomRichView).

1.3.1.3.77 TSVRichViewEdit.OnStyleConversion

Occurs while executing ApplyStyleConversion method of RichViewEdit⁵⁹, RVHeader⁶³, RVFooter⁶¹, or RVNote⁶⁴ (these TRichViewEdit controls are specified in the Sender parameter)

property OnStyleConversion: TRVStyleConversionEvent;

This event allows you to create custom conversion procedure of text styles for the selected text. This event is called for all selected text items allowing to change their text style.

It can be useful to implement commands like "make bold", "apply font", "change text color", etc.

See the event of the same name in the TRichView help file (event of TCustomRichViewEdit).

See also:

- OnParaStyleConversion⁽¹¹⁶⁾

1.3.1.3.78 TSRichViewEdit.OnTableIconClick

Occurs when the user clicked a table "icon" (rectangle in the top left corner of table)

property OnTableIconClick: TSRVMouseEvent⁽²⁷⁶⁾;

Table icon is displayed when the user moves the mouse pointer above the table, if ViewProperty⁽⁶⁸⁾.UseTableIcons⁽¹⁶³⁾=True.

This table is returned by GetTableIconItem⁽⁸⁹⁾. A document where this table is located is returned by GetTableIconRVData⁽⁸⁹⁾.

1.3.1.3.79 TSRichViewEdit.OnTextFound

Occurs as a result of RichViewEdit⁽⁵⁹⁾.SearchText.

property OnTextFound: TRVTextFoundEvent;

See the event of the same name in the TRichView help file (event of TCustomRichViewEdit).

1.3.1.3.80 TSRichViewEdit.OnURLNeeded

An obsolete version of OnWriteHyperlink⁽¹²²⁾

property OnURLNeeded: TRVURLNeededEvent;

See the event of the same name in the TRichView help file (event of TCustomRichView).

1.3.1.3.81 TSRichViewEdit.OnVMenuClickButton

Occurs when the user clicks one of buttons of the vertical menu (toolbar).

property OnVMenuClickButton: TSRVClickButtonEvent⁽²⁷¹⁾;

ToolButton – the clicked button (one of MenuVButtons⁽⁵⁵⁾), or *nil*, if the user clicked outside any button.

See also:

- OnHMenuClickButton⁽¹⁰⁹⁾
- OnVMenuEnterButton⁽¹²¹⁾
- MenuVertical⁽⁵⁶⁾

1.3.1.3.82 TSRichViewEdit.OnVMenuEnterButton

The event occurs when the user moves the mouse pointer to one of the buttons of the vertical menu.

property OnVMenuEnterButton: TSRVClickButtonEvent⁽²⁷¹⁾;

ToolButton – the button under the mouse pointer, one of MenuVButtons⁽⁵⁵⁾.

See also:

- OnHMenuEnterButton⁽¹⁰⁹⁾

- OnVMenuClickButton⁽¹²¹⁾
- MenuVertical⁽⁵⁶⁾

1.3.1.3.83 TSRichViewEdit.OnVScrolled

Occurs on changing position of vertical scrollbar.

property OnVScrolled: TNotifyEvent;

This event occurs when the user changes position of vertical scrollbar, or when new value is assigned to VScrollPos⁽⁶⁹⁾ property.

See also:

- OnHScrolled⁽¹⁰⁹⁾

1.3.1.3.84 TSRichViewEdit.OnWriteHyperlink

Occurs when RichViewEdit⁽⁵⁹⁾ wants to save hypertext link to HTML, RTF, or DocX file.

property OnWriteHyperlink: TRVWriteHyperlink;

See the event of the same name in the TRichView help file (event of TCustomRichView).

1.3.1.3.85 TSRichViewEdit.OnWriteObjectProperties

Occurs when RichViewEdit⁽⁵⁹⁾ wants saves and object (for example, image) to HTML, RTF, or DocX file. This event allows saving additional object properties.

property OnWriteObjectProperties: TRVWriteObjectPropertiesEvent;

See the event of the same name in the TRichView help file (event of TCustomRichView).

1.3.1.3.86 TSRichViewEdit.OnZoomChanged

Occurs after ViewProperty⁽⁶⁸⁾.ZoomPercent⁽¹⁶⁷⁾ or .ZoomMode⁽¹⁶⁵⁾ are changed.

property OnZoomChanged: TNotifyEvent;

See also:

- OnZoomViewChanged⁽¹²²⁾

1.3.1.3.87 TSRichViewEdit.OnZoomViewChanged

Occurs when zooming properties are changed.

property OnZoomViewChanged: TNotifyEvent;

This even occurs when one of the following properties of ViewProperty⁽⁶⁸⁾ is changed:

- ZoomPercentEdit⁽¹⁶⁸⁾
- ZoomModeEdit⁽¹⁶⁶⁾
- ZoomPercentIN⁽¹⁶⁸⁾
- ZoomModeIN⁽¹⁶⁶⁾
- ZoomPercentOUT⁽¹⁶⁸⁾
- ZoomModeOUT⁽¹⁶⁶⁾

See also:

- OnZoomChanged⁽¹²²⁾

1.3.1.4 Classes of properties

Classes of TSVRichViewEdit⁽⁴⁰⁾ Properties

- TSVRVBackgroundProperty⁽¹²³⁾ – properties of the editor's background; the class of BackgroundProperty⁽⁴⁸⁾ property.
- TSVRVPropertyTBH⁽¹⁴⁷⁾ (inherited from TSVRVCustomPropertyTB⁽¹²⁶⁾) – properties of the toolbar in the horizontal scrollbar area; the class of MenuHorizontal⁽⁵⁵⁾ property;
- TSVRVPropertyTBV⁽¹⁴⁸⁾ (inherited from TSVRVCustomPropertyTB⁽¹²⁶⁾) – properties of the toolbar in the vertical scrollbar area; the class of MenuVertical⁽⁵⁶⁾ property;
- TSVRVLineNumberProperty⁽¹³¹⁾ – properties of line numbers; the class of LineNumberProperty⁽⁵⁴⁾;
- TSVRVPageProperty⁽¹³³⁾ – page properties; most of them affect page appearance both on the screen and on paper; the class of PageProperty⁽⁵⁸⁾ property.
- TSVRVViewProperty⁽¹⁵³⁾ – visual page properties; most of them affect page appearance only on the screen; the class of ViewProperty⁽⁶⁸⁾ property.

1.3.1.4.1 TSVRVBackgroundProperty

TSVRVBackgroundProperty is the class for TSVRichViewEdit⁽⁴⁰⁾.BackgroundProperty⁽⁴⁸⁾ property.

Unit SclRView.

Syntax

```
TSVRVBackgroundProperty = class (TPersistent)
```

Hierarchy

TObject

TPersistent

Properties for area around pages

This class defines properties for an advanced background filling. An advanced background is a combination of a gradient filling and a picture. It is activated if Visible⁽¹²⁶⁾=True.

A gradient is defined in FillType⁽¹²⁴⁾, ColorBegin⁽¹²⁴⁾, ColorMiddle⁽¹²⁴⁾, ColorEnd⁽¹²⁴⁾, PercentMiddle⁽¹²⁵⁾ properties.

A picture is defined in Picture⁽¹²⁵⁾, PicturePosition⁽¹²⁶⁾, and TransparentColor⁽¹²⁶⁾ properties.

Properties for pages

GlobalPageBackgroundColor⁽¹²⁵⁾ allows changing color of pages.

1.3.1.4.1.1 Properties

In TSVRVBackgroundProperty

- ColorBegin⁽¹²⁴⁾
- ColorEnd⁽¹²⁴⁾
- ColorMiddle⁽¹²⁴⁾
- FillType⁽¹²⁴⁾
- PercentMiddle⁽¹²⁵⁾

- Picture ⁽¹²⁵⁾
- PicturePosition ⁽¹²⁶⁾
- TransparentColor ⁽¹²⁶⁾
- Visible ⁽¹²⁶⁾

Specifies the initial color of gradient filling.

property ColorBegin: TColor;

If FillType ⁽¹²⁴⁾ = *srvtfFlat*, this property defines a background color.

This color is also used for background of zooming panel, if it does not use a gradient.

If VCL theme is active (in Delphi XE2 or newer), and this color is a system color, a corresponding theme color is used.

Default value:

\$00E7BE9F

Specifies the ending color of gradient filling

property ColorEnd: TColor;

If FillType ⁽¹²⁴⁾ = *srvtfFlat*, this property is not used.

If VCL theme is active (in Delphi XE2 or newer), and this color is a system color, a corresponding theme color is used.

Default value:

\$00CD9165

Specifies the middle color of gradient filling

property ColorMiddle : TColor;

If FillType ⁽¹²⁴⁾ = *srvtfFlat*, this property is not used.

If **ColorMiddle** = *cNone*, this color is ignored, a gradient is drawn from ColorBegin ⁽¹²⁴⁾ to ColorEnd ⁽¹²⁴⁾.

If VCL theme is active (in Delphi XE2 or newer), and this color is a system color, a corresponding theme color is used.

Default value:

\$00BD865D

See also:

- PercentMiddle ⁽¹²⁵⁾

Specifies the background filling mode.

type

TSRVFillType = (srvtfFlat, srvtfGradientH, srvtfGradientV);

property FillType : TSRVFillType;

Value	Meaning
-------	---------

<i>srvtfFlat</i>	A plain colored background, the color is ColorBegin ⁽¹²⁴⁾ .
<i>srvtfGradientH</i>	A horizontal gradient, the background is filled with horizontal gradient using three colors: from ColorBegin ⁽¹²⁴⁾ through ColorMiddle ⁽¹²⁴⁾ to ColorEnd ⁽¹²⁴⁾ . If ColorMiddle = cNone, only two colors are used.
<i>srvtfGradientV</i>	A vertical gradient, the background is filled with vertical gradient using three colors: from ColorBegin ⁽¹²⁴⁾ through ColorMiddle ⁽¹²⁴⁾ to ColorEnd ⁽¹²⁴⁾ . If ColorMiddle = cNone, only two colors are used.

ColorMiddle⁽¹²⁴⁾ is shifted by the percent specified in PercentMiddle⁽¹²⁵⁾.

Default value:

srvtfFlat

Specifies a page background color.

property GlobalPageBackgroundColor: TColor;

Unlike all other properties of BackgroundProperty⁽⁴⁸⁾, this property affects pages, not the area around them.

This property (if not equal to *cNone*) overrides the value of RichViewEdit⁽⁵⁹⁾.Color. Unlike RichViewEdit⁽⁵⁹⁾.Color, this property is not stored in RVF files.

This property has a lower priority than the editor's ReadModeProperty⁽⁵⁹⁾.PageColor⁽¹⁵²⁾.

Default value:

cNone

Specifies the shift of the middle color, percent of the gradient width.

property PercentMiddle: Single;

This value defines the position where a gradient color becomes equal to ColorMiddle⁽¹²⁴⁾.

Default value:

50.0

Specifies the picture to display above the gradient.

property Picture: TPicture;

The position of this picture is defined in PicturePosition⁽¹²⁶⁾.

Specifies the position of Picture⁽¹²⁵⁾.

type

```
TSRVPicturePosition = (srvtpNone, srvtpTopLeft, srvtpCenter,
    srvtpTile, srvtpStretch, srvtpScale);
```

property PicturePosition: TSRVPicturePosition;

Value	Meaning
<i>srvtpNone</i>	picture is not displayed
<i>srvtpTopLeft</i>	picture is in the top left corner
<i>srvtpCenter</i>	picture in the center
<i>srvtpTile</i>	picture is tiled
<i>srvtpStretch</i>	picture is stretched to the window size
<i>srvtpScale</i>	picture is stretched to the window size, proportionally

Default value:

srvtpNone

Specifies a color of Picture⁽¹²⁵⁾ that will be transparent when it is drawn.

property ColorBegin: TColor;

This property is used only if Picture⁽¹²⁵⁾ contains TBitmap.

Default value:

clNone (no transparency)

Disables/enables background displaying.

property Visible : Boolean;

If the value is *False*, all properties of this class are ignored and background is filled with TSRichViewEdit⁽⁴⁰⁾.Color.

A zooming panel⁽¹⁶⁷⁾ is always filled according to BackgroundProperty, even if **Visible**=*False*.

Default value:

False

1.3.1.4.2 TSRVCustomPropertyTB

TSRVCustomPropertyTB is the base class for TSRVPropertyTBH⁽¹⁴⁷⁾ and TSRVPropertyTBV⁽¹⁴⁸⁾.

Unit SRVToolBar.

Syntax

```
TSRVCustomPropertyTB = class (TPersistent)
```

Hierarchy

TObject

TPersistent

Description

This class is not used directly. It is a base class for types of `MenuHorizontal`⁽⁵⁵⁾ and `MenuVertical`⁽⁵⁶⁾ properties of `TSRichViewEdit`⁽⁴⁰⁾.

This class defines properties of toolbars. All properties are protected.

Some of these properties make sense only for toolbars displayed in tool windows, these properties are not published in `TSRVPropertyTBH`⁽¹⁴⁷⁾ and `TSRVPropertyTBV`⁽¹⁴⁸⁾.

Properties

This class contains almost the same set of properties as `TSRVToolBar`⁽¹⁸⁸⁾ and `TSRVToolWindow`⁽¹⁹⁶⁾ components, but a hint area is hidden.

Buttons are defined in `Buttons`⁽¹²⁸⁾ collection (or in external collections, like in case of `MenuHorizontal`⁽⁵⁵⁾ and `MenuVertical`⁽⁵⁶⁾ properties). A size of buttons is defined in `ButtonWidth`⁽¹²⁸⁾ and `ButtonHeight`⁽¹²⁸⁾ properties. A spacing between buttons is specified in `Spacer`⁽¹³⁰⁾. A horizontal positions of buttons is defined by `AlignButton`⁽¹²⁸⁾ and `Indent`⁽¹³⁰⁾ properties.

The toolbar can have a border, its width is specified in `BorderWidth`⁽¹²⁸⁾.

The toolbar color is `Color`⁽¹²⁹⁾. Colors of disabled, pressed and highlighted buttons are specified in `ColorDisable`⁽¹²⁹⁾, `ColorDown`⁽¹²⁹⁾, and `ColorSelect`⁽¹²⁹⁾ properties. A pen of frame around the toolbar is defined in `PenFrame`⁽¹³⁰⁾.

1.3.1.4.2.1 Properties

In `TSRVCustomPropertyTB`

- `BorderWidth`⁽¹²⁸⁾
- `BorderWidth`⁽¹²⁸⁾
- `ButtonHeight`⁽¹²⁸⁾
- `Buttons`⁽¹²⁸⁾
- `ButtonWidth`⁽¹²⁸⁾
- `Color`⁽¹²⁹⁾
- `ColorDisable`⁽¹²⁹⁾
- `ColorDown`⁽¹²⁹⁾
- `ColorSelect`⁽¹²⁹⁾
- `ImageList`⁽¹³⁰⁾
- `Indent`⁽¹³⁰⁾
- `PenFrame`⁽¹³⁰⁾
- `ScaleImagesForDPI`⁽¹³⁰⁾
- `Spacer`⁽¹³⁰⁾
- ▶ `SRVToolBar`⁽¹³¹⁾

Defines the position of buttons in the toolbar window.

property AlignButton: TSRVAlignButton⁽²⁷⁰⁾;

See also

Indent⁽¹³⁰⁾

Default value:

srvabCenter

Specifies a width of the toolbar border (spacing around the toolbar)

property BorderWidth: Integer;

A border (see PenFrame⁽¹³⁰⁾) is drawn in this area. The rest of space (if BorderWidth>the pen's width) is painted with Color⁽¹²⁹⁾.

This value is measured in pixels at 96 pixels per inch; the actual size scaled according to the current screen pixel density.

Default value:

1

See also:

- Spacer⁽¹³⁰⁾

Sets the height of tool buttons (TSRVToolButton⁽²⁷¹⁾)

property ButtonHeight: TRVPixel96Length;

This value is measured in pixels at 96 pixels per inch; the actual size scaled according to the current screen pixel density.

Default value:

20

See also:

- ButtonWidth⁽¹²⁸⁾

Collection of tool buttons (TSRVToolButton⁽²⁷¹⁾).

property Buttons: TSRVButtonCollection⁽²⁷⁰⁾;

For horizontal menu⁽⁵⁵⁾ and vertical menu⁽⁵⁶⁾ of TSRichViewEdit⁽⁴⁰⁾, this property is not published. For them, buttons are available in MenuHButtons⁽⁵⁵⁾ and MenuVButtons⁽⁵⁵⁾ properties of TSRichViewEdit⁽⁴⁰⁾.

Width of tool buttons (TSRVToolButton⁽²⁷¹⁾).

property ButtonWidth: TRVPixel96Length;

This value is measured in pixels at 96 pixels per inch; the actual size scaled according to the current screen pixel density.

Default value:

26

See also:

- ButtonHeight⁽¹²⁸⁾

Specifies the toolbar background color.

property Color: TColor;

This color is used for background of:

- the border around the toolbar (see BorderWidth⁽¹²⁸⁾)
- area between the buttons (see Spacer⁽¹³⁰⁾)
- tool buttons (if the button is in a normal state, i.e. not highlighted, pressed or disabled).

Default value:

c/White

See also:

- ColorDisable⁽¹²⁹⁾
- ColorDown⁽¹²⁹⁾
- ColorSelect⁽¹²⁹⁾

Background color of disabled tool buttons.

property ColorDisable: TColor;

This color is used for buttons having Enabled⁽²⁷²⁾=False.

Default value:

clSilver

See also:

- Color⁽¹²⁹⁾
- ColorDown⁽¹²⁹⁾
- ColorSelect⁽¹²⁹⁾

Background color of pressed tool buttons.

property ColorDown: TColor;

Default value:

\$00EED2C1

See also:

- Color⁽¹²⁹⁾
- ColorDisable⁽¹²⁹⁾
- ColorSelect⁽¹²⁹⁾

Background color of highlighted tool buttons (buttons under the mouse pointer).

property ColorSelect: TColor;

Default value:

\$00E2B598

See also:

- Color⁽¹²⁹⁾
- ColorDisable⁽¹²⁹⁾

- ColorDown ⁽¹²⁹⁾

A reference to image list for images in tool buttons ⁽¹²⁸⁾.

property ImageList: TCustomImageList;

This image list must contain buttons glyphs for all buttons states (normal, pressed, disabled) (you can use the same image for all of them).

When a toolbar displayed on a monitor that has pixel density (DPI) higher than 96, the toolbar draws images stretched, proportionally to the screen DPI. To turn off stretching, assign ScaleImagesForDPI ⁽¹³⁰⁾ = *False*.

Changes the position of buttons in the toolbar window.

property Indent: Integer;

If AlignButton ⁽¹²⁸⁾ = *svabLeft*, buttons are shifted by Indent to the right.

If AlignButton ⁽¹²⁸⁾ = *svabRight*, buttons are shifted by Indent to the left.

If AlignButton ⁽¹²⁸⁾ = *svabCenter*, buttons are shifted by Indent/2 to the left.

Default value:

0

Specifies a pen for drawing lines in the tool bar.

property PenFrame: TPen;

This pen is used for drawing border around the toolbar. The width of this border is defined as a pen's width, not in BorderWidth ⁽¹²⁸⁾.

Default value:

Style=*psSolid*, Width=1, Color=**\$00C56A31**;

Specify whether images from ImageList ⁽¹³⁰⁾ are displayed stretched according to the screen pixel density (DPI).

property ScaleImagesForDPI: Boolean;

If *True*, toolbar images are stretched from 96 DPI to the screen DPI.

If *False*, toolbar images are displayed as they are. This mode is useful if ImageList ⁽¹³⁰⁾ contains images that corresponds to the current DPI.

Default value

True

Specifies the distance between tool buttons ⁽¹²⁸⁾.

property Spacer: TRVPixel96Length;

This value is measured in pixels at 96 pixels per inch; the actual size scaled according to the current screen pixel density.

Default value:

0

Returns a reference to the toolbar.

```
SRVToolBar: TSRVToolBar(188);
```

This property can be used to get the toolbar position. Please do not modify properties of this toolbar.

1.3.1.4.3 TSRVLineNumberProperty

TSRVLineNumberProperty is the class for TSVRichViewEdit⁽⁴⁰⁾.LineNumberProperty⁽⁵⁴⁾ property.

Unit SclRView.

Syntax

```
TSRVLineNumberProperty = class(TPersistent)
```

Hierarchy

TObject

TPersistent

Description

Line numbers are shown (both when displaying on the screen and when printing) if Visible⁽¹³³⁾=*True*.

Numbers are counted on each page from StartFrom⁽¹³²⁾. Every Step⁽¹³²⁾-th line number is shown. Other line numbers are not shown, or shown as ticks, depending Ticks⁽¹³²⁾ property.

Line numbers are displayed to the left side of the main text area (or to the right side, if BiDiMode⁽⁴⁸⁾=*rvbdRightToLeft*). A distance from the main text area is defined in DistanceFromText⁽¹³¹⁾.

The font is defined in Font⁽¹³²⁾.

1.3.1.4.3.1 Properties

In TSRVLineNumberProperty

- DistanceFromText⁽¹³¹⁾
- Font⁽¹³²⁾
- StartFrom⁽¹³²⁾
- Step⁽¹³²⁾
- TablesAsLines⁽¹³²⁾
- Ticks⁽¹³²⁾
- Visible⁽¹³³⁾

Specifies a distance between line numbers and the main text area.

```
property Visible: Boolean;
```

This value is measured in pixels in the resolution 96 dpi.

Default value:

5

Specifies the font for drawing line numbers.

type

```
TSRVLineNumberFont = class (TFont);
```

property Font: TSRVLineNumberFont;

Default value:

'Tahoma', 8, *clBtnShadow*

Specifies the value of the first number on a page and the increment.

property StartFrom: Integer;

Default value:

1

property Step: Integer;

Only every **Step**-th line number is drawn as a number. Other numbers are not drawn, or drawn as ticks, depending on Ticks⁽¹³²⁾ property.

If Ticks⁽¹³²⁾=*True*, the first number on the page is always shown.

Examples:

- Step=1; line numbers: 1 2 3 4 5, etc.
- Step=10, Ticks=*False*; line numbers: 10 20 30, etc. (other line numbers are not shown)
- Step=10, Ticks=*True*; line numbers: 1 . . . | 10 | 20, etc.

Default value:

1

Specifies whether tables are counted as lines.

property TablesAsLines: Boolean;

If *True*, each table is counted as one line (although line numbers are not drawn for tables). This method is compatible with the results returned by GetCurrentLineCol⁽⁸³⁾.

If *False*, tables are ignored when counting lines (like in Microsoft Word)

Default value:

True

Shows or hides ticks.

property Ticks: Boolean;

If *True*, and Step⁽¹³²⁾>1, all intermediate line numbers are drawn as ticks. Additionally, a vertical line is drawn between text and line numbers.

If *False*, intermediate line numbers are not drawn.

Default value:

False

Shows or hides line numbers.

property Visible: Boolean;

Default value:

False

1.3.1.4.4 TSRVPageProperty

TSRVPageProperty is the class for TSRichViewEdit⁽⁴⁰⁾.PageProperty⁽⁵⁸⁾ property.

Unit SclRView.

Syntax

```
TSRVPageProperty = class (TPersistent)
```

Hierarchy

TObject

TPersistent

Properties

Page sizes and margins

The following properties define the page size: PageFormat⁽¹³⁸⁾ and Orientation⁽¹³⁸⁾.

PageWidth⁽¹⁴¹⁾ and PageHeight⁽¹³⁹⁾ are usually calculated automatically, but they can define a page size if PageFormat⁽¹³⁸⁾=*srvfmCustom*.

The following properties define margins: LeftMargin⁽¹³⁷⁾, TopMargin⁽¹⁴²⁾, RightMargin⁽¹⁴¹⁾, BottomMargin⁽¹³⁵⁾.

MirrorMargins⁽¹³⁸⁾ allows switching left and right margins on even pages.

All sizes are measured in TSRichViewEdit.UnitsProgram⁽⁶⁸⁾.

Viewing modes

The following properties are used to implement different viewing modes: AutoWidth⁽¹³⁵⁾ and PageViewMode⁽¹⁴⁰⁾.

Typical modes:

- "Normal mode" (one long page, width depends on PageWidth⁽¹⁴¹⁾): AutoWidth⁽¹³⁵⁾=*False*, PageViewMode⁽¹⁴⁰⁾=*False*.
- "Web mode" (one long page, width depends on the window size): AutoWidth⁽¹³⁵⁾=*True*, PageViewMode⁽¹⁴⁰⁾=*False*.
- "Page View Mode": AutoWidth⁽¹³⁵⁾=*False*, PageViewMode⁽¹⁴⁰⁾=*True*.

A border can be drawn around pages, see BorderPen⁽¹³⁵⁾. A printable area can be shown, see PrintableAreaPen⁽¹⁴¹⁾ (the current printer cannot print beyond this area).

The caret is drawn using CaretPen⁽¹³⁵⁾.

Page numbering

Page numbers are printed, if `PageNoVisible`⁽¹⁴⁰⁾ = `True`. This is a simple option to display page numbering. The alternative (recommended) option is inserting "page number" field (TRVPageNumberItemInfo item) in a header or a footer⁽⁶⁶⁾.

They are printed at the top or the bottom of pages, depending on `PageNoVAlign`⁽¹⁴⁰⁾ property. A horizontal alignment is specified in `PageNoHAlign`⁽¹³⁹⁾, font is specified in `PageNoFont`⁽¹³⁹⁾.

The numbering is started from `PageNoFromNumber`⁽¹³⁹⁾. You can omit page numbers on one or more first pages, using `PageNoFirst`⁽¹³⁹⁾.

`PageNoFromNumber`⁽¹³⁹⁾ is also used as a starting value for "page number" fields.

Header and footer

Headers⁽⁶⁶⁾ are displayed, if `HeaderVisible`⁽¹³⁷⁾ = `True`. Headers position is defined in `HeaderY`⁽¹³⁷⁾.

Footers⁽⁶⁶⁾ are displayed, if `FooterVisible`⁽¹³⁶⁾ = `True`. Footers position is defined in `FooterY`⁽¹³⁶⁾.

If `TitlePage`⁽¹⁴¹⁾ = `True`, special headers are used for the first page. If `FacingPages`⁽¹³⁶⁾ = `True`, special headers are used for even pages.

See also:

`TSRVViewProperty`⁽¹⁵³⁾.

1.3.1.4.4.1 Properties

In TSRVPageProperty

- `AutoWidth`⁽¹³⁵⁾
- `BorderPen`⁽¹³⁵⁾
- `BottomMargin`⁽¹³⁵⁾
- `CaretPen`⁽¹³⁵⁾
- `FacingPages`⁽¹³⁶⁾
- `FooterVisible`⁽¹³⁶⁾
- `FooterY`⁽¹³⁶⁾
- `HeaderVisible`⁽¹³⁷⁾
- `HeaderY`⁽¹³⁷⁾
- `LeftMargin`⁽¹³⁷⁾
- `MirrorMargins`⁽¹³⁸⁾
- `Orientation`⁽¹³⁸⁾
- `PageFormat`⁽¹³⁸⁾
- `PageHeight`⁽¹³⁹⁾
- `PageNoFirst`⁽¹³⁹⁾
- `PageNoFont`⁽¹³⁹⁾
- `PageNoFromNumber`⁽¹³⁹⁾
- `PageNoHAlign`⁽¹³⁹⁾
- `PageNoVAlign`⁽¹⁴⁰⁾
- `PageNoVisible`⁽¹⁴⁰⁾
- `PageViewMod`⁽¹⁴⁰⁾
- `PageWidth`⁽¹⁴¹⁾
- `PrintableAreaPen`⁽¹⁴¹⁾
- `RightMargin`⁽¹⁴¹⁾

- TitlePage⁽¹⁴¹⁾
- TopMargin⁽¹⁴²⁾
- UsePageOrders⁽¹⁴²⁾

Specifies whether page width is automatically calculated depending on the editor's client width.

property AutoWidth: Boolean;

If this property is *True*, a document occupies all available width (except for the double PagePosProperty⁽⁵⁸⁾.HPadding⁽¹⁴⁵⁾), a horizontal scrollbar is never enabled. In a read mode⁽¹⁵⁰⁾, document also occupies all available height (except for the double PagePosProperty⁽⁵⁸⁾.VPadding⁽¹⁴⁵⁾). In this mode, zooming⁽¹⁶⁷⁾ is applied to a document content and is not applied to sizes of pages. Page size⁽¹⁴¹⁾ and format⁽¹³⁸⁾ specified are ignored in this mode.

This property may be used when implementing custom view modes, for example, "web mode" and "read mode".

Default value:

False

Specifies pen properties for drawing border around pages.

type

```
TSRVBorderPen = class (TPen);
```

property BorderPen: TSRVBorderPen;

To disable a page border drawing, assign BorderPen.Style = *psClear*.

Default value:

Color = `$003E3E3E`, Width = 1.

Specifies a bottom margin of pages.

property BottomMargin: TRVLength;

This value is measured in UnitsProgram⁽⁶⁸⁾ (mm by default).

This property can be changed as an editing (undoable) operation, see TSRichViewEdit⁽⁴⁰⁾.SetFloatPropertyEd⁽⁹⁶⁾.

Default value:

25

Specifies pen properties for drawing the caret.

type

```
TSRVCaretPen = class (TPen)
```

property CaretPen: TSRVCaretPen;

If **CaretPen.Width** = 0 (default), the system caret thickness is used (defined in the "Ease of Access" of the Settings app (or the Control Panel, depending in the Windows version)).

CaretPen.Color and **Mode** are ignored (the caret always uses **Mode** = *pmNot*)

Specifies whether special headers and footers are used for even pages.

property FacingPages: Boolean;

This property can be changed as an editing (undoable) operation, see `TSRichViewEdit`⁽⁴⁰⁾.
`.SetIntPropertyEd`⁽⁹⁶⁾.

A header for even pages is stored in `SRichViewEdit.Subdocuments`⁽⁶⁶⁾ [`srvhftEvenPagesHeader`].

A footer for even pages is stored in `SRichViewEdit.Subdocuments`⁽⁶⁶⁾ [`srvhftEvenPagesFooter`].

FacingPages turns on/off displaying headers and footers for even pages, but does not affect saving: headers and footers are loaded and saved regardless of this property.

Headers are shown only if `HeaderVisible`⁽¹³⁷⁾=`True`, footers are shown only if `FooterVisible`⁽¹³⁶⁾=`True`.

Default value:

False

See also:

- `TitlePage`⁽¹⁴¹⁾
- `MirrorMargins`⁽¹³⁸⁾

Shows or hides footers.

property FooterVisible: Boolean;

Footers are stored in `TSRichViewEdit`⁽⁴⁰⁾.`Subdocuments`⁽⁶⁶⁾ and edited in `TSRichViewEdit.RVFooter`⁽⁶¹⁾.

If `FooterVisible`=`False`, footers are not displayed, not loaded and not saved to files.

Default value:

True

See also:

- `FooterY`⁽¹³⁶⁾
- `HeaderVisible`⁽¹³⁷⁾
- `TitlePage`⁽¹⁴¹⁾
- `FacingPages`⁽¹³⁶⁾

Specifies a distance between the bottom of the page and the bottom of the footer.

property FooterY: TRVLength;

This value is measured in `UnitsProgram`⁽⁶⁸⁾ (mm by default).

A footer is shown only if `FooterVisible`⁽¹³⁶⁾=`True`.

Footers are stored in `TSRichViewEdit`⁽⁴⁰⁾.`Subdocuments` and edited in `TSRichViewEdit.RVFooter`⁽⁶¹⁾.

This value is used for all types of footers.

This property can be changed as an editing (undoable) operation, see `TSRichViewEdit`⁽⁴⁰⁾.
`.SetFloatPropertyEd`⁽⁹⁶⁾.

Default value:

10

See also

- HeaderY⁽¹³⁷⁾

Shows or hides headers.

property HeaderVisible: Boolean;

Headers are stored in TSRichViewEdit⁽⁴⁰⁾.Subdocuments⁽⁶⁶⁾ and edited in TSRichViewEdit.RVHeader⁽⁶³⁾.

If HeaderVisible=False, headers are not displayed, not loaded and not saved to files.

Default value:

True

See also:

- HeaderY⁽¹³⁷⁾
- FooterVisible⁽¹³⁶⁾
- TitlePage⁽¹⁴¹⁾
- FacingPages⁽¹³⁶⁾

Specifies a distance between the top of the page and the top of the header.

property HeaderY: TRVLength;

This value is measured in UnitsProgram⁽⁶⁸⁾ (mm by default).

A header is shown only if HeaderVisible⁽¹³⁷⁾=True.

Headers are stored in TSRichViewEdit⁽⁴⁰⁾.Subdocuments and edited in TSRichViewEdit.RVHeader⁽⁶³⁾.

This value is used for all types of headers.

This property can be changed as an editing (undoable) operation, see TSRichViewEdit⁽⁴⁰⁾.SetFloatPropertyEd⁽⁹⁶⁾.

Default value:

10

See also

- FooterY⁽¹³⁶⁾

Specifies a left margin of pages.

property LeftMargin: TRVLength;

This value is measured in UnitsProgram⁽⁶⁸⁾ (mm by default).

This property can be changed as an editing (undoable) operation, see TSRichViewEdit⁽⁴⁰⁾.SetFloatPropertyEd⁽⁹⁶⁾.

Default value:

35

Instructs to exchange left and right margins on even pages.

property MirrorMargins: Boolean;

If MirrorMargins = *True*, then:

- for odd pages, LeftMargin⁽¹³⁷⁾ defines the left margin, RightMargin⁽¹⁴¹⁾ defines the right margin; for even pages, LeftMargin⁽¹³⁷⁾ defines the right margin, RightMargin⁽¹⁴¹⁾ defines the left margin;
- if page numbering is visible⁽¹⁴⁰⁾, left and right alignment⁽¹³⁹⁾ of page numbers are exchanged on even pages.

This property can be changed as an editing (undoable) operation, see T⁽⁴⁰⁾SRichViewEdit⁽⁹⁶⁾.SetIntPropertyEd⁽⁹⁶⁾.

Default value:

False

See also:

- FacingPages⁽¹³⁶⁾

Specifies a page orientation

property Orientation: TPrinterOrientation;

Portrait or landscape.

Assigning to this property sets the proper values to PageWidth⁽¹⁴¹⁾ and PageHeight⁽¹³⁹⁾ properties, if PageFormat⁽¹³⁸⁾ <> *srvfmCustom*.

This property can be changed as an editing (undoable) operation, see T⁽⁴⁰⁾SRichViewEdit⁽⁹⁶⁾.SetIntPropertyEd⁽⁹⁶⁾.

Default value:

poPortrait

Specifies a page format (size).

property PageFormat: T⁽²⁷⁷⁾SRVPageFormat⁽²⁷⁷⁾;

Assigning to this property changes values of PageWidth⁽¹⁴¹⁾ and PageHeight⁽¹³⁹⁾ properties as well (if the new value <> *srvfmCustom*). This change depends on Orientation⁽¹³⁸⁾ property.

When this property is changed, OnPageFormatChanged⁽¹¹⁴⁾ occurs.

If this value is equal to *srvfmCustom* (user-defined paper format), page size is defined in PageWidth⁽¹⁴¹⁾ and PageHeight⁽¹³⁹⁾ properties.

This property can be changed as an editing (undoable) operation, see T⁽⁴⁰⁾SRichViewEdit⁽⁹⁶⁾.SetIntPropertyEd⁽⁹⁶⁾.

Default value

srvfmA4

See also:

- ConvertPageFormatToPageSize⁽¹⁴²⁾
- ConvertPageSizeToPageFormat⁽¹⁴²⁾

Specifies a page height.

property PageHeight: TRVLength;

This value is measured in UnitsProgram⁽⁶⁸⁾ (mm by default).

This property can be changed as an editing (undoable) operation, see TSVRichViewEdit⁽⁴⁰⁾.SetFloatPropertyEd⁽⁹⁶⁾.

Default value:

297 (corresponds to A4 in portrait Orientation⁽¹³⁸⁾)

See also:

- PageWidth⁽¹⁴¹⁾
- PageFormat⁽¹³⁸⁾
- ViewProperty⁽⁶⁸⁾.PageHeight⁽¹⁶¹⁾

Specifies the index of the first page where page numbers are drawn, from 1.

property PageNoFirst: Integer;

Default value:

1 (all pages have page numbers, if PageNoVisible⁽¹⁴⁰⁾=True).

Specifies the font for drawing page numbers.

type

TSRVPageNoFont = **class** (TFont)

property PageNoFont: TSRVPageNoFont;

Default value:

'Arial', 8

See also

- PageNoVisible⁽¹⁴⁰⁾

Specifies the number of the first page.

property PageNoFromNumber: Integer;

This is the starting value for page numbering.

It is used:

- for drawing page numbers, if PageNoVisible⁽¹⁴⁰⁾=True
- for "page number" fields (TRVPageNumberItemInfo items)

Default value:

1

Specifies a horizontal alignment of page numbers.

type

TSRVHAlign = (srvhaLeft, srvhaCenter, srvhaRight);

property PageNoHAlign: TSRVHAlign;

If `MirrorMargins` ⁽¹³⁸⁾ = `True`, this property defines the alignment for odd pages; for even pages, left and right alignments are exchanged.

Default value:

srvvaCenter

See also

- `PageNoVisible` ⁽¹⁴⁰⁾
- `PageNoVAlign` ⁽¹⁴⁰⁾

Specifies a vertical position of page numbers.

type

```
TSRVVAlign = (srvvaUp, srvvaDown);
```

property `PageNoVAlign`: `TSRVVAlign`;

Default value:

srvvaDown (bottom of pages)

See also

- `PageNoVisible` ⁽¹⁴⁰⁾
- `PageNoHAlign` ⁽¹³⁹⁾

Shows/hides page numbers.

property `PageNoVisible`: `Boolean`;

This is a simple option to display page numbering. These page numbers are not saved to RTF, RVF or DocX.

The alternative (recommended) option is inserting "page number field (`TRVPageNumberItemInfo` item) in a header or a footer ⁽⁶⁶⁾.

Default value:

True

See also

- `PageNoVAlign` ⁽¹⁴⁰⁾
- `PageNoHAlign` ⁽¹³⁹⁾
- `PageNoFirst` ⁽¹³⁹⁾
- `PageNoFromNumber` ⁽¹³⁹⁾

Disables/enables page-view mode.

property `PageViewMode`: `Boolean`;

If this property is *False*, documents are displayed without separation on pages. `PageHeight` ⁽¹³⁹⁾ is calculated to fit `DocumentHeight` of `RichViewEdit` component.

This property may be used when implementing custom view modes.

This property must be *True* in a read mode ⁽¹⁵⁰⁾.

Default value:

True

Specifies a page width.

property PageWidth: TRVLength;

This value is measured in UnitsProgram⁽⁶⁸⁾ (mm by default).

This property can be changed as an editing (undoable) operation, see TSRichViewEdit⁽⁴⁰⁾.SetFloatPropertyEd⁽⁹⁶⁾.

Default value:

210 (corresponds to A4 in portrait Orientation⁽¹³⁸⁾)

See also:

- PageHeight⁽¹³⁹⁾
- PageFormat⁽¹³⁸⁾
- ViewProperty⁽⁶⁸⁾.PageWidth⁽¹⁶¹⁾

This pen is used for drawing a rectangle around the printable area on the page.

type

TSRVPAPen = **class** (TPen)

property PrintableAreaPen : TSRVPAPen;

Printer cannot print beyond the area shown on this rectangle, see WinAPI function GetDeviceCaps, PHYSICAL*** constants.

This property is similar to the property of TRVPrintPreview of the same name.

Default value:

Color=**clRed**, Style=*psClear* (rectangle is invisible)

Specifies a right margin of pages.

property RightMargin: TRVLength;

This value is measured in UnitsProgram⁽⁶⁸⁾ (mm by default).

This property can be changed as an editing (undoable) operation, see TSRichViewEdit⁽⁴⁰⁾.SetFloatPropertyEd⁽⁹⁶⁾.

Default value:

15

Specifies whether special headers and footers are used for the first page.

property TitlePage: Boolean;

This property can be changed as an editing (undoable) operation, see TSRichViewEdit⁽⁴⁰⁾.SetIntPropertyEd⁽⁹⁶⁾.

A header for the first page is stored in SRichViewEdit.Subdocuments⁽⁶⁶⁾ [*srvhftFirstPageHeader*].

A footer for the first page is stored in SRichViewEdit.Subdocuments⁽⁶⁶⁾ [*srvhftFirstPageFooter*].

TitlePage turns on/off displaying headers and footers for the first page, but does not affect saving: headers and footers are loaded and saved regardless of this property.

Headers are shown only if HeaderVisible⁽¹³⁷⁾=*True*, footers are shown only if FooterVisible⁽¹³⁶⁾=*True*.

Default value:*False***See also:**

- FacingPages ⁽¹³⁶⁾

Specifies a top margin of pages.

property TopMargin: TRVLength;

This value is measured in UnitsProgram ⁽⁶⁸⁾ (mm by default).

This property can be changed as an editing (undoable) operation, see TSVRichViewEdit ⁽⁴⁰⁾.
 .SetFloatPropertyEd ⁽⁹⁶⁾.

Default value:*25*

Specifies whether to use custom Z-order when drawing pages.

property UsePageOrders: Boolean;

Assigning *True* to this property calls CalculateAllPagePositions ⁽⁷²⁾, if
 TSVRichViewEdit.ViewProperty ⁽⁶⁸⁾.FreePosPages ⁽¹⁵⁷⁾ = *True*.

Default value:*False*

1.3.1.4.4.2 Methods

In TSVPageProperty

ConvertPageFormatToPageFormat ⁽¹⁴²⁾

ConvertPageFormatToPageFormat ⁽¹⁴²⁾

Converts **PageFormat** to **PaperWidth** and **PaperHeight**, in portrait orientation.

procedure ConvertPageFormatToPageFormat (PageFormat: TSVPageFormat ⁽²⁷⁷⁾;
 UnitsPaper: TRVUnits; **var** PaperWidth, PaperHeight: TRVLength);

PaperWidth and **PaperHeight** are measured in **UnitsPaper** ("pixels" assume 96 dpi).

If **PageFormat**=*svfmCustom*, PageWidth ⁽¹⁴¹⁾ and PageHeight ⁽¹³⁹⁾ are returned (always in landscape orientation)

Returns a page format by the specified values of **PaperWidth** and **PaperHeight**.

function ConvertPageFormatToPageFormat (UnitsPaper: TRVUnits;
 PaperWidth, PaperHeight: TRVLength): TSVPageFormat ⁽²⁷⁷⁾;

PaperWidth and **PaperHeight** must be specified in portrait orientation.

PaperWidth and **PaperHeight** are measured in **UnitsPaper** ("pixels" assume 96 dpi).

If a standard format of this size cannot be found, *svfmCustom* is returned.

1.3.1.4.5 TSRVPagePositionProperty

TSRVPagePositionProperty is the class for TRichViewEdit⁽⁴⁰⁾.PagePosProperty⁽⁵⁸⁾ property

Unit SclRView.

Syntax

```
TSRVPagePositionProperty = class (TPersistent)
```

Hierarchy

TObject

TPersistent

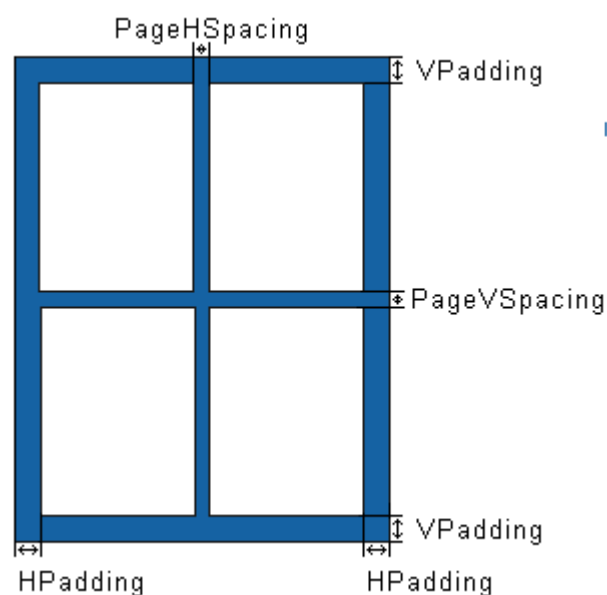
Properties

This class contains properties controlling arrangement of pages in TRichViewEdit⁽⁴⁰⁾, if ViewProperty⁽⁶⁸⁾.FreePosPage⁽¹⁵⁷⁾ = False.

HPadding and VPadding⁽¹⁴⁵⁾ define distances between sides of a scrollable area and outermost pages.

PageHSpacing and PageVSpacing⁽¹⁴⁶⁾ define distances between pages.

You can remove some of these distances using HiddenDistances⁽¹⁴⁵⁾ property.



You can align pages (columns of pages) horizontally using AlignPageH⁽¹⁴⁴⁾ property.

You can align pages (rows of pages) vertically using AlignPageV⁽¹⁴⁴⁾ property.

By default, each column have as many pages as a page width and ZoomPercent⁽¹⁶⁷⁾ allow. You can limit it using MaxPageColCount⁽¹⁴⁶⁾ property.

1.3.1.4.5.1 Properties

In TSRVPagePositionProperty

- AlignPageH⁽¹⁴⁴⁾

- AlignPageV⁽¹⁴⁴⁾
- HiddenDistances⁽¹⁴⁵⁾
- HPadding⁽¹⁴⁵⁾
- MaxPageColCount⁽¹⁴⁶⁾
- PageHSpacing⁽¹⁴⁶⁾
- PageVSpacing⁽¹⁴⁶⁾
- VPadding⁽¹⁴⁵⁾

Specifies the horizontal alignment of pages.

type

```
TSRVAlignPageH =  
    (srvaphLeft, srvaphRight, srvaphCenter);
```

property AlignPageH : TSRVAlignPageH;

A minimum distance from the left and the right sides of the scrollable area to a page is specified in HPadding⁽¹⁴⁵⁾. If a page width is smaller than the rest of space, pages are aligned according to the value of this property (if there are multiple columns of pages, columns as a whole are aligned according to this property).

If TSRichViewEdit⁽⁴⁰⁾.PageProperty⁽⁵⁸⁾.AutoWidth⁽¹³⁵⁾=*True*, the page occupies all available window width (excluding the double HPadding⁽¹⁴⁵⁾), so this alignment makes sense only if TSRichViewEdit⁽⁴⁰⁾.PageProperty⁽⁵⁸⁾.AutoWidth⁽¹³⁵⁾=*False*.

if TSRichViewEdit⁽⁴⁰⁾.BiDiMode⁽⁴⁸⁾ = *rvbdRightToLeft*, left and right values of this property are exchanged.

Call TSRichViewEdit⁽⁴⁰⁾.Format⁽⁸²⁾ after changing value of this property.

Default value:

srvaphCenter

Specifies the vertical alignment of pages.

type

```
TSRVAlignPageV = (srvapvTop, srvapvCenter);
```

property AlignPageV : TSRVAlignPageV;

A minimum distance from the top and the bottom sides of the scrollable area to a page is specified in VPadding⁽¹⁴⁵⁾. If a page height is smaller than the rest of space, pages are aligned according to the value of this property (if there are multiple rows of pages, rows as a whole are aligned according to this property).

Call TSRichViewEdit⁽⁴⁰⁾.Format⁽⁸²⁾ after changing value of this property.

Allows showing/hiding distances between pages.

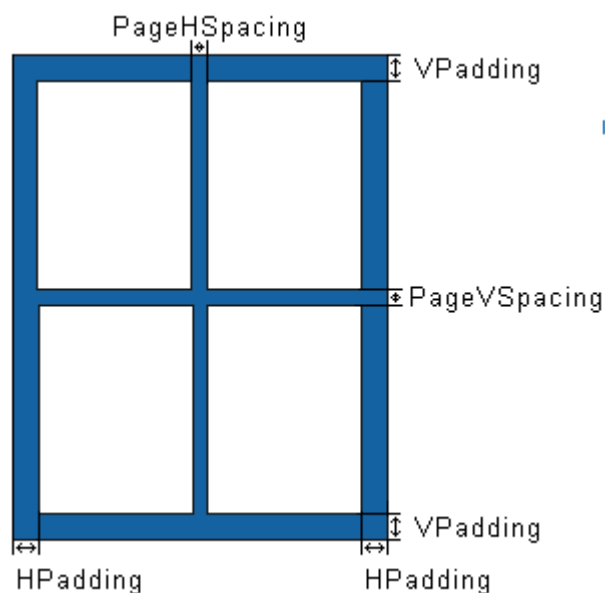
type

```
TSRVPagePositionDistance = (  
    srvppdHPadding, srvppdVPadding,  
    srvppdPageHSpacing, srvppdPageVSpacing);  
TSRVPagePositionDistances = set of TSRVPagePositionDistance;
```

property HiddenDistances: TSRVPagePositionDistances;

Include values in this property to hide the corresponding distance in `TSRichViewEdit`⁽⁴⁰⁾.

Value	Distance
<i>srvppdHPadding</i>	HPadding ⁽¹⁴⁵⁾
<i>srvppdVPadding</i>	VPadding ⁽¹⁴⁵⁾
<i>srvppdPageHSpacing</i>	PageHSpacing ⁽¹⁴⁶⁾
<i>srvppdPageVSpacing</i>	PageVSpacing ⁽¹⁴⁶⁾



Call `TSRichViewEdit`⁽⁴⁰⁾.`Format`⁽⁸²⁾ after changing value of this property.

Default value:

[]

Minimal distance between outermost pages and sides of the editor's scrollable area, horizontal and vertical.

property HPadding: TRVPixel96Length;

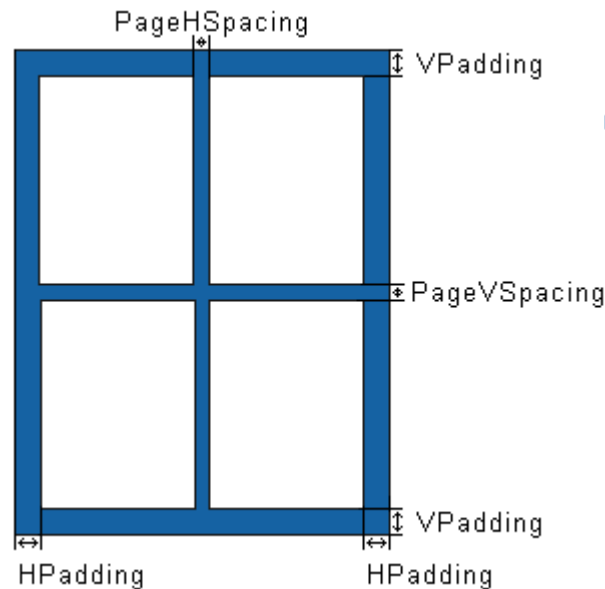
property VPadding: TRVPixel96Length;

HPadding is a minimal possible distance from the left/right side of the scrollable area to a page.

VPadding is a minimal possible distance from the top/bottom side of the scrollable area to a page.

These values are measured in pixels in 96 DPI (i.e. 1 pixel = 1/96 of an inch). When displaying on the screen, the actual distance is recalculated according to the screen DPI. Zooming does not affect the result.

To remove these distances, it's not necessary to assign 0 to these properties. Instead, you can include *srvppdHPadding* and/or *srvppdVPadding* in HiddenDistances⁽¹⁴⁵⁾.



Call `TSRichViewEdit(40).Format(82)` after changing values of these properties.

Default value:

15

Specifies the maximum count of columns for page arrangement.

property `MaxPageColCount: Integer;`

If zooming⁽¹⁶⁷⁾ allows, the editor tries to display several pages in a row. This value defines the maximum count of pages in a row (0 means unlimited).

This property is ignored in a read mode⁽¹⁵⁰⁾.

Default value:

0

Distance between pages, horizontal and vertical.

property `PageHSpacing: TRVPixel96Length;`

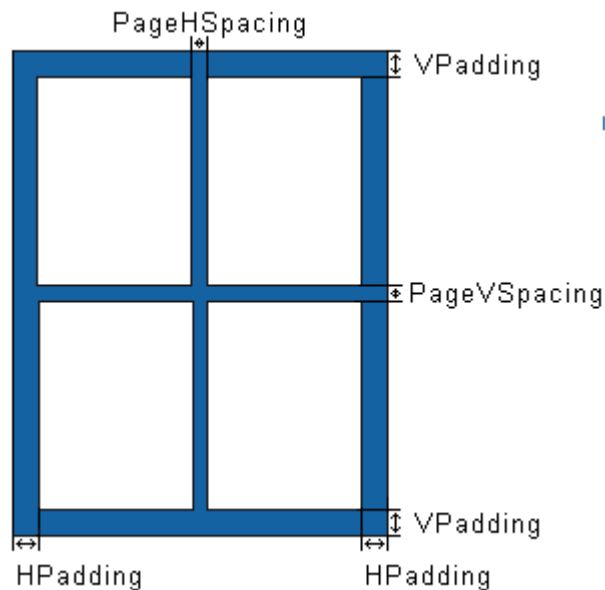
property `PageVSpacing: TRVPixel96Length;`

PageHSpacing is a distance between columns of pages.

PageVPadding is a distance between rows of pages.

These values are measured in pixels in 96 DPI (i.e. 1 pixel = 1/96 of an inch). When displaying on the screen, the actual distance is recalculated according to the screen DPI. Zooming does not affect the result.

To remove these distances, it's not necessary to assign 0 to these properties. Instead, you can include *srvppdPageHSpacing* and/or *srvppdPageVSpacing* in *HiddenDistances* ⁽¹⁴⁵⁾.



Call *TSRichViewEdit* ⁽⁴⁰⁾.*Format* ⁽⁸²⁾ after changing values of these properties.

Default value:

12

1.3.1.4.6 TSRVPropertyTBH

TSRVPropertyTBH is the class for *MenuHorizontal* ⁽⁵⁵⁾ property of *TSRichViewEdit* ⁽⁴⁰⁾

Unit *SclRView*.

Syntax

```
TSRVPropertyTBH = class (TSRVCustomPropertyTB (126))
```

Hierarchy

TObject

TPersistent

TSRVCustomPropertyTB ⁽¹²⁶⁾

Description

This class describes properties of "horizontal menu" (toolbar in the area of horizontal scrollbar).

For example, this toolbar can be used to define viewing modes ("Web mode", "Normal mode", "Print mode").

This toolbar can be placed to the left or to the right of the scrollbar, depending on *Align* ⁽¹⁴⁸⁾ property. It is shown only if *Visible* ⁽¹⁴⁸⁾ = *True*.

Since this is a toolbar, not a tool window, all properties related to a hint area are not available. A height of buttons is calculated automatically.

Buttons are defined in MenuHButtons⁽⁵⁵⁾ property of TSRichViewEdit⁽⁴⁰⁾.

When the user moves the mouse to a button, TSRichViewEdit.OnHMenuEnterButton⁽¹⁰⁹⁾ event occurs. When a button is clicked, TSRichViewEdit.OnHMenuClickButton⁽¹⁰⁹⁾ event occurs.

See also:

- TSRVPropertyTBV⁽¹⁴⁸⁾

1.3.1.4.6.1 Properties

In TSRVPropertyTBH

- Align⁽¹⁴⁸⁾
- Visible⁽¹⁴⁸⁾

Derived from TSRVCustomPropertyTB⁽¹²⁶⁾

- BorderWidth⁽¹²⁸⁾
- ButtonHeight⁽¹²⁸⁾
- Buttons⁽¹²⁸⁾
- ButtonWidth⁽¹²⁸⁾
- Color⁽¹²⁹⁾
- ColorDisable⁽¹²⁹⁾
- ColorDown⁽¹²⁹⁾
- ColorSelect⁽¹²⁹⁾
- ImageList⁽¹³⁰⁾
- PenFrame⁽¹³⁰⁾
- ScaleImagesForDPI⁽¹³⁰⁾
- Spacer⁽¹³⁰⁾

Defines the position of the toolbar relative to the horizontal scrollbar.

type

```
TSRVTBAlignH = (srvtbaLeft, srvtbaRight);
```

property Align : TSRVTBAlignH;

Default value:

srvtbaLeft (to the left of the scrollbar)

Specifies whether the toolbar is visible.

property Visible: Boolean;

Default value:

False (the toolbar is hidden)

1.3.1.4.7 TSRVPropertyTBV

TSRVPropertyTBV is the class for MenuVertical⁽⁵⁶⁾ property of TSRichViewEdit⁽⁴⁰⁾

Unit SclRView.

Syntax


```
TSRVPropertyTBV = class (TSRVCustomPropertyTB126)
```

Hierarchy

TObject

TPersistent

*TSRVCustomPropertyTB*¹²⁶

Description

This class describes properties of "vertical menu" (toolbar in the area of vertical scrollbar).

For example, this toolbar can be used for searching in the document.

This toolbar can be placed above or below the scrollbar, depending on *Align*¹⁴⁹ property. It is shown only if *Visible*¹⁵⁰ = *True*.

Since this is a toolbar, not a tool window, all properties related to a hint area are not available. A width of buttons is calculated automatically.

Buttons are defined in *MenuVButtons*⁵⁵ property of *TSRichViewEdit*⁴⁰.

When the user moves the mouse to a button, *TSRichViewEdit.OnVMenuEnterButton*¹²¹ event occurs. When a button is clicked, *TSRichViewEdit.OnVMenuClickButton*¹²¹ event occurs.

See also:

- *TSRVPropertyTBVH*¹⁴⁷

1.3.1.4.7.1 Properties

In *TSRVPropertyTBV*

- *Align*¹⁴⁹
- *Visible*¹⁵⁰

Derived from *TSRVCustomPropertyTB*¹²⁶

- *BorderWidth*¹²⁸
- *ButtonHeight*¹²⁸
- *Buttons*¹²⁸
- *ButtonWidth*¹²⁸
- *Color*¹²⁹
- *ColorDisable*¹²⁹
- *ColorDown*¹²⁹
- *ColorSelect*¹²⁹
- *ImageList*¹³⁰
- *PenFrame*¹³⁰
- *ScaleImagesForDPI*¹³⁰
- *Spacer*¹³⁰

Defines the position of the toolbar relative to the vertical scrollbar.

type

```
TSRVTBAlignV = (srvtbaTop, srvtbaBottom);
```

property Align : TSRVTBAlignV;

Default value:

srvtbaTop (above the the scrollbar)

Specifies whether the toolbar is visible.

property Visible: Boolean;

Default value:

False (the toolbar is hidden)

1.3.1.4.8 TSRVReadModeProperty

TSRVReadModeProperty is the class for ReadModeProperty⁽⁵⁹⁾ property of TSRichViewEdit⁽⁴⁰⁾

Unit SclRView.

Syntax

TSRVReadModeProperty = **class** (TPersistent)

Hierarchy

TObject

TPersistent

Description

This class defines properties for displaying pages in a way convenient for reading.

It is applied when you assign Active⁽¹⁵¹⁾ = *True*. Before activating a read mode, assign the parent editor's PageProperty⁽¹⁴⁰⁾.PageViewMode⁽¹⁴⁰⁾ = *True*.

The results depend on the parent editor's PageProperty⁽⁵⁸⁾.AutoWidth⁽¹³⁵⁾.

Side to side mode

If AutoWidth⁽¹³⁵⁾ = *False*, pages are displayed at their normal size, i.e. pages look identically to pages displayed without a read mode.

However, there are the following differences:

- all pages are arranged in a horizontal row (the parent editor's PagePosProperty⁽⁵⁸⁾.MaxPageColCount⁽¹⁴⁶⁾ is ignored)
- the horizontal scroll bar scrolls by whole pages; the vertical scroll bar is disabled; so all visible pages are shown completely
- the parent editor's ViewProperty⁽⁶⁸⁾.ZoomPercent⁽¹⁶⁷⁾ becomes read-only; it is auto-calculated to show at least MinPageColCount⁽¹⁵¹⁾ pages completely.
- no more than MaxPageColCount⁽¹⁵¹⁾ pages are displayed at the same time.

Normal read mode

If AutoWidth⁽¹³⁵⁾ = *True*, each page occupies all available space in the editor (excluding the parent editor's PagePosProperty⁽⁵⁸⁾.HPadding and VPadding⁽¹⁴⁵⁾).

Additionally:

- all pages are arranged in a horizontal row; only one page is visible (all column count properties are ignored)

- the horizontal scroll bar scrolls by whole pages; the vertical scroll bar is disabled; so a visible page is shown completely
- headers and footers⁽⁶⁶⁾ are not displayed

Visual properties

In a read mode, special page transition effects are available. They are defined in PageFlipEffect⁽¹⁵²⁾ property.

You can hide a page background picture⁽¹⁵¹⁾ and define a special page background color⁽¹⁵²⁾.

1.3.1.4.8.1 Properties

In TSRVReadModeProperty

- Active⁽¹⁵¹⁾
- HidePageBackgroundImage⁽¹⁵¹⁾
- MaxPageColCount⁽¹⁵¹⁾
- MinPageColCount⁽¹⁵¹⁾
- PageColor⁽¹⁵²⁾
- PageFlipEffect⁽¹⁵²⁾

Activates or deactivates a read mode.

property Active: Boolean;

Before assigning True to this property, assign the parent editor's PageProperty⁽¹⁴⁰⁾.PageViewMode⁽¹⁴⁰⁾ = True.

Default value:

False

Allows hiding a background images of pages in a read mode.

property HidePageBackgroundImage: Boolean;

If **HidePageBackgroundImage** = True, the editor's RichViewEdit⁽⁵⁹⁾.BackgroundBitmap is not shown when a read mode is active⁽¹⁵¹⁾.

This property is not applied when printing pages from a read mode.

See also

- PageColor⁽¹⁵²⁾

Default value:

c/None

Defines a number of visible pages in a read mode.

property MinPageColCount: Integer;

property MaxPageColCount: Integer;

These properties are applied when a read mode is active⁽⁵⁹⁾.

MinPageColCount defines the minimal count of visible pages.

If **MaxPageColCount** > 0, it defines the maximal allowed count of visible pages.

For example, if you want to implement a two-page "book" view, assign 2 to the both properties.

Default value:

- MinPageColCount: 1
- MaxPageColCount: 0

Allows overriding background color of pages in a read mode.

property PageColor: TColor;

If **PageColor** <> *c/None*, it is used as a background page color when a read mode is active ⁽¹⁵¹⁾.

This property has higher priority than the editor's BackgroundProperty ⁽⁴⁸⁾.
 .GlobalPageBackgroundColor ⁽¹²⁵⁾. However, it is not applied when printing pages from a read mode.

See also

- HidePageBackgroundImage ⁽¹⁵¹⁾

Default value:

c/None

Defines a page transition animation in a read mode.

type

```
TSRVPageFlipEffect = (srvpfeNone, srvpfeScroll,  
    srvpfeStack, srvpfePopUp, srvpfeFade);
```

property PageFlipEffect: TSRVPageFlipEffect;

Value	Meaning
<i>srvpfeNone</i>	No animation. Fast page transition.
<i>srvpfeScroll</i>	Pages slide from/to outside
<i>srvpfeStack</i>	Pages slide as if they are stacked in outermost visible columns
<i>srvpfePopUp</i>	Like <i>srvpfeStack</i> , but page pop-up effect is amplified.
<i>srvpfeFade</i>	Pages gradually appear on new places (requires Delphi 2009+).

Page transition animation is used:

- when pages are scrolled using a mouse wheel
- when pages are scrolled using a horizontal scrollbar by 1 position

Default value:

srvpfeStack

1.3.1.4.9 TSRVViewProperty

TSRVViewProperty is the class for `TSRichViewEdit`⁽⁴⁰⁾.`ViewProperty`⁽⁶⁸⁾ property.

Unit `SclRView`.

Syntax

```
TSRVViewProperty = class (TPersistent)
```

Hierarchy

TObject

TPersistent

Description

Free page positioning mode

Free page positioning mode is activated in `FreePosPage`⁽¹⁵⁷⁾ property. In this mode, you can define position, zooming and Z-order for specific pages.

Zooming and page arrangement

The current zooming is defined in `ZoomMode`⁽¹⁶⁵⁾ and `ZoomPercent`⁽¹⁶⁷⁾ properties.

In a preview mode (`ViewMode`⁽¹⁶⁴⁾ = `srvvmPreviewMode`), clicking switches these properties between `ZoomModeIN`⁽¹⁶⁶⁾, `ZoomPercentIN`⁽¹⁶⁸⁾ and `ZoomModeOUT`⁽¹⁶⁶⁾, `ZoomPercentOUT`⁽¹⁶⁸⁾.

When switching back to an editing mode (`ViewMode`⁽¹⁶⁴⁾ = `srvvmEditMode`), they are set to `ZoomModeEdit`⁽¹⁶⁶⁾, `ZoomPercentEdit`⁽¹⁶⁸⁾.

The user can change zooming:

- using a mouse wheel, if `MouseWheelZooming`⁽¹⁶¹⁾ = `True`.
- using a special panel, if `ZoomPanelVisible`⁽¹⁶⁷⁾ = `True`.

`ZoomPercent`⁽¹⁶⁷⁾ can be assigned in the range `ZoomMin..ZoomMax`⁽¹⁶⁵⁾.

If zooming allows, pages may be arranged in columns, according to `MaxPageColCount`⁽¹⁴⁶⁾ property.

Table icons

A table "icon" (a rectangle at the top left corner) is displayed when the user moves the mouse pointer inside a table (`TRVTableItemInfo`), if `UseTableIcons`⁽¹⁶³⁾ = `True`.

Even when the user moves the mouse pointer outside a table, this icon is still visible for `TableIconDelay`⁽¹⁶²⁾.

When the user right-clicks this icon, `TableIconPopupMenu`⁽¹⁶²⁾ is displayed. Besides, `TSRichViewEdit.OnTableIconClick`⁽¹²¹⁾ occurs on clicking.

The table for the displayed icon is returned by `TSRichViewEdit.GetTableIconItem`⁽⁸⁹⁾. A document where this table is located is returned by `TSRichViewEdit.GetTableIconRVData`⁽⁸⁹⁾.

Rectangle around the document area

You can display a rectangle between the document area and margins, if you set `MarginsRectVisible`⁽¹⁶¹⁾ = `True`. This rectangle is drawn using `MarginsPen`⁽¹⁶¹⁾.

Hints on vertical scrolling

A hint appear on scrolling, if `ShowScrollHint`⁽¹⁶²⁾ = `True`. This hint contains two lines describing the page at the scrolling position:

- `HintPrefixText`⁽¹⁵⁹⁾ followed by the page number
- text from the beginning of page

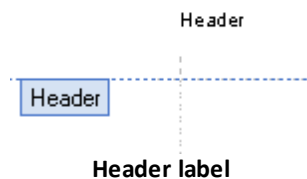
This text is drawn with `HintFont`⁽¹⁵⁹⁾.

Header and footer

When the header is being edited, a line is shown above the main document. This line uses `HeaderPen`⁽¹⁵⁸⁾. Additionally, a label is displayed.

VCL Themes

`UseVCLThemes`⁽¹⁶⁴⁾ allows using colors of the current VCL theme.



This label has text `Texts`⁽¹⁶²⁾.`NormalPageHeader`, font `HeaderTitleFont`⁽¹⁵⁸⁾, color `HeaderTitleColor`⁽¹⁵⁸⁾, border color `HeaderPen`⁽¹⁵⁸⁾.`Color`. If `Texts`⁽¹⁶²⁾.`NormalPageHeader` is empty, this label is not displayed.

A footers and the main document have similar properties. For the main document, a line and a label are invisible by default.

When a footer or a header is being edited, the main document can be shaded, if `MainDocumentShade`⁽¹⁵⁹⁾ = `True`. When the main document is edited, a header and a footer can be shaded if `HeaderFooterShade`⁽¹⁵⁷⁾ = `True`.

Page size

The page size, *as it is shown on the screen*, is returned in `PageWidth`⁽¹⁶¹⁾ and `PageHeight`⁽¹⁶¹⁾ properties.

They are measured in `SRichViewEdit`'s `UnitsProgram`⁽⁶⁸⁾ and do not depend on scaling. The same values in pixels are returned in `PageWidth100Pix`⁽⁵⁸⁾ and `PageHeight100Pix`⁽⁵⁷⁾ properties of `SRichViewEdit`.

However, these properties depend on viewing modes. If `PageProperty`⁽⁵⁸⁾.`AutoWidth`⁽¹³⁵⁾ = `False` and `PageProperty`⁽⁵⁸⁾.`PageViewMode`⁽¹⁴⁰⁾ = `True`, these properties return the same values as `PageProperty`⁽⁵⁸⁾'s `PageWidth`⁽¹⁴¹⁾ and `PageHeight`⁽¹³⁹⁾. When `AutoWidth` or `PageViewMode` are changed, the screen view does not correspond to the paper size, and these properties become different.

1.3.1.4.9.1 Properties

In `TSRVViewProperty`

- `AutoVScrollBarPosition`⁽¹⁵⁵⁾
- `DarkMode`⁽¹⁵⁶⁾

- DarkModeUI ⁽¹⁵⁶⁾
- FooterPen ⁽¹⁵⁶⁾
- FooterTitleColor ⁽¹⁵⁶⁾
- FooterTitleFont ⁽¹⁵⁷⁾
- FreePosPage ⁽¹⁵⁷⁾
- HeaderFooterShade ⁽¹⁵⁷⁾
- HeaderPen ⁽¹⁵⁸⁾
- HeaderTitleColor ⁽¹⁵⁸⁾
- HeaderTitleFont ⁽¹⁵⁸⁾
- HintFont ⁽¹⁵⁹⁾
- HintPrefixText ⁽¹⁵⁹⁾
- IconStyle ⁽¹⁵⁹⁾
- MainDocumentShade ⁽¹⁵⁹⁾
- MainPen ⁽¹⁶⁰⁾
- MainTitleColor ⁽¹⁶⁰⁾
- MainTitleFont ⁽¹⁶⁰⁾
- MarginsPen ⁽¹⁶¹⁾
- MarginsRectVisible ⁽¹⁶¹⁾
- MouseWheelZoom ⁽¹⁶¹⁾
- ▶ PageHeight ⁽¹⁶¹⁾
- ▶ PageWidth ⁽¹⁶¹⁾
- ShowScrollHint ⁽¹⁶²⁾
- TableIconDelay ⁽¹⁶²⁾
- TableIconPopupMenu ⁽¹⁶²⁾
- Texts ⁽¹⁶²⁾
- UseTableIcons ⁽¹⁶³⁾
- UseVCLThemes ⁽¹⁶⁴⁾
- ViewMode ⁽¹⁶⁴⁾
- WheelStep ⁽¹⁶⁴⁾
- WorkArea ⁽¹⁶⁴⁾
- ZoomFont ⁽¹⁶⁵⁾
- ZoomMenu ⁽¹⁶⁵⁾
- ZoomMin, ZoomMax ⁽¹⁶⁵⁾
- ZoomMode ⁽¹⁶⁵⁾
- ZoomModeEdit ⁽¹⁶⁶⁾
- ZoomModeIN ⁽¹⁶⁶⁾
- ZoomModeOUT ⁽¹⁶⁶⁾
- ZoomPanelFont ⁽¹⁶⁶⁾
- ZoomPanelVisible ⁽¹⁶⁷⁾
- ZoomPercent ⁽¹⁶⁷⁾
- ZoomPercentEdit ⁽¹⁶⁸⁾
- ZoomPercentIN ⁽¹⁶⁸⁾
- ZoomPercentOUT ⁽¹⁶⁸⁾

Specifies where a vertical scrollbar is located if a right-to-left document is loaded

property AutoVScrollBarPosition: Boolean;

By default, if BiDiMode ⁽⁴⁸⁾ = *rvbdRightToLeft*, a vertical scrollbar is moved to the left side.

If you assign *False* to this property, this scrollbar will always be at the right side.

Default value:

True

Inverts luminance of all colors in document.

property `DarkMode: Boolean;`

If *True*, luminance of all colors in the document is inverted. Dark colors become light colors, black becomes white, white becomes black.

Warning: be careful with this property: it affects both drawing on the screen and printing. Black backgrounds causes overuse of printer ink.

This property does not affect to drawing controls around pages, see `DarkModeUI` ⁽¹⁵⁶⁾.

Default value:

False

Inverts luminance of all colors in user interface

property `DarkModeUI: Boolean;`

If *True*, luminance of all colors in controls around pages is inverted.

This property does not affect drawing on pages, see `DarkMode` ⁽¹⁵⁶⁾.

Scroll bars use “dark mode” drawing only if you change `ScrollBarControlStyle` ⁽⁶⁹⁾ to *svcsSimple*.

Default value:

False

A pen for drawing a line between the main document and the footer, when a footer (`TSRichViewEdit.RVFooter` ⁽⁶¹⁾) is being edited.

type

```
TSRVHeaderFooterPen = class (TPen);
```

property `FooterPen: TSRVHeaderFooterPen;`

The color of this pen is also used for drawing border around header label, see `Texts` ⁽¹⁶²⁾.

In Delphi XE2 or newer, if a system color is specified for `FooterPen.Color`, and custom styles are applied, then a corresponding style color is used instead.

Default value:

1, *psDot*, \$00CA784B

See also:

- `HeaderPen` ⁽¹⁵⁸⁾
- `MainPen` ⁽¹⁶⁰⁾

A background color of label displayed when a footer (`TSRichViewEdit.RVFooter` ⁽⁶¹⁾) is being edited.

property `FooterTitleColor: TColor;`

A border around this label is drawn using `FooterPen` ⁽¹⁵⁶⁾.Color.

A text of this label (Texts⁽¹⁶²⁾.NormalPageFooter, FirstPageFooter, EvenPageFooter, OddPageFooter) is drawn using FooterTitleFont⁽¹⁵⁷⁾.

In Delphi XE2 or newer, if a system color is specified for this property, and custom styles are applied, then a corresponding style color is used instead.

Default value:

\$00F3E2D5

See also:

- HeaderTitleColor⁽¹⁵⁸⁾
- MainTitleColor⁽¹⁶⁰⁾

A font of label title displayed when a footer (TSRichViewEdit.RVFooter⁽⁶¹⁾) is being edited.

type

```
TSRVHeaderFooterFont = class (TFont);
```

property FooterTitleFont : TSRVHeaderFooterFont;

This font is used to draw a footer title (Texts⁽¹⁶²⁾.NormalPageFooter, FirstPageFooter, EvenPageFooter, OddPageFooter) on a background filled with FooterTitleColor⁽¹⁵⁶⁾.

In Delphi XE2 or newer, if a system color is specified for FooterTitleFont.Color, and custom styles are applied, then a corresponding style color is used instead.

Default value:

'Tahoma', 8, [], \$009A480B

See also:

- HeaderTitleFont⁽¹⁵⁸⁾
- MainTitleFont⁽¹⁶⁰⁾

Enables/disables free page positioning.

property FreePosPage: Boolean;

If *False*, positions, scaling and Z-order of all pages are calculated automatically.

If *True*, they are requested in OnGetPagePos⁽¹⁰⁸⁾ event.

Default value:

False

Determines whether a header and a footer are shaded when the main document is being edited.

property HeaderFooterShade: Boolean;

Default value:

True

See also

- MainDocumentShade⁽¹⁵⁹⁾

A pen for drawing a line between the main document and the header, when a header (TSRichViewEdit.RVHeader⁶³) is being edited.

type

```
TSRVHeaderFooterPen = class (TPen);
```

property HeaderPen: TSRVHeaderFooterPen;

The color of this pen is also used for drawing border around header label, see Texts¹⁶².

In Delphi XE2 or newer, if a system color is specified for HeaderPen.Color, and custom styles are applied, then a corresponding style color is used instead.

Default value:

1, *psDot*, \$00CA784B

See also:

- FooterPen¹⁵⁶
- MainPen¹⁶⁰

A background color of label displayed when a header (TSRichViewEdit.RVHeader⁶³) is being edited.

property HeaderTitleColor: TColor;

A border around this label is drawn using HeaderPen¹⁵⁸.Color.

A text of this label (Texts¹⁶².NormalPageHeader, FirstPageHeader, EvenPageHeader, OddPageHeader) is drawn using HeaderTitleFont¹⁵⁸.

In Delphi XE2 or newer, if a system color is specified for this property, and custom styles are applied, then a corresponding style color is used instead.

Default value:

\$00F3E2D5

See also:

- FooterTitleColor¹⁵⁶
- MainTitleColor¹⁶⁰

A font of label title displayed when a header (TSRichViewEdit.RVHeader⁶³) is being edited.

type

```
TSRVHeaderFooterFont = class (TFont);
```

property HeaderTitleFont : TSRVHeaderFooterFont;

This font is used to draw a header title (Texts¹⁶².NormalPageHeader, FirstPageHeader, EvenPageHeader, OddPageHeader) on a background filled with HeaderTitleColor¹⁵⁸.

In Delphi XE2 or newer, if a system color is specified for HeaderTitleFont.Color, and custom styles are applied, then a corresponding style color is used instead.

Default value:

'Tahoma', 8, [], \$009A480B

See also:

- FooterTitleFont¹⁵⁷

- MainTitleFont⁽¹⁶⁰⁾

A font for hints appearing on vertical scrolling.

type

```
TSRVHintFont = class (TFont);
```

property HintFont: TSRVHintFont;

This hint appears if ShowScrollHint⁽¹⁶²⁾=*True*.

In Delphi XE2 or newer, if a system color is specified for HintFont.Color, and custom styles are applied, then a corresponding style color is used instead.

Default value:

'Arial', 10, clInfoText

Specifies the text for the first line of a popup hint appearing on vertical scrolling.

property HintPrefixText: TRVUnicodeString;

HintPrefixText is displayed before the page number.

This hint appears if ShowScrollHint⁽¹⁶²⁾=*True*.

Default value:

'Page '



Allows to choose an set of icons used in TSRichViewEdit⁽⁴⁰⁾ control.

type

```
TSRVIconStyle = (srvisNormal, srvisFlat);
```

property IconStyle: TSRVIconStyle;

TSRichViewEdit allows choosing different icons for drawing in a zooming panel (if ZoomPanelVisible⁽¹⁶⁷⁾=*True*)

Value	Meaning
<i>srvisNormal</i>	Standard icons 
<i>srvisFlat</i>	Simplified flat white icons 

Default value:

srvisNormal

Determines whether the main document is shaded when a header or a footer is being edited.

property MainDocumentShade: Boolean;

Default value:

False

See also

- HeaderFooterShade⁽¹⁵⁷⁾

A pen for drawing lines between the main document and the header and the footer, when the main document (TSRichViewEdit.RichViewEdit⁽⁵⁹⁾) is being edited.

type

```
TSRVMainPen = class (TPen);
```

property MainPen: TSRVMainPen;

The color of this pen is also used for drawing border around the main document label, see Texts⁽¹⁶²⁾.

In Delphi XE2 or newer, if a system color is specified for MainPen.Color, and custom styles are applied, then a corresponding style color is used instead.

Default value:

1, psClear, \$00CA784B (invisible line)

See also:

- HeaderPen⁽¹⁵⁸⁾
- FooterPen⁽¹⁵⁶⁾

A background color of a label displayed when the main document (TSRichViewEdit.RichViewEdit⁽⁵⁹⁾) is being edited.

property MainTitleColor: TColor;

A border around this label is drawn using MainPen⁽¹⁶⁰⁾.Color.

A text of this label (Texts⁽¹⁶²⁾.MainDocument) is drawn using MainTitleFont⁽¹⁶⁰⁾.

In Delphi XE2 or newer, if a system color is specified for this property, and custom styles are applied, then a corresponding style color is used instead.

Default value:

\$00F3E2D5

See also:

- HeaderTitleColor⁽¹⁵⁸⁾
- FooterTitleColor⁽¹⁵⁶⁾

A font of label title displayed when the main document (TSRichViewEdit.RichViewEdit⁽⁵⁹⁾) is being edited.

type

```
TSRVHeaderFooterFont = class (TFont);
```

property MainTitleFont : TSRVHeaderFooterFont;

This font is used to draw Texts⁽¹⁶²⁾.MainDocument on a background filled with MainTitleColor⁽¹⁶⁰⁾.

In Delphi XE2 or newer, if a system color is specified for MainTitleFont.Color, and custom styles are applied, then a corresponding style color is used instead.

Default value:

'Tahoma', 8, [], \$009A480B

See also:

- HeaderTitleFont⁽¹⁵⁸⁾
- FooterTitleFont⁽¹⁵⁷⁾

Specifies a pen properties for drawing rectangle around the document area.

property BoundTextPen: TSRVMarginsPen;

This rectangle shows margins. It is drawn only if MarginsRectVisible⁽¹⁶¹⁾=*True*.

In Delphi XE2 or newer, if a system color is specified for MarginsPen.Color, and custom styles are applied, then a corresponding style color is used instead.

Default value:

psDot, clSilver

Specifies whether the component draws a rectangle around the document area.

property MarginsRectVisible: Boolean;

This rectangle is drawn using MarginsPen⁽¹⁶¹⁾.

Specifies whether pages can be zoomed in/out using the mouse wheel (with pressed **Ctrl** key)

property MouseWheelZoom: Boolean;

If *True*, **Ctrl**+mouse wheel changes ZoomMode⁽¹⁶⁵⁾ to *rvzmCustom* and increases/decreases ZoomPercent⁽¹⁶⁷⁾.

Default value:

True

See also:

- MouseWheelZoomMin, MouseWheelZoomMax, MouseWheelZoomSpeed global variables⁽²⁸⁰⁾

Returns the page height as it is shown on the screen.

property PageHeight: TRVLength;

This property is measured in SRichViewEdit's UnitsProgram⁽⁶⁸⁾. The same value in pixels is returned in PageHeight100Pix⁽⁵⁷⁾ property of SRichViewEdit.

This property does not depend on scaling, but depends on viewing modes (PageProperty⁽⁵⁸⁾.AutoWidth⁽¹³⁵⁾ and PageProperty⁽⁵⁸⁾.PageViewMode⁽¹⁴⁰⁾). This property is equal to PageProperty⁽⁵⁸⁾.PageHeight⁽¹³⁹⁾ only if AutoWidth=*False* and PageViewMode=*True*.

Returns the page width as it is shown on the screen.

property PageWidth: TRVLength;

This property is measured in SRichViewEdit's UnitsProgram⁽⁶⁸⁾. The same value in pixels is returned in PageWidth100Pix⁽⁵⁸⁾ property of SRichViewEdit.

This property does not depend on scaling, but depends on viewing modes (PageProperty⁽⁵⁸⁾.AutoWidth⁽¹³⁵⁾ and PageProperty⁽⁵⁸⁾.PageViewMode⁽¹⁴⁰⁾). This property is equal to PageProperty⁽⁵⁸⁾.PageWidth⁽¹⁴¹⁾ only if AutoWidth=*False* and PageViewMode=*True*.

Specifies whether a popup hint appears on scrolling.

property ShowScrollHint: Boolean;

This hint displays on the first line: HintPrefixText⁽¹⁵⁹⁾, then the page number.

On the second line, a text from the beginning of this page is displayed.

HintFont⁽¹⁵⁹⁾ is used.

Default value:

True

Specifies the period of time for which a table "icon" is still visible after the user moved the mouse pointer outside a table.

property TableIconDelay: LongWord;

A table "icon" (a rectangle at the top left corner) is displayed when the user moves the mouse pointer inside a table (TRVTableItemInfo), if UseTableIcons⁽¹⁶³⁾=*True*.

Even when the user moves the mouse pointer outside a table, this icon is still visible for this period of time.

This value is measured in milliseconds.

Default value:

3000

See also:

- TSRichViewEdit.OnTableIconClick⁽¹²¹⁾
- TableIconPopupMenu⁽¹⁶²⁾

A reference to the context menu of a table "icon".

property TableIconPopupMenu: TPopupMenu;

A table "icon" (a rectangle at the top left corner) is displayed when the user moves the mouse pointer inside a table (TRVTableItemInfo), if UseTableIcons⁽¹⁶³⁾=*True*.

See also:

- TSRichViewEdit.OnTableIconClick⁽¹²¹⁾
- TableIconDelay⁽¹⁶²⁾

A list of text strings displayed in TSRichViewEdit⁽⁴⁰⁾.

type

TSRVTexts = **class** (TPersistent)

property Texts: TSRVTexts;

TSRVTexts has the properties listed in the table below.

Property	Meaning	Default Value	Related Properties of TSVViewProperty
NormalPageHeader	Title for the normal header, if PageProperty ⁽⁵⁸⁾	'Header'	HeaderTitleFont ⁽¹⁵⁸⁾ , HeaderTitleColor ⁽¹⁵⁸⁾ ,

	.FacingPages ⁽¹³⁶⁾ =False		HeaderPen ⁽¹⁵⁸⁾ .Color
NormalPageFooter	Title for the normal footer, if FacingPages=False	'Footer'	FooterTitleFont ⁽¹⁵⁷⁾ , FooterTitleColor ⁽¹⁵⁶⁾ , FooterPen ⁽¹⁵⁶⁾ .Color
FirstPageHeader	Title for the first page header	'First Page Header'	HeaderTitleFont ⁽¹⁵⁸⁾ , HeaderTitleColor ⁽¹⁵⁸⁾ , HeaderPen ⁽¹⁵⁸⁾ .Color
FirstPageFooter	Title for the first page footer	'First Page Footer'	FooterTitleFont ⁽¹⁵⁷⁾ , FooterTitleColor ⁽¹⁵⁶⁾ , FooterPen ⁽¹⁵⁶⁾ .Color
EvenPageHeader	Title for the even pages header	'Even Page Header'	HeaderTitleFont ⁽¹⁵⁸⁾ , HeaderTitleColor ⁽¹⁵⁸⁾ , HeaderPen ⁽¹⁵⁸⁾ .Color
EvenPageFooter	Title for the even pages footer	'Even Page Footer'	FooterTitleFont ⁽¹⁵⁷⁾ , FooterTitleColor ⁽¹⁵⁶⁾ , FooterPen ⁽¹⁵⁶⁾ .Color
OddPageHeader	Title for the normal header, if FacingPages=True	'Odd Page Header'	HeaderTitleFont ⁽¹⁵⁸⁾ , HeaderTitleColor ⁽¹⁵⁸⁾ , HeaderPen ⁽¹⁵⁸⁾ .Color
OddPageFooter	Title for the normal footer, if FacingPages=True	'Odd Page Footer'	FooterTitleFont ⁽¹⁵⁷⁾ , FooterTitleColor ⁽¹⁵⁶⁾ , FooterPen ⁽¹⁵⁶⁾ .Color
MainDocument	Title for the main document	''	MainTitleFont ⁽¹⁶⁰⁾ , MainTitleColor ⁽¹⁶⁰⁾ , MainPen ⁽¹⁶⁰⁾ .Color

These strings are used to draw labels when editing a header, a footer, or a main document. If a string is empty, the corresponding label is not displayed.

Specifies whether table "icons" should be displayed for tables.

property UseTableIcons: Boolean;

A table "icon" (a rectangle at the top left corner) is displayed when the user moves the mouse pointer inside a table (TRVTableItemInfo), if UseTableIcons=True.

Even when the user moves the mouse pointer outside a table, this icon is still visible for TableIconDelay⁽¹⁶²⁾.

When the user right-clicks this icon, TableIconPopupMenu⁽¹⁶²⁾ is displayed. Besides, TSRichViewEdit.OnTableIconClick⁽¹²¹⁾ occurs on clicking.

The table for the displayed icon is returned by TSRichViewEdit.GetTableIconItem⁽⁸⁹⁾. A document where this table is located is returned by TSRichViewEdit.GetTableIconRVData⁽⁸⁹⁾.

Default value:

True

Allows using VCL theme colors (for Delphi XE2 and newer).

property UseVCLThemes: Boolean;

If *True*, VCL theme colors are used instead of system colors in documents. This property is ignored for Delphi versions prior to XE2.

This property does not affect user interface objects in the editor. They always use VCL themes, if available.

Default value:

False

Switches between editing and preview modes.

type

```
TSRVViewMode = (srvvmEditMode, srvvmPreviewMode);
```

property ViewMode: TSRVViewMode;

When this property is changed, `TSRichViewEdit.OnChangeViewModeAfter`⁽¹⁰⁴⁾ and `TSRichViewEdit.OnChangeViewModeBefore`⁽¹⁰⁵⁾ events occur.

Default value:

srvvmEditMode

Defines how much the document will be scrolled when the user turns a mouse wheel.

property WheelStep: Integer;

This property provides access to the property of the same name of `TSRichViewEdit`⁽⁴⁰⁾.
`.RichViewEdit`⁽⁵⁹⁾.

Default value:

3

Restricts the region available for drawing page content.

type

```
TSRVWorkArea = (srvwaTextRect, srvwaPageRect);
```

property WorkArea : TSRVWorkArea;

Value	Meaning
<i>srvwaTextRect</i>	The output area is limited by the document area (nothing can be drawn on margins)
<i>srvwaPageRect</i>	The output area is limited by the page area (nothing can be drawn beyond the page)

Default value:

srvwaPageRect

A font for drawing text (zooming percent) in a zooming mode.

type

```
TSRVZoomPanelFont = (TFont)
```

property ZoomPanelFont: TSRVZoomPanelFont;

A zooming mode is initiated if `TSRichViewEdit.PageProperty(58).AutoWidth(135) = True`, by a mouse wheel⁽¹⁶¹⁾ or a track bar on a zooming panel⁽¹⁶⁷⁾.

In this mode, the editor content is shaded with white color, and a current zooming percent is displayed using this font.

The specified font size corresponds to 100% zoom. The text size is increased/decreased according to the zooming percent.

Default value:

'Tahoma', 20, clWindowText

A reference to the popup menu shown when the user clicks on the text area of a zooming panel.

property ZoomMenu: TPopupMenu;

See also

- ZoomPanelVisible⁽¹⁶⁷⁾
- ZoomPanelFont⁽¹⁶⁶⁾
- TableIconMenu⁽¹⁶²⁾

The properties define the range of zooming.

property ZoomMin: Integer;

property ZoomMax: Integer;

ZoomPercent⁽¹⁶⁷⁾ can be set to a value in this range.

Default value:

20..500

The current zooming mode.

property ZoomMode: TRVZoomMode;

Full page / Page width / Custom. See also ZoomPercent⁽¹⁶⁷⁾.

This value is switched between ZoomModeIN⁽¹⁶⁶⁾, ZoomModeOUT⁽¹⁶⁶⁾, ZoomModeEdit⁽¹⁶⁶⁾.

When this property is changed, `TSRichViewEdit.OnZoomChanged(122)` occurs.

Default value:

rvzmPageWidth

This value is assigned to ZoomMode⁽¹⁶⁵⁾ when switching ViewMode⁽¹⁶⁴⁾ to *srvvmEditMode*.

property ZoomModeEdit: TRVZoomMode;

If this property is equal to *rvzmCustom*, the corresponding zooming is defined in ZoomPercentEdit⁽¹⁶⁸⁾.

When this property is changed, TSVRichViewEdit.OnZoomViewChanged⁽¹²²⁾ event occurs.

Default value

rvzmPageWidth

This value is assigned to ZoomMode⁽¹⁶⁵⁾ when switching ViewMode⁽¹⁶⁴⁾ to *srvvmPreviewMode*, or in preview mode when the user clicks the page and the cursor looks like "-" magnifier.

property ZoomModeIN: TRVZoomMode;

In preview mode the mouse pointer looks like a magnifier. On clicking, ZoomMode⁽¹⁶⁵⁾ is switched between ZoomModeIN/ZoomModeOUT⁽¹⁶⁶⁾.

If this property is equal to *rvzmCustom*, the corresponding zooming is defined in ZoomPercentIN⁽¹⁶⁸⁾.

When this property is changed, TSVRichViewEdit.OnZoomViewChanged⁽¹²²⁾ event occurs.

Default value:

rvzmFullPage.

This value is assigned to ZoomMode⁽¹⁶⁵⁾ in preview mode when the user clicks the page and the cursor looks like "+" magnifier.

property ZoomModeOUT: TRVZoomMode;

In preview mode the mouse pointer looks like a magnifier. On clicking, ZoomMode⁽¹⁶⁵⁾ is switched between ZoomModeIN⁽¹⁶⁶⁾/ZoomModeOUT.

If this property is equal to *rvzmCustom*, the corresponding zooming is defined in ZoomPercentOUT⁽¹⁶⁸⁾.

When this property is changed, TSVRichViewEdit.OnZoomViewChanged⁽¹²²⁾ event occurs.

Default value:

rvzmCustom

A font for drawing text (zooming percent) in a zooming panel.

type

```
TSRVZoomPanelFont = (TFont)
```

property ZoomPanelFont: TSVRVZoomPanelFont;

A zooming panel is shown if ZoomPanelVisible⁽¹⁶⁷⁾=*True*.

Default value:

'Tahoma', 8, clWindowText

Disables a gradient background fill in a zooming panel.

property ZoomPanelUsesGradient: Boolean;

The zooming panel background has a vertical gradient, if the editor's BackgroundProperty⁽⁴⁸⁾ defines it. You can disable a gradient fill by assigning *False* to this property.

A flat fill uses BackgroundProperty⁽⁴⁸⁾.ColorBegin⁽¹²⁴⁾.

Default value:

True

See also

- ZoomPanelVisible⁽¹⁶⁷⁾

Shows/hides a zooming panel.

property ZoomPanelVisible: Boolean;

This panel is located in the bottom right corner, to the right of the horizontal scrollbar. It contains a track bar to change ZoomPercent⁽¹⁶⁷⁾ in the range ZoomMin..ZoomMax⁽¹⁶⁵⁾.

A background of this panel is drawn using the parent TSRichViewEdit⁽⁴⁰⁾'s BackgroundProperty⁽⁴⁸⁾. If a gradient fill is enabled, it is always vertical in the zooming panel. Even if a gradient fill is enabled in BackgroundProperty⁽⁴⁸⁾, you can disable it by assigning *False* to ZoomPanelUsesGradient⁽¹⁶⁷⁾ property. A flat fill uses BackgroundProperty⁽⁴⁸⁾.ColorBegin⁽¹²⁴⁾.

If ZoomMenu⁽¹⁶⁵⁾ is assigned, it is displayed when the user clicks a text area of the panel. This text is drawn using ZoomPanelFont⁽¹⁶⁶⁾.

The appearance of the track bar depends on the value of IconStyle⁽¹⁵⁹⁾ property.

Default value:

False

See also

- ZoomMenu⁽¹⁶⁵⁾
- ZoomPanelFont⁽¹⁶⁶⁾
- IconStyle⁽¹⁵⁹⁾

The page zooming percent.

property ZoomPercent: Single;

If ZoomMode⁽¹⁶⁵⁾ = *rvzmCustom*, this value is assigned by the user. Otherwise it is calculated automatically.

Assigning to this property resets ZoomMode⁽¹⁶⁵⁾ to *rvzmCustom*.

When this property is changed, TSRichViewEdit.OnZoomChanged⁽¹²²⁾ occurs.

This property can be changed in the range ZoomMin..ZoomMax⁽¹⁶⁵⁾. If the parent editor's PageProperty⁽⁵⁸⁾.AutoWidth⁽¹³⁵⁾ = *True*, there is an additional auto-calculated upper limit, to prevent displaying too large content on too small pages.

If the parent editor's PageProperty⁽⁵⁸⁾.AutoWidth⁽¹³⁵⁾ = *True*, zooming is applied to the document content, while sizes of pages depends on the editor size. Otherwise, zooming is applied to pages (both page size and content).

The result of zooming depend on the screen pixel depth (DPI). First, page sizes are calculated at 96 DPI. Next, the actual sizes are calculated as:

$$\text{SizePix} := \text{Size100Pix} * (\text{ZoomPercent} / 100) * (<\text{Screen DPI}> / 96)$$

If the application supports multiple monitors having different DPI (in Delphi 10.1+), <Screen DPI> is the DPI of the monitor containing the form that contains this editor. Otherwise, this is Screen.PixelsPerInch (i.e. DPI of the primary monitor). This value can be overridden by RichViewPixelsPerInch global variable.

Default value:

100

See also:

- MouseWheelZoom⁽¹⁶¹⁾
- ZoomPanelVisible⁽¹⁶⁷⁾

This value is assigned to ZoomPercent⁽¹⁶⁷⁾ when switching ViewMode⁽¹⁶⁴⁾ to *srvvmEditMode*, if ZoomModeEdit⁽¹⁶⁶⁾ = *rvzmCustom*.

property ZoomPercentEdit: Single;

When this property is changed, TSRichViewEdit.OnZoomViewChanged⁽¹²²⁾ event occurs.

Default value:

100

This value is assigned to ZoomPercent⁽¹⁶⁷⁾ when switching ViewMode⁽¹⁶⁴⁾ to *srvvmPreviewMode*, or in preview mode when the user clicks the page and the cursor looks like "-" magnifier, if ZoomModeIN⁽¹⁶⁶⁾ = *rvzmCustom*.

property ZoomPercentIN: Single;

When this property is changed, TSRichViewEdit.OnZoomViewChanged⁽¹²²⁾ event occurs.

Default value:

100

This value is assigned to ZoomPercent⁽¹⁶⁷⁾ when the user clicks the page and the cursor looks like "+" magnifier, if ZoomModeOUT⁽¹⁶⁶⁾ = *rvzmCustom*.

property ZoomPercentOUT: Single;

When this property is changed, TSRichViewEdit.OnZoomViewChanged⁽¹²²⁾ event occurs.

Default value:

100

1.3.2 TDBSRichViewEdit

TDBSRichViewEdit is a word processing component linked to a database field.

Unit DBSRVE.

Syntax

TDBSRichViewEdit = **class**(TSRichViewEdit⁽⁴⁰⁾);

Hierarchy

TObject
TPersistent
TComponent
TControl
TWinControl
TCustomControl
TSRichViewEdit⁽⁴⁰⁾

Description

Use a TDBSRichViewEdit object to enable users to edit a database field in a word processing control.

If the application doesn't require the data-aware capabilities of TDBSRichViewEdit, use a TSRichViewEdit⁽⁴⁰⁾ component instead, to conserve system resources.

Alternative way: use LiveBindings to link TSRichViewEdit.Document⁽⁵¹⁾ to a database field.

Properties

TDBSRichViewEdit overrides RichViewEdit⁽¹⁷³⁾ and ExternalRV⁽¹⁷³⁾ properties of TSRichViewEdit⁽⁴⁰⁾. They have the same meaning, but TDBSRichViewEdit type.

Other properties and events provide access to properties and events of RichViewEdit⁽¹⁷³⁾.

1.3.2.1 Properties

In TDBSRichViewEdit

- AutoDisplay⁽¹⁷²⁾
- DataField⁽¹⁷²⁾
- DataSource⁽¹⁷²⁾
- ExternalRV⁽¹⁷³⁾
- FieldFormat⁽¹⁷³⁾
- IgnoreEscape⁽¹⁷³⁾
- RichViewEdit⁽¹⁷³⁾

Derived from TSRichViewEdit⁽⁴⁰⁾

- AcceptDragDropFormats⁽⁴⁷⁾
- AcceptPasteFormats⁽⁴⁷⁾
- ▶ ActiveEditor⁽⁴⁷⁾
- AnimationMode⁽⁴⁷⁾
- BackgroundProperty⁽⁴⁸⁾
- BiDiMode⁽⁴⁸⁾
- ▶ BottomMarginPix⁽⁴⁸⁾
- CanScroll⁽⁴⁸⁾
- CanUpdate⁽⁴⁸⁾
- CanUpdatePosition⁽⁴⁸⁾
- CanUpdateMargin⁽⁴⁹⁾

- CaretBlinkTime ⁽⁴⁹⁾
- ▶ CaretPos ⁽⁴⁹⁾
- CaretVisible ⁽⁴⁹⁾
- CPEventKind ⁽⁴⁹⁾
- ▶ CurControl ⁽⁵⁰⁾
- ▶ CurrentNote ⁽⁵⁰⁾
- ▶ CurrentNoteParentEditor ⁽⁵⁰⁾
- CurrentPage ⁽⁵⁰⁾
- ExternalRVStyle ⁽⁵¹⁾
- ▶ FirstPageNo ⁽⁵¹⁾
- FixMarginsMode ⁽⁵²⁾
- ▶ HMaxScrollPos ⁽⁵²⁾
- HScrollBar ⁽⁵²⁾
- HScrollBarSchemeIndex ⁽⁵³⁾
- HScrollPos ⁽⁵³⁾
- HTMLReadProperties ⁽⁵³⁾
- HTMLSaveProperties ⁽⁵³⁾
- ▶ LastPageNo ⁽⁵³⁾
- ▶ LeftMarginPix ⁽⁵⁴⁾
- MarkdownProperties ⁽⁵⁴⁾
- MaxPrintedItemNo ⁽⁵⁴⁾
- MaxPrintedOffsInItem ⁽⁵⁴⁾
- MenuHButtons ⁽⁵⁵⁾
- MenuHorizontal ⁽⁵⁵⁾
- MenuVButtons ⁽⁵⁵⁾
- MenuVertical ⁽⁵⁶⁾
- MinPrintedItemNo ⁽⁵⁶⁾
- MinPrintedOffInItem ⁽⁵⁶⁾
- ▶ OffsetX ⁽⁵⁶⁾
- ▶ OffsetY ⁽⁵⁷⁾
- ▶ PageCount ⁽⁵⁷⁾
- ▶ PageHeightPix, PageHeight100Pix ⁽⁵⁷⁾
- ▶ PageNoMouseDown ⁽⁵⁷⁾
- ▶ PageNoMouseMove ⁽⁵⁷⁾
- PageProperty ⁽⁵⁸⁾
- ▶ PageWidthPix, PageWidth100Pix ⁽⁵⁸⁾
- RangeSearch ⁽⁵⁸⁾
- RangeSearchFirstY ⁽⁵⁹⁾
- RangeSearchLastY ⁽⁵⁹⁾
- ReadModeProperty ⁽⁵⁹⁾
- ReadOnly ⁽⁵⁹⁾
- RichViewEdit ⁽⁵⁹⁾
- ▶ RightMarginPix ⁽⁶⁰⁾
- RTFOptions ⁽⁶⁰⁾
- RTFReadProperties ⁽⁶⁰⁾
- RVBackgroundBitmap ⁽⁶¹⁾
- RVBackgroundStyle ⁽⁶¹⁾
- RVColor ⁽⁶¹⁾

- RVEditorOptions⁶¹
- RVFooter⁶¹
- RVFOptions⁶²
- RVFParaStylesReadMode⁶³
- RVFTextStylesReadMode⁶³
- RVHeader⁶³
- RVNote⁶⁴
- RVOptions⁶⁴
- ▶ ScrolledPage⁶⁵
- SkinManager⁶⁵
- SmartPopupProperties⁶⁵
- SmartPopupVisible⁶⁵
- StyleTemplateInsertMode⁶⁵
- TabNavigation⁶⁷
- TextEngine⁶⁷
- ▶ TopMarginPix⁶⁷
- UnitsProgram⁶⁸
- UseDrawHyperlinksEvent⁶⁸
- UseStyleTemplates⁶⁸
- Version⁶⁸
- ViewProperty⁶⁸
- ▶ VMaxScrollPos⁶⁸
- VScrollBar⁶⁹
- VScrollBarSchemeIndex⁶⁹
- VScrollPos⁶⁹

Derived from TCustomControl

- Align
- Anchors
- Constraints
- Ctl3D
- DragMode
- Enabled
- Height
- HelpContext
- HelpKeyword (D6+)
- HelpType (D6+)
- Hint
- Left
- Name
- ParentCtl3D
- ParentShowHint
- PopupMenu
- ShowHint
- TabOrder
- TabStop
- Tag
- Top

- Touch (D2010+)
- Visible
- Width

1.3.2.1.1 TDBSRichViewEdit.AutoDisplay

Determines whether to automatically display the contents of field in the TDBSRichViewEdit control.

property AutoDisplay: Boolean;

This property provides access to the RichViewEdit⁽¹⁷³⁾'s property of the same name.

If AutoDisplay is *True*, the TDBSRichViewEdit control automatically displays new data when the underlying BLOB field changes (such as when moving to a new record).

If AutoDisplay is *False*, the control shows only the field name whenever the underlying BLOB field changes. To display the data, the user can double-click on the control.

The effect of AutoDisplay is not purely cosmetic. When AutoDisplay is *False*, if the data changes, the content of TDBSRichViewEdit document changes to the name of the field (written in the 0th text and paragraph style). Thus, if AutoDisplay is *False*, applications should be cautious about using the methods for saving contents to ascertain the value of the underlying field.

Calling the RichViewEdit⁽¹⁷³⁾.LoadField method causes component to update to the current value of the BLOB field. This change will also be reflected in the appearance of the control on screen.

Change the value of AutoDisplay to *False* if the automatic loading of BLOB fields seems to take too long.

1.3.2.1.2 TDBSRichViewEdit.DataField

Specifies the field from which the TDBSRichViewEdit control displays data.

property DataField: string;

This property provides access to the RichViewEdit⁽¹⁷³⁾'s property of the same name.

Use DataField to bind the TDBSRichViewEdit control to the field in the dataset. To fully specify database field, both the dataset and the field within that dataset must be defined. The DataSource⁽¹⁷²⁾ property of the TDBSRichViewEdit control specifies the dataset which contains the DataField.

1.3.2.1.3 TDBSRichViewEdit.DataSource

Links the TDBSRichViewEdit control to the dataset that contains the field it represents.

property DataSource : TDataSource;

This property provides access to the RichViewEdit⁽¹⁷³⁾'s property of the same name.

Use DataSource to link the TDBSRichViewEdit control to the dataset in which the data can be found. To fully specify database field for the TDBSRichViewEdit control, both the dataset and the field within that dataset must be defined. Use the DataField⁽¹⁷²⁾ property to specify the particular field within the dataset.

1.3.2.1.4 TDBSRichViewEdit.ExternalRV

Reference to an external TDBRichViewEdit component used to render documents in this TSBRichViewEdit component.

property ExternalRV: TDBRichViewEdit;

Nil by default (invisible internal RichViewEdit⁽¹⁷³⁾ is used).

1.3.2.1.5 TDBSRichViewEdit.FieldFormat

Format for saving data in the database field

property FieldFormat: TRVDBFieldFormat;

This property allows changing the field saving format to RTF or plain text.

This property provides access to the RichViewEdit⁽¹⁷³⁾'s property of the same name.

Default value:

rvdbRVF

1.3.2.1.6 TDBSRichViewEdit.IgnoreEscape

Disallows processing of **Escape** key

property IgnoreEscape: Boolean;

This property provides access to the RichViewEdit⁽¹⁷³⁾'s property of the same name.

By default, **Escape** cancels editing in TDBSRichViewEdit.

If you set this property to *True*, **Escape** will do nothing.

Default value:

False

1.3.2.1.7 TDBSRichViewEdit.RichViewEdit

Returns TDBRichViewEdit component used to render documents.

property RichViewEdit: TDBRichViewEdit;

TDBSRichViewEdit can use two types of TDBRichViewEdit:

- internal: TDBRichViewEdit component is created inside TDBSRichViewEdit. Users cannot see it, it is responsible only for document rendering.
- external: refers to external TDBRichViewEdit component (ExternalRV⁽¹⁷³⁾). Take a note that width and height of this DBRichViewEdit are set by TDBSRichViewEdit.

1.3.3 TSRVPageScroll

TSRVPageScroll is a component for displaying thumbnails of pages for a document from TSBRichViewEdit⁽⁴⁰⁾ component.

Unit SRVPageScroll.

Syntax

```
TSRVPageScroll = class (TRVScroller)
```

Hierarchy

TObject
TPersistent
TComponent
TControl
TWinControl
TCustomControl
TRVScroller

Description

The source document is specified in `SRichViewEdit` ⁽¹⁸²⁾.

Thumbnails are arranged vertically or horizontally, depending on `ScrollType` ⁽¹⁸⁰⁾. A number of columns (or rows) is specified in `Columns` ⁽¹⁷⁷⁾.

A size of thumbnails is defined in `PageWidth` ⁽¹⁸⁰⁾, `PageHeight` ⁽¹⁷⁹⁾, and `PageProportions` ⁽¹⁸⁰⁾ properties.

Distances between and around thumbnails are specified in `PageBreakHeight` ⁽¹⁷⁸⁾, `PageBreakWidth` ⁽¹⁷⁹⁾, `MarginVertical` ⁽¹⁷⁸⁾, `MarginHorizontal` ⁽¹⁷⁸⁾ properties.

In addition to thumbnails, the component can display rectangles containing page numbers. They are displayed if `PageNoVisible` ⁽¹⁸⁰⁾ = `True`. Their properties are `AlignText` ⁽¹⁷⁵⁾, `BackgroundTextColor` ⁽¹⁷⁶⁾, `PageNoFont` ⁽¹⁷⁹⁾.

A background is filled with `Color`. Thumbnails have borders drawn using `PageBorderColor` ⁽¹⁷⁸⁾ and `SelectColor` ⁽¹⁸¹⁾. A shadow can be displayed using the properties `ShadowColor` ⁽¹⁸¹⁾ and `ShadowWidth` ⁽¹⁸¹⁾.

Thumbnail quality may be increased by assigning `SmoothToombnails` ⁽¹⁸²⁾ = `True` (however, it slows down repainting).

1.3.3.1 Properties

In `TSRVPageScroll`

- `AlignText` ⁽¹⁷⁵⁾
- `AutoScrolled` ⁽¹⁷⁶⁾
- `AutoUpdate` ⁽¹⁷⁶⁾
- `BackgroundTextColor` ⁽¹⁷⁶⁾
- `CachedPagesCount` ⁽¹⁷⁷⁾
- `CacheScrollLimit` ⁽¹⁷⁷⁾
- `CacheScrollLimit` ⁽¹⁷⁷⁾
- `Columns` ⁽¹⁷⁷⁾
- `MarginHorizontal` ⁽¹⁷⁸⁾
- `MarginVertical` ⁽¹⁷⁸⁾
- `PageBorderColor` ⁽¹⁷⁸⁾
- `PageBreakHeight` ⁽¹⁷⁸⁾
- `PageBreakWidth` ⁽¹⁷⁹⁾
- `PageHeight` ⁽¹⁷⁹⁾
- `PageNoFont` ⁽¹⁷⁹⁾

- PageNoVisible¹⁸⁰
- PageProportions¹⁸⁰
- PageWidth¹⁸⁰
- ScrollType¹⁸⁰
- SelectColor¹⁸¹
- ShadowColor¹⁸¹
- ShadowWidth¹⁸¹
- SmoothScroll¹⁸¹
- SmoothThumbnails¹⁸²
- SRichViewEdit¹⁸²

Derived from TRVScroller

- BorderStyle
- ▶ HScrollMax
 - HScrollPos
- HScrollVisible
- ▶ InplaceEditor (not used in this component)
 - NoVScroll
- Tracking
- UseXPThemes
- ▶ VScrollMax
 - VScrollPos
- VScrollVisible
- WheelStep

Derived from TCustomControl

- Align
- Anchors
- Color
- Constraints
- Ctl3D
- Height
- HelpContext
- Hint
- ParentCtl3D
- PopupMenu
- ShowHint
- TabOrder
- TabStop
- Touch (D2010+)
- Visible
- Width

1.3.3.1.1 TSRVPageScroll.AlignText

Specifies the position of page number relative to the page.

type

```
TPSAlignText = (psatLeft, psatRight, psatTop, psatBottom);
```

property AlignText : TPSAlignText ;

Page numbers are displayed if PageNoVisible⁽¹⁸⁰⁾=True.

Default value:

psatBottom (below pages)

See also:

- BackgroundTextColor⁽¹⁷⁶⁾
- PageNoFont⁽¹⁷⁹⁾
- OnDrawPageNumber⁽¹⁸⁵⁾

1.3.3.1.2 TSRVPageScroll.AutoScrolled

Specifies whether the component should be scrolled when another page becomes visible in SRichViewEdit⁽¹⁸²⁾.

property AutoScrolled: Boolean;

If True, the component is scrolled to make SRichViewEdit.ScrolledPage⁽⁶⁵⁾ visible.

Default value:

True

1.3.3.1.3 TSRVPageScroll.AutoUpdate

Enables/disables automatic update of thumbnails when the content of SRichViewEdit⁽¹⁸²⁾ is changed.

property AutoUpdate: Boolean;

Assign False to this value if you know that the content of SRichViewEdit⁽¹⁸²⁾ cannot be changed, or to make editing in SRichViewEdit⁽¹⁸²⁾ faster.

If False, call UpdateCache⁽¹⁸⁴⁾ to update thumbnails.

Default value:

True

1.3.3.1.4 TSRVPageScroll.BackgroundColor

Specifies the background color below page numbers.

property BackgroundTextColor : TColor;

Page numbers are displayed if PageNoVisible⁽¹⁸⁰⁾=True.

Default value:

cBtnFace

See also:

- AlignText⁽¹⁷⁵⁾
- PageNoFont⁽¹⁷⁹⁾
- OnDrawPageNumber⁽¹⁸⁵⁾

1.3.3.1.5 TSRVPageScroll.CachedPagesCount

Specifies the maximal page cache size.

property `CachedPagesCount: Integer;`

The cache stores page images. If the cache is large, more memory is used, but drawing is faster.

Default value:

10

See also:

- `CacheScrollLimit`⁽¹⁷⁷⁾

1.3.3.1.6 TSRVPageScroll.CacheScrollLimit

Specifies how the page cache is updated when the component is scrolled.

property `CacheScrollLimit: Integer;`

When the cache is completely filled (see `CachedPageCount`⁽¹⁷⁷⁾), it is updated when the component is scrolled by at least `CacheScrollLimit` number of pages.

Default value:

1

1.3.3.1.7 TSRVPageScroll.ChangeDelay

Redrawing delay, in milliseconds.

property `ChangeDelay: Integer;`

This property specifies the delay between when `SRichViewEdit`⁽¹⁸²⁾ is updated and when this `TSRVPageScroll` component is repainted.

0 means immediate repainting, it may slow down response time.

Default value:

50

1.3.3.1.8 TSRVPageScroll.Columns

Specifies the number of columns (if `ScrollType`⁽¹⁸⁰⁾=`psstVertical`) or rows (if `ScrollType`⁽¹⁸⁰⁾=`psstHorizontal`),

property `Columns: Integer;`

For example, if `SRichViewEdit`⁽¹⁸²⁾.`PageCount`⁽⁵⁷⁾=5 and `Columns`=2, thumbnails are displayed in the order shown below.

If `ScrollType`⁽¹⁸⁰⁾=`psstVertical`:

1 2

3 4

5

If `ScrollType`⁽¹⁸⁰⁾=`psstHorizontal`:

1 3 5

2 4

A distance between columns is defined in `PageBreakWidth`⁽¹⁷⁹⁾, a distance between rows is defined in `PageBreakHeight`⁽¹⁷⁸⁾.

Default value:

1

1.3.3.1.9 TSRVPageScroll.MarginHorizontal

Specifies the left and the right margins.

property `MarginHorizontal` : Integer;

This is a distance from the left and from the right sides of the scrollable area to thumbnails.

Default value:

15

See also:

`MarginVertical`⁽¹⁷⁸⁾

`PageBreakWidth`⁽¹⁷⁹⁾

1.3.3.1.10 TSRVPageScroll.MarginVertical

Specifies the top and the bottom margins.

property `MarginVertical` : Integer;

This is a distance from the top and from the bottom sides of the scrollable area to thumbnails.

Default value:

15

See also:

`MarginHorizontal`⁽¹⁷⁸⁾

`PageBreakHeight`⁽¹⁷⁸⁾

1.3.3.1.11 TSRVPageScroll.PageBorderColor

Defines the page border color.

property `PageBorderColor` : TColor;

This is a border color for all thumbnails, except for `SRichViewEdit.CurrentPage`⁽⁵⁰⁾. For this page, `SelectColor`⁽¹⁸¹⁾ is used.

You can draw this border yourself, using `OnDrawPageBorder`⁽¹⁸⁵⁾ event.

1.3.3.1.12 TSRVPageScroll.PageBreakHeight

Specifies the vertical distance between page thumbnails.

property `PageBreakHeight` : Integer;

Thumbnails are displayed in several rows in the following cases:

- `ScrollType`⁽¹⁸⁰⁾ = `psstVertical`, or
- `ScrollType`⁽¹⁸⁰⁾ = `pssHorizontal` and `Columns`⁽¹⁷⁷⁾ > 1

Default value:

20

See also:

- PageBreakWidth⁽¹⁷⁹⁾
- MarginVertical⁽¹⁷⁸⁾

1.3.3.1.13 TSRVPageScroll.PageBreakWidth

Specifies the horizontal distance between page thumbnails.

property PageBreakWidth: Integer;

Thumbnails are displayed in several columns in the following cases:

- ScrollType⁽¹⁸⁰⁾ = *pssHorizontal*, or
- ScrollType⁽¹⁸⁰⁾ = *psstVertical* and Columns⁽¹⁷⁷⁾ > 1

Default value:

25

See also:

- PageBreakHeight⁽¹⁷⁸⁾
- MarginHorizontal⁽¹⁷⁸⁾

1.3.3.1.14 TSRVPageScroll.PageHeight

Specifies the [maximal] height of page thumbnails.

property PageHeight: Integer;

If PageProportions⁽¹⁸⁰⁾ = *False*, all thumbnails are created with the following size: PageWidth⁽¹⁸⁰⁾ x PageHeight.

If PageProportions⁽¹⁸⁰⁾ = *True*, thumbnails have the proportions of the original pages:

- ScrollType⁽¹⁸⁰⁾ = *pssHorizontal*, thumbnails' width is calculated automatically by PageHeight;
- ScrollType⁽¹⁸⁰⁾ = *psstVertical*, thumbnails' height is calculated automatically by PageWidth⁽¹⁸⁰⁾.

Default value

200

1.3.3.1.15 TSRVPageScroll.PageNoFont

Specifies the font for displaying page numbers.

type

```
TSRVPageNoFont = class (TFont)
```

property PageNoFont: TSRVPageNoFont;

Page numbers are displayed if PageNoVisible⁽¹⁸⁰⁾ = *True*.

Default value:

'Arial', 8

See also:

- BackgroundTextColor⁽¹⁷⁶⁾
- AlignText⁽¹⁷⁵⁾
- OnDrawPageNumber⁽¹⁸⁵⁾

1.3.3.1.16 TSRVPageScroll.PageNoVisible

Shows or hides page numbers

property PageNoVisible: Boolean;

Default value:

True

See also:

- BackgroundTextColor⁽¹⁷⁶⁾
- PageNoFont⁽¹⁷⁹⁾
- AlignText⁽¹⁷⁵⁾
- OnDrawPageNumber⁽¹⁸⁵⁾

1.3.3.1.17 TSRVPageScroll.PageProportions

Specifies whether thumbnails keep proportions of the original pages.

property PageProportions : Boolean;

Default value:

True

See also:

- PageWidth⁽¹⁸⁰⁾
- PageHeight⁽¹⁷⁹⁾

1.3.3.1.18 TSRVPageScroll.PageWidth

Specifies the [maximal] width of page thumbnails.

property PageWidth: Integer;

If PageProportions⁽¹⁸⁰⁾=*False*, all thumbnails are created with the following size: PageWidth x PageHeight⁽¹⁷⁹⁾.

If PageProportions⁽¹⁸⁰⁾=*True*, thumbnails have the proportions of the original pages:

- ScrollType⁽¹⁸⁰⁾=*pssHorizontal*, thumbnails' width is calculated automatically by PageHeight⁽¹⁷⁹⁾;
- ScrollType⁽¹⁸⁰⁾=*psstVertical*, thumbnails' height is calculated automatically by PageWidth.

Default value

150

1.3.3.1.19 TSRVPageScroll.ScrollType

Specifies how page thumbnails are arranged.

type

TPSScrollType = (psstHorizontal, psstVertical);

property ScrollType: TPSScrollType;

Mode	Meaning
<i>psstHorizontal</i>	Thumbnails are displayed in columns, each column has Columns ⁽¹⁷⁷⁾ rows

<i>psstVertical</i>	Thumbnails are displayed in rows, each row has Columns ⁽¹⁷⁷⁾ columns
---------------------	---

Default value:

psstVertical

1.3.3.1.20 TSRVPageScroll.SelectColor

Defines the border color for the first visible page in SRichViewEdit⁽¹⁸²⁾.

property SelectColor: TColor;

This is a border color for SRichViewEdit.CurrentPage⁽⁵⁰⁾. Other pages use PageBorderColor⁽¹⁷⁸⁾.

Default value:

clRed

See also:

- AutoScrolled⁽¹⁷⁶⁾

1.3.3.1.21 TSRVPageScroll.ShadowColor

Specifies the color of shadow cast by page thumbnails.

property ShadowColor: TColor;

A shadow is displayed if ShadowWidth⁽¹⁸¹⁾>0.

Default value:

\$0099A8AC

1.3.3.1.22 TSRVPageScroll.ShadowWidth

Specifies the width of shadow cast by page thumbnails.

property ShadowWidth: Integer;

A shadow color is ShadowColor⁽¹⁸¹⁾.

Default value:

3

1.3.3.1.23 TSRVPageScroll.SmoothScroll

Enables/disables smooth scrolling to the page in SRichViewEdit⁽¹⁸²⁾.

property SmoothScroll: Boolean;

Then the user clicks on a page thumbnail, the linked editor is scrolled show this page. If **SmoothScroll** = *True*, scrolling will be smooth.

Default value:

True

1.3.3.1.24 TSRVPageScroll.SmoothThumbnails

Enables/disables smoothed thumbnails of page images.

property SmoothThumbnails: Boolean;

Assign *True* to this properties to generate smoothed page images. They look better, but the component is updated slower.

Default value:

False

1.3.3.1.25 TSRVPageScroll.SRichViewEdit

Specifies the link to the editor containing the source document.

property SRichViewEdit: TSRichViewEdit⁽⁴⁰⁾;

Thumbnails can be displayed only if this property is assigned.

1.3.3.2 Methods

In TSRVPageScroll

BeginUpdate⁽¹⁸²⁾
 EndUpdate⁽¹⁸²⁾
 First⁽¹⁸³⁾
 GetPageNoToXY⁽¹⁸³⁾
 Last⁽¹⁸³⁾
 Next⁽¹⁸³⁾
 Prev⁽¹⁸⁴⁾
 ScrollToPage⁽¹⁸⁴⁾
 UpdateCache⁽¹⁸⁴⁾

1.3.3.2.1 TSRVPageScroll.BeginUpdate

Disables repainting until EndUpdate⁽¹⁸²⁾ is called.

procedure BeginUpdate;

Calls to BeginUpdate and EndUpdate⁽¹⁸²⁾ may be nested.

Call BeginUpdate before multiple changes in SRichViewEdit⁽¹⁸²⁾ to make them faster.

1.3.3.2.2 TSRVPageScroll.EndUpdate

Enables repainting blocked by BeginUpdate⁽¹⁸²⁾.

procedure EndUpdate;

1.3.3.2.3 TSRVPageScroll.First

Moves to the first page.

procedure First;

This method changes `SRichViewEdit(182).CurrentPage(50)`. This page becomes highlighted in this component, but not necessary the component is scrolled to this page (because it scrolls to show `SRichViewEdit(182).ScrolledPage(65)`, not `SRichViewEdit(182).CurrentPage(50)`). Call `ScrollToPage(184)` (`SRichViewEdit(182).CurrentPage(50)`) to scroll to the page containing the caret.

See also:

- Prev⁽¹⁸⁴⁾
- Next⁽¹⁸³⁾
- Last⁽¹⁸³⁾

1.3.3.2.4 TSRVPageScroll.GetPageAt

Returns the index of page displayed at the specified position.

function GetPageAt(X, Y: Integer; isNearest : Boolean) : Integer;

X, Y – client coordinates (relative to the top left corner of the client area of the component's window).

if **isNearest=False**, the method returns the index of page at the specified coordinates (from 1), or 0 if these coordinates are outside of any page.

if **isNearest=True**, the method returns the index of the nearest page to the specified coordinates (from 1).

1.3.3.2.5 TSRVPageScroll.Last

Moves to the last page.

procedure Last;

This method changes `SRichViewEdit(182).CurrentPage(50)`. This page becomes highlighted in this component, but not necessary the component is scrolled to this page (because it scrolls to show `SRichViewEdit(182).ScrolledPage(65)`, not `SRichViewEdit(182).CurrentPage(50)`). Call `ScrollToPage(184)` (`SRichViewEdit(182).CurrentPage(50)`) to scroll to the page containing the caret.

See also:

- First⁽¹⁸³⁾
- Prev⁽¹⁸⁴⁾
- Next⁽¹⁸³⁾

1.3.3.2.6 TSRVPageScroll.Next

Moves to the next page.

procedure Next;

This method changes `SRichViewEdit(182).CurrentPage(50)`. This page becomes highlighted in this component, but not necessary the component is scrolled to this page (because it scrolls to show `SRichViewEdit(182).ScrolledPage(65)`, not `SRichViewEdit(182).CurrentPage(50)`). Call `ScrollToPage(184)` (`SRichViewEdit(182).CurrentPage(50)`) to scroll to the page containing the caret.

See also:

- First¹⁸³
- Prev¹⁸⁴
- Last¹⁸³

1.3.3.2.7 TSRVPageScroll.Prev

Moves to the previous page.

procedure Prev;

This method changes SRichViewEdit¹⁸².CurrentPage⁵⁰. This page becomes highlighted in this component, but not necessary the component is scrolled to this page (because it scrolls to show SRichViewEdit¹⁸².ScrolledPage⁶⁵, not SRichViewEdit¹⁸².CurrentPage⁵⁰). Call ScrollToPage¹⁸⁴ (SRichViewEdit¹⁸².CurrentPage⁵⁰) to scroll to the page containing the caret.

See also:

- First¹⁸³
- Next¹⁸³
- Last¹⁸³

1.3.3.2.8 TSRVPageScroll.ScrollToPage

Scrolls to the specified page.

procedure ScrollToPage (PageNo : Integer);

This method scrolls this component, without scrolling SRichViewEdit¹⁸².

1.3.3.2.9 TSRVPageScroll.UpdateCache

Updates the cached images of page thumbnails.

procedure UpdateCache;

Normally, the cache is updated automatically. This method is useful if AutoUpdate¹⁷⁶=False.

1.3.3.3 Events**In TSRVPageScroll**

- OnDrawPageBorder¹⁸⁵
- OnDrawPageNumber¹⁸⁵
- OnDrawSelectedPage¹⁸⁶

Derived from TRVScroller

- OnHScrolled
- OnVScrolled

Derived from TCustomControl

- OnDbClick
- OnClick
- OnDbClick

- OnDragDrop
- OnDragOver
- OnEndDrag
- OnEnter
- OnExit
- OnGesture
- OnKeyPress
- OnMouseMove

1.3.3.3.1 TSRVPageScroll.OnDrawPageBorder

Allows a custom drawing of borders of page thumbnails.

type

```
TPSDrawPageBorderEvent = procedure (Sender: TObject; Canvas: TCanvas;  
    const R: TRect; PageNo: Integer) of object;
```

property OnDrawPageBorder : TPSDrawPageBorderEvent ;

Canvas – canvas where to draw.

R – location of the page thumbnail.

PagNo – index of the thumbnails' page, from 1.

By default, all thumbnails have a border of PageBorderColor⁽¹⁷⁸⁾ color. If this event is assigned, a default border is not drawn. But the selected page is still highlighted with SelectColor⁽¹⁸¹⁾ color, use OnDrawSelectedPage⁽¹⁸⁶⁾ to override it.

1.3.3.3.2 TSRVPageScroll.OnDrawPageNumber

Allows a custom drawing of page numbers.

type

```
TPSDrawPageNumberEvent = procedure (Sender: TObject; Canvas: TCanvas;  
    const R: TRect; PageNo : Integer; isSelect : Boolean) of object;
```

property OnDrawPageNumber: TPSDrawPageNumberEvent;

By default, page numbers are displayed if PageNoVisible⁽¹⁸⁰⁾=True.

They are displayed in the location specified in AlignText⁽¹⁷⁵⁾, drawing using BackgroundTextColor⁽¹⁷⁶⁾ and PageNoFont⁽¹⁷⁹⁾ properties.

If this event is assigned, the default drawing is not performed.

Canvas – canvas where to draw.

R – location of the page number, calculated according to AlignText⁽¹⁷⁵⁾ and PageNoFont⁽¹⁷⁹⁾.

PagNo – page number to draw.

isSelect specifies whether this page is highlighted.

1.3.3.3.3 TSRVPageScroll.OnDrawSelectedPage

Allows a custom drawing of highlighting of the selected thumbnail.

type

```
TPSDrawSelectedPageEvent = procedure (Sender: TObject; Canvas: TCanvas;  
    const R: TRect; PageNo: Integer) of object;
```

property OnDrawSelectedPage: TPSDrawSelectedPageEvent;

Canvas – canvas where to draw.

R – location of the page thumbnail.

PagNo – index of the thumbnails' page, from 1.

This drawing occurs after drawing the default page border, see OnDrawPageBorder⁽¹⁸⁵⁾.

By default, highlighted page have a border of SelectColor⁽¹⁸¹⁾ color. If this event is assigned, this default border is not drawn

1.3.4 TScIRVRuler

TScIRVRuler is a horizontal or vertical ruler.

Unit ScIRVRuler.

Syntax

```
TScIRVRuler = class (TRVRuler);
```

Hierarchy

TObject
TPersistent
TComponent
TControl
TWinControl
TCustomControl
TCustomRuler
TRuler
TRVRuler

Description

The class is based on the TRVRuler class and adapts the ruler for work with TSRichViewEdit⁽⁴⁰⁾.

It introduces new properties and methods for work as a vertical ruler and for assigning margins.

TRVRuler is included in RichViewActions.

The ruler works with the editor specified in SRichViewEdit⁽¹⁸⁷⁾ property.

1.3.4.1 Properties

In TScIRVRuler

■ SRichViewEdit⁽¹⁸⁷⁾

Derived from TRVRuler

...

1.3.4.1.1 TScIRVRuler.SRichViewEdit

A reference to the editor.

property SRichViewEdit: TScRichViewEdit⁴⁰;

The ruler works with this editor.

1.3.4.2 Methods

In TScIRVRuler

GetPageProperties¹⁸⁷

Scrolled¹⁸⁸

SetPageProperties¹⁸⁸

Derived from TRVRuler

...

1.3.4.2.1 TScIRVRuler.CheckMargins

Assigns the ruler properties from the linked SRichViewEdit¹⁸⁷.

procedure CheckMargins;

1.3.4.2.2 TScIRVRuler.GetPageProperties

The procedure returns margins and page size in the specified units. The returned values depend on the ruler type (horizontal/vertical).

procedure GetPageProperties(Units: TRulerUnits;
var AMargin1, AMargin2, APageWidth: Extended);

Input parameters

Units – type of units: centimeters, millimeters, pixels etc. The output parameters are measured in these units.

Output parameters

AMargin1 – left margin (LeftMargin) for horizontal ruler, or top margin (TopMargin) for vertical ruler.

AMargin2 – right margin (RightMargin) for horizontal ruler, bottom margin (BottomMargin) for vertical ruler.

APageWidth – page width (PageWidth) for horizontal ruler, page height (PageHeight) for vertical ruler.

1.3.4.2.3 TScIRVRuler.Scrolled

Updates the ruler's margins.

procedure Scrolled;

This method must be called if `SRichViewEdit(188).CanUpdate(48)=False`. Otherwise, margins are updated automatically.

1.3.4.2.4 TScIRVRuler.SetPageProperties

The procedure changes margins and page size. The changed properties depend on the ruler type (horizontal/vertical).

procedure SetPageProperties(Units: TRulerUnits;
AMargin1, AMargin2, APageWidth: Extended);

Units – type of units: centimeters, millimeters, pixels etc. Other parameters are measured in these units.

AMargin1 – left margin (LeftMargin) for horizontal ruler, or top margin (TopMargin) for vertical ruler.

AMargin2 – right margin (RightMargin) for horizontal ruler, bottom margin (BottomMargin) for vertical ruler.

APageWidth – page width (PageWidth) for horizontal ruler, page height (PageHeight) for vertical ruler.

1.3.4.3 Events

In TScIRVRuler

■ OnAfterMarginChanged⁽¹⁸⁸⁾

Derived from TRVRuler

...

1.3.4.3.1 TScIRVRuler.OnAfterMarginChanged

Occurs after the ruler's margins are changed.

property OnAfterMarginChanged: TNotifyEvent;

1.3.5 TSVToolBar

TSRVToolBar is a simple toolbar component.

Unit SRVToolBar.

Syntax

TSRVToolBar = **class**(TCustomControl)

Hierarchy

TObject

TPersistent

TComponent
TControl
TWinControl
TCustomControl

Description

TSRVToolbar is a container for buttons (TSRVToolButton⁽²⁷¹⁾).

The main features:

- all buttons in the toolbar have the same height and width;
- buttons are arranged in rows and columns;
- buttons can have different images for different states (normal, highlighted, disabled);
- button captions can be displayed in the bottom or the top part of the toolbar.

This component can be used directly, or inside a floating tool window (TSRVToolWindow⁽¹⁹⁶⁾), or inside a scrollbar area of TSRichViewEdit⁽⁴⁰⁾ component (see TSRVPropertyTBH⁽¹⁴⁷⁾ and TSRVPropertyTBV⁽¹⁴⁸⁾).

Properties

This class contains almost the same set of properties as TSRVCustomPropertyTB⁽¹²⁶⁾ class and TSRVToolWindow⁽¹⁹⁶⁾ component.

A toolbar consists of buttons and a hint area.

The hint area can be placed at the top or at the bottom of the toolbar, depending on AlignText⁽¹⁹⁰⁾ property. A height of this area is defined in HintHeight⁽¹⁹³⁾ property. This area can work as a "Cancel" button. When none of buttons is highlighted, it displays HintCancel⁽¹⁹³⁾. Otherwise, it displays Caption⁽²⁷²⁾ of tool button. A font for text displayed in this area is specified in HintFont⁽¹⁹³⁾.

Buttons are defined in Buttons⁽¹⁹¹⁾ collection. Buttons are arranged in rows and columns, the number of columns is specified in ButtonsCol⁽¹⁹¹⁾. A size of buttons is defined in ButtonWidth⁽¹⁹²⁾ and ButtonHeight⁽¹⁹¹⁾ properties. A spacing between buttons is specified in Spacer⁽¹⁹⁵⁾. A horizontal positions of buttons is defined by AlignButton⁽¹⁹⁰⁾ and Indent⁽¹⁹⁴⁾ properties.

The toolbar can have a border, its width is specified in BorderWidth⁽¹⁹¹⁾.

The toolbar color is Color⁽¹⁹²⁾. Colors of disabled, pressed and highlighted buttons are specified in ColorDisable⁽¹⁹²⁾, ColorDown⁽¹⁹²⁾, and ColorSelect⁽¹⁹³⁾ properties. A pen of frame around the toolbar is defined in PenFrame⁽¹⁹⁴⁾.

1.3.5.1 Properties

In TSRVToolBar

- AlignButton⁽¹⁹⁰⁾
- AlignText⁽¹⁹⁰⁾
- AutoSize⁽¹⁹¹⁾
- BorderWidth⁽¹⁹¹⁾
- ButtonsCol⁽¹⁹¹⁾
- ButtonHeight⁽¹⁹¹⁾
- Buttons⁽¹⁹¹⁾

- ButtonWidth ⁽¹⁹²⁾
- Color ⁽¹⁹²⁾
- ColorDisable ⁽¹⁹²⁾
- ColorDown ⁽¹⁹²⁾
- ColorSelect ⁽¹⁹³⁾
- HintCancel ⁽¹⁹³⁾
- HintFont ⁽¹⁹³⁾
- HintHeight ⁽¹⁹³⁾
- ImageList ⁽¹⁹⁴⁾
- Indent ⁽¹⁹⁴⁾
- PenFrame ⁽¹⁹⁴⁾
- ScaleImagesForDPI ⁽¹⁹⁴⁾
- Spacer ⁽¹⁹⁵⁾

Derived from TCustomControl

- Align
- Anchors
- Cursor
- Height
- HelpContext
- HelpKeyword (D6+)
- HelpType (D6+)
- Hint
- Left
- Name
- Tag
- Top
- Width

1.3.5.1.1 TSRVToolBar.AlignButton

Defines the position of buttons in the toolbar window.

property AlignButton: TSRVAlignButton ⁽²⁷⁰⁾;

See also

Indent ⁽¹⁹⁴⁾

Default value:

srvabCenter

1.3.5.1.2 TSRVToolBar.AlignText

Defines the position of text with the captions ⁽²⁷²⁾ of the highlighted button (button below the mouse pointer)

property AlignText: TSRVAlignText ⁽²⁷⁰⁾;

When none of the buttons are highlighted, HintCancel ⁽¹⁹³⁾ is displayed in this area.

Default value:

srvatBottom

1.3.5.1.3 TSRVToolBar.AutoSize

Allows autocalculation of the toolbar size.

property AutoSize: Boolean;

Default value:

False

1.3.5.1.4 TSRVToolBar.BorderWidth

Specifies a width of the toolbar border (spacing around the toolbar)

property BorderWidth: TRVPixel96Length;

A border (see PenFrame⁽¹⁹⁴⁾) is drawn in this area. The rest of space (if BorderWidth>the pen's width) is painted with Color⁽¹⁹²⁾.

This value is measured in pixels at 96 pixels per inch; the actual size scaled according to the current screen pixel density.

Default value:

1

See also:

- Spacer⁽¹⁹⁵⁾
- HintHeight⁽¹⁹³⁾

1.3.5.1.5 TSRVToolBar.ButtonCol

Specifies the maximal number of columns.

property ButtonCol: Integer;

ButtonCol specifies the number of buttons what may be placed in a line.

Default value:

-1 (unlimited)

1.3.5.1.6 TSRVToolBar.ButtonHeight

Sets the height of tool buttons (TSRVToolButton⁽²⁷¹⁾), in pixels.

property ButtonHeight: TRVPixel96Length;

This value is measured in pixels at 96 pixels per inch; the actual size scaled according to the current screen pixel density.

Default value:

26

See also:

- ButtonWidth⁽¹⁹²⁾

1.3.5.1.7 TSRVToolBar.Buttons

Collection of tool buttons (TSRVToolButton⁽²⁷¹⁾).

property Buttons: TSRVButtonCollection⁽²⁷⁰⁾;

1.3.5.1.8 TSRVToolBar.ButtonWidth

Sets the width of tool buttons (TSRVToolButton⁽²⁷¹⁾), in pixels.

property ButtonWidth: TRVPixel96Length;

This value is measured in pixels at 96 pixels per inch; the actual size scaled according to the current screen pixel density.

Default value:

26

See also:

- ButtonHeight⁽¹⁹¹⁾

1.3.5.1.9 TSRVToolBar.Color

Specifies the toolbar background color.

property Color: TColor;

This color is used for background of:

- the border around the toolbar (see BorderWidth⁽¹⁹¹⁾)
- area between the buttons (see Spacer⁽¹⁹⁵⁾)
- hint area (see HintHeight⁽¹⁹³⁾)
- tool buttons (if the button is in a normal state, i.e. not highlighted, pressed or disabled).

Default value:

clWindow

See also:

- ColorDisable⁽¹⁹²⁾
- ColorDown⁽¹⁹²⁾
- ColorSelect⁽¹⁹³⁾

1.3.5.1.10 TSRVToolBar.ColorDisable

Background color of disabled tool buttons⁽¹⁹¹⁾.

property ColorDisable: TColor;

This color is used for buttons having Enabled⁽²⁷²⁾=*False*.

Default value:

clSilver

See also:

- Color⁽¹⁹²⁾
- ColorDown⁽¹⁹²⁾
- ColorSelect⁽¹⁹³⁾

1.3.5.1.11 TSRVToolBar.ColorDown

Background color of pressed tool buttons⁽¹⁹¹⁾.

property ColorDown: TColor;

Default value:

\$00EED2C1

See also:

- Color⁽¹⁹²⁾
- ColorDisable⁽¹⁹²⁾
- ColorSelect⁽¹⁹³⁾

1.3.5.1.12 TSRVToolBar.ColorSelect

Background color of highlighted tool buttons⁽¹⁹¹⁾ (buttons under the mouse pointer).

property ColorSelect: TColor;

Default value:

\$00E2B598

See also:

- Color⁽¹⁹²⁾
- ColorDisable⁽¹⁹²⁾
- ColorDown⁽¹⁹²⁾

1.3.5.1.13 TSRVToolBar.HintCancel

Specifies the text to display in the hint area when none of the buttons is highlighted.

property HintCancel: TRVUnicodeString;

When this text is displayed, clicking on the hint area closes the tool window (and OnClickButton⁽¹⁹⁵⁾ occurs with *nil* parameter).

Default value:

'Cancel'

1.3.5.1.14 TSRVToolBar.HintFont

Specifies the font for the hint area.

type

```
TSRVTBFont = class(TFont);
```

property HintFont: TSRVTBFont;

This font is used for captions of tool buttons and HintCancel⁽¹⁹³⁾.

Default value:

default properties of TFont

See also:

- HintHeight⁽¹⁹³⁾

1.3.5.1.15 TSRVToolBar.HintHeight

Specifies the height of the hint area.

property HintHeight: TRVPixel96Length;

This value is measured in pixels at 96 pixels per inch; the actual size scaled according to the current screen pixel density.

See also:

- AlignText⁽¹⁹³⁾
- HintFont⁽¹⁹³⁾

1.3.5.1.16 TSRVToolBar.ImageList

A reference to image list for images in tool buttons⁽¹⁹¹⁾.

property ImageList: TCustomImageList;

This image list must contain buttons glyphs for all buttons states (normal, pressed, disabled) (you can use the same image for all of them).

When a toolbar displayed on a monitor that has pixel density (DPI) higher than 96, the toolbar draws images stretched, proportionally to the screen DPI. To turn off stretching, assign ScaleImagesForDPI⁽¹⁹⁴⁾ = *False*.

1.3.5.1.17 TSRVToolBar.Indent

Changes the position of buttons in the toolbar window.

property Indent: Integer;

If AlignButton⁽¹⁹⁰⁾=*srvabLeft*, buttons are shifted by Indent to the right.

If AlignButton⁽¹⁹⁰⁾=*srvabRight*, buttons are shifted by Indent to the left.

If AlignButton⁽¹⁹⁰⁾=*srvabCenter*, buttons are shifted by Indent/2 to the left.

Default value:

0

1.3.5.1.18 TSRVToolBar.PenFrame

Specifies a pen for drawing lines in the tool bar.

property PenFrame: TPen;

This pen is used for drawing border around the toolbar. The width of this border is defined as a pen's width, not in BorderWidth⁽¹⁹¹⁾.

Default value:

Style=*psSolid*, Width=1, Color=\$00C56A31;

1.3.5.1.19 TSRVToolBar.ScaleImagesForDPI

Specify whether images from ImageList⁽¹⁹⁴⁾ are displayed stretched according to the screen pixel density (DPI).

property ScaleImagesForDPI: Boolean;

If *True*, toolbar images are stretched from 96 DPI to the screen DPI.

If *False*, toolbar images are displayed as they are. This mode is useful if ImageList⁽¹⁹⁴⁾ contains images that corresponds to the current DPI.

Default value

True

1.3.5.1.20 TSRVToolBar.Spacer

Specifies the distance between tool buttons ⁽¹⁹¹⁾.

property `Spacer: TRVPixel96Length;`

This value is measured in pixels at 96 pixels per inch; the actual size scaled according to the current screen pixel density.

Default value:

0

1.3.5.2 Events

In TSRVToolBar

- `OnClickButton` ⁽¹⁹⁵⁾
- `OnDrawButtonBackground` ⁽¹⁹⁵⁾
- `OnEnter` ⁽¹⁹⁵⁾
- `OnEnterButton` ⁽¹⁹⁶⁾
- `OnLeave` ⁽¹⁹⁶⁾

1.3.5.2.1 TSRVToolBar.OnClickButton

Occurs when user clicks inside the toolbar.

property `OnClickButton: TSRVClickButtonEvent` ⁽²⁷¹⁾;

ToolButton – the clicked button (one of `Buttons` ⁽¹⁹¹⁾), or *nil*, if the user clicked outside any button (for example, in a hint area).

1.3.5.2.2 TSRVToolBar.OnDrawButtonBackground

Allows drawing a background of a toolbar button

type

```
TSRVDDrawButtonEvent = procedure (Sender: TObject; ToolButton: TSRVToolButton (2)
    ACanvas: TCanvas; const ARect: TRect; Selected: Boolean;
    var DoDefault: Boolean) of object;
```

property `OnDrawButtonBackground: TSRVDDrawButtonEvent;`

ToolButton – the button to draw (one of `Buttons` ⁽¹⁹¹⁾)

The button must be drawn in the rectangle **ARect** on **ACanvas**.

You can check if the button is pressed: **ToolButton.Down**. If the button is below the mouse pointer, **Selected** = *True*.

If you draw the button background yourself, assign **DoDefault** = *False* to cancel default drawing by the toolbar.

1.3.5.2.3 TSRVToolBar.OnEnter

Occurs when user moves the mouse pointer into the toolbar area.

property `OnEnter: TNotifyEvent;`

See also:

- OnLeave⁽¹⁹⁶⁾

1.3.5.2.4 TSRVToolBar.OnEnterButton

Occurs when user moves the mouse pointer into the button area.

property OnEnterButton: TSRVClickButtonEvent⁽²⁷¹⁾;

ToolButton – the button below the mouse pointer, one of Buttons⁽¹⁹¹⁾.

1.3.5.2.5 TSRVToolBar.OnLeave

Occurs when user moves the mouse pointer out of the toolbar area.

property OnLeave: TNotifyEvent;

See also:

- OnEnter⁽¹⁹⁵⁾

1.3.6 TSRVToolWindow

TSRVToolWindow shows a toolbar inside a popup window.

Unit SRVToolWindow.

Syntax

TSRVToolWindow = **class** (TComponent)

Hierarchy

TObject

TPersistent

TComponent

Description

TSRVToolWindow shows a toolbar inside a popup window, when you call Execute⁽²⁰²⁾ method. The window is closed without any action when it loses focus or when user clicks outside any button on the toolbar

TSRVToolWindow uses TSRVToolBar⁽¹⁸⁸⁾ inside.

Usually TSRVToolWindow is used as a submenu for buttons in TSVRichViewEdit⁽⁴⁰⁾'s MenuVertical⁽⁵⁶⁾.

Properties

This class contains almost the same set of properties as TSRVCustomPropertyTB⁽¹²⁶⁾ class and TSRVToolBar⁽¹⁸⁸⁾ component.

A toolbar consists of buttons and a hint area.

The hint area can be placed at the top or at the bottom of the toolbar, depending on AlignText⁽¹⁹⁷⁾ property. A height of this area is defined in HintHeight⁽²⁰⁰⁾ property. This area can work as a "Cancel" button. When none of buttons is highlighted, it displays HintCancel⁽²⁰⁰⁾. Otherwise, it displays Caption⁽²⁷²⁾ of tool button. A font for text displayed in this area is specified in HintFont⁽²⁰⁰⁾.

Buttons are defined in Buttons⁽¹⁹⁸⁾ collection. Buttons are arranged in rows and columns, the number of columns is specified in ButtonsCol⁽¹⁹⁸⁾. A size of buttons is defined in ButtonWidth⁽¹⁹⁸⁾ and ButtonHeight⁽¹⁹⁸⁾ properties. A spacing between buttons is specified in Spacer⁽²⁰¹⁾.

The toolbar can have a border, its width is specified in BorderWidth⁽¹⁹⁸⁾.

The toolbar color is Color⁽¹⁹⁹⁾. Colors of disabled, pressed and highlighted buttons are specified in ColorDisable⁽¹⁹⁹⁾, ColorDown⁽¹⁹⁹⁾, and ColorSelect⁽²⁰⁰⁾ properties. A pen of frame around the toolbar is defined in PenFrame⁽²⁰¹⁾.

1.3.6.1 Properties

In TSRVToolWindow

- AlignText⁽¹⁹⁷⁾
- BorderWidth⁽¹⁹⁸⁾
- ButtonCol⁽¹⁹⁸⁾
- ButtonHeight⁽¹⁹⁸⁾
- Buttons⁽¹⁹⁸⁾
- ButtonWidth⁽¹⁹⁸⁾
- Color⁽¹⁹⁹⁾
- ColorDisable⁽¹⁹⁹⁾
- ColorDown⁽¹⁹⁹⁾
- ColorSelect⁽²⁰⁰⁾
- HintCancel⁽²⁰⁰⁾
- HintFont⁽²⁰⁰⁾
- HintHeight⁽²⁰⁰⁾
- ImageList⁽²⁰¹⁾
- PenFrame⁽²⁰¹⁾
- ScaleImagesForDPI⁽²⁰¹⁾
- Spacer⁽²⁰¹⁾

Derived from TComponent

- Name
- Tag

1.3.6.1.1 TSRVToolWindow.AlignText

Defines the position of text with the captions⁽²⁷²⁾ of the highlighted button (button below the mouse pointer)

property AlignText: TSRVAlignText⁽²⁷⁰⁾;

When none of the buttons are highlighted, HintCancel⁽²⁰⁰⁾ is displayed in this area.

Default value:

srvatBottom

1.3.6.1.2 TSRVToolWindow.BorderWidth

Specifies a width of the toolbar border (spacing around the toolbar)

property BorderWidth: TRVPixel96Length;

A border (see PenFrame⁽²⁰¹⁾) is drawn in this area. The rest of space (if BorderWidth>the pen's width) is painted with Color⁽¹⁹⁹⁾.

This value is measured in pixels at 96 pixels per inch; the actual size scaled according to the current screen pixel density.

Default value:

1

See also:

- Spacer⁽²⁰¹⁾
- HintHeight⁽²⁰⁰⁾

1.3.6.1.3 TSRVToolWindow.ButtonCol

Specifies the maximal number of columns.

property ButtonCol: Integer;

ButtonCol specifies the number of buttons what may be placed in a line.

Default value:

-1 (unlimited)

1.3.6.1.4 TSRVToolWindow.ButtonHeight

Sets the height of tool buttons (TSRVToolButton⁽²⁷¹⁾), in pixels.

property ButtonHeight: TRVPixel96Length;

This value is measured in pixels at 96 pixels per inch; the actual size scaled according to the current screen pixel density.

Default value:

26

See also:

- ButtonWidth⁽¹⁹⁸⁾

1.3.6.1.5 TSRVToolWindow.Buttons

Collection of tool buttons (TSRVToolButton⁽²⁷¹⁾).

property Buttons: TSRVButtonCollection⁽²⁷⁰⁾;

1.3.6.1.6 TSRVToolWindow.ButtonWidth

Sets the width of tool buttons (TSRVToolButton⁽²⁷¹⁾), in pixels.

property ButtonWidth: TRVPixel96Length;

This value is measured in pixels at 96 pixels per inch; the actual size scaled according to the current screen pixel density.

Default value:

26

See also:

- ButtonHeight⁽¹⁹⁸⁾

1.3.6.1.7 TSRVToolWindow.Color

Specifies the toolbar background color.

property Color: TColor;

This color is used for background of:

- the border around the toolbar (see BorderWidth⁽¹⁹⁸⁾)
- area between the buttons (see Spacer⁽²⁰¹⁾)
- hint area (see HintHeight⁽²⁰⁰⁾)
- tool buttons (if the button is in a normal state, i.e. not highlighted, pressed or disabled).

Default value:

c/Window

See also:

- ColorDisable⁽¹⁹⁹⁾
- ColorDown⁽¹⁹⁹⁾
- ColorSelect⁽²⁰⁰⁾

1.3.6.1.8 TSRVToolWindow.ColorDisable

Background color of disabled tool buttons⁽¹⁹⁸⁾.

property ColorDisable: TColor;

This color is used for buttons having Enabled⁽²⁷²⁾=False.

Default value:

c/Silver

See also:

- Color⁽¹⁹⁹⁾
- ColorDown⁽¹⁹⁹⁾
- ColorSelect⁽²⁰⁰⁾

1.3.6.1.9 TSRVToolWindow.ColorDown

Background color of pressed tool buttons⁽¹⁹⁸⁾.

property ColorDown: TColor;

Default value:

\$00EED2C1

See also:

- Color⁽¹⁹⁹⁾
- ColorDisable⁽¹⁹⁹⁾
- ColorSelect⁽²⁰⁰⁾

1.3.6.1.10 TSRVToolWindow.ColorSelect

Background color of highlighted tool buttons⁽¹⁹⁸⁾ (buttons under the mouse pointer).

property ColorSelect: TColor;

Default value:

\$00E2B598

See also:

- Color⁽¹⁹⁹⁾
- ColorDisable⁽¹⁹⁹⁾
- ColorDown⁽¹⁹⁹⁾

1.3.6.1.11 TSRVToolWindow.HintCancel

Specifies the text to display in the hint area when none of the buttons is highlighted.

property HintCancel: TRVUnicodeString;

When this text is displayed, clicking on the hint area closes the tool window (and OnClickButton⁽²⁰²⁾ occurs with *nil* parameter).

Default value:

'Cancel'

1.3.6.1.12 TSRVToolWindow.HintFont

Specifies the font for the hint area.

type

```
TSRVTBFont = class(TFont);
```

property HintFont: TSRVTBFont;

This font is used for captions of tool buttons and HintCancel⁽²⁰⁰⁾.

Default value:

default properties of TFont

See also:

- HintHeight⁽²⁰⁰⁾

1.3.6.1.13 TSRVToolWindow.HintHeight

Specifies the height of the hint area.

property HintHeight: TRVPixel96Length;

This value is measured in pixels at 96 pixels per inch; the actual size scaled according to the current screen pixel density.

See also:

- AlignText⁽²⁰⁰⁾
- HintFont⁽²⁰⁰⁾

1.3.6.1.14 TSRVToolWindow.ImageList

A reference to image list for images in tool buttons ⁽¹⁹⁸⁾.

property ImageList: TCustomImageList;

This image list must contain buttons glyphs for all buttons states (normal, pressed, disabled) (you can use the same image for all of them).

When a toolbar displayed on a monitor that has pixel density (DPI) higher than 96, the toolbar draws images stretched, proportionally to the screen DPI. To turn off stretching, assign ScaleImagesForDPI ⁽²⁰¹⁾ = *False*.

1.3.6.1.15 TSRVToolWindow.PenFrame

Specifies a pen for drawing lines in the tool bar.

property PenFrame: TPen;

This pen is used for drawing border around the toolbar. The width of this border is defined as a pen's width, not in BorderWidth ⁽¹⁹⁸⁾.

Default value:

Style=*psSolid*, Width=1, Color=*\$00C56A31*;

1.3.6.1.16 TSRVToolWindow.ScaleImagesForDPI

Specify whether images from ImageList ⁽²⁰¹⁾ are displayed stretched according to the screen pixel density (DPI).

property ScaleImagesForDPI: Boolean;

If *True*, toolbar images are stretched from 96 DPI to the screen DPI.

If *False*, toolbar images are displayed as they are. This mode is useful if ImageList ⁽²⁰¹⁾ contains images that corresponds to the current DPI.

Default value

True

1.3.6.1.17 TSRVToolWindow.Spacer

Specifies the distance between tool buttons ⁽¹⁹⁸⁾.

property Spacer: TRVPixel96Length;

This value is measured in pixels at 96 pixels per inch; the actual size scaled according to the current screen pixel density.

Default value:

0

1.3.6.2 Methods

In TSRVToolWindow

Execute ⁽²⁰²⁾

1.3.6.2.1 TSRVToolWindow.Execute

The methods show a window with a toolbar.

```
procedure Execute(X, Y : Integer); overload;  
procedure Execute(R : TRect); overload;
```

The window remains open until it loses focus or one of buttons is clicked.

X,Y – screen coordinates of the top left corner of the window. The window will be shown exactly at these coordinates.

R – screen coordinates specifying a rectangle of the button which called this window. The window will be positioned close to this button without overlapping it, and without going beyond the screen. This method sets AlignText⁽¹⁹⁷⁾ property depending on the window position.

1.3.6.3 Events

In TSRVToolWindow

- OnClickButton⁽²⁰²⁾
- OnEnter⁽²⁰²⁾
- OnEnterButton⁽²⁰²⁾
- OnLeave⁽²⁰³⁾
- OnSelectEvent⁽²⁰³⁾

1.3.6.3.1 TSRVToolWindow.OnClickButton

Occurs when user clicks inside the toolbar.

```
property OnClickButton: TSRVClickButtonEvent(271);
```

ToolButton – the clicked button (one of Buttons⁽¹⁹⁸⁾), or *nil*, if the user clicked outside any button (for example, in a hint area).

1.3.6.3.2 TSRVToolWindow.OnEnter

Occurs when user moves the mouse pointer into the toolbar area.

```
property OnEnter: TNotifyEvent;
```

See also:

- OnLeave⁽²⁰³⁾

1.3.6.3.3 TSRVToolWindow.OnEnterButton

Occurs when user moves the mouse pointer into the button area.

```
property OnEnterButton: TSRVClickButtonEvent(271);
```

ToolButton – the button below the mouse pointer, one of Buttons⁽¹⁹⁸⁾.

1.3.6.3.4 TSRVToolWindow.OnLeave

Occurs when user moves the mouse pointer out of the toolbar area.

property OnLeave: TNotifyEvent;

See also:

- OnEnter⁽²⁰²⁾

1.3.6.3.5 TSRVToolWindow.OnSelectEvent

Occurs when user clicked a tool button or closed the window.

property OnSelectEvent: TSRVClickButtonEvent⁽²⁷¹⁾;

The window is closed (and the action is canceled) when the user clicks on the window outside any button, or when it loses the input focus.

1.3.7 TSRVPrint

TSRVPrint allows printing documents from TSVRichViewEdit⁽⁴⁰⁾ in several modes, including posters.

Unit SRVPrint.

Syntax

TSRVPrint = **class** (TComponent)

Hierarchy

TObject

TPersistent

TComponent

Description

The main property of TRVPrint is PrintMode⁽²¹³⁾. It defines how the document in SRichViewEdit⁽²¹⁶⁾ is printed.

- *srvpAuto* – the printer paper size and orientation are set automatically;
- *srvpStretchToFit* – the current printer paper size and orientation are used;
- *srvpStretchIfLarger* is like *srvpStretchToFit* if the document page size is larger than the printer paper size. Otherwise, the document is printed at the top left corner of paper.
- *srvpTiles* – if possible, prints several copies of the document page on the same paper sheet;
- *srvpPoster* – allows printing posters: each page can be printed on several paper sheets; this is the most complex mode, it is described below.
- *srvpGrid* – allows printing several document pages on the same paper sheet.

The component uses the current printer (specified in Printer.PrinterIndex, where Printer is a function defined in the Printers unit). When the parameters of the current printer are changed, or the current printer is changed, call Update⁽²²¹⁾ to recalculate properties according to the printer.

Posters

Sizes

A document is prepared in `TSRichViewEdit`⁽⁴⁰⁾ component. Its size is defined in `TSRichViewEdit`⁽⁴⁰⁾'s page properties⁽⁵⁸⁾.

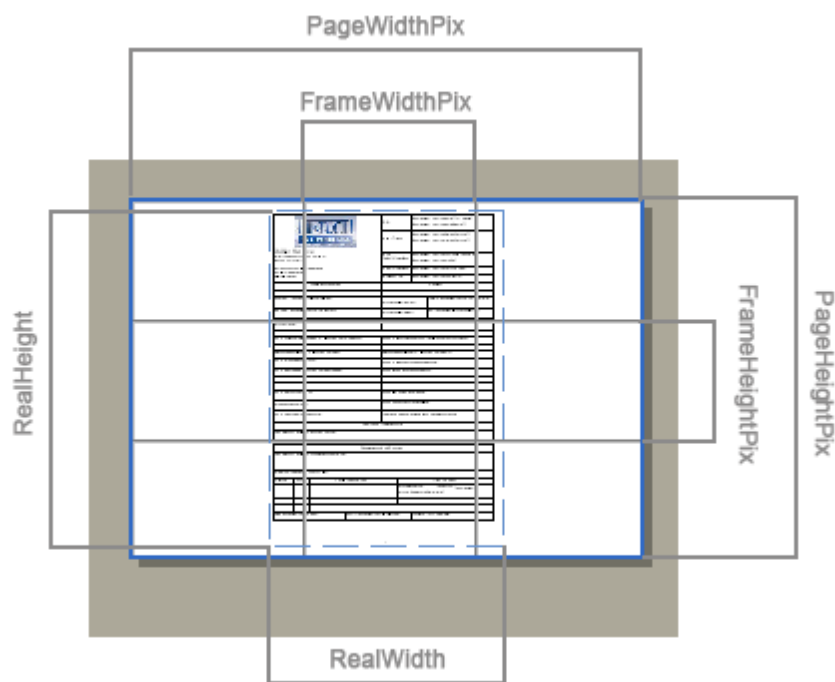
Each page of this document is scaled to `RealWidth`⁽²¹⁵⁾ x `RealHeight`⁽²¹⁴⁾ (see below how they are calculated).

Each page of this document is printed as a poster. The poster consists of frames. Frame is one paper sheet, it has size of the printer's paper (`FrameWidthPix`⁽²⁰⁹⁾ x `FrameHeightPix`⁽²⁰⁸⁾, orientation `Orientation`⁽²¹¹⁾).

The poster consists of $\text{Ceil}(\text{FrameCountX}^{(207)}) \times \text{Ceil}(\text{FrameCountY}^{(208)})$ frames (so, each page of the original document will be printed on $\text{Ceil}(\text{FrameCountX}^{(207)}) * \text{Ceil}(\text{FrameCountY}^{(208)})$ printer pages).

The poster size in pixels is `PageWidthPix`⁽²¹³⁾ x `PageHeightPix`⁽²¹²⁾ (i.e. $\text{Ceil}(\text{FrameCountX}^{(207)}) * \text{FrameWidthPix}^{(209)} \times \text{Ceil}(\text{FrameCountY}^{(208)}) * \text{FrameHeightPix}^{(208)}$).

`FrameCountX`⁽²⁰⁷⁾ and `FrameCountY`⁽²⁰⁸⁾ can be fractional values (if `IntegerCount`⁽²⁰⁹⁾ = *False*), it affects calculation of `RealWidth`⁽²¹⁵⁾ and `RealHeight`⁽²¹⁴⁾ (see below).



Sizes in posters

How the document page is scaled

As it was said above, each page of the original document is scaled to `RealWidth`⁽²¹⁵⁾ x `RealHeight`⁽²¹⁴⁾.

If `ScaleImage`⁽²¹⁵⁾ = *True*, pages are scaled to the full poster size (`RealWidth`⁽²¹⁵⁾ = `PageWidthPix`⁽²¹³⁾, `RealHeight`⁽²¹⁴⁾ = `PageHeightPix`⁽²¹²⁾), without keeping the original proportions.

If `ScaleImage`⁽²¹⁵⁾ = `False`, pages are scaled proportionally to the maximal size that fits in `TotalFrameWPix`⁽²¹⁶⁾ x `TotalFrameHPix`⁽²¹⁶⁾ (so `RealWidth`⁽²¹⁵⁾ ≤ `TotalFrameWPix`⁽²¹⁶⁾, `RealHeight`⁽²¹⁴⁾ ≤ `TotalFrameHPix`⁽²¹⁶⁾).

`TotalFrameWPix`⁽²¹⁶⁾ and `TotalFrameHPix`⁽²¹⁶⁾ are similar to `PageWidthPix`⁽²¹³⁾ and `PageHeightPix`⁽²¹²⁾, but calculated without rounding, i.e. they are equal to `FrameCountX`⁽²⁰⁷⁾ * `FrameWidthPix`⁽²⁰⁹⁾ x `FrameCountY`⁽²⁰⁸⁾ * `FrameHeightPix`⁽²⁰⁸⁾.

If `ScaleImage`⁽²¹⁵⁾ = `False`, the document page can be printed in the top left corner or in the center of the poster, depending on the `Center`⁽²⁰⁶⁾ property.

See also: `UsePhysicalOffsets`⁽²¹⁶⁾.

Printing and previewing

The main printing methods are `Print`⁽²¹⁸⁾ and `PrintPages`⁽²²⁰⁾. See also `PrintFrame`⁽²¹⁹⁾, `PrintFrames`⁽²¹⁹⁾, `PrintFramesEx`⁽²²⁰⁾.

A print preview is displayed by `TSRVPreview`⁽²²²⁾ component.

You can provide a visual indication of printing progress using `OnSendingToPrinter`⁽²²¹⁾ event.

To make printing output compatible with metafiles, assign `MetafileCompatibility`⁽²¹⁰⁾ = `True` (and probably, `NoMetafiles`⁽²¹⁰⁾ = `True`). It may be necessary for some virtual printers

1.3.7.1 Properties

In TSRVPrint

- `AutoUpdate`⁽²⁰⁶⁾
- `Center`⁽²⁰⁶⁾
- `ClipMargins`⁽²⁰⁷⁾
- `FrameCountX`⁽²⁰⁷⁾
- `FrameCountY`⁽²⁰⁸⁾
- ▶ `FrameHeightPix`⁽²⁰⁸⁾
- ▶ `FrameWidthPix`⁽²⁰⁹⁾
- `IntegerCount`⁽²⁰⁹⁾
- `MetafileCompatibility`⁽²¹⁰⁾
- `NoMetafiles`⁽²¹⁰⁾
- ▶ `OffXPix`⁽²¹⁰⁾
- ▶ `OffYPix`⁽²¹¹⁾
- `Orientation`⁽²¹¹⁾
- `PageColCount`⁽²¹²⁾
- `PageFormat`⁽²¹²⁾
- `PageHeightPix`⁽²¹²⁾
- `PageRowCount`⁽²¹³⁾
- `PageWidthPix`⁽²¹³⁾
- `PrintMode`⁽²¹³⁾
- ▶ `RealHeight`⁽²¹⁴⁾
- ▶ `RealWidth`⁽²¹⁵⁾

- ScaleImage⁽²¹⁵⁾
- SRichViewEdit⁽²¹⁶⁾
- ▶ TotalFrameHPix⁽²¹⁶⁾
- ▶ TotalFrameWPix⁽²¹⁶⁾
- UsePhysicalOffsets⁽²¹⁶⁾

Derived from TComponent

- Name
- Tag

1.3.7.1.1 TSRVPrint.AutoUpdate

Enables an autoupdate mode.

property AutoUpdate: Boolean;

If *True*, TSRVPrint recalculates its parameters when the content of SRichViewEdit⁽²¹⁶⁾ is changed. Recalculation occurs on OnChange⁽¹⁰⁴⁾ and OnPageCountChanged⁽¹¹³⁾ events.

If the document was changed when *AutoUpdate=False*, call Update⁽²²¹⁾.

Default value:

False

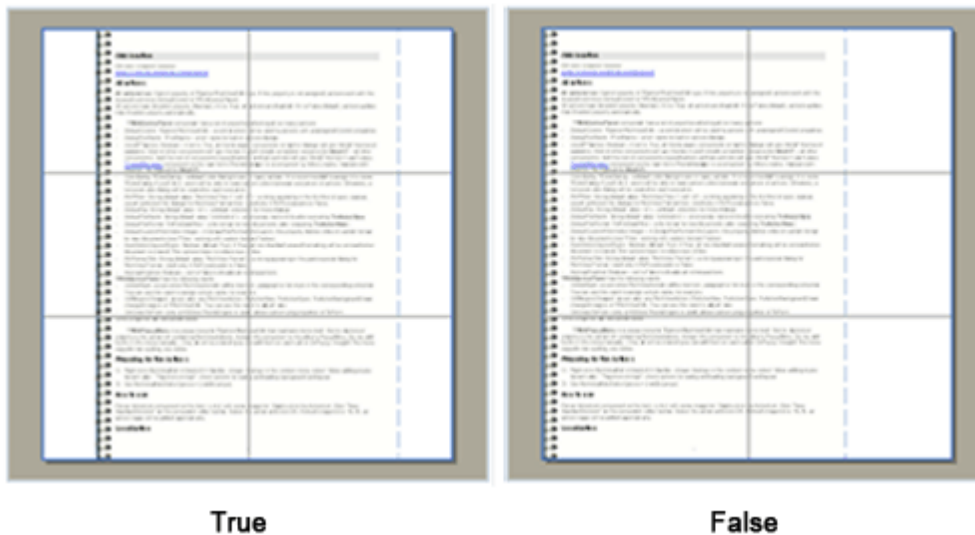
1.3.7.1.2 TSRVPrint.Center

In the poster mode: the property places the document image in the center relative to the poster.

In the grid mode: the property places the document pages in the center of the paper sheet.

property Center: Boolean;

Center



A posters is printed if PrintMode⁽²¹³⁾ = *srvpPoster*. For posters, this property makes sense only if ScaleImage⁽²¹⁵⁾ = *False*.

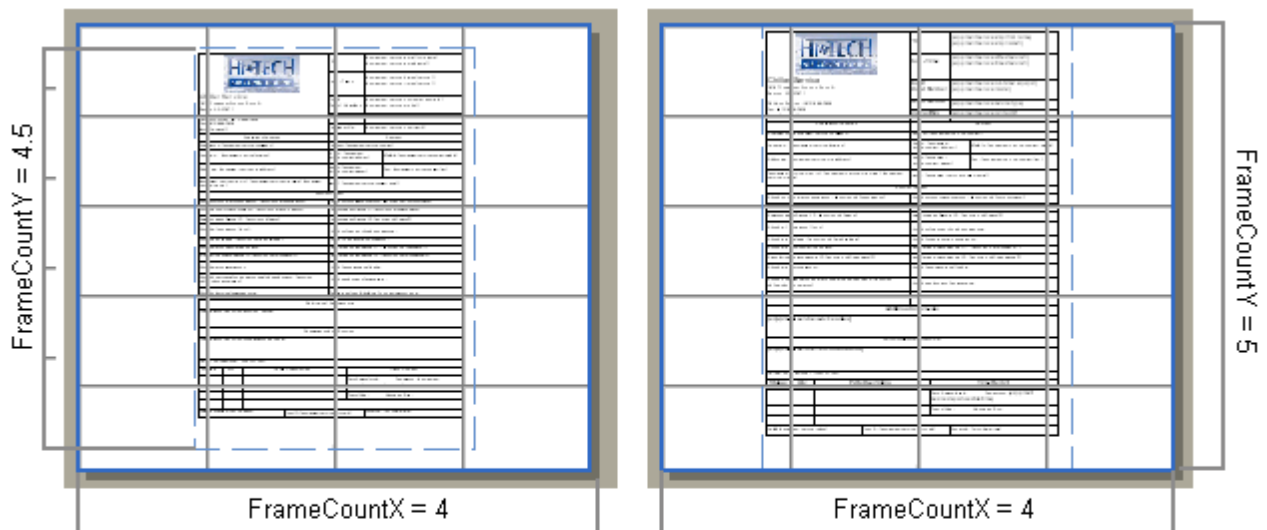
A grid is printed if PrintMode⁽²¹³⁾ = *srvpGrid*.

Default value:*True***1.3.7.1.3 TSRVPrint.ClipMargins**

Restricts the output area.

property ClipMargins: Boolean;If this property is *True*, printing on margins is not allowed (clipped).**Default value:***True***1.3.7.1.4 TSRVPrint.FrameCountX**

Count of frames in a poster, horizontally.

property FrameCountX: Extended;Posters are printed if PrintMode⁽²¹³⁾ = *srvpPoster*.Assigning to this property changes the poster width (PageWidthPix⁽²¹³⁾) and may change RealWidth⁽²¹⁵⁾ and RealHeight⁽²¹⁴⁾.Assigning to this property resets PageFormat⁽²¹²⁾ to *srvfmCustom* and IntegerCount⁽²⁰⁹⁾ to *False* (value may be fractional).**Frame Count**

PageFormat = A1
 Printer page format = A4
 Orientation = poLandscape
 IntegerCount = False
 ScaleImage = False

Frame Count

1.3.7.1.5 TSRVPrint.FrameCountY

Count of frames in a poster, vertically.

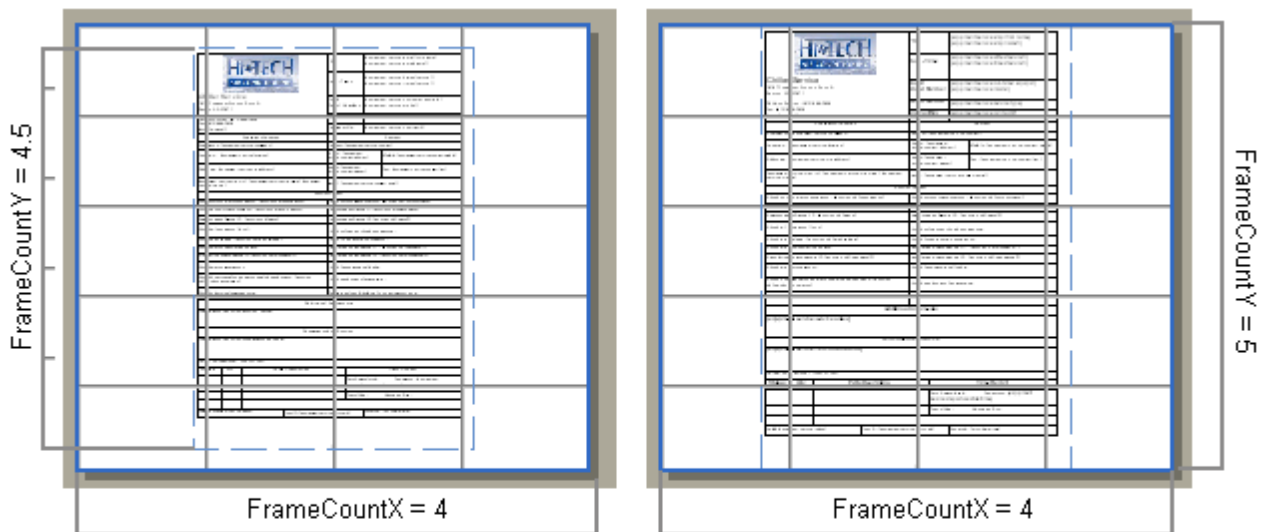
property FrameCountY: Extended;

Posters are printed if PrintMode⁽²¹³⁾ = *srvpPoster*.

Assigning to this property changes the poster height (PageHeightPix⁽²¹²⁾) and may change RealWidth⁽²¹⁵⁾ and RealHeight⁽²¹⁴⁾.

Assigning to this property resets PageFormat⁽²¹²⁾ to *srvfmCustom* and IntegerCount⁽²⁰⁹⁾ to *False* (value may be fractional).

Frame Count



PageFormat = A1
 Printer page format = A4
 Orientation = poLandscape
 IntegerCount = False
 ScaleImage = False

Frame Count

1.3.7.1.6 TSRVPrint.FrameHeightPix

Returns the height of one frame of a poster, in pixels.

property FrameHeightPix: Integer;

Posters are printed if PrintMode⁽²¹³⁾ = *srvpPoster*.

Each page of a poster is printed as several frames.

If the default printer is set, the printer paper height is used.

Default value:

1

See also:

- FrameWidthPix⁽²⁰⁹⁾
- the picture in the topic about TSRVPrint⁽²⁰³⁾

1.3.7.1.7 TSRVPrint.FrameWidthPix

Returns the height of one frame of a poster, in pixels.

property `FrameWidthPix : Integer;`

Posters are printed if `PrintMode`⁽²¹³⁾ = `srvpPoster`.

Each page of a poster is printed as several frames.

If the default printer is set, the printer paper width is used.

Default value:

1

See also:

- `FrameHeightPix`⁽²⁰⁸⁾
- the picture in the topic about `TSRVPrint`⁽²⁰³⁾

1.3.7.1.8 TSRVPrint.IntegerCount

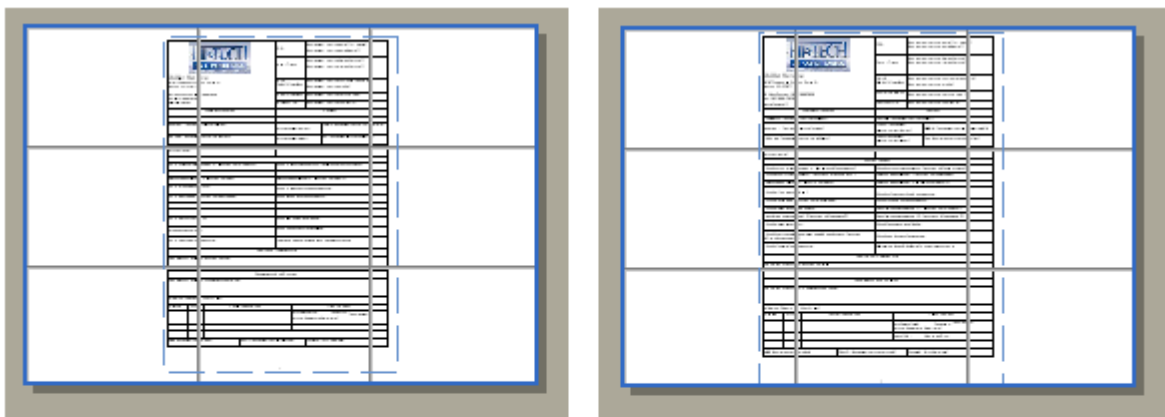
Specifies whether to round values of `FrameCountX`⁽²⁰⁷⁾ and `FrameCountY`⁽²⁰⁸⁾ up.

property `IntegerCount : Boolean;`

Posters are printed if `PrintMode`⁽²¹³⁾ = `srvpPoster`.

This property allows to use integer numbers of frame counts for a poster.

IntegerCount



`IntegerCount = False`
`FrameCountX = 2.8`
`FrameCountY = 2.8`

`IntegerCount = True`
`FrameCountX = 3`
`FrameCountY = 3`

`PageFormat = A1`
`Printer page format = A4`
`Orientation = poLandscape`
`ScaleImage = False`

IntegerCount Example

Default value:

False

1.3.7.1.9 TSRVPrint.MetafileCompatibility

Specifies whether the component's printing output is compatible with metafiles.

property MetafileCompatibility: Boolean;

If **MetafileCompatibility**=*False*, printing output may be incompatible with metafiles; the component may use font glyph indexes instead of character codes when printing text. Such metafiles will be displayed incorrectly on computers without this font or with another version of this font.

If **MetafileCompatibility**=*True*, the component always uses character codes when printing text, to create metafile-compatible output.

Default value:

False

See also

- TRVAControlPanel.MetafileCompatibility (in the RichViewActions help file)

1.3.7.1.10 TSRVPrint.NoMetafiles

Specifies whether the component prints metafiles as bitmaps.

property MetafileCompatibility: Boolean;

If **NoMetafiles**=*False*, metafiles in documents are printed as they are.

If **NoMetafiles**=*True*, metafiles are printed as bitmaps. In this mode, printing quality may be lower (pixelated graphics), however, it may be useful for printing on printers that do not handle embedded metafiles properly.

Default value:

False

1.3.7.1.11 TSRVPrint.OffXPix

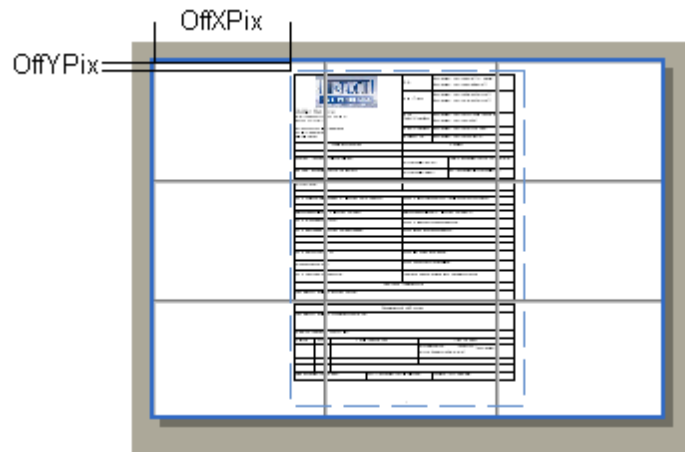
Returns the horizontal shift of the document page on the poster, pixels.

property OffXPix: Extended;

Posters are printed if PrintMode⁽²¹³⁾ = *srvpPoster*.

The shift is nonzero if

- the document page is not scaled to the full poster size (ScaleImage⁽²¹⁵⁾=*False*) and
- the page is printed at the center of the poster (Center⁽²⁰⁶⁾=*True*) and
- the page width (RealWidth⁽²¹⁵⁾) is smaller than the poster width (PageWidthPix⁽²¹³⁾).



OffXPix and OffYPix

1.3.7.1.12 TSRVPrint.OffYPix

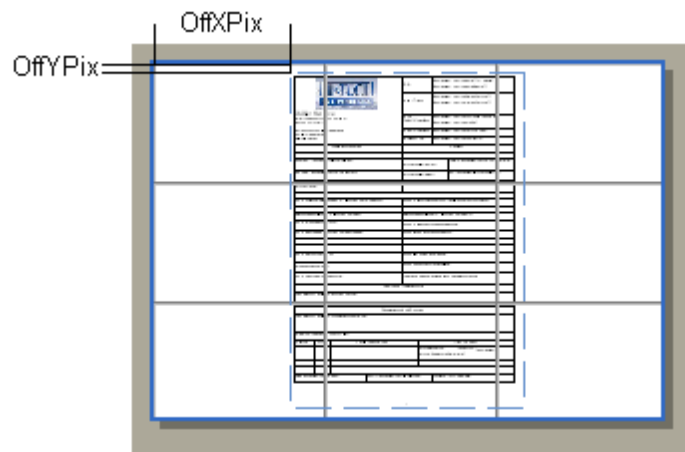
Returns the vertical shift of the document page on the poster, pixels.

property OffYPix: Extended;

Posters are printed if PrintMode⁽²¹³⁾ = *srvpPoster*.

The shift is nonzero if

- the document page is not scaled to the full poster size (ScaleImage⁽²¹⁵⁾ = *False*) and
- the page is printed at the center of the poster (Center⁽²⁰⁶⁾ = *True*) and
- the page height (RealHeight⁽²¹⁴⁾) is smaller than the poster height (PageHeightPix⁽²¹²⁾).



OffXPix and OffYPix

1.3.7.1.13 TSRVPrint.Orientation

Specifies the orientation of frames in a poster.

property Orientation: TPrinterOrientation;

Posters are printed if PrintMode⁽²¹³⁾ = *srvpPoster*.

Default value:

poPortrait

1.3.7.1.14 TSRVPrint.PageColCount

Number of document pages on one paper sheet in horizontal direction, in *srvpGrid* mode.

property PageCount: Integer;

This property is used if PrintMode⁽²¹³⁾ = *srvpGrid*.

In this mode, PageRowCount⁽²¹³⁾ x PageColCount document pages are printed on a single paper sheet.

Default value:

1

See also:

Center⁽²⁰⁶⁾

1.3.7.1.15 TSRVPrint.PageFormat

The poster's page format.

property PageFormat: TSRVPageFormat⁽²⁷⁷⁾;

Posters are printed if PrintMode⁽²¹³⁾ = *srvpPoster*.

The poster consists of frames, each frame has size of the printer paper.

This is the page format as of the whole poster, containing all frames.

Assigning to this property changes PageWidthPix⁽²¹³⁾ and PageHeightPix⁽²¹²⁾. Values of FrameCountX⁽²⁰⁷⁾ and FrameCountY⁽²⁰⁸⁾ are recalculated to fit the given format (taking the printer's paper size into account). If IntegerCount⁽²⁰⁹⁾=True, they are calculated as integer values. This calculation depends on UsePhysicalOffsets⁽²¹⁶⁾ property.

When you assign value to FrameCountX⁽²⁰⁷⁾, FrameCountY⁽²⁰⁸⁾, PageWidthPix⁽²¹³⁾, or PageHeightPix⁽²¹²⁾, this property is reset to *srvfmCustom*.

Default value:

srvfmA4

1.3.7.1.16 TSRVPrint.PageHeightPix

The poster height, pixels

property PageHeightPix: Extended;

Posters are printed if PrintMode⁽²¹³⁾ = *srvpPoster*.

A value of this property can be assigned or calculated. The formula PageHeightPix = Ceil(FrameCountY⁽²⁰⁸⁾*FrameHeightPix⁽²⁰⁸⁾ is used.

Assigning to this property resets PageFormat⁽²¹²⁾ to *srvfmCustom*.

Assigning any value (except for *srvfmCustom*) to PageFormat⁽²¹²⁾ changes a value of this property.

Assigning to FrameCountY⁽²⁰⁸⁾ changes value of this property.

See also:

- PageWidthPix⁽²¹³⁾
- the picture in the topic about TSRVPrint⁽²⁰³⁾

1.3.7.1.17 TSRVPrint.PageRowCount

Number of document pages on one paper sheet in vertical direction, in *srvpGrid* mode.

property PageCount: Integer;

This property is used if PrintMode⁽²¹³⁾ = *srvpGrid*.

In this mode, PageRowCount x PageColCount⁽²¹²⁾ document pages are printed on a single paper sheet.

Default value:

1

See also:

Center⁽²⁰⁶⁾

1.3.7.1.18 TSRVPrint.PageWidthPix

The poster height, pixels

property PageWidthPix: Extended;

Posters are printed if PrintMode⁽²¹³⁾ = *srvpPoster*.

A value of this property can be assigned or calculated. The formula PageWidthPix = Ceil(FrameCountX⁽²⁰⁷⁾)*FrameWidthPix⁽²⁰⁹⁾ is used.

Assigning to this property resets PageFormat⁽²¹²⁾ to *srvfmCustom*.

Assigning any value (except for *srvfmCustom*) to PageFormat⁽²¹²⁾ changes a value of this property.

Assigning to FrameCountX⁽²⁰⁷⁾ changes value of this property.

See also:

- PageHeightPix⁽²¹²⁾
- the picture in the topic about TSRVPrint⁽²⁰³⁾

1.3.7.1.19 TSRVPrint.PrintMode

Specifies how the document from SRichViewEdit⁽²¹⁶⁾ will be printed.

type

```
TSRVPrintMode = (srvpAuto, srvpStretchToFit, srvpStretchIfLarger,
  srvpTiles,  srvpPoster, srvpGrid);
```

property PrintMode: TSRVPrintMode;

Value	Meaning
<i>srvpAuto</i>	A printer paper size and orientation are set automatically; if the printer supports the page size specified for the document, it is used (orientation is set automatically too). If the printer does not support the document page size, this mode works like <i>srvpStretchIfLarger</i> mode.

<i>srvpStretchToFit</i>	The current printer paper size and orientation are used; on printing, the document is stretched proportionally to fit the page size and the orientation chosen for the printer.
<i>srvpStretchIfLarger</i>	This mode is like <i>srvpStretchToFit</i> if the document page size is larger than the printer paper size. Otherwise, the document is printed at the top left corner of paper sheet.
<i>srvpTiles</i>	<p>The current printer paper size and orientation are used. If possible, prints several copies of the document page on the same paper sheet; for example, if the document page size is A6, and the printer page size is A4, it prints 4 copies (if orientations are the same).</p> <p>If document page is larger than paper (possibly because of different orientation), the component shrinks it to fit the paper size, and tries to print several copies of this shrunk document page.</p>
<i>srvpPoster</i>	<p>Allows printing posters (one page on several paper sheets).</p> <p>It's recommended to assign <code>UsePhysicalOffsets⁽²¹⁶⁾ = False</code> in this mode.</p>
<i>srvpGrid</i>	<p>The current printer paper size and orientation are used.</p> <p><code>PageRowCount⁽²¹³⁾ x PageColCount⁽²¹²⁾</code> document pages are printed on each paper sheet as a grid. These pages are scaled proportionally to fit; unlike <i>srvpTiles</i> mode, these pages are not copies of the same page.</p>

Default value:*srvpPoster***1.3.7.1.20 TSRVPrint.RealHeight**

Returns the height of the scaled page of the source document.

property `RealHeight`: `Extended`;Posters are printed if `PrintMode(213) = srvpPoster`.If `ScaleImage(215) = True`, each page of the source document is scaled to the full poster size without keeping proportions, so `RealHeight = PageHeightPix(212)`.If `ScaleImage(215) = False`, each page is scaled proportionally to the maximal size that fits in `TotalFrameWPix(216) x TotalFrameHPix(216)`, so `RealHeight ≤ TotalFrameHPix(216)`.**See also:**

- `RealWidth(215)`
- the picture in the topic about `TSRVPrint(203)`

1.3.7.1.21 TSRVPrint.RealWidth

Returns the width of the scaled page of the source document.

property RealWidth: Extended;

Posters are printed if PrintMode⁽²¹³⁾ = *srvpPoster*.

If ScaleImage⁽²¹⁵⁾ = *True*, each page of the source document is scaled to the full poster size without keeping proportions, so RealWidth=PageWidthPix⁽²¹³⁾.

If ScaleImage⁽²¹⁵⁾ = *False*, each page is scaled proportionally to the maximal size that fits in TotalFrameWPix⁽²¹⁶⁾ x TotalFrameHPix⁽²¹⁶⁾, so RealWidth<=TotalFrameWPix⁽²¹⁶⁾.

See also:

- RealHeight⁽²¹⁴⁾
- the picture in the topic about TSRVPrint⁽²⁰³⁾

1.3.7.1.22 TSRVPrint.ScaleImage

Specifies whether the source page image must be stretched to the full poster size.

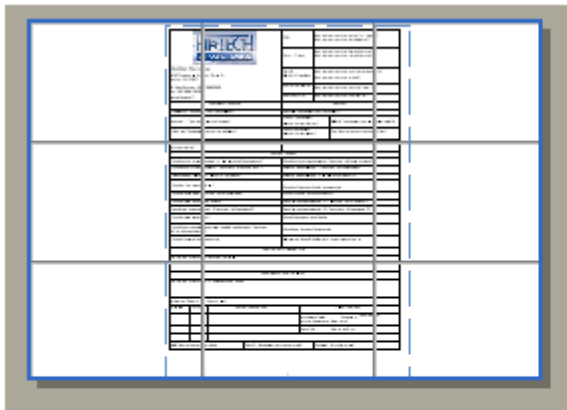
property ScaleImage: Boolean;

Posters are printed if PrintMode⁽²¹³⁾ = *srvpPoster*.

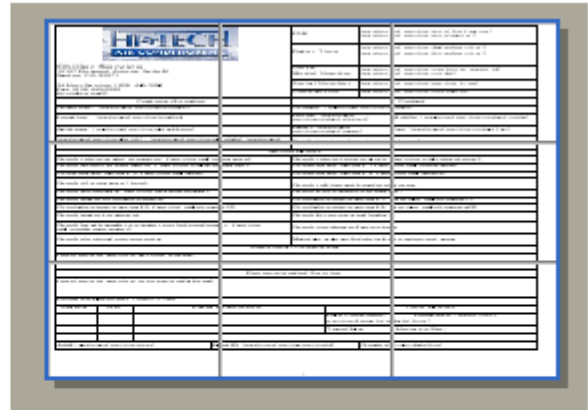
If ScaleImage = *True*, pages are scaled to the full poster size (RealWidth⁽²¹⁵⁾ will be PageWidthPix⁽²¹³⁾, RealHeight⁽²¹⁴⁾ will be PageHeightPix⁽²¹²⁾), without keeping the original proportions.

If ScaleImage = *False*, pages are scaled proportionally to the maximal size that fits in TotalFrameWPix⁽²¹⁶⁾ x TotalFrameHPix⁽²¹⁶⁾ (RealWidth⁽²¹⁵⁾ will be <=TotalFrameWPix⁽²¹⁶⁾, RealHeight⁽²¹⁴⁾ <=TotalFrameHPix⁽²¹⁶⁾).

ScaleImage



ScaleImage = False



ScaleImage = True

PageFormat = A1
Printer page format = A4
Orientation = poLandscape
FrameCountX = 3
FrameCountY = 3

ScaleImage Example

Default value:

False

1.3.7.1.23 TSRVPrint.SRichViewEdit

Specifies a link to the editor containing the document for printing.

property `SRichViewEdit: TSRichViewEdit(40);`

TSRVPrint component can print only if an editor is specified.

1.3.7.1.24 TSRVPrint.TotalFrameHPix

Returns `FrameCountY(208) * FrameHeightPix(208)`.

property `TotalFrameHPix: Extended;`

Posters are printed if `PrintMode(213) = srvpPoster`.

If `ScaleImage(215) = False`, pages of the source document are scaled proportionally to the maximal size that fits in `TotalFrameWPix(216) x TotalFrameHPix`.

1.3.7.1.25 TSRVPrint.TotalFrameWPix

Returns `FrameCountX(207) * FrameWidthPix(209)`.

property `TotalFrameWPix: Extended;`

Posters are printed if `PrintMode(213) = srvpPoster`.

If `ScaleImage(215) = False`, pages of the source document are scaled proportionally to the maximal size that fits in `TotalFrameWPix x TotalFrameHPix(216)`.

1.3.7.1.26 TSRVPrint.UsePhysicalOffsets

Specifies whether the component attempts to print on "physical margins".

property `UsePhysicalOffsets: Boolean;`

Physical margins are areas at the left, top, right and bottom sides of paper where the printer cannot print.

If *True*, TSRVPrint does not print on physical margins. It also affects how a number of frames for a poster is calculated. For example, you want to print A4 poster on A4 paper. If you set the current printer paper size to A4, and set `PageFormat(212)` to *srvfmA4*, you may expect that the poster will be printed on 1 paper sheet. But you will find that a poster is printed on 2 x 2 paper sheets, because `FrameCountX(207)` and `FrameCountY(208)` are greater than 1. It happens because the poster size is a pure A4 size, and the printer's page size is A4 minus physical offsets, so the poster does not fit the paper. If you want to define a poster size more precisely, assign `FrameCountX(207)` and `FrameCountY(208)` properties directly, instead of changing `PageFormat(212)`.

If *False*, TSRVPrint prints even on physical margins. In the example above (A4 poster on A4 paper), a poster will be printed on one paper sheet. But this value is highly not recommended for posters, because everything printed on these margins will be blank. This value is ok for non-poster printing (because usually document margins are larger than physical margins).

Default value:

True

1.3.7.2 Methods

In TSRVPrint

BeginUpdate⁽²¹⁷⁾
 CheckMargin⁽²¹⁷⁾
 EndUpdate⁽²¹⁸⁾
 OptimalOrientation⁽²¹⁸⁾
 PaperCodeToFormatPage⁽²¹⁸⁾
 Print⁽²¹⁸⁾
 PrintFrame⁽²¹⁹⁾
 PrintFrames⁽²¹⁹⁾
 PrintFramesEx⁽²²⁰⁾
 PrintPages⁽²²⁰⁾
 Update⁽²²¹⁾

1.3.7.2.1 TSRVPrint.BeginUpdate

Disables recalculation of parameters until EndUpdate⁽²¹⁸⁾ is called.

procedure BeginUpdate;

Each call to BeginUpdate must be followed by the call to EndUpdate⁽²¹⁸⁾. Calls may be nested.

Between calls of BeginUpdate and EndUpdate⁽²¹⁸⁾, assignments to TSRVPrint's properties do not invoke recalculations of other properties, and OnUpdateData⁽²²¹⁾ event is not called. For example, assignment to FrameCountX⁽²⁰⁷⁾ will not lead to recalculation of PageWidthPix⁽²¹³⁾.

1.3.7.2.2 TSRVPrint.CheckMargin

Compares margins with minimal possible values and returns adjusted values.

function CheckMargin(**var** NewLeft, NewTop, NewRight, NewBottom: Integer;
 ALeft, ATop, ARight, ABottom: Integer) : Boolean;

This methods compares the margins specified in **ALeft**, **ATop**, **ARight**, **ABottom** parameters with minimal margins supported by the current printer.

The results are written to **NewLeft**, **NewTop**, **NewRight**, **NewBottom** parameters. If the corresponding margin is too small, a minimal supported value is returned. Otherwise, the value of the original parameter is returned.

All values are in pixels.

Any of the input parameters (**ALeft**, **ATop**, **ARight**, **ABottom**) can be equal to -1. Such parameters are ignored.

This method can be used in TSRichViewEdit.OnMarginsChanged⁽¹¹¹⁾ event.

1.3.7.2.3 TSRVPrint.EndUpdate

Enables automatic recalculation of parameters that was disabled by BeginUpdate⁽²¹⁷⁾.

procedure EndUpdate;

Each call to BeginUpdate⁽²¹⁷⁾ must be followed by call to EndUpdate. Calls may be nested.

If EndUpdate is called as many times as BeginUpdate⁽²¹⁷⁾, the last call performs all necessary recalculation of properties (by calling Update⁽²²¹⁾).

1.3.7.2.4 TSRVPrint.OptimalOrientation

Returns an optimal frame orientation.

function OptimalOrientation : TPrinterOrientation;

Returns which value of Orientation⁽²¹¹⁾ would be the best (less count of frames in a poster).

Posters are printed if PrintMode⁽²¹³⁾ = *srvpPoster*.

1.3.7.2.5 TSRVPrint.PaperCodeToPageFormat

This method converts DMPAPER_*** values (defined in the Printers unit) to TSRVPageFormat⁽²⁷⁷⁾ values.

function PaperCodeToPageFormat(PaperSizeCode: Integer): TSRVPageFormat⁽²⁷⁷⁾;

PaperSizeCode must be one of constants like DMPAPER_A2, DMPAPER_A3, DMPAPER_A4, etc.

1.3.7.2.6 TSRVPrint.Print

Prints the whole document.

procedure Print(Title: **String**; Copies: Integer; Collate: Boolean;
PageSet: TRVPageSet = rvpsAll; Ascending: Boolean = True);

The method prints the document from SRichViewEdit⁽²¹⁶⁾ according to PrintMode⁽²¹³⁾.

The properties SRichViewEdit⁽²¹⁶⁾.MinPrintedItemNo⁽⁵⁶⁾ and MaxPrintedItemNo⁽⁵⁴⁾ are taken into account.

Title – document title for the Print Manager.

Copies – count of copies to print.

Collate – if more than one copy is printed, defines whether to print them together or not (if **Collate**, pages are printed like 1, 2, 3, 1, 2, 3. If not **Collate**, pages are printed like 1, 1, 2, 2, 3, 3).

PageSet – set of pages for printing (all/odd/even).

Ascending switches normal/reverse order of printing (*True*: 1, 2, 3; *False*: 3, 2, 1).

Print(Title, Copies, Collate, PageSet, Ascending) is equivalent to PrintPages⁽²²⁰⁾(1, SRichViewEdit⁽⁵⁷⁾.PageCount⁽⁵⁷⁾, Title, Copies, Collate, PageSet, Ascending).

About the grid printing mode (multiple pages on the same paper sheet)

Printing odd and even pages, and printing in reverse order are not supported if `PrintMode`⁽²¹³⁾ = `srvpGrid`.

About the poster printing mode (each page on the multiple paper sheets)

For posters (`PrintMode`⁽²¹³⁾ = `srvpPoster`), if **Collate** = `False`, printing order depends on the printer capabilities. In this mode, if **Collate** = `False`, pages are never collated. If the printer itself supports printing multiple copies, fragments of each page are not collated as well (page 1 fragment 1, page 1 fragment 1, page 1 fragment 2, page 1 fragment 2, then the same for page 2). But if the printer does not support multiple copies, TSRVPrint collates page fragments (page 1 fragment 1, page 1 fragment 2, page 1 fragment 1, page 1 fragment 2, then the same for page 2).

For posters, fragments of each page are always printed in normal order, **Ascending** may only reverse the order for pages of the source editor (`SRichViewEdit`⁽²¹⁶⁾). "Odd/even pages" means pages of the source editor; all fragments of pages specified in **PageSet** are always printed. Because of this, you cannot use the procedure described below for printing posters on both sides of paper sheets.

How to print on two sides of paper:

1. Print odd pages in normal order.
2. Flip the sheets over, then reinsert them into the printer. If page count is odd, do not insert the last page.
3. Print even pages in reverse order.

See also:

- `OnSendingToPrinter`⁽²²¹⁾

1.3.7.2.7 TSRVPrint.PrintFrame

Prints the specified frame of the specified page, in the current printing job.

function `PrintFrame(PageNo, FrameIndex : Integer) : Boolean;`

PageNo – index of the page (from 1).

FrameIndex – index of the frame to print, in range from 1 to `Ceil(FrameCountX`⁽²⁰⁷⁾
`*Ceil(FrameCountY`⁽²⁰⁸⁾`)`.

This method does not start a new printing job.

Posters are printed if `PrintMode`⁽²¹³⁾ = `srvpPoster`. In other modes, the method prints the specified page (and **FrameIndex** must be 1).

This method is rarely needed. The main printing methods are `Print`⁽²¹⁸⁾ and `PrintPages`⁽²²⁰⁾.

The method calls `OnSendingToPrinter`⁽²²¹⁾ (`SRichViewEdit`⁽⁴⁰⁾, **PageNo**, **FrameIndex**, *rvpsProceeding*).

Return value:

True if the frame was printed. *False* if **FrameIndex** is not in the proper range.

1.3.7.2.8 TSRVPrint.PrintFrames

Prints one page.

procedure `PrintFrames(Title: String; PageNo: Integer);`

Title – document title for the Print Manager.

PageNo – index of the page (from 1).

This method starts a new printing job, calls `PrintFramesEx`⁽²²⁰⁾, finishes the printing job.

This method is rarely needed. The main printing methods are `Print`⁽²¹⁸⁾ and `PrintPages`⁽²²⁰⁾.

1.3.7.2.9 TSRVPrint.PrintFramesEx

Prints one page in the current printing job.

```
procedure PrintFramesEx (PageNo: Integer);
```

PageNo – index of the page (from 1).

If `PrintMode`⁽²¹³⁾ = *srvpPoster*, the method prints all frames of this page by calling `PrintFrame`⁽²¹⁹⁾. For other modes, it prints the specified page.

This method is rarely needed. The main printing methods are `Print`⁽²¹⁸⁾ and `PrintPages`⁽²²⁰⁾.

1.3.7.2.10 TSRVPrint.PrintPages

Prints the specified range of pages.

```
procedure PrintPages (firstPgNo, lastPgNo: Integer; Title: String;  
  Copies: Integer; Collate: Boolean;  
  PageSet: TRVPageSet = rvpsAll; Ascending: Boolean = True);
```

The method prints the document from `SRichViewEdit`⁽²¹⁶⁾ according to `PrintMode`⁽²¹³⁾.

The properties `SRichViewEdit`⁽²¹⁶⁾.`MinPrintedItemNo`⁽⁵⁶⁾ and `MaxPrintedItemNo`⁽⁵⁴⁾ are taken into account.

firstPgNo – index of the first page to print (from 1).

lastPgNo – index of the last page to print (from 1).

Title – document title for the Print Manager.

Copies – count of copies to print.

PageSet – set of pages for printing (all/odd/even).

Ascending switches normal/reverse order of printing (*True*: 1, 2, 3; *False*: 3, 2, 1). Printing in reverse order is not supported if `PrintMode`⁽²¹³⁾ = *srvpGrid*.

Collate – if more than one copy is printed, defines whether to print them together or not (if **Collate**, pages are printed like 1, 2, 3, 1, 2, 3. If not **Collate**, pages are printed like 1, 1, 2, 2, 3, 3).

In the topic about the `Print`⁽²¹⁸⁾ method, you can read the notes about using these parameters in grid and poster modes.

How to print on two sides of paper:

1. Print odd pages in normal order.
2. Flip the sheets over, then reinsert them into the printer. If page count is odd, do not insert the last page.
3. Print even pages in reverse order.

See also:

- `OnSendingToPrinter`⁽²²¹⁾

1.3.7.2.11 TSRVPrint.Update

Performs calculations necessary for printing or previewing.

procedure Update;

Normally, all necessary calculations are performed automatically, when you modify properties of this component.

An explicit call is necessary in the following cases:

- the current printer or its properties were changed;
- a document in `SRichViewEdit`⁽²¹⁶⁾ was changed (and `AutoUpdate`⁽²⁰⁶⁾ = *False*, or this change cannot be detected by `TSRVPrint`).

This method is called by `EndUpdate`⁽²¹⁸⁾, when it enables recalculations disabled by `BeginUpdate`⁽²¹⁷⁾.

This method calls `OnUpdateData`⁽²²¹⁾ event.

1.3.7.3 Events

In TSRVPrint

- `OnSendingToPrinter`⁽²²¹⁾
- `OnUpdateData`⁽²²¹⁾

1.3.7.3.1 TSRVPrint.OnSendingToPrinter

Occurs on printing.

type

```
TSRVPrintingEvent = procedure (Sender: TSRichViewEdit;
    PageCompleted, FrameCompleted: Integer;
    Step: TRVPrintingStep) of object;
```

property OnSendingToPrinter: TSRVPrintingEvent;

This event is called by the printing methods.

The main printing methods are `Print`⁽²¹⁸⁾ and `PrintPages`⁽²²⁰⁾. Initially, they call this event with the following parameters: (`SRichViewEdit`⁽²¹⁶⁾, 0, 0, *rvpsStarting*). Then they call this event after printing each frame (non-poster modes assume one frame per page), with **Step**=*rvpsProceeding*. Finally, they call it with the parameters: (`SRichViewEdit`⁽²¹⁶⁾, 0, 0, *rvpsFinished*).

PageCompleted (index of page containing the completed frame, from 1) and **FrameCompleted** (index of the completed frame, from 1) are used if **Step**=*rvpsProceeding*.

See also:

- `TSRichViewEdit.OnPrinting`⁽¹¹⁶⁾

1.3.7.3.2 TSRVPrint.OnUpdateData

Occurs when `TSRVPrint`'s properties are recalculated according the the changes of other properties, changes in document in `SRichViewEdit`⁽²¹⁶⁾, or a call to `Update`⁽²²¹⁾.

property OnUpdateData: TNotifyEvent;

1.3.8 TSRVPreview

TSRVPreview displays a print preview for TSRVPrint⁽²⁰³⁾.

Unit SRVPreview.

Syntax

```
TSRVPreview = class (TCustomRVPrintPreview)
```

Hierarchy

TObject
TPersistent
TComponent
TControl
TWinControl
TCustomControl
TRVScroller
TCustomRVPrintPreview

Description

The source TSRVPrint⁽²⁰³⁾ component is specified in SRVPrint⁽²²⁶⁾ property. TSRVPreview displays a single page of the source document, specified in PageNo⁽²²⁵⁾ property. This page may consist of several frames (paper sheets), if a poster is printed.

When SRVPrint⁽²²⁶⁾ prints a poster, this component displays all frames (paper sheets) comprising this page. A border around all frames is shown using BorderColor⁽²²⁴⁾ and BorderWidth⁽²²⁴⁾ properties. Borders between frames are shown using FrameBorderColor⁽²²⁴⁾ and FrameBorderWidth⁽²²⁴⁾. A position of the original page is shown using PageBorderColor⁽²²⁵⁾, PageBorderWidth⁽²²⁵⁾, PageBorderStyle⁽²²⁵⁾ properties.

When SRVPrint prints in another mode, this component displays a single paper sheet. A border around it is shown using BorderColor⁽²²⁴⁾ and BorderWidth⁽²²⁴⁾ properties. A position of the original page is shown using PageBorderColor⁽²²⁵⁾, PageBorderWidth⁽²²⁵⁾, PageBorderStyle⁽²²⁵⁾ properties (one sheet may contain several pages, if tiles are printed).

Paper sheets inside the main border (drawn using BorderColor⁽²²⁴⁾ and BorderWidth⁽²²⁴⁾) drop a shadow with the parameters ShadowColor and ShadowWidth. The rest of the component's window is filled with Color.

1.3.8.1 Properties

In TSRVPreview

- BorderColor⁽²²⁴⁾
- BorderWidth⁽²²⁴⁾
- FrameBorderColor⁽²²⁴⁾
- FrameBorderWidth⁽²²⁴⁾
- PageBorderColor⁽²²⁵⁾
- PageBorderStyle⁽²²⁵⁾
- PageBorderWidth⁽²²⁵⁾
- PageNo⁽²²⁵⁾

■ SRVPrint ²²⁶

Derived from TCustomRVPrintPreview

- ClickMode
- Color
- MarginsPen
- PrintableAreaPen
- ShadowColor
- ShadowWidth
- ZoomInCursor
- ZoomMode
- ZoomOutCursor
- ZoomPercent

Derived from TRVScroller

- BorderStyle
- ▶ HScrollMax
- HScrollPos
- HScrollVisible
- ▶ InplaceEditor (not used in this component)
- NoVScroll
- Tracking
- UseXPThemes
- ▶ VScrollMax
- VScrollPos
- VScrollVisible
- WheelStep

Derived from TCustomControl

- Align
- Anchors
- Constraints
- Ctl3D
- Height
- HelpContext
- Hint
- ParentCtl3D
- PopupMenu
- ShowHint
- TabOrder
- TabStop
- Touch (D2010+)
- Visible
- Width

1.3.8.1.1 TSRVPreview.BorderColor

Specifies the color of border around the paper.

property BorderColor: TColor;

When previewing posters, this is the color of border around all frames (all paper sheets).

When previewing in other modes, this is the color of border around the paper sheet.

The border's width is BorderWidth⁽²²⁴⁾.

Default value:

clHighlight

1.3.8.1.2 TSRVPreview.BorderWidth

Specifies the width of border around the paper.

property BorderWidth: Integer;

When previewing posters, this is the width of border around all frames (all paper sheets).

When previewing in other modes, this is the width of border around the paper sheet.

The border's color is BorderColor⁽²²⁴⁾.

Default value:

2

1.3.8.1.3 TSRVPreview.FrameBorderColor

Specifies the color of borders between poster frames.

property FrameBorderColor: TColor;

When previewing posters, this is a color of lines between the poster's frames (paper sheets).

Not used in other modes.

The width of these lines is FrameBorderWidth⁽²²⁴⁾.

Default value:

clBtnShadow

1.3.8.1.4 TSRVPreview.FrameBorderWidth

Specifies the width of borders between poster frames.

property FrameBorderWidth: Integer;

When previewing posters, this is a width of lines between the poster's frames (paper sheets).

Not used in other modes.

The color of these lines is FrameBorderColor⁽²²⁴⁾.

Default value:

1

1.3.8.1.5 TSRVPreview.PageBorderColor

Specifies the color of border around the page of the source document.

property PageBorderColor: TColor;

This border shows the location of the page of the source document on the paper sheet(s).

When previewing a poster, this page can occupy several frames (paper sheets).

The width of this border is PageBorderWidth⁽²²⁵⁾, its style is PageBorderStyle⁽²²⁵⁾.

Default value:

\$00CD9165

1.3.8.1.6 TSRVPreview.PageBorderStyle

Specifies the style of border around the page of the source document.

property PageBorderStyle: TPenStyle;

This border shows the location of the page of the source document on the paper sheet(s).

When previewing a poster, this page can occupy several frames (paper sheets).

The width of this border is PageBorderWidth⁽²²⁵⁾, its color is PageBorderColor⁽²²⁵⁾.

Default value:

psDash

1.3.8.1.7 TSRVPreview.PageBorderWidth

Specifies the width of border around the page of the source document.

property PageBorderWidth: Integer;

This border shows the location of the page of the source document on the paper sheet(s).

When previewing a poster, this page can occupy several frames (paper sheets).

The style of this border is PageBorderStyle⁽²²⁵⁾, its color is PageBorderColor⁽²²⁵⁾.

Default value:

1

1.3.8.1.8 TSRVPreview.PageNo

Specifies the index of the page to display.

property PageNo: Integer;

This value must be in range 1..⁽²²⁶⁾SRVPrint⁽²²⁶⁾.SRichViewEdit⁽²¹⁶⁾.PageCount⁽⁵⁷⁾.

See also:

- First⁽²²⁶⁾
- Prev⁽²²⁶⁾
- Next⁽²²⁶⁾
- Last⁽²²⁶⁾

1.3.8.1.9 TSRVPreview.SRVPrint

Contains a link to a [TSRVPrint](#)⁽²⁰³⁾ component.

property `SRVPrint: TSRVPrint(203);`

This property provides a connection to [TSRVPrint](#)⁽²⁰³⁾ component.

TSRVPreview component can display a preview only if the this [TSRVPrint](#)⁽²⁰³⁾ component is ready.

1.3.8.2 Methods

In TSRVPreview

[First](#)⁽²²⁶⁾

[Last](#)⁽²²⁶⁾

[Next](#)⁽²²⁶⁾

[Prev](#)⁽²²⁶⁾

1.3.8.2.1 TSRVPreview.First

Switches the current page ([PageNo](#)⁽²²⁵⁾) to the first page.

procedure `First;`

1.3.8.2.2 TSRVPreview.Last

Switches the current page ([PageNo](#)⁽²²⁵⁾) to the last page.

procedure `Last;`

1.3.8.2.3 TSRVPreview.Next

Switches the current page ([PageNo](#)⁽²²⁵⁾) to the next page.

procedure `Next;`

1.3.8.2.4 TSRVPreview.Prev

Switches the current page ([PageNo](#)⁽²²⁵⁾) to the previous page.

procedure `Prev;`

1.3.8.3 Events

In TSRVPreview

■ [OnAfterPaint](#)⁽²²⁷⁾

■ [OnBeforePaint](#)⁽²²⁷⁾

Derived from TCustomRVPrintPreview

■ [OnHScrolled](#)

■ [OnVScrolled](#)

■ [OnZoomChanged](#)

Derived from TCustomControl

■ OnGesture (D2010+)

1.3.8.3.1 TSRVPreview.OnAfterPaint

Occurs when the component paints itself, after all the default painting.

property OnAfterPaint: TSRVOnPaint²⁶⁹;

This event allows drawing additional information in this component.

See also:

- OnBeforePaint²²⁷

1.3.8.3.2 TSRVPreview.OnBeforePaint

Occurs when the component paints itself, before painting the content of pages.

property OnBeforePaint: TSRVOnPaint²⁶⁹;

This event allows drawing additional information in this component.

This event allows drawing only beyond the area specified in **ARect** parameter, because any drawing inside this area will be overridden. This event may be considered redundant, because OnAfterPaint²²⁷ allows drawing both inside and outside of this area.

Since a clipping region is **ARect**, you need to create a new clipping region to draw outside **ARect**.

Example:

```
procedure TForm1.SRVPreview1BeforePaint(Sender: TObject; ACanvas: TCanvas;
  ARect: TRect);
var
  HRgn: Cardinal;
begin
  // creating a clipping region for drawing anywhere in SRVPreview1
  HRgn := CreateRectRgn(0, 0, SRVPreview1.Width, SRVPreview1.Height);
  SelectClipRgn(ACanvas.Handle, HRgn);
  // <drawing here>
  DeleteObject(HRgn);
  // restoring the original region
  HRgn := CreateRectRgn(ARect.Left, ARect.Top, ARect.Right, ARect.Bottom);
  SelectClipRgn(ACanvas.Handle, HRgn);
  DeleteObject(HRgn);
end;
```

1.3.9 TSRVSkinManager

TSRVSkinManager contains a collection of skins and allows applying them to visual components from the ScaleRichView package.

Unit SRVSkinManager.

Syntax

```
TSRVSkinManager = class (TComponent)
```

Hierarchy

TObject

TPersistent

TComponent

Description

This component can apply skins to:

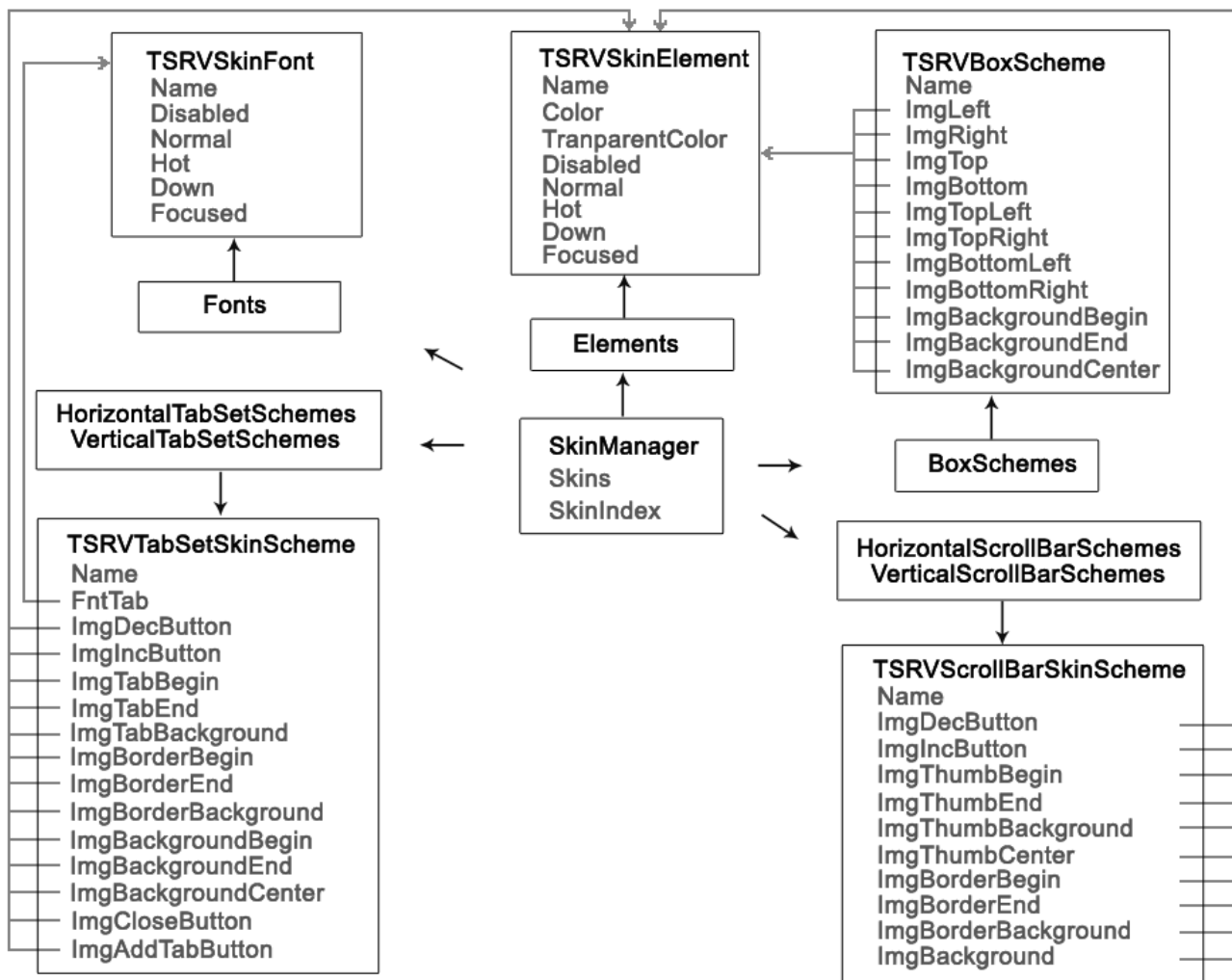
- scrollbars of `TSRichViewEdit`⁽⁴⁰⁾ and `TDBSRichViewEdit`⁽¹⁶⁸⁾;
- `TSRVScrollBar`⁽²⁴²⁾;
- `TSRVTabSet`⁽²⁴⁸⁾.

Properties

`Skins`⁽²²⁹⁾ is a collection of skins. `SkinIndex`⁽²²⁹⁾ allows choosing a skin from this collection. This skin is returned in the `CurrentSkin`⁽²²⁹⁾ property and applied to all controls linked to this skin manager.

Scheme

This scheme shows a skin manager component and its properties.



The main properties of each skin (i.e. of each item in Skins⁽²²⁹⁾ collection) are schemes:

- HorizontalScrollBarSchemes⁽²³¹⁾ – scheme for horizontal scrollbars;
- VerticalScrollBarSchemes⁽²³²⁾ – scheme for vertical scrollbars;
- HorizontalTabSetSchemes⁽²³¹⁾ – scheme for horizontal tab sets;
- VerticalTabSetSchemes⁽²³²⁾ – scheme for vertical tab sets;
- BoxSchemes⁽²³¹⁾ – scheme for rectangular areas of different controls.

Each scheme contains references to Elements⁽²³¹⁾ (collection of images) and Fonts⁽²³¹⁾ (collection of fonts).

1.3.9.1 Properties

In TSRVSkinManager

► CurrentSkin⁽²²⁹⁾
 ■ SkinIndex⁽²²⁹⁾
 ■ Skins⁽²²⁹⁾

1.3.9.1.1 TSRVSkinManager.CurrentSkin

Returns the active skin.

property CurrentSkin: TSRVSkin⁽²³⁰⁾;

If SkinIndex⁽²²⁹⁾ is in the range 0..Skins⁽²²⁹⁾.Count-1, this property returns Skins⁽²²⁹⁾[SkinIndex⁽²²⁹⁾]. Otherwise, it returns *nil*.

1.3.9.1.2 TSRVSkinManager.SkinIndex

Specifies the active skin.

property SkinIndex: Integer;

Zero or positive value defines the index in Skins⁽²²⁹⁾ collection.

-1 means default (no-skin) mode.

Default value:

-1

1.3.9.1.3 TSRVSkinManager.Skins

A collection of skins.

property Skins: TSRVSkinCollection⁽²³⁴⁾;

See also:

SkinIndex⁽²²⁹⁾

1.3.9.2 Classes of properties

Classes of TSRVSkinManager⁽²²⁷⁾ Properties

TSRVSkinCollection⁽²³⁴⁾ is collection of TSRVSkin⁽²³⁰⁾ items. This is a type of Skins⁽²²⁹⁾ property.

Classes of TSRVSkin⁽²³⁰⁾ Properties

TSRVSkin has basic properties of the following classes:

- TSRVSkinElementCollection⁽²³⁷⁾ – collection of TSRVSkinElement⁽²³⁶⁾ items; a type of Elements⁽²³¹⁾ property;
- TSRVSkinFontCollection⁽²³⁵⁾ – collection of TSRVSkinFont⁽²³⁴⁾ items; a type of Fonts⁽²³¹⁾ property;

All other properties of TSRVSkin contains indexes of items in the collections above (an index in Elements, if it needs to fill some area; index in Fonts, if it needs to draw some text).

TSRVSkin has scheme properties of the following classes:

- TSRVBoxSchemeCollection⁽²³³⁾ – collection of TSRVBoxScheme⁽²³²⁾ items; a type of BoxSchemes⁽²³¹⁾ property; schemes defining skins for different types of controls;
- TSRVScrollBarSkinSchemeCollection⁽²³⁹⁾ – collection of TSRVScrollBarSkinScheme⁽²³⁷⁾ items; a type of HorizontalScrollBarSchemes⁽²³¹⁾ and VerticalScrollBarSchemes⁽²³²⁾ properties; schemes defining skins for TSRVScrollBar⁽²⁴²⁾;
- TSRVTabSetSkinSchemeCollection⁽²⁴¹⁾ – collection of TSRVTabSetSkinScheme⁽²⁴⁰⁾ items; a type of HorizontalTabSetSchemes⁽²³¹⁾ and VerticalTabSetSchemes⁽²³²⁾ properties; schemes defining skins for TSRVTabSet⁽²⁴⁸⁾;

1.3.9.2.1 TSRVSkin

TSRVSkin contains properties defining a visual appearance of skinnable controls.

This is a class of items in the collection TSRVSkinManager⁽²²⁷⁾.Skins⁽²²⁹⁾.

Unit SRVSkinManager.

Syntax

```
TSRVSkin = class (TCollectionItem)
```

Hierarchy

TObject

TPersistent

TCollectionItem.

Properties

Name⁽²³²⁾ is the skin name.

Elements⁽²³¹⁾ is a collection of images, Fonts⁽²³¹⁾ is a collection of fonts.

Other properties are collections of "schemes". A scheme defines which Elements⁽²³¹⁾ and Fonts⁽²³¹⁾ are used for different parts of skinnable objects:

- HorizontalScrollBarSchemes⁽²³¹⁾ – for horizontal scrollbars;
- VerticalScrollBarSchemes⁽²³²⁾ – for vertical scrollbars;
- HorizontalTabSetSchemes⁽²³¹⁾ – for horizontal tab sets;
- VerticalTabSetSchemes⁽²³²⁾ – for vertical tab sets;
- BoxSchemes⁽²³¹⁾ – for rectangular areas of different controls.

1.3.9.2.1.1 Properties

In TSRVSkin

- BoxSchemes⁽²³¹⁾
- Elements⁽²³¹⁾
- Fonts⁽²³¹⁾
- HorizontalScrollBarSchemes⁽²³¹⁾
- HorizontalTabSetSchemes⁽²³¹⁾
- Name⁽²³²⁾
- VerticalScrollBarSchemes⁽²³²⁾
- VerticalTabSetSchemes⁽²³²⁾

Specifies which Elements⁽²³¹⁾ are used to draw specific parts of rectangular areas of different controls.

property BoxSchemes: TSRVBoxSchemeCollection⁽²³³⁾;

This is a collection of TSRVBoxScheme⁽²³²⁾ items. Each item has properties defining which Elements⁽²³¹⁾ are used to draw specific parts of a rectangular area of a control.

A collection of images used in skinnable objects.

property Elements: TSRVSkinElementCollection⁽²³⁷⁾;

See also

- TSRVSkinElement⁽²³⁶⁾ (class of items)

A collection of fonts used in skinnable objects.

property Fonts: TSRVSkinFontCollection⁽²³⁵⁾;

See also

- TSRVSkinFont⁽²³⁴⁾ (class of items)

Specifies which Elements⁽²³¹⁾ are used to draw specific parts of a horizontal scrollbar.

property HorizontalScrollBarSchemes: TSRVScrollBarSkinSchemeCollection⁽²³⁹⁾;

This is a collection of TSRVScrollBarSkinScheme⁽²³⁷⁾ items. Each item has properties defining which Elements⁽²³¹⁾ are used to draw specific parts of a horizontal scrollbar.

TSRVScrollBar⁽²⁴²⁾ has SkinSchemeIndex⁽²⁴⁶⁾ property specifying the index in this collection (used if TSRVScrollBar⁽²⁴²⁾.Kind = *sbHorizontal*).

TSRRichViewEdit⁽⁴⁰⁾ has HScrollBarSchemeIndex⁽⁵³⁾ property specifying the index in this collection for applying to horizontal scrollbar.

Specifies which Elements⁽²³¹⁾ are used to draw specific parts of a horizontal tab set control.

property HorizontalTabSetSchemes: TSRVTabSetSkinSchemeCollection⁽²⁴¹⁾;

This is a collection of TSRVTabSetSkinScheme⁽²⁴⁰⁾ items. Each item has properties defining which Elements⁽²³¹⁾ are used to draw specific parts of a horizontal tab set control.

TSRVTabSet⁽²⁴⁸⁾ has SkinSchemeIndex⁽²⁵⁶⁾ property specifying the index in this collection (used if TSRVTabSet⁽²⁴⁸⁾.Kind = *srvtskHorizontal*).

Specifies the skin name.

property Name: TRVUnicodeString;

This text is shown at design time when editing TSRVSkinManager⁽²²⁷⁾.Skins⁽²²⁹⁾ collection.

Specifies which Elements⁽²³¹⁾ are used to draw specific parts of a vertical scrollbar.

property VerticalScrollBarSchemes: TSRVScrollBarSkinSchemeCollection⁽²³⁹⁾;

This is a collection of TSRVScrollBarSkinScheme⁽²³⁷⁾ items. Each item has properties defining which Elements⁽²³¹⁾ are used to draw specific parts of a vertical scrollbar.

TSRVScrollBar⁽²⁴²⁾ has SkinSchemeIndex⁽²⁴⁶⁾ property specifying the index in this collection (used if TSRVScrollBar⁽²⁴²⁾.Kind = *sbVertical*).

TSRRichViewEdit⁽⁴⁰⁾ has VScrollBarSchemeIndex⁽⁶⁹⁾ property specifying the index in this collection for applying to vertical scrollbar.

Specifies which Elements⁽²³¹⁾ are used to draw specific parts of a vertical tab set control.

property VerticalTabSetSchemes: TSRVTabSetSkinSchemeCollection⁽²⁴¹⁾;

This is a collection of TSRVTabSetSkinScheme⁽²⁴⁰⁾ items. Each item has properties defining which Elements⁽²³¹⁾ are used to draw specific parts of a vertical tab set control.

TSRVTabSet⁽²⁴⁸⁾ has SkinSchemeIndex⁽²⁵⁶⁾ property specifying the index in this collection (used if TSRVTabSet⁽²⁴⁸⁾.Kind = *srvtskVertical*).

1.3.9.2.2 TSRVBoxScheme

TSRVBoxScheme is a class defining a visual appearance of rectangular areas of different controls.

This is a class of items in the following collection:

- TSRVSkinManager⁽²²⁷⁾.Skins⁽²²⁹⁾[i].BoxSchemes⁽²³¹⁾;

Unit SRVSkinManager.

Syntax

TSRVBoxScheme = **class** (TCollectionItem)

Hierarchy

TObject

TPersistent

TCollectionItem

Properties

Name: TRVUnicodeString – a name of this scheme. This text is displayed at design time when editing collections.

Other properties define indices in `TSRVSkinManager(227).Skins(229)[i].Elements(231)` collection. All these properties are Integers, their default value is -1.

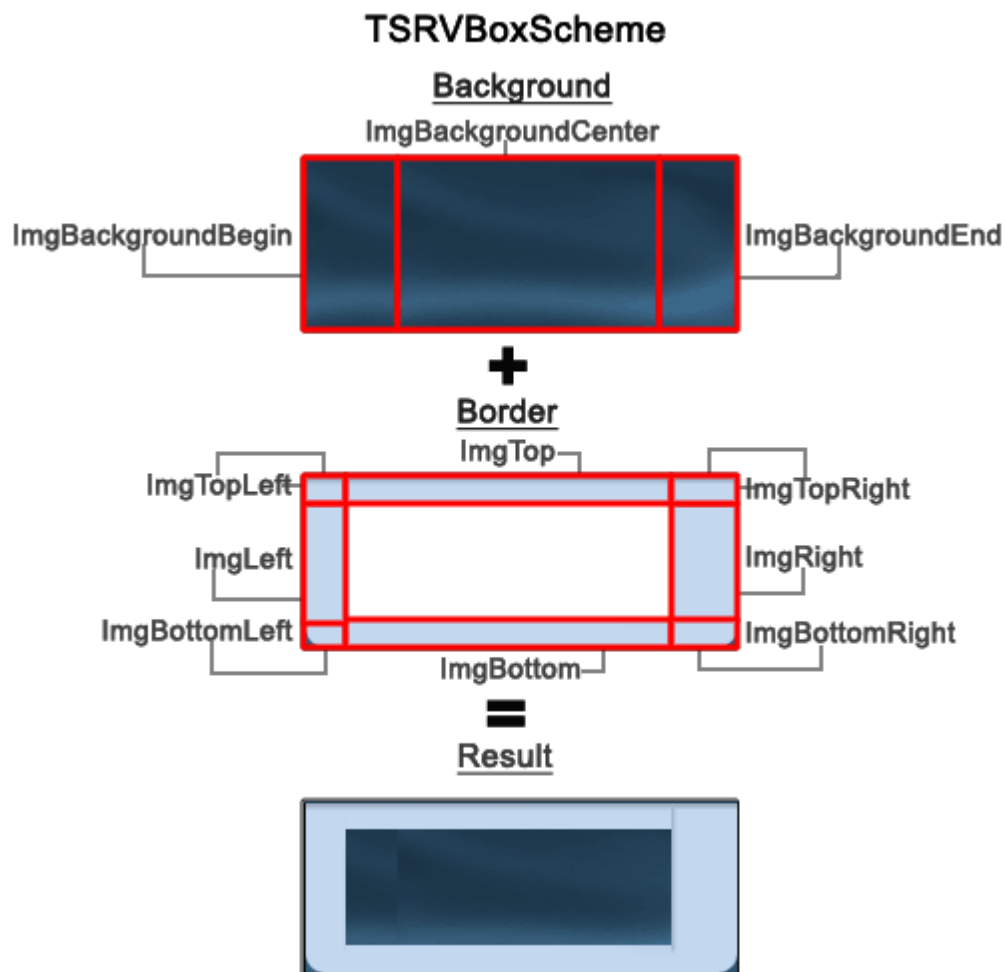
Background:

- **ImgBackgroundBegin** – left side;
- **ImgBackgroundEnd** – right side;
- **ImgBackgroundCenter** – middle of background (stretched).

Border:

- **ImgLeft, ImgRight, ImgTop, ImgBottom** – left/right/top/bottom sides (stretched);
- **ImgTopLeft, ImgTopRight, ImgBottomLeft, ImgBottomRight** – corners.

Example



1.3.9.2.3 TSRVBoxSchemeCollection

`TSRVBoxSchemeCollection` is a class of a collection of skin schemes for rectangular elements of different controls.

This is a class of the following collections:

- `TSRVSkinManager(227).Skins(229)[i].BoxSchemes(231)`;

Unit SRVSkinManager.

Syntax

```
TSRVBoxSchemeCollection = class (TCollection)
```

Hierarchy

TObject

TPersistent

TCollection

Properties

- **Items**[Index: Integer]: TSRVBoxScheme⁽²³²⁾;
- Count

1.3.9.2.4 TSRVSkinCollection

TSRVSkinCollection is a class of a collection of skins.

This is a class of the collection TSRVSkinManager⁽²²⁷⁾.Skins⁽²²⁹⁾.

Unit SRVSkinManager.

Syntax

```
TSRVSkinCollection = class (TCollection)
```

Hierarchy

TObject

TPersistent

TCollection

Properties

- **Items**[Index: Integer]: TSRVSkin⁽²³⁰⁾;
- Count

1.3.9.2.5 TSRVSkinFont

TSRVSkinFont is a class defining properties for fonts used in skinnable objects.

This is a class of items in the collection TSRVSkinManager⁽²²⁷⁾.Skins⁽²²⁹⁾[i].Fonts⁽²³¹⁾.

Unit SRVSkinManager.

Syntax

```
TSRVSkinFont = class (TCollectionItem)
```

Hierarchy

TObject

TPersistent

TCollectionItem

Properties

- **Name:** TRVUnicodeString – a caption of this item. This text is displayed at design time when editing TSRVSkinManager⁽²²⁷⁾.Skins⁽²²⁹⁾[i].Fonts⁽²³¹⁾ collection.
- **Disabled:** TSRVFont – a font used when the object is disabled (where TSRVFont = class(TFont)).
- **Normal:** TSRVFont – a font used when the object is in a normal state.
- **Hot:** TSRVFont – a font used when the object is below the mouse pointer.
- **Down:** TSRVFont – a font used when the object is clicked.
- **Focused:** TSRVFont – a font used when the object is in a focused state.

Example

TSRVSkinFont



1.3.9.2.6 TSRVSkinFontCollection

TSRVSkinFontCollection is a class of a collection of font items for applying to skinnable objects.

This is a class of the collection TSRVSkinManager⁽²²⁷⁾.Skins⁽²²⁹⁾[i].Fonts⁽²³¹⁾.

Unit SRVSkinManager.

Syntax

```
TSRVSkinFontCollection = class (TCollection)
```

Hierarchy

TObject

TPersistent

TCollection

Properties

- **Items**[Index: Integer]: TSRVSkinFont⁽²³⁴⁾;
- **Count**

1.3.9.2.7 TSRVSkinElement

TSRVSkinElement is a class defining properties for skinnable objects. Font properties are defined separately in TSRVSkinFont⁽²³⁴⁾.

This is a class of items in the collection TSRVSkinManager⁽²²⁷⁾.Skins⁽²²⁹⁾[i].Elements⁽²³¹⁾.

Unit SRVSkinManager.

Syntax

```
TSRVSkinElement = class (TCollectionItem)
```

Hierarchy

TObject

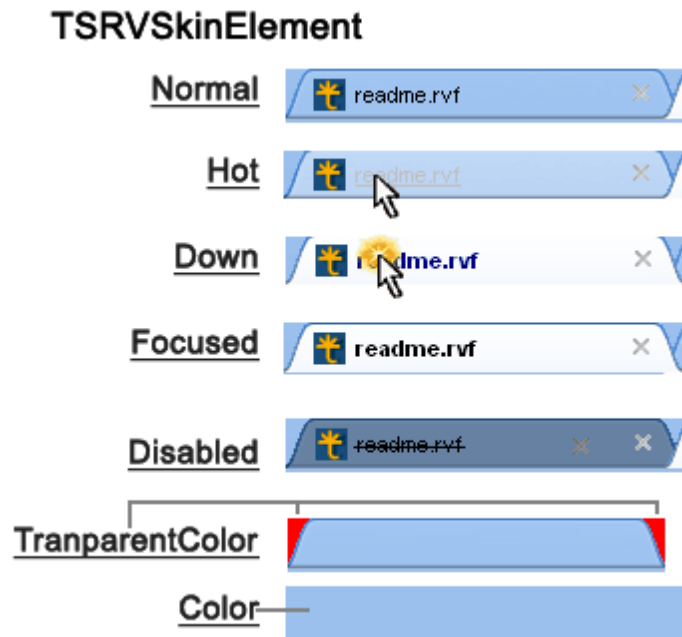
TPersistent

TCollectionItem

Properties

- **Name:** TRVUnicodeString – a caption of this item. This text is displayed at design time when editing TSRVSkinManager⁽²²⁷⁾.Skins⁽²²⁹⁾[i].Elements⁽²³¹⁾ collection.
- **Color:** TColor – a background color (usually used when images are not defined). Default value: *c/White*.
- **TransparentColor** – color of images that will be transparent when images are drawn. Used for Disabled, Normal, Hot, Down, Focused pictures, if they contain TBitmap. Default value: *c/None* (no bitmap transparency).
- **Disabled:** TPicture – a picture used when the object is disabled.
- **Normal:** TPicture – a picture used when the object is in a normal state.
- **Hot:** TPicture – a picture used when the object is below the mouse pointer.
- **Down:** TPicture – a picture used when the object is clicked.
- **Focused:** TPicture – a picture used when the object is in a focused state.

Example



1.3.9.2.8 TSRVSkinElementCollection

TSRVSkinElementCollection is a class of a collection of image items for applying to skinnable objects.

This is a class of the collection `TSRVSkinManager(227).Skins(229)[i].Elements(231)`.

Unit SRVSkinManager.

Syntax

```
TSRVSkinElementCollection = class (TCollection)
```

Hierarchy

TObject

TPersistent

TCollection

Properties

- **Items**[Index: Integer]: TSRVSkinElement⁽²³⁶⁾;
- **Count**

1.3.9.2.9 TSRVScrollBarSkinScheme

TSRVScrollBarSkinScheme is a class defining a visual appearance of scrollbars.

This is a class of items in the following collections:

- `TSRVSkinManager(227).Skins(229)[i].HorizontalScrollBarSchemes(231)`;
- `TSRVSkinManager(227).Skins(229)[i].VerticalScrollBarSchemes(232)`.

Unit SRVSkinManager.

Syntax

```
TSRVScrollBarSkinScheme = class (TCollectionItem)
```

Hierarchy

TObject

TPersistent

TCollectionItem

Properties

Name: TRVUnicodeString – a name of this scheme. This text is displayed at design time when editing collections.

Other properties define indices in `TSRVSkinManager(227).Skins(229)[i].Elements(231)` collection. All these properties are Integers, their default value is -1.

Buttons:

- **ImgDecButton** for the left/top scrollbar button;
- **ImgIncButton** for the right/bottom scrollbar button.

Thumbnail:

- **ImgThumbBegin** for the left/top side of thumbnail;
- **ImgThumbEnd** for the right/bottom side of thumbnail;
- **ImgThumbBackground** for the thumbnail background;
- **ImgThumbCenter** for drawing in the middle of thumbnail, above the `ImgThumbBackground`.

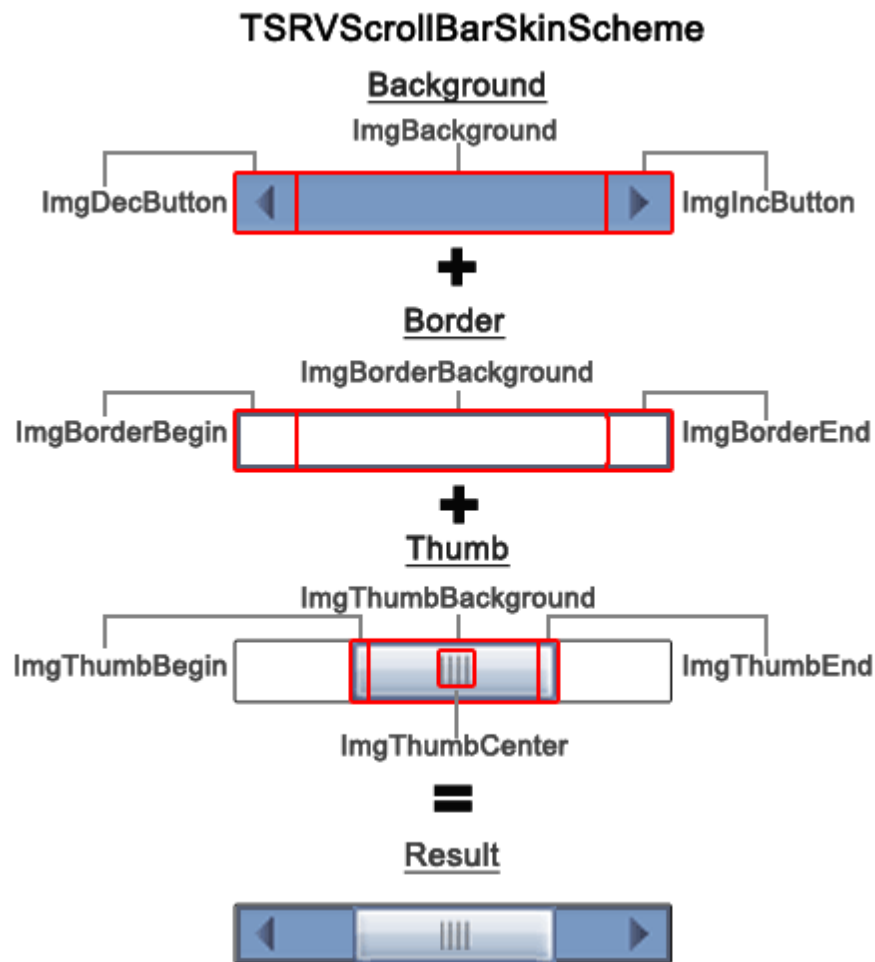
Border (the border is drawn above all other parts):

- **ImgBorderBegin** for the left/top side of the scrollbar border;
- **ImgBorderEnd** for the right/bottom side of the scrollbar border;
- **ImgBorderBackground** for the middle part of the scrollbar border (since the border is drawn after all other content, this image must have a transparent area in the middle).

Background:

- **ImgBackground** for the area between the buttons and the thumbnail.

Example



1.3.9.2.10 TSRVScrollBarSkinSchemeCollection

TSRVScrollBarSkinSchemeCollection is a class of a collection of skin schemes for scrollbars.

This is a class of the following collections:

- **TSRVSkinManager**⁽²²⁷⁾.**Skins**⁽²²⁹⁾[i].**HorizontalScrollBarSchemes**⁽²³¹⁾;
- **TSRVSkinManager**⁽²²⁷⁾.**Skins**⁽²²⁹⁾[i].**VerticalScrollBarSchemes**⁽²³²⁾.

Unit SRVSkinManager.

Syntax

```
TSRVScrollBarSkinSchemeCollection = class (TCollection)
```

Hierarchy

TObject

TPersistent

TCollection

Properties

- **Items**[Index: Integer]: **TSRVScrollBarSkinScheme**⁽²³⁷⁾;

- Count

1.3.9.2.11 TSRVTabSetSkinScheme

TSRVTabSetSkinScheme is a class defining a visual appearance of tab sets.

This is a class of items in the following collections:

- TSRVSkinManager⁽²²⁷⁾.Skins⁽²²⁹⁾[i].HorizontalTabSetSchemes⁽²³¹⁾;
- TSRVSkinManager⁽²²⁷⁾.Skins⁽²²⁹⁾[i].VerticalTabSetSchemes⁽²³²⁾.

Unit SRVSkinManager.

Syntax

```
TSRVTabSetSkinScheme = class (TCollectionItem)
```

Hierarchy

TObject

TPersistent

TCollectionItem

Properties

Name: TRVUnicodeString – a name of this scheme. This text is displayed at design time when editing collections.

FontTab: Integer specifies the index in the TSRVSkinManager⁽²²⁷⁾.Skins⁽²²⁹⁾[i].Fonts⁽²³¹⁾ collection. It defines the font for text in the tab. If you use a vertical tab, this font must be a TrueType/OpenType. The default value is -1.

Other properties define indices in TSRVSkinManager⁽²²⁷⁾.Skins⁽²²⁹⁾[i].Elements⁽²³¹⁾ collection. All these properties are Integers, their default value is -1.

Buttons on the tab set:

- **ImgDecButton** for the button to scroll to the beginning;
- **ImgIncButton** for the button to scroll to the end;
- **ImgAddTabButton** for the button for adding a new tab.

Tab:

- **ImgTabBegin** for the left/top side of tab;
- **ImgTabEnd** for the right/bottom side of tab;
- **ImgTabBackground** for the tab background;
- **ImgCloseButton** for a button for closing the tab.

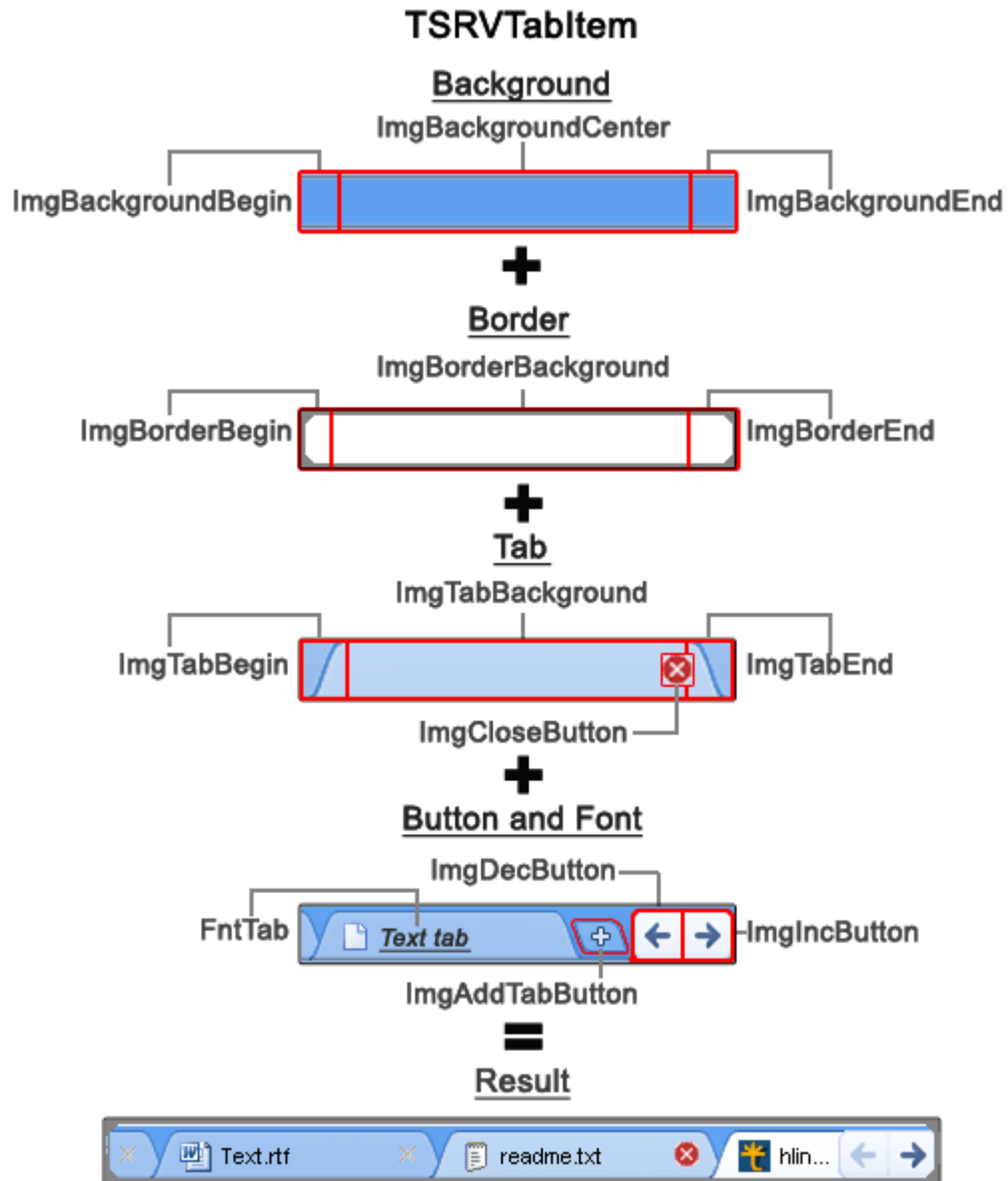
Border (the border is drawn above all other parts):

- **ImgBorderBegin** for the left/top side of the control border;
- **ImgBorderEnd** for the right/bottom side of the control border;
- **ImgBorderBackground** for the middle part of the control border (since the border is drawn after all other content, this image must have a transparent area in the middle).

Background:

- **ImgBackgroundBegin** for the left/top side of the control background;
- **ImgBackgroundEnd** for the right/bottom side of the control background;
- **ImgBackgroundCenter** for the middle part of the control background.

Example



1.3.9.2.12 TSRVTabSetSkinSchemeCollection

TSRVTabSetSkinSchemeCollection is a class of a collection of skin schemes for tab set controls.

This is a class of the following collections:

- `TSRVSkinManager(227).Skins(229)[i].HorizontalTabSetSchemes(231)`;
- `TSRVSkinManager(227).Skins(229)[i].VerticalTabSetSchemes(232)`.

Unit SRVSkinManager.

Syntax

```
TSRVTabSetSkinSchemeCollection = class (TCollection)
```

Hierarchy

TObject

TPersistent

TCollection

Properties

- **Items**[Index: Integer]: TSRVTabSetSkinScheme⁽²⁴⁰⁾;
- Count

1.3.10 TSRVScrollBar

TSRVScrollBar is a scrollbar, which is used to scroll the contents of a window, form, or control.

Unit SRVSBBar.

Syntax

```
TSRVScrollBar = class (TSRVCustomControl(334))
```

Hierarchy

TObject

TPersistent

TComponent

TControl

TWinControl

TCustomControl

TSRVCustomControl⁽³³⁴⁾

Description

This is a skinnable version of TScrollBar. These scrollbars are used in TSRichViewEdit⁽⁴⁰⁾.

Visual appearance

If skins are not defined, the scrollbar appearance depends on SRVControlStyle⁽²⁴⁷⁾ property:

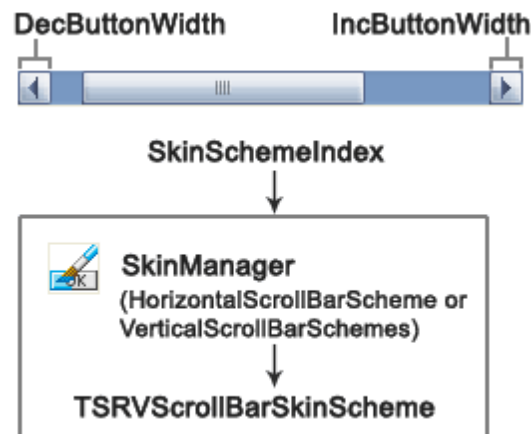
srvcsSimple (if RVControlsPainter.Theme = *rvctPaleBlue*):



srvcsClassic:

Appearance depends on UseXPThemes⁽²⁴⁷⁾, Flat⁽²⁴⁴⁾ and Color properties.

TSRVScrollBar is a skinnable component. Skins are used when SkinManager⁽²⁴⁶⁾ and SkinSchemeIndex⁽²⁴⁶⁾ properties are assigned. You can draw various parts of the control yourself using custom drawing events⁽²⁴⁸⁾.



If this control is not inserted in `TSRichViewEdit` ⁽⁴⁰⁾ control, we recommend to assign `True` to its `DoubleBuffered` property to prevent flickering.

1.3.10.1 Properties

In TSRVScrollBar

- `DecButtonWidth` ⁽²⁴⁴⁾
- `Flat` ⁽²⁴⁴⁾
- `FocusBlinkDelay` ⁽²⁴⁴⁾
- `IncButtonWidth` ⁽²⁴⁵⁾
- `Kind` ⁽²⁴⁵⁾
- `LargeChange` ⁽²⁴⁶⁾
- `PageSize` ⁽²⁴⁵⁾
- `Position` ⁽²⁴⁵⁾
- `ScrollingDelay` ⁽²⁴⁶⁾
- `SmallChange` ⁽²⁴⁶⁾
- `UseXPThemes` ⁽²⁴⁷⁾

Inherited from TSRVCustomControl ⁽³³⁴⁾

- `AutoDarkMode`
- `DarkMode`
- ▶ `MouseIn`
- `SkinManager` ⁽²⁴⁶⁾
- `SkinSchemeIndex` ⁽²⁴⁶⁾
- `SRVControlStyle` ⁽²⁴⁷⁾

Inherited from TCustomControl

- `Align`
- `Anchors`
- `BiDiMode`
- `Color`
- `Constraints`
- `DoubleBuffered`

- DragCursor
- DragKind
- DragMode
- Enabled
- Kind
- LargeChange
- Max
- Min
- ParentBiDiMode
- ParentCtl3D
- ParentShowHint
- PopupMenu
- ShowHint
- SmallChange
- TabOrder
- TabStop
- Visible

1.3.10.1.1 TSRVScrollBar.DecButtonWidth

Specifies width of the button for scrolling to the beginning.

property DecButtonWidth: TRVPixel96Length;

If value of this property is 0, a system default value is used. Values less than 5 are treated as 5.

This value is measured in pixels at 96 pixels per inch; the actual size scaled according to the current screen pixel density.

Default value

0

1.3.10.1.2 TSRVScrollBar.Flat

Makes the scrollbar "flat".

property Flat: Boolean;

If *True*, the scrollbar looks flat and does not has a frame around it.

This flat appearance is used if:

- Windows themes are not active (see UseXPThemes⁽²⁴⁷⁾), and
- a skin is not applied (see SkinManager⁽³³⁴⁾, SkinSchemeIndex⁽³³⁴⁾), and
- SRVControlStyle⁽²⁴⁷⁾ = *srvcsClassic*

Default value

False

1.3.10.1.3 TSRVScrollBar.FocusBlinkDelay

Specifies the delay for blinking when the scroll bar is focused.

property FocusBlinkDelay: Integer;

This property is used only in a skin mode, when a focused image is defined for the thumb tab.

Default value

500

1.3.10.1.4 TSRVScrollBar.IncButtonWidth

Specifies width of the button for scrolling to the end.

property IncButtonWidth: TRVPixel96Length;

If value of this property is 0, a system default value is used. Values less than 5 are treated as 5.

This value is used only in a skin mode.

This value is measured in pixels at 96 pixels per inch; the actual size scaled according to the current screen pixel density.

Default value

0

1.3.10.1.5 TSRVScrollBar.Kind

Specifies whether the scroll bar is horizontal or vertical.

property Kind: TScrollBarKind;

Set Kind to indicate the orientation of the scroll bar.

Possible values: *sbHorizontal*, *sbVertical*.

Default value

sbHorizontal

1.3.10.1.6 TSRVScrollBar.PageSize

Specifies the size of the thumb tab.

property PageSize: Integer;

PageSize is the size of the thumb tab, measured in the same units as Position⁽²⁴⁵⁾, Min, and Max (not pixels).

0 means the default size.

Default value

0

1.3.10.1.7 TSRVScrollBar.Position

Indicates the current position of the scroll bar.

property Position: Integer;

This value is in range Min..Max-PageSize⁽²⁴⁵⁾.

Default value

0

1.3.10.1.8 TSRVScrollBar.ScrollingDelay

Defines a scrolling speed.

property `ScrollingDelay: Integer;`

This is an interval between scrolling events, ms. Successive scrolling events occur, for example, when the user presses a scrolling button.

Default value

40

1.3.10.1.9 TSRVScrollBar.SkinManager

Determines which skin manager is associated with the scrollbar.

property `SkinManager: TSRVSkinManager(227);`

SkinManager defines a visual appearance of this control. If it is not specified, the control has a default (no-skin) view.

See also:

- SkinSchemeIndex⁽²⁵⁶⁾

1.3.10.1.10 TSRVScrollBar.SkinSchemeIndex

Determines which skin scheme is used for this scrollbar.

property `SkinSchemeIndex: Integer;`

If Kind=*sbHorizontal*, this is an index in SkinManager⁽²⁴⁶⁾.CurrentSkin⁽²²⁹⁾.HorizontalScrollBarSchemes⁽²³¹⁾ collection.

If Kind=*sbVertical*, this is an index in SkinManager⁽²⁴⁶⁾.CurrentSkin⁽²²⁹⁾.VerticalScrollBarSchemes⁽²³²⁾ collection.

If SkinManager⁽²⁴⁶⁾ = *nil*, or SkinManager⁽²⁴⁶⁾.CurrentSkin⁽²²⁹⁾ = *nil*, or this index is not valid for this collection (negative or too large), then this scrollbar has a default (no-skin) appearance.

Default value

0

1.3.10.1.11 TSRVScrollBar.SmallChange, LargeChange

The properties defines changes of the current position of the scroll bar.

property `SmallChange: TScrollBarInc;`

property `LargeChange: TScrollBarInc;`

SmallChange determines how much Position⁽²⁴⁵⁾ changes when the user clicks the arrow buttons on the scroll bar or presses the arrow keys on the keyboard.

LargeChange determines how much Position⁽²⁴⁵⁾ changes when the user clicks the scroll bar on either side of the thumb tab or presses PgUp or PgDn.

Default value

1

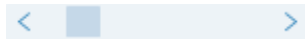
1.3.10.1.12 TSRVScrollBar.SRVControlStyle

Define appearance of this scrollbar.

property SRVControlStyle: TSRVControlStyle⁽²⁷⁵⁾;

This property is used if a skin is not assigned (see SkinManager⁽²⁴⁶⁾ and SkinSchemeIndex⁽²⁴⁶⁾).

srvcsSimple (an example view if RVControlsPainter.Theme = *rvctPaleBlue*):



srvcsClassic:

If UseXPThemes⁽²⁴⁷⁾ = *True*, the scrollbar uses VCL themes (if available) or Windows themes (if available). Otherwise, the scrollbar has 3d look. When the scrollbar is inserted in a TSVRichViewEdit⁽⁴⁰⁾ document, theme drawing is scaled not completely correctly. We recommend using *srvcsSimple* instead.

Default value:

srvcsClassic

This is the only control in ScaleRichView/RSRVControls that has **SRVControlStyle** = *srvcsClassic* by default.

1.3.10.1.13 TSRVScrollBar.UseXPThemes

Specifies whether the scrollbar uses VCL and Windows themes (visual styles), if they are available.

property UseXPThemes: Boolean;

VCL/Windows themes are used by the scrollbar if:

- it is not skinned (SkinManager⁽²⁴⁶⁾ = nil, or SkinManager⁽²⁴⁶⁾.CurrentSkin⁽²²⁹⁾ = nil, SkinSchemeIndex⁽²⁴⁶⁾ is not valid) and
- SRVControlStyle⁽²⁴⁷⁾ = *srvcsClassic*

If a VCL theme is active (in Delphi XE2 or newer), it is used for drawing. Otherwise, Windows theme is used.

Note: when this scrollbar is inserted in a TSVRichViewEdit⁽⁴⁰⁾ document, theme drawing is scaled not completely correctly. We recommend using SRVControlStyle⁽²⁴⁷⁾ = *srvcsSimple* instead.

Default value

True

1.3.10.2 Events

In TSRVScrollBar

- OnDrawArea⁽²⁴⁸⁾
- OnDrawBorder⁽²⁴⁸⁾
- OnDrawDecButton⁽²⁴⁸⁾
- OnDrawIncButton⁽²⁴⁸⁾
- OnDrawThumbTab⁽²⁴⁸⁾

Inherited from TScrollBar

- OnChange

- OnContextPopup
- OnDragDrop
- OnDragOver
- OnEndDock
- OnEndDrag
- OnEnter
- OnExit
- OnKeyDown
- OnKeyPress
- OnKeyUp
- OnScroll
- OnStartDock
- OnStartDrag

1.3.10.2.1 TSRVScrollBar.OnDraw...

Events for custom drawing of the scrollbar.

```

property OnDrawBorder: TSRVDrawScrollBarEvent267;
property OnDrawDecButton: TSRVDrawScrollBarEvent267;
property OnDrawIncButton: TSRVDrawScrollBarEvent267;
property OnDrawThumbTab: TSRVDrawScrollBarEvent267;
property OnDrawArea: TSRVDrawScrollBarAreaEvent267;

```

Use OnDrawArea to draw the scrollbar background (areas between the "inc/dec buttons" and the thumb tab).

Use OnDrawBackground to draw the control background. This event is called before all other custom drawing events.

Use OnDrawThumbTab to draw the thumb tab.

Use OnDrawDecButton to draw the button for scrolling to the beginning, and OnDrawIncButton to draw the button for scrolling to the end.

These events occur only if `SRVControlStyle247 = srvcsClassic`.

1.3.11 TSRVTabSet

The TSRVTabSet component presents horizontal or vertical tabs users can click to initiate actions.

Unit SRVTabSet.

Syntax

```
TSRVTabSet = class (TCustomControl);
```

Hierarchy

```

TObject
TPersistent
TComponent
TControl
TWinControl
TCustomControl

```

Tabs

You create a set of tabs for the tab set control when you specify a list of items as the value of the `Tabs` ⁽²⁵⁸⁾ property. One tab is created for each item.

To determine which tab is currently selected or to use code to select a tab, use the `TabIndex` ⁽²⁵⁷⁾ property. To find out which tab is the first/last visible tab in the tab set control use the `FirstTabIndex` ⁽²⁵²⁾/`LastTabIndex` ⁽²⁵³⁾ properties. To make the tab visible, use `ScrollToTab` ⁽²⁵⁸⁾.

The user can close tabs, if closing buttons are displayed. Properties of these buttons are specified in `CloseButton` ⁽²⁵²⁾ property.

Tabs can have icons from `ImageList` ⁽²⁵²⁾.

Layout

Tabs are displayed horizontally or vertically, depending on the value of `Kind` ⁽²⁵³⁾ property.

By default, tabs are positioned at the top (for a horizontal ruler) / the left (for a vertical ruler) of the control. To move them to bottom / right, assign `True` to `OppositeTabPosition` ⁽²⁵⁵⁾.

By default, left-to-right text is displayed from left to right (for a horizontal ruler) / from bottom to top (for a vertical ruler). To display it upside down / from top to bottom, assign `True` to `MirrorText` ⁽²⁵⁴⁾.

By default, tabs are ordered from left to right. Using `BiDiMode` property, you can order them from from right to left (`BiDiMode` does not affect tab order in a vertical ruler).

The following properties affect the size of tabs: `AutoWidth` ⁽²⁵¹⁾, `MinTabSize` ⁽²⁵⁴⁾, `MaxTabSize` ⁽²⁵⁴⁾, `TabHeight` ⁽²⁵⁷⁾.

The following properties affect the layout of each tab: `Indent` ⁽²⁵³⁾, `Margins` ⁽²⁵³⁾, `MirrorText` ⁽²⁵⁴⁾, `TextAlignH` ⁽²⁵⁸⁾, `TextAlignV` ⁽²⁵⁸⁾.

Visual appearance

If skins are not defined, the tab set appearance depends on `SRVControlStyle` ⁽²⁵⁷⁾ property:

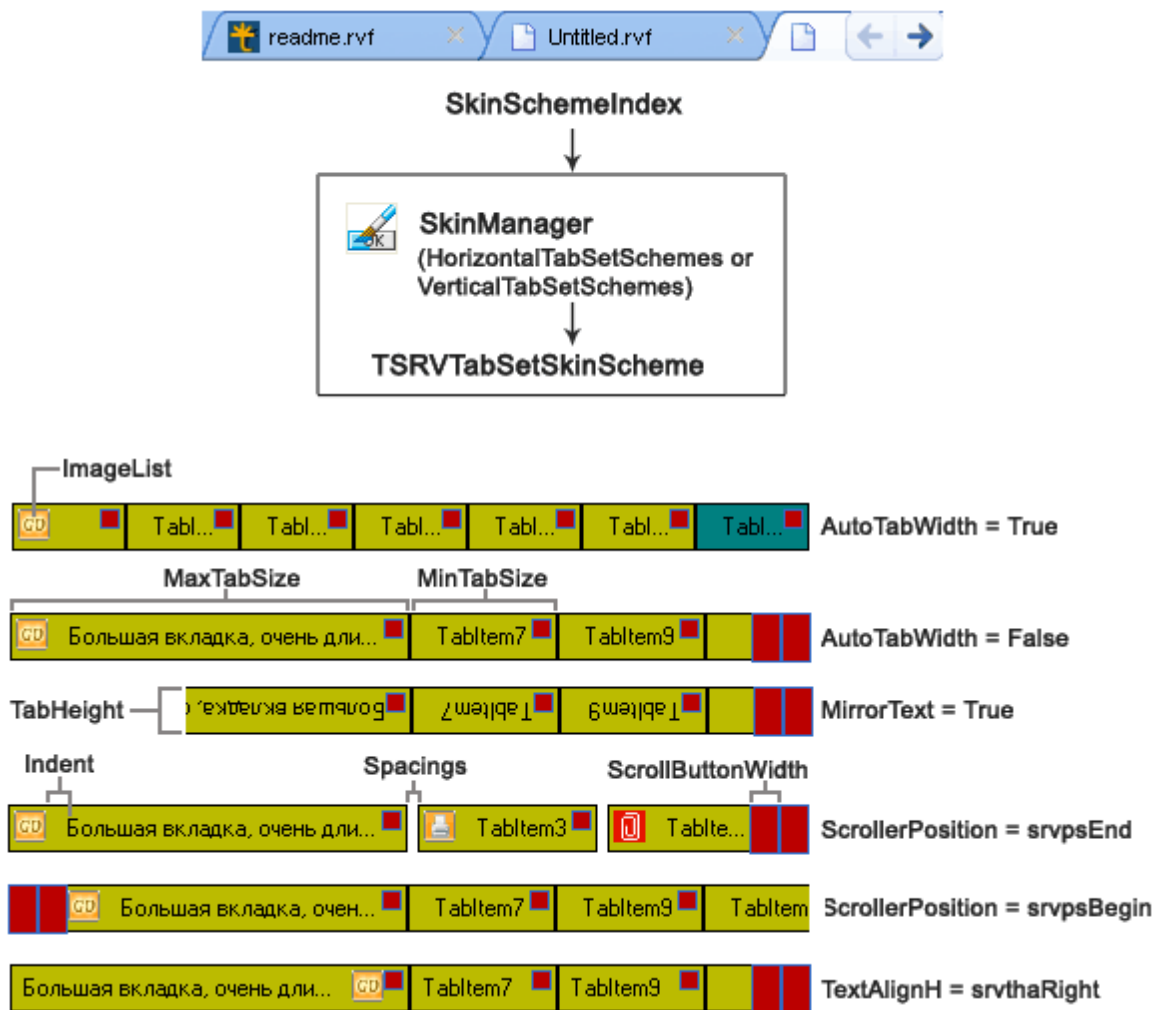
srvcsSimple (if `RVControlsPainter.Theme = rvctPaleBlue`):



srvcsClassic:



`TSRVTabSet` is a skinnable component. Skins are used when `SkinManager` ⁽²⁵⁶⁾ and `SkinSchemeIndex` ⁽²⁵⁶⁾ properties are assigned. You can draw various parts of the control yourself using custom drawing events ⁽²⁵⁹⁾.



Rearranging tabs

Tabs can be moved with the mouse, if `CanMoveTabs`⁽²⁵²⁾ is `True`. When moving, `OnMoveTab`⁽²⁶⁰⁾ event occurs.

1.3.11.1 Properties

In TSRVTabSet

- `AutoDarkMode`⁽²⁵²⁾
- `AutoTabWidth`⁽²⁵¹⁾
- `CanMoveTabs`⁽²⁵²⁾
- `CloseButton`⁽²⁵²⁾
- `DarkMode`⁽²⁵²⁾
- ▶ `FirstTabIndex`⁽²⁵²⁾
- `ImageList`⁽²⁵²⁾
- `Indent`⁽²⁵³⁾
- `Kind`⁽²⁵³⁾
- ▶ `LastTabIndex`⁽²⁵³⁾

- Margins ⁽²⁵³⁾
- MaxTabSize ⁽²⁵⁴⁾
- MinTabSize ⁽²⁵⁴⁾
- MirrorText ⁽²⁵⁴⁾
- OppositeTabPosition ⁽²⁵⁵⁾
- ScrollButtonWidth ⁽²⁵⁵⁾
- ScrollerMenu ⁽²⁵⁵⁾
- ScrollerPosition ⁽²⁵⁵⁾
- ScrollingDelay ⁽²⁵⁶⁾
- SkinManager ⁽²⁵⁶⁾
- SkinSchemeIndex ⁽²⁵⁶⁾
- Spacings ⁽²⁵⁶⁾
- SRVControlStyle ⁽²⁵⁷⁾
- TabHeight ⁽²⁵⁷⁾
- TabIndex ⁽²⁵⁷⁾
- Tabs ⁽²⁵⁸⁾
- TextAlignH ⁽²⁵⁸⁾
- TextAlignV ⁽²⁵⁸⁾

Inherited from TCustomControl

- Align
- Anchors
- Color
- Constraints
- Cursor
- Enabled
- Font
- Height
- Hint
- Left
- PopupMenu
- ShowHint
- TabStop
- Top
- Width

1.3.11.1.1 TSRVTabSet.AutoTabWidth

Specifies how widths of tabs are calculated.

property AutoTabWidth: Boolean;

If *True*, all tabs have widths wide enough to show their content (icon and text).

If *False*, widths of all tabs are calculated as $\text{SRVTabSet.Width} / \text{SRVTabSet.Tabs}^{(258)}.Count$.

In the both modes, MinTabSize ⁽²⁵⁴⁾ and MaxTabSize ⁽²⁵⁴⁾ are taken into account.

Default value

True

1.3.11.1.2 TSRVTabSet.CanMoveTabs

Allows rearranging tabs with the mouse.

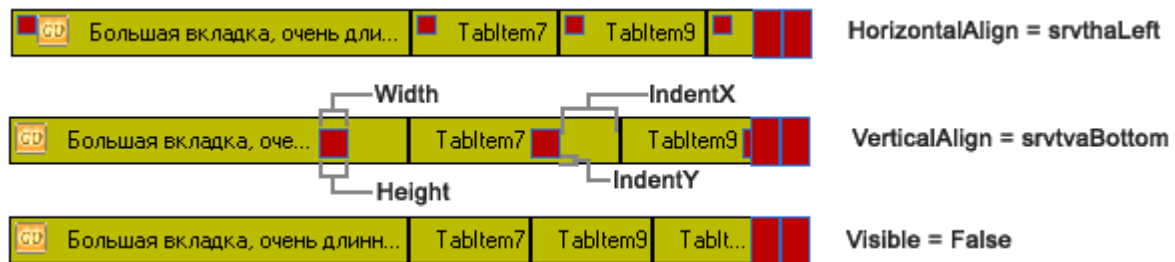
property CanMoveTabs: Boolean;

When moving tab, OnMoveTab⁽²⁶⁰⁾ event occurs (when two tabs are exchanged).

1.3.11.1.3 TSRVTabSet.CloseButton

Contains properties of tab closing buttons.

property CloseButton: TSRVCloseButton⁽²⁶¹⁾;



1.3.11.1.4 TSRVTabSet.DarkMode, AutoDarkMode

The property allow implementing a "dark mode".

property AutoTabWidth: Boolean;

These properties are used if SRVControlStyle⁽²⁵⁷⁾ = *srvcsSimple*.

If **AutoDarkMode** = *True*, and the control is inserted in a dark-mode editor (TSRichViewEdit⁽⁴⁰⁾ or TRichView), luminance of all colors is inverted.

Otherwise, **DarkMode** property is used (if *True*, luminance of all colors are inverted).

Default values:

AutoDarkMode: *True*

DarkMode: *False*

1.3.11.1.5 TSRVTabSet.FirstTabIndex

Returns the index of the first visible tab.

property FirstTabIndex: Integer;

See also:

LastTabIndex⁽²⁵³⁾

1.3.11.1.6 TSRVTabSet.ImageList

Determines the image list for drawing icons in tabs of the tab set control.

property ImageList: TCustomImageList;

Use **ImageList** to provide a customized list of bitmaps that can be displayed close to a tab's text. Individual tabs specify the image from this list that should appear by setting their ImageIndex⁽²⁶⁴⁾ property.

1.3.11.1.7 TSRVTabSet.Indent

Adds spacing before text in each tab.

property Indent: TRVPixel96Length;

If the tab has an icon, this is a distance between the icon and the text.

If the tab does not have an icon, this is a distance from the tab's Margin⁽²⁵³⁾ to text.

This value is measured in pixels at 96 pixels per inch; the actual size scaled according to the current screen pixel density.

Default value:

5

1.3.11.1.8 TSRVTabSet.Kind

Specifies whether the tab set control is horizontal or vertical.

type

```
TSRVTabSetKind = (srvtskHorizontal, srvtskVertical);
```

property Kind: TSRVTabSetKind;

A vertical tab is rotated by 90° counterclockwise from a horizontal position.

Default value:

srvtskHorizontal

See also:

- MirrorText⁽²⁵⁴⁾
- OppositeTabPosition⁽²⁵⁵⁾

1.3.11.1.9 TSRVTabSet.LastTabIndex

Returns the index of the last visible tab.

property LastTabIndex: Integer;

See also:

FirstTabIndex⁽²⁵²⁾

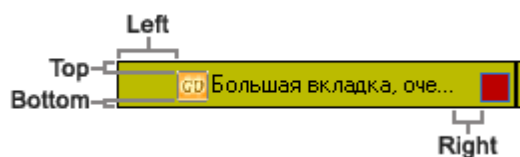
1.3.11.1.10 TSRVTabSet.Margins

Specifies margins for each tab.

property Margins: TSRVMargins;

The class TSRVMargins has Left, Right, Top, Bottom: TRVPixel96Length properties (0 by default).

These values are measured in pixels at 96 pixels per inch; the actual size scaled according to the current screen pixel density.



See also:

- TabHeight⁽²⁵⁷⁾
- Indent⁽²⁵³⁾

1.3.11.1.11 TSRVTabSet.MaxTabSize

Specifies the maximum allowed tab width.

property MaxTabSize: TRVPixel96Length;

For horizontal⁽²⁵³⁾ tab set controls, this property defines a maximum width of tabs.

For vertical⁽²⁵³⁾ tab set controls, this property defines a maximum height of tabs.

This value is measured in pixels at 96 pixels per inch; the actual size scaled according to the current screen pixel density.

Default value:

200

See also:

- AutoTabWidth⁽²⁵¹⁾
- MinTabSize⁽²⁵⁴⁾

1.3.11.1.12 TSRVTabSet.MinTabSize

Specifies the minimum allowed tab width.

property MinTabSize: TRVPixel96Length;

For horizontal⁽²⁵³⁾ tab set controls, this property defines a minimum width of tabs.

For vertical⁽²⁵³⁾ tab set controls, this property defines a minimum height of tabs.

This value is measured in pixels at 96 pixels per inch; the actual size scaled according to the current screen pixel density.

Default value:

0

See also:

- AutoTabWidth⁽²⁵¹⁾
- MaxTabSize⁽²⁵⁴⁾

1.3.11.1.13 TSRVTabSet.MirrorText

Allows flipping text in all tabs (vertically for horizontal tab set control, horizontally otherwise)

property MirrorText: Boolean;

Note: this property is named inaccurately. Actually, the text is not mirrored by rotated by 180°.

This property also inverts TextAlignH⁽²⁵⁸⁾, TextAlignV⁽²⁵⁸⁾, and the icons positions relative to texts.

Default value:

False

See also:

- Kind⁽²⁵³⁾

1.3.11.1.14 TSRVTabSet.OppositeTabPosition

Displays tabs at the opposite side of the tab set control.

property OppositeTabPosition: Boolean;

By default, tabs are located at the top side (for a horizontal ²⁵³ tab set) or at the left side (for a vertical ²⁵³ tab set) of the control. This property moves them to the opposite side (bottom / right).

Default value:

False

See also:

- Kind ²⁵³
- MirrorText ²⁵⁴

1.3.11.1.15 TSRVTabSet.ScrollButtonWidth

Defines the width of scrolling buttons.

property ScrollButtonWidth: TRVPixel96Length;

Scrolling buttons appear when not all tabs can be shown without scrolling.

For horizontal ²⁵³ tab set controls, this property defines a width of the buttons.

For vertical ²⁵³ tab set controls, this property defines a height of the buttons.

This value is measured in pixels at 96 pixels per inch; the actual size scaled according to the current screen pixel density.

Default value:

15

See also:

- ScrollerPosition ²⁵⁵

1.3.11.1.16 TSRVTabSet.ScrollerMenu

This property is reserved for future use.

property ScrollerMenu: TPopupMenu;

1.3.11.1.17 TSRVTabSet.ScrollerPosition

Defines the position of scrolling buttons.

type

```
TSRVScrollerPosition = (srvpsBegin, srvpsEnd);
```

property ScrollerPosition: TSRVScrollerPosition;

For horizontal ²⁵³ tab set controls, RTL BiDiMode inverts value of this property.

Scrolling buttons appear when not all tabs can be shown without scrolling.

Depending on this property, they appear before or after tabs.

1.3.11.1.18 TSRVTabSet.ScrollingDelay

Defines the tab scrolling speed.

property ScrollingDelay: Integer;

This property sets a pause between redrawing on scrolling, in ms.

Default value:

100

1.3.11.1.19 TSRVTabSet.SkinManager

Determines which skin manager is associated with the tab set control.

property SkinManager: TSRVSkinManager⁽²²⁷⁾;

SkinManager defines a visual appearance of this tab set control. If it is not specified, the control has a default (no-skin) view.

See also:

- SkinSchemeIndex⁽²⁵⁶⁾

1.3.11.1.20 TSRVTabSet.SkinSchemeIndex

Determines which skin scheme is used for this tab set control.

property SkinSchemeIndex: Integer;

If Kind⁽²⁵³⁾ = *srvtSkHorizontal*, this is an index in SkinManager⁽²⁵⁶⁾.CurrentSkin⁽²²⁹⁾.HorizontalTabSetSchemes⁽²³¹⁾ collection.

If Kind⁽²⁵³⁾ = *srvtSkVertical*, this is an index in SkinManager⁽²⁵⁶⁾.CurrentSkin⁽²²⁹⁾.VerticalTabSetSchemes⁽²³²⁾ collection.

If SkinManager⁽²⁵⁶⁾ = *nil*, or SkinManager⁽²⁵⁶⁾.CurrentSkin⁽²²⁹⁾ = *nil*, or this index is not valid for this collection (negative or too large), then this tab set control has a default (no-skin) appearance.

Individual tabs can override this property with their own SkinSchemeIndex⁽²⁶⁵⁾.

Default value

0

1.3.11.1.21 TSRVTabSet.Spacings

Specifies a distance between tabs.

property Spacings: TRVPixel96Length;

This value can be negative, if you want tabs to overlap.

This value is measured in pixels at 96 pixels per inch; the actual size scaled according to the current screen pixel density.

Default value:

0

1.3.11.1.22 TSRVTabSet.SRVControlStyle

Define appearance of this tab set.

property `SRVControlStyle: TSRVControlStyle(275);`

This property is used if a skin is not assigned (see `SkinManager(256)` and `SkinSchemeIndex(256)`).

srvcsSimple (if `RVControlsPainter.Theme = rvctPaleBlue`):



srvcsClassic:



Default value:

srvcsSimple

1.3.11.1.23 TSRVTabSet.TabHeight

Defines the height of tabs.

property `TabHeight: TRVPixel96Length;`

For horizontal⁽²⁵³⁾ tab set controls, this property defines the height of tabs.

For vertical⁽²⁵³⁾ tab set controls, this property defines the widths of tabs.

This value is measured in pixels at 96 pixels per inch; the actual size scaled according to the current screen pixel density.

See also:

- `Margins(253)`
- `Indent(253)`
- `TextAlignV(258)`

1.3.11.1.24 TSRVTabSet.TabIndex

Determines which tab of a tab set control is currently selected.

property `TabIndex: Integer;`

The value of `TabIndex` can be from -1 to one less than the number of `Tabs(258)`. `TabIndex` is an index in the `Tabs(258)` collection property: a value of 0 indicates the first tab in the tab set control, 1 is the second tab, and so on. If no tabs are selected, `TabIndex` has a value of -1.

When a value is assigned to `TabIndex`, `OnChange(259)` event occurs, just as if the user had clicked on a tab.

1.3.11.1.25 TSRVTabSet.Tabs

Contains the list of tabs displayed in the tab set control.

property Tabs: TSRVTabCollection⁽²⁶⁶⁾;

Properties of each tab is defined in TSRVTabItem⁽²⁶³⁾ class.

1.3.11.1.26 TSRVTabSet.TextAlignH

Defines a horizontal alignment of text and icon in each tab.

property TextAlignH: TSRVTHAlign⁽²⁸⁰⁾;

Meaning of this property is not changed if Kind⁽²⁵³⁾ = *svtskVertical*: this property always defines a horizontal alignment.

MirrorText⁽²⁵⁴⁾ inverts value of this property. For a horizontal ruler, RTL BiDiMode inverts it as well.

Default value:

svthaLeft

See also:

TextAlignV⁽²⁵⁸⁾

1.3.11.1.27 TSRVTabSet.TextAlignV

Defines a vertical alignment of text and icon in each tab.

property TextAlignV: TSRVTVAlign⁽²⁸⁰⁾;

Meaning of this property is not changed if Kind⁽²⁵³⁾ = *svtskVertical*: this property always defines a vertical alignment.

MirrorText⁽²⁵⁴⁾ inverts value of this property. For a vertical ruler, RTL BiDiMode inverts it as well.

Default value:

svtvaCenter

See also:

TextAlignH⁽²⁵⁸⁾

TabHeight⁽²⁵⁷⁾

1.3.11.2 Methods

In TSRVTabSet

ScrollToTab⁽²⁵⁸⁾

1.3.11.2.1 TSRVTabSet.ScrollToTab

Scrolls tabs in the control to make Tabs⁽²⁵⁸⁾ [TabIndex] visible.

procedure ScrollToTab (TabIndex : Integer);

1.3.11.3 Events

In TSRVTabSet

- OnChange ⁽²⁵⁹⁾
- OnCloseTab ⁽²⁵⁹⁾
- OnDrawBackground ⁽²⁵⁹⁾
- OnDrawBorder ⁽²⁵⁹⁾
- OnDrawCloseButton ⁽²⁵⁹⁾
- OnDrawDecButton ⁽²⁵⁹⁾
- OnDrawIncButton ⁽²⁵⁹⁾
- OnDrawTab ⁽²⁵⁹⁾
- OnMoveTab ⁽²⁶⁰⁾

Inherited from TCustomControl

- OnClick
- OnEnter
- OnExit
- OnMouseDown
- OnMouseMove
- OnMouseUp
- OnKeyDown
- OnKeyUp

1.3.11.3.1 TSRVTabSet.OnChange

Occurs after a new tab is selected.

property OnChange: TNotifyEvent;

This event occurs when value of TabIndex ⁽²⁵⁷⁾ property is changed.

1.3.11.3.2 TSRVTabSet.OnCloseTab

Occurs when the user attempts to close a tab.

type

```
TSRVCloseTabEvent = procedure (Sender: TSRVTabSet (248); TabIndex: Integer;
    var CanClose : Boolean) of object;
```

property OnCloseTab: TSRVCloseTabEvent;

Parameters

TabIndex – the index of the tab to close (in the Tabs ⁽²⁵⁸⁾ collection).

Set **CanClose** to *False* to prevent closing.

When a tab is closed, its object is removed from Tabs ⁽²⁵⁸⁾ and freed.

1.3.11.3.3 TSRVTabSet.OnDraw...

Events for custom drawing of the tab set control.

property OnDrawBackground: TSRVDrawTabSetEvent ⁽²⁶⁹⁾;

property OnDrawBorder: TSRVDrawTabSetEvent ⁽²⁶⁹⁾;

```

property OnDrawDecButton: TSRVDrawTabSetEvent269;
property OnDrawIncButton: TSRVDrawTabSetEvent269;
property OnDrawTab: TSRVDrawTabEvent268;
property OnDrawCloseButton: TSRVDrawTabEvent268;

```

Drawing the tab set control

Use OnDrawBackground to draw the control background. This event is called before all other custom drawing events.

Use OnDrawBorder to draw the control border. This event is called after all other custom drawing events.

Use OnDrawDecButton to draw the button for scrolling tabs to the beginning, and OnDrawIncButton to draw the button for scrolling tabs to the end.

Drawing tabs

Use OnDrawTab to draw a tab, and OnDrawCloseButton to draw a tab closing button.

1.3.11.3.4 TSRVTabSet.OnMoveTab

Occurs when two tabs exchange their places as a result of tab moving.

```

type
  TSRVTabMoveEvent = procedure (Sender: TSRVTabSet248; NewTabIndex, OldTabIndex : Integer);
property OnTabMove: TSRVTabMoveEvent;

```

NewTabIndex, OldTabIndex – indexes of exchanged tabs (in the Tabs²⁵⁸ collection).

(this event can be interpreted as moving a tab from **OldTabIndex** to **NewTabIndex** position; but since this event occurs only for adjacent tabs, it is convenient to think about it as about exchanging places).

This event occurs only when tab rearranging is allowed, see CanMoveTabs²⁵² property.

Example

Assuming that each tab corresponds to one item in Editors array. When moving tabs, we need to update this array:

```

procedure TForm3.SRVTabSet1TabMove(Sender: TSRVTabSet248; NewTabIndex,
  OldTabIndex: Integer);
var
  tmp: TSRichViewEdit40;
begin
  tmp := Editors[OldTabIndex];
  Editors[OldTabIndex] := Editors[NewTabIndex];
  Editors[NewTabIndex] := tmp;
end;

```

1.3.11.4 Classes of properties

Classes of TSRVTabSet²⁴⁸ properties

- TSRVTabCollection²⁶⁶ – collection of TSRVTabItem²⁶³ items; a type of Tabs²⁵⁸ property;
- TSRVCloseButton²⁶¹ – properties of a tab closing button.

1.3.11.4.1 TSRVCloseButton

TSRVCloseButton contains properties of closing buttons that can be displayed on all tabs of TSRVTabSet⁽²⁴⁸⁾ or TSRVImagesScroll⁽³⁵³⁾.

Unit SRVTabSet.

Syntax

```
TSRVCloseButton = class (TPersistent)
```

Hierarchy

TObject

TPersistent

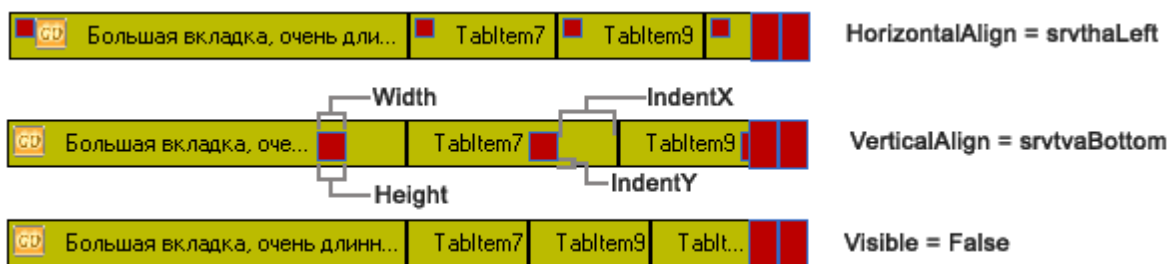
Properties

Closing buttons are shown if Visible⁽²⁶³⁾ = *True*. If the tab set control has only a single tab, a closing button is displayed only if AllowCloseLastTab⁽²⁶²⁾ = *True*.

Closing buttons can be shown either on all tabs, or only on a highlighted tab, or only on an active tab, depending on Visibility⁽²⁶³⁾.

Positions of closing buttons are defined with the properties HorizontalAlign⁽²⁶²⁾, VerticalAlign⁽²⁶³⁾, IndentX⁽²⁶²⁾, IndentY⁽²⁶²⁾.

Size of closing buttons is defined with the properties Width⁽²⁶³⁾ and Height⁽²⁶²⁾.



1.3.11.4.1.1 Properties

In TSRVCloseButton

- AllowCloseLastTab⁽²⁶²⁾
- Height⁽²⁶²⁾
- HorizontalAlign⁽²⁶²⁾
- IndentX⁽²⁶²⁾
- IndentY⁽²⁶²⁾
- VerticalAlign⁽²⁶³⁾
- Visibility⁽²⁶³⁾
- Visible⁽²⁶³⁾
- Width⁽²⁶³⁾

Specifies whether a closing button is displayed on the last tab (of `TSRVTabSet`⁽²⁴⁸⁾ control) or on the last image (of `TSRVImagesScroll`⁽³⁵³⁾ control).

property `AllowCloseLastTab`: Boolean;

If *True* (and `Visible`⁽²⁶³⁾ is *True* too), a closing button is displayed even if the control contains a single tab / image.

Default value:

True

Specifies the height of the closing button.

property `Height`: `TRVPixel96Length`;

This value is measured in pixels at 96 pixels per inch; the actual size scaled according to the current screen pixel density.

Default value:

10

Specifies the horizontal position of the tab closing button.

property `HorizontalAlign`: `TSRVTHAlign`⁽²⁸⁰⁾;

Meaning of this property is not changed if the tab set's `Kind`⁽²⁵³⁾ = *srvtskVertical*: this property always defines a horizontal position.

For a horizontal ruler, RTL tab set's `BiDiMode` inverts value of this property.

Default value:

srvthaRight

Specifies the horizontal indent added to the button position.

property `IndentX`: `TRVPixel96Length`;

If `HorizontalAlign`⁽²⁶²⁾ = *srvthaRight*, the button is shifted to the left by this value. Otherwise, it is shifted to the right.

This value is measured in pixels at 96 pixels per inch; the actual size scaled according to the current screen pixel density.

Default value:

3

Specifies the vertical indent added to the button position.

property `IndentY`: `TRVPixel96Length`;

If `VerticalAlign`⁽²⁶³⁾ = *srvthaBottom*, the button is shifted to the top by this value. Otherwise, it is shifted to the bottom.

This value is measured in pixels at 96 pixels per inch; the actual size scaled according to the current screen pixel density.

Default value:

3

Specifies the vertical position of the tab closing button.

property VerticalAlign: TSRVTVAlign⁽²⁸⁰⁾;

Meaning of this property is not changed if the tab set's Kind⁽²⁵³⁾ = *srvtSkVertical*: this property always defines a vertical position.

For a vertical ruler, RTL tab set's BiDiMode inverts value of this property.

Default value:

srvtvaTop

Specifies when tab closing buttons are displayed.

type

TSRVTSCloseButtonVisibility = (srvcbvAlways, srvcbvHot, srvcbvActive);

property Visibility: TSRVTSCloseButtonVisibility;

Closing buttons are displayed only if Visible⁽²⁶³⁾ is *True*.

Visibility	Meaning
<i>srvcbvAlways</i>	Buttons are displayed on all tabs
<i>srvcbvHot</i>	A button is displayed only on the tab below the mouse pointer
<i>srvcbvActive</i>	A button is displayed only on the active tab (Tabs ⁽²⁵⁸⁾ [TabIndex ⁽²⁵⁷⁾])

Default value:

srvcbvAlways

Specifies whether tabs have closing buttons.

property Visible: Boolean;

Default value:

True

Specifies the width of the closing button.

property Width: TRVPixel96Length;

This value is measured in pixels at 96 pixels per inch; the actual size scaled according to the current screen pixel density.

Default value:

10

1.3.11.4.2 TSRVTabItem

TSRVTabItem is an individual tab in TSRVTabSet⁽²⁴⁸⁾ control.

This is a class of items in the collection TSRVTabSet⁽²⁴⁸⁾.Tabs⁽²⁵⁸⁾.

Unit SRVTabSet.

Syntax

```
TSRVTabItem = class (TCollectionItem)
```

Hierarchy

TObject

TPersistent

TCollectionItem.

Properties

The tab has Text⁽²⁶⁵⁾ and an optional icon, defined in ImageIndex⁽²⁶⁴⁾.

Normally, in a skin mode, appearance of all tabs is defined in the tab set control. You can override these settings with SkinSchemeIndex⁽²⁶⁵⁾ and/or SkinFontIndex⁽²⁶⁴⁾ properties.

The tab can be hidden (Visible⁽²⁶⁵⁾ property) or disabled (Enabled⁽²⁶⁴⁾ property).

1.3.11.4.2.1 Properties

In TSRVTabItem

- Enabled⁽²⁶⁴⁾
- ImageIndex⁽²⁶⁴⁾
- SkinFontIndex⁽²⁶⁴⁾
- SkinSchemeIndex⁽²⁶⁵⁾
- Tag⁽²⁶⁵⁾
- Text⁽²⁶⁵⁾
- Visible⁽²⁶⁵⁾

Controls whether the tab responds to mouse and keyboard events.

```
property Enabled: Boolean;
```

Default value:

True

Determines which image is displayed as the icon for the tab.

```
property ImageIndex: Integer;
```

Set ImageIndex to associate the tab with one of the images in the ImageList⁽²⁵²⁾ property of the tab set control.

Default value:

-1 (no image)

Specifies which font is used for this tab in the skin mode.

```
property SkinFontIndex: Integer;
```

In a no-skin mode, a tab's font is defined in the Font property of the tab set control.

In a skin mode, a tab's fonts are defined in TabSet.SkinManager⁽²⁵⁶⁾.CurrentSkin⁽²²⁹⁾.Fonts⁽²³¹⁾ [FontIndex],

1.3.11.4.3 TSRVTabCollection

TSRVTabCollection maintains the collection of tabs that appear in TSRVTabSet⁽²⁴⁸⁾ control.

This is a class of TSRVTabSet⁽²⁴⁸⁾.Tabs⁽²⁵⁸⁾ collection.

Unit SRVTabSet.

Syntax

```
TSRVTabCollection = class (TCollection)
```

Hierarchy

TObject

TPersistent

TCollection

Properties

- **Items**[Index: Integer]: TSRVTabItem⁽²⁶³⁾;
- **Count**

1.4 Types

Types Related to Custom Drawing

Events:

- TSRVDrawScrollBarAreaEvent⁽²⁶⁷⁾ for drawing TSRVScrollBar⁽²⁴²⁾ background area;
- TSRVDrawScrollBarEvent⁽²⁶⁷⁾ for drawing TSRVScrollBar⁽²⁴²⁾ elements;
- TSRVDrawTabEvent⁽²⁶⁸⁾ for drawing tabs and a closing button in TSRVTabSet⁽²⁴⁸⁾;
- TSRVDrawTabSetEvent⁽²⁶⁹⁾ for drawing TSRVTabSet⁽²⁴⁸⁾ elements.

Other types:

- TSRVDrawState, TSRVDrawStates⁽²⁶⁸⁾ – states of a painted object.

Types of Properties of ToolBars and ToolWindows

Types of properties:

- TSRVButtonCollection⁽²⁷⁰⁾ – collection of TSRVToolButton⁽²⁷¹⁾
- TSRVAlignButton⁽²⁷⁰⁾ defines the position of buttons;
- TSRVAlignText⁽²⁷⁰⁾ defines the position of text.

Types of events:

- TSRVClickButtonEvent⁽²⁷¹⁾ occurs when a toolbar button is clicked.

Other Types

Types of events:

- TSRVMouseEvent⁽²⁷⁶⁾ – a type of mouse events in TSRichViewEdit⁽⁴⁰⁾;

Types of properties:

TSRVControlStyle⁽²⁷⁵⁾ defines a visual appearance of controls (if skins⁽²²⁷⁾ are not applied);

TSRVFloatProperty⁽²⁷⁵⁾ identifies a TSRichViewEdit⁽⁴⁰⁾ of an floating point type;

TSRVIntProperty⁽²⁷⁶⁾ identifies a TSRichViewEdit⁽⁴⁰⁾ of an floating point type;
 TSRVPageFormat⁽²⁷⁷⁾ – a page size;
 TSRVTHAlign, TSRVTVAlign⁽²⁸⁰⁾ – a position in TSRVTabSet⁽²⁴⁸⁾.

1.4.1 Custom drawing types

Types Related to Custom Drawing

Events:

- TSRVDrawScrollBarAreaEvent⁽²⁶⁷⁾ for drawing TSRVScrollBar⁽²⁴²⁾ background area;
- TSRVDrawScrollBarEvent⁽²⁶⁷⁾ for drawing TSRVScrollBar⁽²⁴²⁾ elements;
- TSRVDrawTabEvent⁽²⁶⁸⁾ for drawing tabs and a closing button in TSRVTabSet⁽²⁴⁸⁾;
- TSRVDrawTabSetEvent⁽²⁶⁹⁾ for drawing TSRVTabSet⁽²⁴⁸⁾ elements.

Other types:

- TSRVDrawState, TSRVDrawStates⁽²⁶⁸⁾ – states of a painted object.

1.4.1.1 TSRVDrawScrollBarAreaEvent type

A type of the following events of TSRVScrollBar⁽²⁴²⁾:

- OnDrawArea⁽²⁴⁸⁾

Unit SRVSBAR.

type

```
TSRVDrawScrollBarAreaEvent = procedure (Sender: TSRVScrollBar(242);
  Canvas: TCanvas; ARect, PaintRect: TRect; isFirstArea: Boolean;
  State: TSRVDrawStates(268); var DoDefault : Boolean) of object;
```

Parameters

Canvas – a canvas where to draw.

ARect – a rectangle defining the position of the object to draw.

APaintRect – a rectangle that needs to be redrawn. You can use this parameter to optimize drawing.

isFirstArea – *True* if this event is called for the background area between the "dec button" and the thumb tab.

State – a set describing the current object state.

Set **DoDefault** to *False* to prevent the default drawing of this object.

1.4.1.2 TSRVDrawScrollBarEvent type

A type of the following events of TSRVScrollBar⁽²⁴²⁾:

- OnDrawBorder⁽²⁴⁸⁾
- OnDrawDecButton
- OnDrawThumbTab

Unit SRVSBAR.

type

```
TSRVDrawScrollBarEvent = procedure (Sender: TSRVScrollBar(242);
  Canvas: TCanvas; ARect, PaintRect: TRect;
  State: TSRVDrawStates(268); var DoDefault : Boolean) of object;
```

Parameters

Canvas – a canvas where to draw.

ARect – a rectangle defining the position of the object to draw.

APaintRect – a rectangle that needs to be redrawn. You can use this parameter to optimize drawing.

State – a set describing the current object state.

Set **DoDefault** to *False* to prevent the default drawing of this object.

1.4.1.3 TSRVDrawState, TSRVDrawStates types

Indicates state information that can influence how an object is drawn.

Unit SRVSkinManager.

type

```
TSRVDrawState = (srvdsHot, srvdsDown, srvdFocused, srvdsEnabled);
TSRVDrawStates = set of TSRVDrawState;
```

State	Meaning
<i>srvdsHot</i>	the object is below the mouse pointer
<i>srvdsDown</i>	the object is clicked
<i>srvdFocused</i>	the object is focused (or belongs to the focused control)
<i>srvdsEnabled</i>	the object is not disabled

1.4.1.4 TSRVDrawTabEvent type

A type of the following events of TSRVTabSet⁽²⁴⁸⁾:

- OnDrawTab⁽²⁵⁹⁾
- OnDrawCloseButton

Unit SRVPreview.

type

```
TSRVDrawTabEvent = procedure (Sender: TSRVTabSet(248); Canvas: TCanvas;
  ARect, PaintRect: TRect; TabIndex: Integer; State: TSRVDrawStates(268);
  var DoDefault: Boolean) of object;
```

Parameters

Canvas – a canvas where to draw.

ARect – a rectangle defining the position of the object to draw.

APaintRect – a rectangle that needs to be redrawn. You can use this parameter to optimize drawing.

TabIndex – the index of the tab (in Tabs⁽²⁵⁸⁾) where this drawing should occur.

State – a set describing the current tab state.

Set **DoDefault** to *False* to prevent the default drawing of this tab.

1.4.1.5 TSRVDrawTabSetEvent type

A type of the following events of TSRVTabSet⁽²⁴⁸⁾:

- OnDrawBorder⁽²⁵⁹⁾
- OnDrawDecButton
- OnDrawIncButton
- OnDrawBackground

Unit SRVTabSet.

type

```
TSRVDrawTabSetEvent = procedure (Sender: TSRVTabSet(248); Canvas: TCanvas;
  ARect, PaintRect: TRect; State: TSRVDrawStates(268);
  var DoDefault: Boolean) of object;
```

Parameters

Canvas – a canvas where to draw.

ARect – a rectangle defining the position of the object to draw.

APaintRect – a rectangle that needs to be redrawn. You can use this parameter to optimize drawing.

State – a set describing the current object state.

Set **DoDefault** to *False* to prevent the default drawing of this object.

1.4.1.6 TSRVOnPaint type

A type of OnBeforePaint⁽²²⁷⁾ and OnAfterPaint⁽²²⁷⁾ events of TSRVPreview⁽²²²⁾.

Unit SRVPreview.

type

```
TSRVOnPaint = procedure (Sender: TObject; ACanvas: TCanvas;
  ARect: TRect) of object;
```

ACanvas – canvas where to draw.

ARect – a location of page(s). This area includes TSRVPreview.BorderWidth⁽²²⁴⁾, but does not include a shadow area (TSRVPreview.ShadowWidth).

When these events are called, a background color (beyond **ARect**) and a shadow is already drawn, and the clipping region is set to **ARect**.

1.4.2 ToolBar types

Types of Properties of ToolBars and ToolWindows

Types of properties:

- `TSRVButtonCollection`⁽²⁷⁰⁾ – collection of `TSRVToolButton`⁽²⁷¹⁾
- `TSRVAlignButton`⁽²⁷⁰⁾ defines the position of buttons;
- `TSRVAlignText`⁽²⁷⁰⁾ defines the position of text.

Types of events:

- `TSRVClickButtonEvent`⁽²⁷¹⁾ occurs when a toolbar button is clicked.

1.4.2.1 TSRVAlignButton type

Defines the position of buttons in `TSRVToolBar`⁽¹⁸⁸⁾.

Unit `SRVToolBar`.

type

```
TSRVAlignButton = (srvabLeft, srvabRight, srvabCenter);
```

1.4.2.2 TSRVAlignText type

Defines the position of text in `TSRVToolBar`⁽¹⁸⁸⁾.

Unit `SRVToolBar`.

type

```
TSRVAlignText = (srvatTop, srvatBottom);
```

A text with the description of the button under the mouse pointer can be displayed at the top (above the buttons) or at the bottom (below the buttons) of the toolbar.

1.4.2.3 TSRVButtonCollection

`TSRVButtonCollection` is a collection of `TSRVToolButton`⁽²⁷¹⁾.

Unit `SRVToolBar`.

Syntax

```
TSRVButtonCollection = class (TCollection)
```

Hierarchy

TObject

TPersistent

TCollection

Description

This is a type of `TSRVToolBar.Buttons`⁽¹⁹¹⁾, `TSRVToolWindow.Buttons`⁽¹⁹⁸⁾, `TSRVRichViewEdit.MenuHButtons`⁽⁵⁵⁾, `TSRVRichViewEdit.MenuVButtons`⁽⁵⁵⁾ properties.

1.4.2.4 TSRVClickButtonEvent type

A type for events happened to a toolbar button.

Unit SRVToolBar.

type

```
TSRVClickButtonEvent = procedure (Sender: TObject;  
    ToolButton: TSRVToolButton(271)) of object;
```

1.4.2.5 TSRVToolButton

TSRVToolButton is a class containing properties of a toolbar button.

Unit SRVToolBar.

Syntax

```
TSRVToolButton = class (TCollectionItem)
```

Hierarchy

TObject

TPersistent

TCollectionItem

Description

This is a type of item in the TSRVToolBar.Buttons⁽¹⁹¹⁾, TSRVToolWindow.Buttons⁽¹⁹⁸⁾, TSRichViewEdit.MenuHButtons⁽⁵⁵⁾, TSRichViewEdit.MenuVButtons⁽⁵⁵⁾ collections properties.

Size and colors of buttons are defined in properties of their toolbars/tool windows.

1.4.2.5.1 Properties

In TSRVToolButton

- AllowAllUp⁽²⁷²⁾
- Caption⁽²⁷²⁾
- Down⁽²⁷²⁾
- Enabled⁽²⁷²⁾
- GroupIndex⁽²⁷³⁾
- Hint⁽²⁷²⁾
- ImageIndexDisable⁽²⁷³⁾
- ImageIndexDown⁽²⁷³⁾
- ImageIndexMove⁽²⁷³⁾
- ImageIndexNormal⁽²⁷⁴⁾
- ShowHint⁽²⁷⁴⁾
- Tag⁽²⁷⁴⁾
- Visible⁽²⁷⁴⁾

1.4.2.5.1.1 TSRVToolButton.AllowAllUp

Specifies whether all buttons in the group that contains this button can be unselected at the same time.

property AllowAllUp: Boolean;

Use AllowAllUp with a group of buttons that belong to the same group. Buttons belong to the same group if they have the same value on their GroupIndex⁽²⁷³⁾ property. Buttons in the same group behave in a mutually exclusive fashion: when one button in the group is selected (Down⁽²⁷²⁾=True), all other buttons in the group become unselected (Down⁽²⁷²⁾=False).

If AllowAllUp is *True*, all of the buttons in a group can be unselected at the same time. Clicking the single selected button in the group will deselect that button without selecting another. If AllowAllUp is *False*, exactly one of the buttons in the group must be selected at any time. Clicking a down button won't return the button to its up state.

If GroupIndex⁽²⁷³⁾ is zero, AllowAllUp has no effect.

1.4.2.5.1.2 TSRVToolButton.Caption

Specifies the caption for the button.

property Caption: TRVUnicodeString;

If the toolbar has a hint area (above or below the buttons), buttons captions are shown there.

1.4.2.5.1.3 TSRVToolButton.Enabled

Controls whether the button responds to mouse and keyboard events.

property Enabled : Boolean;

Use Enabled to change the availability of the button to the user. To disable a button, set Enabled to *False*.

Default value:

False

1.4.2.5.1.4 TSRVToolButton.Down

Specifies whether this button is pressed.

property Down: Boolean;

If *True*, this button is pressed. Only buttons with GroupIndex⁽²⁷³⁾ <>0 can be permanently pressed.

1.4.2.5.1.5 TSRVToolButton.Hint

Contains the text string that can appear when the user moves the mouse over the control.

property Hint: TRVUnicodeString;

Use the Hint property to provide a string of help text either as a help hint, or as help text on a particular location such as a status bar. A help hint is displayed only if ShowHint⁽²⁷⁴⁾=*True*.

If the toolbar has a hint area (below or above the buttons), the main property explaining the button for users is Caption⁽²⁷²⁾.

1.4.2.5.1.6 TSRVToolButton.GroupIndex

Allows buttons to work together as a group.

property GroupIndex : Integer;

If GroupIndex is 0, the button works as a command button. When the user clicks such a button, the button appears pressed (in its clicked state) and then returns to its normal up state when the user releases the mouse button.

When GroupIndex is greater than 0, the button works like a checkbox or as a radio button, depending on AllowAllUp⁽²⁷²⁾ property. If AllowAllUp⁽²⁷²⁾ is *True*, the button works like a checkbox. If AllowAllUp⁽²⁷²⁾=*False*, buttons with the same GroupIndex property value (other than 0) work together as a group of radio buttons. Checked buttons have Down⁽²⁷²⁾=*True*.

Default value:

0

1.4.2.5.1.7 TSRVToolButton.ImageIndexDisable

Specifies the index of the image displayed in the button, if Enabled⁽²⁷²⁾=*False*.

property ImageIndexDisable: Integer;

The image list is defined in the corresponding property of the component containing this button:

- TSRVToolBar.ImageList⁽¹⁹⁴⁾
- TSRVToolWindow.ImageList⁽²⁰¹⁾
- TSRVCustomPropertyTB.ImageList⁽¹³⁰⁾

Default value:

-1 (no image)

1.4.2.5.1.8 TSRVToolButton.ImageIndexDown

Specifies the index of the image displayed in the button, if it is pressed (or checked)

property ImageIndexDown: Integer;

The image list is defined in the corresponding property of the component containing this button:

- TSRVToolBar.ImageList⁽¹⁹⁴⁾
- TSRVToolWindow.ImageList⁽²⁰¹⁾
- TSRVCustomPropertyTB.ImageList⁽¹³⁰⁾

Default value:

-1 (no image)

1.4.2.5.1.9 TSRVToolButton.ImageIndexMove

Specifies the index of the image displayed in the button, if it is highlighted (below the mouse pointer)

property ImageIndexMove: Integer;

The image list is defined in the corresponding property of the component containing this button:

- TSRVToolBar.ImageList⁽¹⁹⁴⁾
- TSRVToolWindow.ImageList⁽²⁰¹⁾
- TSRVCustomPropertyTB.ImageList⁽¹³⁰⁾

Default value:

-1 (no image)

1.4.2.5.1.10 TSRVToolButton.ImageIndexNormal

Specifies the index of the image displayed in the button.

property ImageIndexNormal: Integer;

This image is displayed in a normal state (when the button is not disabled, is not highlighted, is not pressed).

The image list is defined in the corresponding property of the component containing this button:

- TSRVToolBar.ImageList⁽¹⁹⁴⁾
- TSRVToolWindow.ImageList⁽²⁰¹⁾
- TSRVCustomPropertyTB.ImageList⁽¹³⁰⁾

Default value:

-1 (no image)

1.4.2.5.1.11 TSRVToolButton.ShowHint

Determines whether a toolbar displays a help hint when the mouse pointer is above this button.

property ShowHint: Boolean;

The help hint's text is defined in Hint⁽²⁷²⁾ property.

Default value:

False

See also:

- Caption⁽²⁷²⁾

1.4.2.5.1.12 TSRVToolButton.Tag

Stores an integer value as part of a button.

property Tag: Longint;

Tag has no predefined meaning. The Tag property is provided for the convenience of developers. It can be used for storing an additional integer value or it can be typecast to any 32-bit value such as a component reference or a pointer.

Default value:

0

1.4.2.5.1.13 TSRVToolButton.Visible

Specifies whether the button is visible on the toolbar.

property Visible: Boolean;

Default value:

True

1.4.3 TSRVControlStyle

Define appearance of controls.

Unit SRVControl.

type

```
TSRVControlStyle = (srvcsSimple, srvcsClassic);
```

srvcsSimple:

A modern, minimalistic and consistent appearance of controls.

It can be customized using RVControlsPainter class from RVControls unit. This class is described in the TRichView help file. The main its property is Theme: TRVControlTheme, where

```
TRVControlTheme = (rvctPaleBlue, rvctSpringGreen, rvctSienna, rvctHighContrast);
```

By changing Theme, you change colors of all controls having SRVControlStyle = *srvcsSimple*.

In Delphi XE2 and newer, this mode uses GDI+ for drawing, so lines are anti-aliased.

In Lazarus, GDI+ drawing is optional, see GDI+ notes ⁽³²⁶⁾.

srvcsClassic:

A classic appearance of controls.

They look like in ScaleRichView 8 and older.

This is the type of the following properties:

- TSRVScrollBar⁽²⁴²⁾.SRVControlStyle⁽²⁴⁷⁾;
- TSRVTabSet⁽²⁴⁸⁾.SRVControlStyle⁽²⁵⁷⁾;
- TSRVCustomControl.SRVControlStyle⁽³³⁴⁾ (published in TSRVButton⁽³⁴²⁾, TSRVCheckBox⁽³⁴⁴⁾, TSRVGroupBox⁽³⁵¹⁾, TSRVPanel⁽³⁶⁴⁾, TSRVRadioButton⁽³⁶⁶⁾);
- TSRVWinControl.SRVControlStyle⁽³³⁹⁾ (published in TSRVEdit⁽³⁴⁸⁾, TSRVMemo⁽³⁶¹⁾).
- TSRichViewEdit⁽⁴⁰⁾.ScrollBarControlStyle⁽⁶⁹⁾

1.4.4 TSRVFloatProperty type

Unit SclRView.

Identifies a property of TSRichViewEdit⁽⁴⁰⁾ of an integer or a floating point type.

type

```
TSRVFloatProperty = (srvfpPPLeftMargin, srvfpPPRightMargin,
  srvfpPPTopMargin, srvfpPPBottomMargin,
  srvfpPPHeaderY, srvfpPPFooterY,
  srvfpPPPPageWidth, srvfpPPPPageHeight);
```

Value	Corresponding property of TSRichViewEdit
<i>srvfpPPLeftMargin</i>	PageProperty ⁽⁵⁸⁾ .LeftMargin ⁽¹³⁷⁾
<i>srvfpPPRightMargin</i>	PageProperty.RightMargin ⁽¹⁴¹⁾
<i>srvfpPPTopMargin</i>	PageProperty.TopMargin ⁽¹⁴²⁾

Value	Corresponding property of T <code>SRichViewEdit</code>
<code>srvfpPPBottomMargin</code>	<code>PageProperty.BottomMargin</code> ⁽¹³⁵⁾
<code>srvfpPPHeaderY</code>	<code>PageProperty.HeaderY</code> ⁽¹³⁷⁾
<code>srvfpPPFooterY</code>	<code>PageProperty.FooterY</code> ⁽¹³⁶⁾
<code>srvfpPPPPageWidth</code>	<code>PageProperty.PageWidth</code> ⁽¹⁴¹⁾
<code>srvfpPPPPageHeight</code>	<code>PageProperty.PageHeight</code> ⁽¹³⁹⁾

A parameter of this type is used in `T(40)SRichViewEdit`.`SetFloatPropertyEd` ⁽⁹⁶⁾.

See also:

- `TSRVIntProperty` ⁽²⁷⁶⁾

1.4.5 T`SRVIntProperty` type

Unit `ScIRView`.

Identifies a property of `T(40)SRichViewEdit` of an integer or an ordinal type.

type

```
TSRVIntProperty = (srvipPPOrientation, srvipPPPPageFormat,
  srvipPPMirrorMargins, srvipPPFacingPages, srvipPPTitlePage);
```

Value	Corresponding property of T <code>SRichViewEdit</code>
<code>srvipPPOrientation</code>	<code>PageProperty</code> ⁽⁵⁸⁾ . <code>Orientation</code> ⁽¹³⁸⁾
<code>srvipPPPPageFormat</code>	<code>PageProperty.PageFormat</code> ⁽¹³⁸⁾
<code>srvipPPMirrorMargins</code> <code>s</code>	<code>PageProperty.MirrorMargins</code> ⁽¹³⁸⁾
<code>srvipPPFacingPages</code>	<code>PageProperty.FacingPages</code> ⁽¹³⁶⁾
<code>srvipPPTitlePage</code>	<code>PageProperty.TitlePage</code> ⁽¹⁴¹⁾

A parameter of this type is used in `T(40)SRichViewEdit`.`SetIntPropertyEd` ⁽⁹⁶⁾.

See also:

- `TSRVFloatProperty` ⁽²⁷⁵⁾

1.4.6 T`SRVMouseEvent` type

A type for several events of `T(40)SRichViewEdit`.

Unit `ScIRView`.

```
T(40)SRVMouseEvent = procedure (Sender: T(40)SRichViewEdit; Button: TMouseButton;
  Shift: TShiftState; X, Y: Integer; PageNo : Integer) of object;
```

X and **Y** are the pixel coordinates of the mouse pointer in the client area of the **Sender**.

Shift contains a set of pressed keys. **Button** identifies a mouse button.

PageNo is a page below the mouse pointer, from 1.

1.4.7 TSRVPageFormat type

A type of page format (size).

Unit SclRView.

type

```
TSRVPageFormat = (
    srvfmCustom, srvfm4A0, srvfm2A0, srvfmA0, srvfmA1, srvfmA2,
    srvfmA3, srvfmA4, srvfmA5, srvfmA6, srvfmA7, srvfmA8,
    srvfmA9, srvfmA10, srvfmB0, srvfmB1, srvfmB2, srvfmB3,
    srvfmB4, srvfmB5, srvfmB6, srvfmB7, srvfmB8, srvfmB9,
    srvfmB10, srvfmC0, srvfmC1, srvfmC2, srvfmC3, srvfmC4,
    srvfmC5, srvfmC6, srvfmC7, srvfmC8, srvfmC9, srvfmC10,
    srvfmLetter, srvfmLegal, srvfmLedger, srvfmTabloid,
    srvfmStatement, srvfmQuarto, srvfmFoolscap, srvfmFolio,
    srvfmExecutive, srvfmMonarch, srvfmGovernmentLetter,
    srvfmPost, srvfmCrown, srvfmLargePost, srvfmDemy,
    srvfmMedium, srvfmRoyal, srvfmElephant, srvfmDoubleDemy,
    srvfmQuadDemy, srvfmIndexCard3_5, srvfmIndexCard4_6,
    srvfmIndexCard5_8, srvfmInternationalBusinessCard,
    srvfmUSBusinessCard, srvfmEmperor, srvfmAntiquarian,
    srvfmGrandEagle, srvfmDoubleElephant, srvfmAtlas,
    srvfmColombier, srvfmImperial, srvfmDoubleLargePost,
    srvfmPrincess, srvfmCartridge, srvfmSheet, srvfmHalfPost,
    srvfmDoublePost, srvfmSuperRoyal, srvfmCopyDraught,
    srvfmPinchedPost, srvfmSmallFoolscap, srvfmBrief, srvfmPott,
    srvfmPA0, srvfmPA1, srvfmPA2, srvfmPA3, srvfmPA4, srvfmPA5,
    srvfmPA6, srvfmPA7, srvfmPA8, srvfmPA9, srvfmPA10, srvfmF4,
    srvfmA0a, srvfmJISB0, srvfmJISB1, srvfmJISB2, srvfmJISB3,
    srvfmJISB4, srvfmJISB5, srvfmJISB6, srvfmJISB7, srvfmJISB8,
    srvfmJISB9, srvfmJISB10, srvfmJISB11, srvfmJISB12,
    srvfmANSI_A, srvfmANSI_B, srvfmANSI_C, srvfmANSI_D,
    srvfmANSI_E, srvfmArch_A, srvfmArch_B, srvfmArch_C,
    srvfmArch_D, srvfmArch_E, srvfmArch_E1);
```

Custom Format

srvfmCustom – user-defined paper format.

ISO "A" Paper Sizes (mm x mm)

srvfmA0 – paper format A0, 841 x 1189

srvfmA1 – paper format A1, 594 x 841

srvfmA2 – paper format A2, 420 x 594

srvfmA3 – paper format A3, 297 x 420

srvfmA4 – paper format A4, 210 x 297

srvfmA5 – paper format A5, 148 x 210

srvfmA6 – paper format A6, 105 x 148

srvfmA7 – paper format A7, 74 x 105

srvfmA8 – paper format A8, 52 x 74

srvfmA9 – paper format A9, 37 x 52

srvfmA10 – paper format A10, 26 x 37

ISO "A" Paper Sizes, German Extensions (mm x mm)

srvfm4A0 – paper format 4A0, 1682 x 2378

srvfm2A0 – paper format 2A0, 1189 x 1682

ISO "B" Paper Sizes (mm x mm)

srvfmB0 – paper format B0, 1000 x 1414

srvfmB1 – paper format B1, 707 x 1000

srvfmB2 – paper format B2, 500 x 707

srvfmB3 – paper format B3, 353 x 500

srvfmB4 – paper format B4, 250 x 353

srvfmB5 – paper format B5, 176 x 250

srvfmB6 – paper format B6, 125 x 176

srvfmB7 – paper format B7, 88 x 125

srvfmB8 – paper format B8, 62 x 88

srvfmB9 – paper format B9, 44 x 62

srvfmB10 – paper format B10, 31 x 44

ISO "C" Paper Sizes (mm x mm)

srvfmC0 – paper format C0, 917 x 1297

srvfmC1 – paper format C1, 648 x 917

srvfmC2 – paper format C2, 458 x 648

srvfmC3 – paper format C3, 324 x 458

srvfmC4 – paper format C4, 229 x 324

srvfmC5 – paper format C5, 162 x 229

srvfmC6 – paper format C6, 114 x 162

srvfmC7 – paper format C7, 81 x 114

srvfmC8 – paper format C8, 57 x 81

srvfmC9 – paper format C9, 40 x 57

srvfmC10 – paper format C10, 28 x 40

North American Paper Sizes (inch x inch)

srvfmLetter – letter paper format, 8.5 x 11

srvfmLegal – legal paper format, 8.5 x 14

srvfmLedger – tabloid (ledger) paper format, 11 x 17

srvfmStatement – statement (half letter) paper format, 5.5 x 8.5

srvfmQuarto – quarto paper format, 8 x 10

srvfmFoolscap – foolscap (folio) paper format, 8 x 13

srvfmFolio – foolscap (folio) paper format, 8 x 13

srvfmExecutive – executive (monarch) paper format, 7.25 x 10.5

srvfmMonarch – executive (monarch) paper format, 7.25 x 10.5

srvfmGovernmentLetter – government-letter paper format, 8 x 10.5

srvfmPost – post paper format, 15.5 x 19.5

srvfmCrown – crown paper format, 15 x 20

srvfmLargePost – large post paper format, 16.5 x 21

srvfmDemy – demy paper format, 17.5 x 22.5

srvfmMedium – medium paper format, 18 x 23

srvfmRoyal – royal paper format, 20 x 25
srvfmElephant – elephant paper format, 23 x 28
srvfmDoubleDemy – double demy paper format, 22.5 x 35
srvfmQuadDemy – quad demy paper format, 35 x 45

Index and Business Cards (inch x inch)

srvfmIndexCard3_5 – index card 3 x 5
srvfmIndexCard4_6 – index card 4 x 6
srvfmIndexCard5_8 – index card 5 x 8
srvfmInternationalBusinessCard – international business card, 2.125 x 3.37
srvfmUSBusinessCard – US business card, 2 x 3.5

Traditional Paper Sizes (inch x inch)

srvfmEmperor – emperor paper format, 48 x 72
srvfmAntiquarian – antiquarian paper format, 31 x 53
srvfmGrandEagle – grand eagle paper format, 28.75 x 42
srvfmDoubleElephant – double elephant paper format, 26.75 x 40
srvfmAtlas – atlas paper format, 26 x 34
srvfmColombier – colombier paper format, 23.5 x 34.5
srvfmImperial – imperial paper format, 22 x 30
srvfmDoubleLargePost – double large post paper format, 21 x 33
srvfmPrincess – princess paper format, 21.5 x 28
srvfmCartridge – cartridge paper format, 21 x 26
srvfmSheet – sheet (half post) paper format, 19.5 x 23.5
srvfmHalfPost – sheet (half post) paper format, 19.5 x 23.5
srvfmDoublePost – double post paper format, 19 x 30.5
srvfmSuperRoyal – super royal paper format, 19 x 27
srvfmCopyDraught – copy draught paper format, 16 x 20
srvfmPinchedPost – pinched post paper format, 14.75 x 18.5
srvfmSmallFoolscap – small foolscap paper format, 13.25 x 16.5
srvfmBrief – brief paper format, 13.5 x 16
srvfmPott – pott paper format, 12.5 x 15

PA Series (ISO 216) (mm x mm)

srvfmPA0 – paper format PA0, 840 x 1120
srvfmPA1 – paper format PA1, 560 x 840
srvfmPA2 – paper format PA2, 420 x 560
srvfmPA3 – paper format PA3, 280 x 420
srvfmPA4 – paper format PA4, 210 x 280
srvfmPA5 – paper format PA5, 140 x 210
srvfmPA6 – paper format PA6, 105 x 140
srvfmPA7 – paper format PA7, 70 x 105
srvfmPA8 – paper format PA8, 52 x 70
srvfmPA9 – paper format PA9, 35 x 52
srvfmPA10 – paper format PA10, 26 x 35

Other Paper Sizes (mm x mm)

srvfmF4 – 210 x 330
srvfmA0a – paper format A0a, 1000 x 1370

JIS B-Series (Japanese) (mm x mm)

srvfmJISB0 – paper format JIS B0, 1030 x 1456
srvfmJISB1 – paper format JIS B1, 728 x 1030
srvfmJISB2 – paper format JIS B2, 515 x 728
srvfmJISB3 – paper format JIS B3, 364 x 515
srvfmJISB4 – paper format JIS B4, 257 x 364
srvfmJISB5 – paper format JIS B5, 182 x 257
srvfmJISB6 – paper format JIS B6, 128 x 182
srvfmJISB7 – paper format JIS B7, 91 x 128
srvfmJISB8 – paper format JIS B8, 64 x 91
srvfmJISB9 – paper format JIS B9, 45 x 64
srvfmJISB10 – paper format JIS B10, 32 x 45
srvfmJISB11 – paper format JIS B11, 22 x 32
srvfmJISB12 – paper format JIS B12, 16 x 22

ANSI Paper Sizes (inch x inch)

srvfmANSI_A – paper format ANSI A, 8.5 x 11
srvfmANSI_B – paper format ANSI B, 11 x 17
srvfmANSI_C – paper format ANSI C, 17 x 22
srvfmANSI_D – paper format ANSI D, 22 x 34
srvfmANSI_E – paper format ANSI E, 34 x 44

Architectural Sizes (inch x inch)

srvfmArch_A – paper size Arch A, 12 x 9
srvfmArch_B – paper size Arch B, 18 x 12
srvfmArch_C – paper size Arch C, 24 x 18
srvfmArch_D – paper size Arch D, 36 x 24
srvfmArch_E – paper size Arch E, 48 x 36
srvfmArch_E1 – paper size Arch E1, 42 x 30

1.4.8 TSRVTHAlign, TSRVTVAlign types

Define the position of tab closing buttons in `TSRVTabSet`⁽²⁴⁸⁾.

Unit `SRVTabSet`.

type

```
TSRVTHAlign = (srvthaLeft, srvthaCenter, srvthaRight);
TSRVTVAlign = (srvtvaTop, srvtvaCenter, srvtvaBottom);
```

These are the types of the following properties:

- `TSRVCloseButton.HorizontalAlign`⁽²⁶²⁾

1.5 Global variables

Unit `SclRView`;

var

```
MouseWheelZoomMin: Integer = 10;
MouseWheelZoomMax: Integer = 500;
MouseWheelZoomSpeed : Single = 1.0;
```

The following variables defines parameters for zooming using the mouse wheel⁽¹⁶¹⁾:

MouseWheelZoomMin – minimal value of ZoomPercent⁽¹⁶⁷⁾

MouseWheelZoomMax – maximal value of ZoomPercent⁽¹⁶⁷⁾


MouseWheelZoomSpeed defines how fast ZoomPercent⁽¹⁶⁷⁾ changes.


2 ScaleRichView Actions


Almost all RichViewActions designed for TRichViewEdit can be used with TScaleRichViewEdit⁽⁴⁰⁾.

However, printing actions (TrvActionPrint, TrvActionQuickPrint, TrvActionPrintPreview, TrvActionPageSetup) cannot be used. Also, it is highly preferred to use ScaleRichView versions of the note actions (do not use TrvActionInsertFootnote, TrvActionInsertEndnote, TrvActionInsertSidenote, TrvActionInsertTextBox, TrvActionEditNote).

Print and preview:

 TsrvActionQuickPrint⁽²⁹³⁾


 TsrvActionPrint⁽²⁹²⁾


 TsrvActionPreview⁽²⁹²⁾

Page setup:

 TsrvActionPageSetup⁽²⁹⁰⁾


 TsrvActionOrientationPortrait⁽²⁸⁷⁾ changes page orientation to "portrait".


 TsrvActionOrientationLandscape⁽²⁸⁶⁾ changes page orientation to "landscape".

 TsrvActionPageFormat⁽²⁸⁸⁾ changes page size.

 TsrvActionLineNumbers⁽²⁸⁶⁾ shows or hides line numbers.


View:


 TsrvActionThumbnails⁽³¹⁰⁾ shows/hides page thumbnails.

 TsrvActionLayoutDraft⁽²⁹⁸⁾ switches the editor to a draft layout.

 TsrvActionLayoutWeb⁽³⁰⁹⁾ switches the editor to a web layout.


 TsrvActionLayoutPrint⁽³⁰⁰⁾ switches the editor to a print layout.


 TsrvActionLayoutSideToSide⁽³⁰⁶⁾ switches the editor to a side-to-side layout.

 TsrvActionLayoutRead⁽³⁰³⁾ switches the editor to a read mode layout.


Zoom:


 TsrvActionZoom⁽³¹²⁾ applies the specified zooming percent to the editor.


 TsrvActionZoomPageWidth⁽³¹³⁾ zooms the editor to fit the page width in the editor width.

 TsrvActionZoomFullPage⁽³¹³⁾ zooms the editor to fit the full page in the editor.

Header and footer:


 TsrvActionEditHeader⁽³¹⁴⁾ starts editing a page header.


 TsrvActionEditFooter⁽³¹⁴⁾ starts editing a page footer.




 TsrvActionEditMain⁽³¹⁵⁾ closes a header/footer editor and returns the main document.

Notes and text boxes:

 TsrvActionInsertFootnote⁽³¹⁹⁾ inserts a footnote.

 TsrvActionInsertEndnote⁽³¹⁸⁾ inserts an endnote.

 TsrvActionInsertSidenote⁽³²¹⁾ inserts a sidenote (a note in a floating box).

-  TsrvActionInsertTextBox⁽³²³⁾ inserts a text box item.
-  TsrvActionEditNote⁽³¹⁸⁾ starts editing a note or a text box.
-  TsrvActionReturnToNote⁽³²⁴⁾ moves the caret to the parent note.

See also:

translating actions⁽²⁸²⁾

2.1 Localization

General information

RichViewActions require localization, even if you use only one UI language.

Initially, the language specified in TRVAControlPanel.Language is used. If TRVAControlPanel component is not used, 'English (US)' is used. This language will be used in dialogs displayed by RichViewActions, but Captions and Hints of the actions themselves are not localized yet. Call RVA_LocalizeForm to localize them.

For changing UI language, use RVA_ChooseLanguage function. The topic about this function contains an example.

Other localization functions are declared in RVALocalize and SRVALocalize units.

If you want to add a translation to you language (for example, Greek):

1. First, a translation RichViewActions for TRichViewEdit must be done.
2. Open SRVAL_EngUS.pas, rename it to SRVAL_Greek.pas. Add SRVAL_Greek in "uses" of SRVALocalize.pas. (If you use Delphi/C++Builder 2009+, save this unit as UTF-8).
3. Translate all text in this unit (do not translate comments!)
4. Change the call of SRVA_RegisterLanguage to SRVA_RegisterLanguage('Greek', @Messages).
5. Send the translated file to svt@trichview.com for including in the main installation of ScaleRichView. Your file will be maintained (it will be corrected when new messages will be added in the future versions of RichViewActions and ScaleRichView)

Procedures and functions

The unit SRVALocalize contains the following procedures and functions:

```
function SRVA_GetS(MsgID: TSRVAMessageID;
  ControlPanel: TRVAControlPanel = nil): TRVALocString;
function SRVA_GetPC(MsgID: TSRVAMessageID;
  ControlPanel: TRVAControlPanel): PChar;
procedure SRVA_LocalizeToolWindow(ToolWindow: TSRVToolWindow(196);
  ControlPanel: TRVAControlPanel = nil);
procedure SRVA_LocalizeSRichViewEdit(Edit: TSRichViewEdit(40);
  ControlPanel: TRVAControlPanel = nil);
function SRVA_GetUnitsName(Units: TRVUnits;
  ControlPanel: TRVAControlPanel): TRVALocString;
```

SRVA_GetS and SRVA_GetPC return the text of the message by its identifier.

SRVA_LocalizeToolWindow localizes a typical search tool window, assuming that it has buttons in the following order:

- go to next/previous page,
- find table
- find picture
- find heading
- find hyperlink
- find text

The following constants are defined:

```
SRV_TOOLWIN_PAGE      = 0;
SRV_TOOLWIN_TABLE     = 1;
SRV_TOOLWIN_PICTURE   = 2;
SRV_TOOLWIN_HEADLINE  = 3;
SRV_TOOLWIN_HYPERLINK = 4;
SRV_TOOLWIN_TEXT      = 5;
```

SRVA_LocalizeSRichViewEdit localizes a TSRichViewEdit control. It changes the following properties of ViewProperty⁽⁶⁸⁾:

- HintPrefixText⁽¹⁵⁹⁾;
- Texts⁽¹⁶²⁾

SRVA_GetUnitsName returns name of the given measuring unit.

2.2 Basic actions

Basic Actions

The following classes are ancestors for actions:

- TsrvAction⁽²⁸³⁾ – the parent action for all ScaleRichView actions;
- TsrvCustomActionThatNeedsDocProps⁽²⁸⁴⁾ – the parent action for ScaleRichView actions than need access to file properties.

2.2.1 TsrvAction

TrvAction is the base class for all RichViewActions working with TSRichViewEdit⁽⁴⁰⁾.

Unit SRVActions;

Syntax

```
TsrvAction = class (TrvCustomAction)
```

▼ Hierarchy

```
TObject
TPersistent
TComponent
TBasicAction
TContainedAction
TCustomAction
TAction
TrvCustomAction
```

Description

This action is not used directly, but all RichViewActions working with TSRichViewEdit⁽⁴⁰⁾ are inherited from it.

2.2.1.1 Properties

In TsrvAction

■ Control⁽²⁸⁴⁾

Inherited from TrvCustomAction

■ Disabled

2.2.1.1.1 TsrvAction.Control

Defines the editor for this action.

```
property Control: T(40)SRichViewEdit;
```

If T⁽⁴⁰⁾SRichViewEdit component is assigned to this property, the action works with this component.

Otherwise, if T⁽⁴⁰⁾SRichViewEdit component is focused, the action works with it.

Otherwise, it works with the first found T⁽⁴⁰⁾SRichViewEdit component.

2.2.2 TsrvCustomActionThatNeedsDocProps

TsrvCustomActionThatNeedsDocProps is the basic class for actions that needs access to the document properties maintained by TrvActionSave.

Unit SRVActions;

Syntax

```
TsrvCustomActionThatNeedsDocProps = class (TsrvAction(283))
```

▼ Hierarchy

TObject
TPersistent
TComponent
TBasicAction
TContainedAction
TCustomAction
TAction
TrvCustomAction
TsrvAction⁽²⁸³⁾

Description

This class is not used directly. The following actions are inherited from it:

- TsrvActionPrint⁽²⁹²⁾
- TsrvActionQuickPrint⁽²⁹³⁾

- TsrvActionPageSetup ⁽²⁹⁰⁾
- TsrvActionPageFormat ⁽²⁸⁸⁾
- TsrvActionLayoutDraft ⁽²⁹⁸⁾
- TsrvActionLayoutPrint ⁽³⁰⁰⁾
- TsrvActionLayoutWeb ⁽³⁰⁹⁾

2.2.2.1 Properties

In TsrvCustomActionThatNeedsDocProps

- ActionSave ⁽²⁸⁵⁾

Inherited from TsrvAction ⁽²⁸³⁾

- Control ⁽²⁸⁴⁾

Inherited from TrvCustomAction

- Disabled

2.2.2.1.1 TsrvCustomActionThatNeedsDocProps.ActionSave




A link to TrvActionSave.

property ActionSave: TrvActionSave;






If this property is not assigned, the action searches for the first TrvActionSave action on the same form/datamodule, and uses it.

2.3 Print, preview, page setup

Print and preview:

-  TsrvActionQuickPrint ⁽²⁹³⁾
-  TsrvActionPrint ⁽²⁹²⁾
-  TsrvActionPreview ⁽²⁹²⁾

Page setup:

-  TsrvActionPageSetup ⁽²⁹⁰⁾
-  TsrvActionOrientationPortrait ⁽²⁸⁷⁾ changes page orientation to "portrait".
-  TsrvActionOrientationLandscape ⁽²⁸⁶⁾ changes page orientation to "landscape".
-  TsrvActionPageFormat ⁽²⁸⁸⁾ changes page size.
-  TsrvActionLineNumbers ⁽²⁸⁶⁾ shows or hides line numbers.

2.3.1 TsrvActionCustomOrientation

TsrvActionCustomOrientation is the base class for actions changing page orientation.

Unit SRVActions;

Syntax

TsrvActionCustomOrientation = **class** (TsrvAction ⁽²⁸³⁾)

▼ Hierarchy

TObject
TPersistent
TComponent
TBasicAction
TContainedAction
TCustomAction
TAction
TrvCustomAction
TsrvAction ⁽²⁸³⁾

Description

This action is not used directly.

The following actions are inherited from it:

- *TsrvActionOrientationPortrait* ⁽²⁸⁷⁾
- *TsrvActionOrientationLandscape* ⁽²⁸⁶⁾

2.3.2 TsrvActionLineNumbers

TsrvActionLineNumbers toggles visibility of line numbers

Unit SRVActions;

Syntax

```
TsrvActionLineNumbers = class (TsrvAction (283))
```

▼ Hierarchy

TObject
TPersistent
TComponent
TBasicAction
TContainedAction
TCustomAction
TAction
TrvCustomAction
TsrvAction ⁽²⁸³⁾

Description

This action changes value of *LineNumberProperty* ⁽⁵⁴⁾.*Visible* ⁽¹³³⁾ property.

2.3.3 TsrvActionOrientationLandscape

TsrvActionOrientationLandscape applies a landscape page orientation the the document in the target editor.

Unit SRVActions;

Syntax

```
TsrvActionOrientationLandscape = class (TsrvAction (283))
```

▼ **Hierarchy**

TObject
TPersistent
TComponent
TBasicAction
TContainedAction
TCustomAction
TAction
TrvCustomAction
TsrvAction⁽²⁸³⁾

Description

This action assigns `PageProperty(58).Orientation(138) := poLandscape`.

If `PageProperty(58).PageFormat(138) = srvfmCustom`, and `PageProperty(58).PageWidth(141) < PageProperty(58).PageHeight(139)`, the action exchanges values of these properties.

The properties are changed as an editing operation (undoable).

See also:

- `TsrvActionOrientationPortrait`⁽²⁸⁷⁾

2.3.4 TsrvActionOrientationPortrait

`TsrvActionOrientationPortrait` applies a portrait page orientation the the document in the target editor.

Unit SRVActions;

Syntax

```
TsrvActionOrientationPortrait = class(TsrvAction(283))
```

▼ **Hierarchy**

TObject
TPersistent
TComponent
TBasicAction
TContainedAction
TCustomAction
TAction
TrvCustomAction
TsrvAction⁽²⁸³⁾

Description

This action assigns `PageProperty(58).Orientation(138) := poPortrait`.

If `PageProperty(58).PageFormat(138) = srvfmCustom`, and `PageProperty(58).PageWidth(141) > PageProperty(58).PageHeight(139)`, the action exchanges values of these properties.

The properties are changed as an editing operation (undoable).

See also:

- [TsrvActionOrientationLandscape](#) ⁽²⁸⁶⁾

2.3.5 TsrvActionPageFormat

[TsrvActionPageFormat](#) changes page size for the document in the target editor.

Unit SRVActions;

Syntax

```
TsrvActionPageFormat = class (TsrvCustomActionThatNeedsDocProps (284))
```

▼ Hierarchy

[TObject](#)
[TPersistent](#)
[TComponent](#)
[TBasicAction](#)
[TContainedAction](#)
[TCustomAction](#)
[TAction](#)
[TrvCustomAction](#)
[TsrvAction](#) ⁽²⁸³⁾
[TsrvCustomActionThatNeedsDocProps](#) ⁽²⁸⁴⁾

Description

This value changes the following properties of the target editor:

- [PageProperty](#) ⁽⁵⁸⁾.[PageFormat](#) ⁽¹³⁸⁾ (see [PageFormat](#) ⁽²⁸⁹⁾)
- [PageProperty](#) ⁽⁵⁸⁾.[PageWidth](#) ⁽¹⁴¹⁾ (see [PageWidth](#) ⁽²⁸⁹⁾)
- [PageProperty](#) ⁽⁵⁸⁾.[PageHeight](#) ⁽¹³⁹⁾ (see [PageHeight](#) ⁽²⁸⁹⁾)

The properties are changed as an editing operation (undoable).

2.3.5.1 Properties

In TsrvActionLayoutPrint

- [PageFormat](#) ⁽²⁸⁹⁾
- [PageHeight](#) ⁽²⁸⁹⁾
- [PageWidth](#) ⁽²⁸⁹⁾
- [Units](#) ⁽²⁸⁹⁾

Inherited from TsrvCustomActionThatNeedsDocProps

- [ActionSave](#) ⁽²⁸⁵⁾

Inherited from TsrvAction ⁽²⁸³⁾

- [Control](#) ⁽²⁸⁴⁾

Inherited from TrvCustomAction

■ Disabled

2.3.5.1.1 TsrvActionPageFormat.PageFormat

Specifies a page format to apply to the document.

property PageFormat: TSRVPageFormat⁽²⁷⁷⁾;

If PageFormat=*srvfmCustom*, PageWidth⁽²⁸⁹⁾ and PageHeight⁽²⁸⁹⁾ properties are used.

When this property is assigned, Caption and Hints are changed.

Default value:

srvfmA4

2.3.5.1.2 TsrvActionPageFormat.PageHeight

Specifies a page height to apply to the document.

property PageHeight: Extended;

This property is used if PageFormat⁽²⁸⁹⁾=*srvfmCustom*.

This property is measured in Units⁽²⁸⁹⁾.

When this property is assigned, Caption and Hints are changed.

See also:

- PageWidth⁽²⁸⁹⁾.

Default value:

297.0 (height of A4 in mm)

2.3.5.1.3 TsrvActionPageFormat.PageWidth

Specifies a page width to apply to the document.

property PageWidth: Extended;

This property is used if PageFormat⁽²⁸⁹⁾=*srvfmCustom*.

This property is measured in Units⁽²⁸⁹⁾.

When this property is assigned, Caption and Hints are changed.

See also:

- PageHeight⁽²⁸⁹⁾.

Default value:

210.0 (width of A4 in mm)

2.3.5.1.4 TsrvActionPageFormat.Units

Specifies measuring units for PageWidth⁽²⁸⁹⁾ and PageHeight⁽²⁸⁹⁾ properties.

property PageWidth: Extended;

When this property is assigned, Caption and Hints are changed.

Default value:

rvuMillimeters

2.3.6 TsrvActionPageSetup

TsrvActionPageSetup is the action for "File | Page Setup" command.

Unit SRVActions;

Syntax

```
TsrvActionPageSetup = class (TsrvCustomActionThatNeedsDocProps284)
```

▼ **Hierarchy**

TObject
TPersistent
TComponent
TBasicAction
TContainedAction
TCustomAction
TAction
TrvCustomAction
*TsrvAction*²⁸³
*TsrvCustomActionThatNeedsDocProps*²⁸⁴

Description

This action displays a page setup dialog.

It changes the following properties of TSVRichViewEdit.PageProperty⁵⁸:

- LeftMargin¹³⁷
- RightMargin¹⁴¹
- TopMargin¹⁴²
- HeaderY¹³⁷
- FooterY¹³⁶
- BottomMargin¹³⁵
- Orientation¹³⁸

The properties above are changed as an editing operation (undoable).

Additionally, it allows changing a paper source for the current printer (if available).

2.3.6.1 Properties

In TsrvActionPageSetup

- PageFormats1..PageFormats6²⁹¹

Inherited from TsrvCustomActionThatNeedsDocProps

- ActionSave²⁸⁵

Inherited from TsrvAction ⁽²⁸³⁾

■ Control ⁽²⁸⁴⁾

Inherited from TrvCustomAction

■ Disabled

2.3.6.1.1 TsrvActionPageSetup.PageFormats*

Lists of page formats initially available in the page setup dialog.

type

```
// ISO A & B
TSRVPageFormatAB = srvfm4A0..srvfmB10;
// ISO C
TSRVPageFormatC = srvfmC0..srvfmC10;
// North American
TSRVPageFormatUSA = srvfmLetter..srvfmQuadDemy;
// Index/business cards & Traditional
TSRVPageFormatCardTrad = srvfmIndexCard3_5..srvfmPott;
// PA & Other & JIS B
TSRVPageFormatPAOtherJISB = srvfmPA0..srvfmJISB12;
// ANSI & Architectural
TSRVPageFormatANSIArch = srvfmANSI_A..srvfmArch_E1;

TSRVPageFormats = set of TSRVPageFormat (277);
TSRVPageFormatsAB = set of TSRVPageFormatAB;
TSRVPageFormatsC = set of TSRVPageFormatC;
TSRVPageFormatsUSA = set of TSRVPageFormatUSA;
TSRVPageFormatsCardTrad = set of TSRVPageFormatCardTrad;
TSRVPageFormatsPAOtherJISB = set of TSRVPageFormatPAOtherJISB;
TSRVPageFormatsANSIArch = set of TSRVPageFormatANSIArch;
```

property PageFormats1: TSRVPageFormatsAB;

property PageFormats2: TSRVPageFormatsC;

property PageFormats3: TSRVPageFormatsUSA;

property PageFormats4: TSRVPageFormatsCardTrad;

property PageFormats5: TSRVPageFormatsPAOtherJISB;

property PageFormats6: TSRVPageFormatsANSIArch;

All these types are subsets of TSRVPageFormat ⁽²⁷⁷⁾ type. Formats listed in these properties are added in the combo box on the page setup dialog. The user can choose any page format using a menu.

Default values:

[*srvfmA4*, *srvfmA5*, *srvfmA5*] in PageFormats1

[*srvfmLetter*, *srvfmLegal*] in PageFormats3

[] in all other

2.3.6.2 Events

In TsrvActionPageSetup

■ OnChange ⁽²⁹²⁾

2.3.6.2.1 TsrvActionPageSetup.OnChange

Occurs after the changes made in the page setup dialog are applied.

property OnChange: TNotifyEvent;

2.3.7 TsrvActionPreview

TsrvActionPreview switches the editor to the preview mode and back.

Unit SRVActions;

Syntax

TsrvActionPreview = **class** (TsrvAction ⁽²⁸³⁾)

▼ Hierarchy

TObject
TPersistent
TComponent
TBasicAction
TContainedAction
TCustomAction
TAction
TrvCustomAction
TsrvAction ⁽²⁸³⁾

Description

This action toggles the value of SRichViewEdit.ViewProperty ⁽⁶⁸⁾.ViewMode ⁽¹⁶⁴⁾

2.3.8 TsrvActionPrint

TsrvActionPrint is the action for "File | Print..." command.

Unit SRVActions;

Syntax

TsrvActionPrint = **class** (TsrvCustomPrintAction ⁽²⁹⁵⁾)

▼ Hierarchy

TObject
TPersistent
TComponent
TBasicAction
TContainedAction
TCustomAction
TAction

*TrvCustomAction**TsrvAction* ⁽²⁸³⁾*TsrvCustomActionThatNeedsDocProps* ⁽²⁸⁴⁾*TsrvCustomPrintAction* ⁽²⁹⁵⁾

Description

This action displays a printing dialog (TPrintDialog) and prints the document from TSRichViewEdit ⁽⁴⁰⁾ component.

The action can print the whole document or the range of pages.

This action is enabled only if at least one printer is installed in the system (Printer.Printers.Count>0).

See also:

- TsrvActionQuickPrint ⁽²⁹³⁾
- TRVAControlPanel.MetafileCompatibility (in the RichViewActions help file)

2.3.8.1 Properties

In TsrvActionPrint

■ Title ⁽²⁹³⁾

Inherited from TsrvCustomActionThatNeedsDocProps ⁽²⁸⁴⁾

■ ActionSave ⁽²⁸⁵⁾

Inherited from TsrvAction ⁽²⁸³⁾

■ Control ⁽²⁸⁴⁾

Inherited from TrvCustomAction

■ Disabled

2.3.8.1.1 TsrvActionPrint.Title

Determines the text that is listed in the Print Manager.

property Title: **string**;

See also:

TsrvActionQuickPrint.Title ⁽²⁹⁴⁾

2.3.9 TsrvActionQuickPrint

TsrvActionQuickPrint is the action for the "quick print" command (usually assigned to a toolbar button).

Unit SRVActions;

Syntax

TsrvActionQuickPrint = **class** (TsrvCustomPrintAction ⁽²⁹⁵⁾)

▼ Hierarchy

TObject
TPersistent
TComponent
TBasicAction
TContainedAction
TCustomAction
TAction
TrvCustomAction
TsrvAction ⁽²⁸³⁾
TsrvCustomActionThatNeedsDocProps ⁽²⁸⁴⁾
TsrvCustomPrintAction ⁽²⁹⁵⁾

Description

This action prints the document from *TSRichViewEdit* ⁽⁴⁰⁾ component without displaying a printing dialog.

This action is enabled only if at least one printer is installed in the system (*Printer.Printers.Count*>0).

See also:

- *TsrvActionPrint* ⁽²⁹²⁾
- *TRVAControlPanel.MetafileCompatibility* (in the *RichViewActions* help file)

2.3.9.1 Properties

In *TsrvActionQuickPrint*

■ *Title* ⁽²⁹⁴⁾

Inherited from *TsrvCustomActionThatNeedsDocProps* ⁽²⁸⁴⁾

■ *ActionSave* ⁽²⁸⁵⁾

Inherited from *TsrvAction* ⁽²⁸³⁾

■ *Control* ⁽²⁸⁴⁾

Inherited from *TrvCustomAction*

■ *Disabled*

2.3.9.1.1 *TsrvActionQuickPrint.Title*

Determines the text that is listed in the Print Manager.

property *Title*: **string**;

See also:

TsrvActionPrint.Title ⁽²⁹³⁾

2.3.10 TsrvCustomPrintAction

TsrvCustomPrintAction is the base class for all printing RichViewActions working with TSRichViewEdit⁽⁴⁰⁾.

Unit SRVActions;

Syntax

```
TsrvCustomPrintAction = class (TsrvCustomActionThatNeedsDocProps(284))
```

▼ Hierarchy

TObject
TPersistent
TComponent
TBasicAction
TContainedAction
TCustomAction
TAction
TrvCustomAction
TsrvAction⁽²⁸³⁾
TsrvCustomActionThatNeedsDocProps⁽²⁸⁴⁾







Description

This action is not used directly, but all printing RichViewActions working with TSRichViewEdit⁽⁴⁰⁾ are inherited from it:




- TsrvActionQuickPrint⁽²⁹³⁾
- TsrvActionPrint⁽²⁹²⁾

2.4 View

View:

-  TsrvActionThumbnails⁽³¹⁰⁾ shows/hides page thumbnails.
-  TsrvActionLayoutDraft⁽²⁹⁸⁾ switches the editor to a draft layout.
-  TsrvActionLayoutWeb⁽³⁰⁹⁾ switches the editor to a web layout.
-  TsrvActionLayoutPrint⁽³⁰⁰⁾ switches the editor to a print layout.
-  TsrvActionLayoutSideToSide⁽³⁰⁶⁾ switches the editor to a side-to-side layout.
-  TsrvActionLayoutRead⁽³⁰³⁾ switches the editor to a read mode layout.

Zoom:

-  TsrvActionZoom⁽³¹²⁾ applies the specified zooming percent to the editor.
-  TsrvActionZoomPageWidth⁽³¹³⁾ zooms the editor to fit the page width in the editor width.
-  TsrvActionZoomFullPage⁽³¹³⁾ zooms the editor to fit the full page in the editor.

2.4.1 TsrvActionCustomLayout

TsrvActionCustomLayout is a base class for actions changing view modes for TSRichViewEdit⁽⁴⁰⁾ control.

Unit SRVActions;

Syntax

```
TsrvActionCustomLayout = class (TsrvCustomActionThatNeedsDocProps284)
```

▼ Hierarchy

TObject
TPersistent
TComponent
TBasicAction
TContainedAction
TCustomAction
TAction
TrvCustomAction
*TsrvAction*²⁸³
*TsrvCustomActionThatNeedsDocProps*²⁸⁴

Description

This action is not used directly.

The following actions are inherited from it:

- TsrvActionLayoutDraft²⁹⁸
- TsrvActionLayoutWeb³⁰⁹
- TsrvActionLayoutPrint³⁰⁰
- TsrvActionLayoutSideToSide³⁰⁶
- TsrvActionLayoutRead³⁰³

After the action changes a layout, OnExecuted²⁹⁷ event occurs.

These action use TrvActionSave to store additional information for the document in the target editor.

Comparison of layouts

Parameter	Draft	Web layout	Print layout	Side to side	Read mode
Multiple pages			+	+	+
Pages are displayed at their original size (with zoom applied)			+	+	
Page width depends on editor width		+			+
Page height depends on editor height					+

Auto zoom				+	
Zoom scales content without scaling pages (zoom does not affect page width)		+			+
Page flip effects				+	+

2.4.1.1 Events

In TsrvActionCustomLayout

■ Control ⁽²⁸⁴⁾

2.4.1.1.1 TsrvActionCustomLayout.OnExecuted

Occurs after the action applies the layout

property OnExecuted: TNotifyEvent;

This event may be useful if you have buttons for document view modes in the scrollbar area of TSRichViewEdit ⁽⁴⁰⁾ control.

You can use this event to synchronize buttons with the actions.

Example:

(assuming that the buttons are in the following order: draft, web layout, print layout)

```
// SRichViewEdit1.OnHMenuClickButton (109)
procedure TfrmMain.SRichViewEdit1HMenuClickButton(Sender: TObject;
  ToolButton: TSVToolButton (271));
begin
  if ToolButton = nil then Exit;
  SRichViewEdit1.CanUpdate (48) := False;
  case (ToolButton.Index) of
    0 : srvActionLayoutDraft1.Execute;
    1 : srvActionLayoutWeb1.Execute;
    2 : srvActionLayoutPrint1.Execute;
    3 : srvActionLayoutSideToSide1.Execute;
    4 : srvActionLayoutRead1.Execute;
  end;
  ActiveEditor.CanUpdate (48) := True;
end;
// After doc view actions are executed
// TsrvActionLayoutDraft (298)
procedure TfrmMain.srvActionLayoutDraft1Executed(Sender: TObject);
begin
  TSVToolButton (271) (SRichViewEdit1.MenuHButtons (55).Items (270) [0]).Down (272) :=
    True;
```

```

end;
// TsrvActionLayoutWeb(309)
procedure TfrmMain.srvActionLayoutWeb1Executed(Sender: TObject);
begin
    TSrvToolButton(271)(SRichViewEdit1.MenuHButtons(55).Items(270)[1]).Down(272) :=
        True;
end;
// TsrvActionLayoutPrint(300)
procedure TfrmMain.srvActionLayoutPrint1Executed(Sender: TObject);
begin
    TSrvToolButton(271)(SRichViewEdit1.MenuHButtons(55).Items(270)[2]).Down(272) :=
        True;
end;

```

2.4.2 TsrvActionCustomZoom

TsrvActionCustomZoom is a base class for actions zooming TSRichViewEdit⁽⁴⁰⁾ control.

Unit SRVActions;

Syntax

```
TsrvActionCustomZoom = class (TsrvAction(283))
```

▼ Hierarchy

TObject
TPersistent
TComponent
TBasicAction
TContainedAction
TCustomAction
TAction
TrvCustomAction
TsrvAction⁽²⁸³⁾

Description

This action is not used directly.

The following actions are inherited from it:

- TsrvActionZoom⁽³¹²⁾
- TsrvActionZoomPageWidth⁽³¹³⁾
- TsrvActionZoomFullPage⁽³¹³⁾

2.4.3 TsrvActionLayoutDraft

TsrvActionLayoutDraft switches TSRichViewEdit⁽⁴⁰⁾ control to a "draft" layout.

Unit SRVActions;

Syntax

```
TsrvActionLayoutDraft = class (TsrvActionCustomLayout(295))
```

▼ Hierarchy

TObject
TPersistent
TComponent
TBasicAction
TContainedAction
TCustomAction
TAction
TrvCustomAction
TsrvAction⁽²⁸³⁾
TsrvCustomActionThatNeedsDocProps⁽²⁸⁴⁾
TsrvActionCustomLayout⁽²⁹⁵⁾

Description

In this mode, a document is shown using the target editor's PageProperty⁽⁵⁸⁾.PageWidth⁽¹⁴¹⁾, but as a long single page.

The main changes made by this action for TSVRichViewEdit control:

Property	Assigned value
PageProperty ⁽⁵⁸⁾ .PageViewMode ⁽¹⁴⁰⁾	<i>False</i>
PageProperty ⁽⁵⁸⁾ .AutoWidth ⁽¹³⁵⁾	<i>False</i>
ReadModeProperty ⁽⁵⁹⁾ .Active ⁽¹⁵¹⁾	<i>False</i>

This action also changes the following properties:

Property	Assigned value
Color	<i>clWhite</i>
PageProperty ⁽⁵⁸⁾ .BorderPen ⁽¹³⁵⁾ .Style	<i>psClear</i>
PagePosProperty ⁽⁵⁸⁾ .HPadding ⁽¹⁴⁵⁾	HPadding ⁽³⁰⁰⁾
BackgroundProperty ⁽⁴⁸⁾ .Visible ⁽¹²⁶⁾	<i>False</i>
ViewProperty ⁽⁶⁸⁾ .ShowScrollHint ⁽¹⁶²⁾	<i>True</i>
PagePosProperty ⁽⁵⁸⁾ .AlignPageH ⁽¹⁴⁴⁾	<i>srvaphLeft</i>
PagePosProperty ⁽⁵⁸⁾ .AlignPageV ⁽¹⁴⁴⁾	<i>srvapvTop</i>
PageProperty ⁽⁵⁸⁾ .HeaderVisible ⁽¹³⁷⁾	<i>False</i>
PageProperty ⁽⁵⁸⁾ .FooterVisible ⁽¹³⁶⁾	<i>False</i>

PagePosProperty ⁽⁵⁸⁾ .HiddenDistances ⁽¹⁴⁵⁾	excludes [srvppdHPadding, srvppdVPadding, srvppdPageHSpacing]
---	---

After the layout is changed, TRVAControlPanel.OnViewChanged event occurs.

See also:

- TsrvActionLayoutWeb⁽³⁰⁹⁾
- TsrvActionLayoutPrint⁽³⁰⁰⁾
- TsrvActionLayoutSideToSide⁽³⁰⁶⁾
- TsrvActionLayoutRead⁽³⁰³⁾

2.4.3.1 Properties

In TsrvActionLayoutDraft

■ HPadding⁽³⁰⁰⁾

Inherited from TsrvCustomActionThatNeedsDocProps⁽²⁸⁴⁾

■ ActionSave⁽²⁸⁵⁾

Inherited from TsrvAction⁽²⁸³⁾

■ Control⁽²⁸⁴⁾

Inherited from TrvCustomAction

■ Disabled

2.4.3.1.1 TsrvActionLayoutDraft.HPadding

A value to assign to SRichViewEdit.PagePosProperty⁽⁵⁸⁾.HPadding⁽¹⁴⁵⁾.

property HPadding: TRVPixel96Length;

The action assigns this value to PagePosProperty⁽⁵⁸⁾.HPadding⁽¹⁴⁵⁾ of the target editor.

Default value

15

2.4.4 TsrvActionLayoutPrint

TsrvActionLayoutPrint switches TSRichViewEdit⁽⁴⁰⁾ control to a "print layout".

Unit SRVActions;

Syntax

```
TsrvActionLayoutPrint = class(TsrvActionCustomLayout(295))
```

▼ Hierarchy

TObject

TPersistent

TComponent

TBasicAction
TContainedAction
TCustomAction
TAction
TrvCustomAction
TsrvAction⁽²⁸³⁾
TsrvCustomActionThatNeedsDocProps⁽²⁸⁴⁾
TsrvActionCustomLayout⁽²⁹⁵⁾

Description

In this mode, a document is shown using the target editor's page size and orientation (defined in *PageProperty*⁽⁵⁸⁾. Pages are arranged in rows and columns (no more than the target editor's *PagePosProperty*⁽⁵⁸⁾.*MaxPageColCount*⁽¹⁴⁶⁾ pages in a row), scrolled vertically.

The main changes made by this action for *TSRichViewEdit* control:

Property	Assigned value
<i>PageProperty</i> ⁽⁵⁸⁾ . <i>PageViewMode</i> ⁽¹⁴⁰⁾	<i>True</i>
<i>PageProperty</i> ⁽⁵⁸⁾ . <i>AutoWidth</i> ⁽¹³⁵⁾	<i>False</i>
<i>ReadModeProperty</i> ⁽⁵⁹⁾ . <i>Active</i> ⁽¹⁵¹⁾	<i>False</i>

This action also changes the following properties:

Property	Assigned value
Color	Color ⁽³⁰²⁾
<i>PageProperty</i> ⁽⁵⁸⁾ . <i>BorderPen</i> ⁽¹³⁵⁾ . <i>Style</i>	<i>psSolid</i>
<i>PagePosProperty</i> ⁽⁵⁸⁾ . <i>HPadding</i> ⁽¹⁴⁵⁾	<i>HPadding</i> ⁽³⁰³⁾
<i>BackgroundProperty</i> ⁽⁴⁸⁾ . <i>Visible</i> ⁽¹²⁶⁾	<i>True</i>
<i>ViewProperty</i> ⁽⁶⁸⁾ . <i>ShowScrollHint</i> ⁽¹⁶²⁾	<i>True</i>
<i>PagePosProperty</i> ⁽⁵⁸⁾ . <i>AlignPageH</i> ⁽¹⁴⁴⁾	<i>srvaphCenter</i>
<i>PagePosProperty</i> ⁽⁵⁸⁾ . <i>AlignPageV</i> ⁽¹⁴⁴⁾	<i>srvaphCenter</i>
<i>PageProperty</i> ⁽⁵⁸⁾ . <i>HeaderVisible</i> ⁽¹³⁷⁾	<i>HeaderVisible</i> ⁽³⁰²⁾
<i>PageProperty</i> ⁽⁵⁸⁾ . <i>FooterVisible</i> ⁽¹³⁶⁾	<i>FooterVisible</i> ⁽³⁰²⁾
<i>PagePosProperty</i> ⁽⁵⁸⁾ . <i>HiddenDistances</i> ⁽¹⁴⁵⁾	excludes [<i>srvppdHPadding</i> , <i>srvppdVPadding</i> , <i>srvppdPageHSpacing</i>]

After the layout is changed, *TRVAControlPanel.OnViewChanged* event occurs.

See also:

- TsrvActionLayoutWeb ⁽³⁰⁹⁾
- TsrvActionLayoutDraft ⁽²⁹⁸⁾
- TsrvActionLayoutSideToSide ⁽³⁰⁶⁾
- TsrvActionLayoutRead ⁽³⁰³⁾

2.4.4.1 Properties

In TsrvActionLayoutPrint

- Color ⁽³⁰²⁾
- FooterVisible ⁽³⁰²⁾
- HeaderVisible ⁽³⁰²⁾
- HPadding ⁽³⁰³⁾

Inherited from TsrvCustomActionThatNeedsDocProps ⁽²⁸⁴⁾

- ActionSave ⁽²⁸⁵⁾

Inherited from TsrvAction ⁽²⁸³⁾

- Control ⁽²⁸⁴⁾

Inherited from TrvCustomAction

- Disabled

2.4.4.1.1 TsrvActionLayoutPrint.Color

A value to assign to SRichViewEdit.Color.

property Color: TColor;

The action assigns this value to Color property of the target editor.

Default value

\$99A8AC

2.4.4.1.2 TsrvActionLayoutPrint.FooterVisible

A value to assign to SRichViewEdit.PageProperty ⁽⁵⁸⁾.FooterVisible ⁽¹³⁶⁾.

property FooterVisible: Boolean;

The action assigns this value to PageProperty ⁽⁵⁸⁾.FooterVisible ⁽¹³⁶⁾ of the target editor.

Default value

True

2.4.4.1.3 TsrvActionLayoutPrint.HeaderVisible

A value to assign to SRichViewEdit.PageProperty ⁽⁵⁸⁾.HeaderVisible ⁽¹³⁷⁾.

property HeaderVisible: Boolean;

The action assigns this value to PageProperty ⁽⁵⁸⁾.HeaderVisible ⁽¹³⁷⁾ of the target editor.

Default value

True

2.4.4.1.4 TsrvActionLayoutPrint.HPadding

A value to assign to `SRichViewEdit.PagePosProperty(58).HPadding(145)`.

property `HPadding: TRVPixel96Length;`

The action assigns this value to `PagePosProperty(58).HPadding(145)` of the target editor.

Default value

15

2.4.5 TsrvActionLayoutRead

`TsrvActionLayoutRead` switches `TSRichViewEdit(40)` control to a "read mode" layout.

Unit `SRVActions;`

Syntax

`TsrvActionLayoutRead = class (TsrvActionCustomLayout(295))`

▼ Hierarchy

TObject
TPersistent
TComponent
TBasicAction
TContainedAction
TCustomAction
TAction
TrvCustomAction
TsrvAction⁽²⁸³⁾
TsrvCustomActionThatNeedsDocProps⁽²⁸⁴⁾
TsrvActionCustomLayout⁽²⁹⁵⁾

Description

In this mode width and height of pages are calculated automatically to fit the editor. Pages are shown completely, are arranged in one row, are scrolled horizontally with a special transition effect.

The main changes made by this action for `TSRichViewEdit` control:

Property	Assigned value
<code>PageProperty⁽⁵⁸⁾.PageViewMode⁽¹⁴⁰⁾</code>	<i>True</i>
<code>PageProperty⁽⁵⁸⁾.AutoWidth⁽¹³⁵⁾</code>	<i>True</i>
<code>ReadModeProperty⁽⁵⁹⁾.Active⁽¹⁵¹⁾</code>	<i>True</i>

This action also changes the following properties:

Property	Assigned value
<code>Color</code>	<code>Color⁽³⁰⁵⁾</code>

PageProperty ⁽⁵⁸⁾ .BorderPen ⁽¹³⁵⁾ .Style	<i>psSolid</i>
PagePosProperty ⁽⁵⁸⁾ .HPadding ⁽¹⁴⁵⁾	HPadding ⁽³⁰³⁾
BackgroundProperty ⁽⁴⁸⁾ .Visible ⁽¹²⁶⁾	<i>True</i>
ViewProperty ⁽⁶⁸⁾ .ShowScrollHint ⁽¹⁶²⁾	<i>True</i>
PagePosProperty ⁽⁵⁸⁾ .AlignPageH ⁽¹⁴⁴⁾	<i>srvaphCenter</i>
PagePosProperty ⁽⁵⁸⁾ .AlignPageV ⁽¹⁴⁴⁾	<i>srvapvCenter</i>
PageProperty ⁽⁵⁸⁾ .HeaderVisible ⁽¹³⁷⁾	<i>False</i>
PageProperty ⁽⁵⁸⁾ .FooterVisible ⁽¹³⁶⁾	<i>False</i>
PagePosProperty ⁽⁵⁸⁾ .HiddenDistances ⁽¹⁴⁵⁾	excludes [srvppdHPadding, srvppdVPadding, srvppdPageHSpacing]

This action also changes the following properties specific for a read mode:

Property	Assigned value
ReadModeProperty ⁽⁵⁹⁾ .PageFlipEffect ⁽¹⁵²⁾	PageFlipEffect ⁽³⁰⁵⁾
ReadModeProperty ⁽⁵⁹⁾ .PageColor ⁽¹⁵²⁾	PageColor ⁽³⁰⁵⁾
ReadModeProperty ⁽⁵⁹⁾ .HidePageBackgroundImage ⁽¹⁵¹⁾	HidePageBackgroundImage ⁽³⁰⁵⁾

After the layout is changed, TRVAControlPanel.OnViewChanged event occurs.

See also:

- TsrvActionLayoutDraft⁽²⁹⁸⁾
- TsrvActionLayoutWeb⁽³⁰⁹⁾
- TsrvActionLayoutPrint⁽³⁰⁰⁾
- TsrvActionLayoutSideToSide⁽³⁰⁶⁾

2.4.5.1 Properties

In TsrvActionLayoutRead

- Color⁽³⁰⁵⁾
- HidePageBackgroundImage⁽³⁰⁵⁾
- HPadding⁽³⁰⁵⁾
- PageColor⁽³⁰⁵⁾
- PageFlipEffect⁽³⁰⁵⁾

Inherited from TsrvCustomActionThatNeedsDocProps⁽²⁸⁴⁾

- ActionSave⁽²⁸⁵⁾

Inherited from TsrvAction ⁽²⁸³⁾

■ Control ⁽²⁸⁴⁾

Inherited from TrvCustomAction

■ Disabled

2.4.5.1.1 TsrvActionLayoutRead.Color

A value to assign to SRichViewEdit.Color.

```
property Color: TColor;
```

The action assigns this value to Color property of the target editor.

Default value

\$99A8AC

2.4.5.1.2 TsrvActionLayoutRead.HidePageBackgroundImage

A value to assign to SRichViewEdit.ReadModeProperty ⁽⁵⁹⁾.HidePageBackgroundImage ⁽¹⁵¹⁾

```
property HidePageBackgroundImage: Boolean;
```

The action assigns this value to ReadModeProperty ⁽⁵⁹⁾.HidePageBackgroundImage ⁽¹⁵¹⁾ property of the target editor.

Default value

False

2.4.5.1.3 TsrvActionLayoutRead.HPadding

A value to assign to SRichViewEdit.PagePosProperty ⁽⁵⁸⁾.HPadding ⁽¹⁴⁵⁾.

```
property HPadding: TRVPixel96Length;
```

The action assigns this value to PagePosProperty ⁽⁵⁸⁾.HPadding ⁽¹⁴⁵⁾ of the target editor.

Default value

15

2.4.5.1.4 TsrvActionLayoutRead.PageColor

A value to assign to SRichViewEdit.ReadModeProperty ⁽⁵⁹⁾.PageColor ⁽¹⁵²⁾

```
property PageColor: TColor;
```

The action assigns this value to ReadModeProperty ⁽⁵⁹⁾.PageColor ⁽¹⁵²⁾ property of the target editor.

Default value

c/None

2.4.5.1.5 TsrvActionLayoutRead.PageFlipEffect

A value to assign to SRichViewEdit.ReadModeProperty ⁽⁵⁹⁾.PageFlipEffect ⁽¹⁵²⁾.

```
property PageFlipEffect: TSRVPageFlipEffect;
```

The action assigns this value to ReadModeProperty ⁽⁵⁹⁾.PageFlipEffect ⁽¹⁵²⁾ of the target editor.

Default value*srvpfeScroll*

2.4.6 TsrvActionLayoutSideToSide

TsrvActionLayoutSideToSide switches TSRichViewEdit⁽⁴⁰⁾ control to a "side to side" layout.

Unit SRVActions;

Syntax

```
TsrvActionLayoutSideToSide = class (TsrvActionCustomLayout(295))
```

▼ Hierarchy

TObject
TPersistent
TComponent
TBasicAction
TContainedAction
TCustomAction
TAction
TrvCustomAction
TsrvAction⁽²⁸³⁾
TsrvCustomActionThatNeedsDocProps⁽²⁸⁴⁾
TsrvActionCustomLayout⁽²⁹⁵⁾

Description

In this mode, a document is shown using the target editor's page size and orientation (defined in PageProperty)⁽⁵⁸⁾. Pages are shown completely, are arranged in one row, are scrolled horizontally with a special transition effect.

The main changes made by this action for TSRichViewEdit control:

Property	Assigned value
PageProperty ⁽⁵⁸⁾ .PageViewMode ⁽¹⁴⁰⁾	<i>True</i>
PageProperty ⁽⁵⁸⁾ .AutoWidth ⁽¹³⁵⁾	<i>False</i>
ReadModeProperty ⁽⁵⁹⁾ .Active ⁽¹⁵¹⁾	<i>True</i>

This action also changes the following properties:

Property	Assigned value
Color	Color ⁽³⁰⁸⁾
PageProperty ⁽⁵⁸⁾ .BorderPen ⁽¹³⁵⁾ .Style	<i>psSolid</i>
PagePosProperty ⁽⁵⁸⁾ .HPadding ⁽¹⁴⁵⁾	HPadding ⁽³⁰⁸⁾
BackgroundProperty ⁽⁴⁸⁾ .Visible ⁽¹²⁶⁾	<i>True</i>

ViewProperty ⁶⁸ .ShowScrollHint ¹⁶²	<i>True</i>
PagePosProperty ⁵⁸ .AlignPageH ¹⁴⁴	<i>srvaphCenter</i>
PagePosProperty ⁵⁸ .AlignPageV ¹⁴⁴	<i>srvapvCenter</i>
PageProperty ⁵⁸ .HeaderVisible ¹³⁷	HeaderVisible ³⁰⁸
PageProperty ⁵⁸ .FooterVisible ¹³⁶	FooterVisible ³⁰⁸
PagePosProperty ⁵⁸ .HiddenDistances ¹⁴⁵	excludes [srvppdHPadding, srvppdVPadding], excluded/includes [srvppdPageHSpacing] depending on NoPageSpacing ³⁰⁸

This action also changes the following properties specific for a read mode:

Property	Assigned value
ReadModeProperty ⁵⁹ .PageFlipEffect ¹⁵²	PageFlipEffect ³⁰⁹
ReadModeProperty ⁵⁹ .PageColor ¹⁵²	<i>c/None</i>
ReadModeProperty ⁵⁹ .HidePageBackgroundImage ¹⁵¹	<i>False</i>

After the layout is changed, TRVAControlPanel.OnViewChanged event occurs.

See also:

- TsrvActionLayoutDraft²⁹⁸
- TsrvActionLayoutWeb³⁰⁹
- TsrvActionLayoutPrint³⁰⁰
- TsrvActionLayoutRead³⁰³

2.4.6.1 Properties

In TsrvActionLayoutSideToSide

- Color³⁰⁸
- FooterVisible³⁰⁸
- HeaderVisible³⁰⁸
- HPadding³⁰⁸
- NoPageSpacing³⁰⁸
- PageFlipEffect³⁰⁹

Inherited from TsrvCustomActionThatNeedsDocProps²⁸⁴

- ActionSave²⁸⁵

Inherited from TsrvAction ⁽²⁸³⁾

■ Control ⁽²⁸⁴⁾

Inherited from TrvCustomAction

■ Disabled

2.4.6.1.1 TsrvActionLayoutSideToSide.Color

A value to assign to SRichViewEdit.Color.

property Color: TColor;

The action assigns this value to Color property of the target editor.

Default value

\$99A8AC

2.4.6.1.2 TsrvActionLayoutSideToSide.FooterVisible

A value to assign to SRichViewEdit.PageProperty ⁽⁵⁸⁾.FooterVisible ⁽¹³⁶⁾.

property FooterVisible: Boolean;

The action assigns this value to PageProperty ⁽⁵⁸⁾.FooterVisible ⁽¹³⁶⁾ of the target editor.

Default value

True

2.4.6.1.3 TsrvActionLayoutSideToSide.HeaderVisible

A value to assign to SRichViewEdit.PageProperty ⁽⁵⁸⁾.HeaderVisible ⁽¹³⁷⁾.

property HeaderVisible: Boolean;

The action assigns this value to PageProperty ⁽⁵⁸⁾.HeaderVisible ⁽¹³⁷⁾ of the target editor.

Default value

True

2.4.6.1.4 TsrvActionLayoutSideToSide.HPadding

A value to assign to SRichViewEdit.PagePosProperty ⁽⁵⁸⁾.HPadding ⁽¹⁴⁵⁾.

property HPadding: TRVPixel96Length;

The action assigns this value to PagePosProperty ⁽⁵⁸⁾.HPadding ⁽¹⁴⁵⁾ of the target editor.

Default value

15

2.4.6.1.5 TsrvActionLayoutSideToSide.NoPageSpacing

Turns spacing between pages on/off.

property NoPageSpacing: Boolean;

Depending on the value of this property, the action excludes/includes *srvppdPageHSpacing* in PagePosProperty ⁽⁵⁸⁾.HiddenDistances ⁽¹⁴⁵⁾ of the target editor.

Default value*False***2.4.6.1.6 TsrvActionLayoutSideToSide.PageFlipEffect**

A value to assign to `SRichViewEdit.ReadModeProperty(59).PageFlipEffect(152)`.

property `PageFlipEffect: TSRVPageFlipEffect;`

The action assigns this value to `ReadModeProperty(59).PageFlipEffect(152)` of the target editor.

Default value*srvpfeStack***2.4.7 TsrvActionLayoutWeb**

`TsrvActionLayoutWeb` switches `TSRichViewEdit(40)` control to a "web layout".

Unit `SRVActions;`

Syntax

```
TsrvActionLayoutWeb = class (TsrvActionCustomLayout(295))
```

▼ Hierarchy*TObject**TPersistent**TComponent**TBasicAction**TContainedAction**TCustomAction**TAction**TrvCustomAction**TsrvAction⁽²⁸³⁾**TsrvCustomActionThatNeedsDocProps⁽²⁸⁴⁾**TsrvActionCustomLayout⁽²⁹⁵⁾***Description**

In this mode, a document is shown in a single page that occupies all available width of the target editor.

The main changes made by this action for `TSRichViewEdit` control:

Property	Assigned value
<code>PageProperty⁽⁵⁸⁾.PageViewMode⁽¹⁴⁰⁾</code>	<i>False</i>
<code>PageProperty⁽⁵⁸⁾.AutoWidth⁽¹³⁵⁾</code>	<i>True</i>
<code>ReadModeProperty⁽⁵⁹⁾.Active⁽¹⁵¹⁾</code>	<i>False</i>

This action also changes the following properties:

Property	Assigned value
Color	<i>clWhite</i>
PageProperty ⁽⁵⁸⁾ .BorderPen ⁽¹³⁵⁾ .Style	<i>psClear</i>
BackgroundProperty ⁽⁴⁸⁾ .Visible ⁽¹²⁶⁾	<i>False</i>
ViewProperty ⁽⁶⁸⁾ .ShowScrollHint ⁽¹⁶²⁾	<i>False</i>
PagePosProperty ⁽⁵⁸⁾ .AlignPageH ⁽¹⁴⁴⁾	<i>srvaphLeft</i>
PagePosProperty ⁽⁵⁸⁾ .AlignPageV ⁽¹⁴⁴⁾	<i>srvapvTop</i>
PageProperty ⁽⁵⁸⁾ .HeaderVisible ⁽¹³⁷⁾	<i>False</i>
PageProperty ⁽⁵⁸⁾ .FooterVisible ⁽¹³⁶⁾	<i>False</i>
PagePosProperty ⁽⁵⁸⁾ .HiddenDistances ⁽¹⁴⁵⁾	includes [<i>srvppdHPadding</i> , <i>srvppdVPadding</i>], excludes [<i>srvppdPageHSpacing</i>]

After the layout is changed, TRVAControlPanel.OnViewChanged event occurs.

See also:

- TsrvActionLayoutPrint⁽³⁰⁰⁾
- TsrvActionLayoutDraft⁽²⁹⁸⁾
- TsrvActionLayoutSideToSide⁽³⁰⁶⁾
- TsrvActionLayoutRead⁽³⁰³⁾

2.4.8 TsrvActionThumbnails

TsrvActionThumbnails shows/hides page thumbnails for the editor.

Unit SRVActions;

Syntax

```
TsrvActionThumbnails = class (TsrvAction(283))
```

▼ Hierarchy

TObject
TPersistent
TComponent
TBasicAction
TContainedAction
TCustomAction
TAction
TrvCustomAction
TsrvAction⁽²⁸³⁾

Description

This action links/unlinks PageScroll⁽³¹¹⁾ to TSRichViewEdit⁽⁴⁰⁾ control.

It assigns PageScroll⁽³¹¹⁾.SRichViewEdit⁽¹⁸²⁾ property, then calls OnExecuted⁽³¹¹⁾. In this event, you can perform additional operations, for example showing/hiding PageScroll⁽³¹¹⁾.

2.4.8.1 Properties

In TsrvActionThumbnails

- PageScroll⁽³¹¹⁾

Inherited from TsrvAction⁽²⁸³⁾

- Control⁽²⁸⁴⁾

Inherited from TrvCustomAction

- Disabled

2.4.8.1.1 TsrvActionThumbnails.PageScroll

A link to TSRVPageScroll⁽¹⁷³⁾ component.

```
property PageScroll: TSRVPageScroll(173);
```

2.4.8.2 Events

In TsrvActionThumbnails

- OnExecuted⁽³¹¹⁾

2.4.8.2.1 TsrvActionThumbnails.OnExecuted

Occurs after the action links/unlinks PageScroll⁽³¹¹⁾ to TSRichViewEdit⁽⁴⁰⁾ control.

```
property OnExecuted: TNotifyEvent;
```

Example:

```
procedure TForm3.srvActionThumbnails1Executed(Sender: TObject);  
begin  
    Splitter1.Visible := TsrvActionThumbnails(310)(Sender).Checked;  
    SRVPageScroll1.Visible := TsrvActionThumbnails(310)(Sender).Checked;  
end;
```

2.4.9 TsrvActionZoom

TsrvActionZoom zooms T^{SRichViewEdit}⁽⁴⁰⁾ control to the specified percent.

Unit SRVActions;

Syntax

```
TsrvActionZoom = class(TsrvActionCustomZoom(298))
```

▼ Hierarchy

TObject
TPersistent
TComponent
TBasicAction
TContainedAction
TCustomAction
TAction
TrvCustomAction
TsrvAction⁽²⁸³⁾
TsrvActionCustomZoom⁽²⁹⁸⁾

Description

This action assigns the following properties of the target editor:

- ViewProperty⁽⁶⁸⁾.ZoomPercent⁽¹⁶⁷⁾ := ZoomPercent⁽³¹²⁾;
- ViewProperty⁽⁶⁸⁾.ZoomMode⁽¹⁶⁵⁾ := *rvzmCustom*.

See also:

- TsrvActionZoomPageWidth⁽³¹³⁾
- TsrvActionZoomFullPage⁽³¹³⁾

2.4.9.1 Properties

Inherited from TsrvActionCustomZoom⁽²⁹⁸⁾

■ ZoomPercent⁽³¹²⁾

Inherited from TsrvAction⁽²⁸³⁾

■ Control⁽²⁸⁴⁾

Inherited from TrvCustomAction

■ Disabled

2.4.9.1.1 TsrvActionZoom.ZoomPercent

Zooming percent.

```
property ZoomPercent: Single;
```

This value is assigned to ViewProperty⁽⁶⁸⁾.ZoomPercent⁽¹⁶⁷⁾ of the target editor.

When you change value of this property, Caption and Hint are updated.

Default value:

100

2.4.10 TsrvActionZoomFullPage

TsrvActionZoomFullPage zooms TSVRichViewEdit⁽⁴⁰⁾ control to fit the full page in the editor.

Unit SRVActions;

Syntax

```
TsrvActionZoomFullPage = class (TsrvActionCustomZoom(298))
```

▼ Hierarchy

TObject
TPersistent
TComponent
TBasicAction
TContainedAction
TCustomAction
TAction
TrvCustomAction
TsrvAction⁽²⁸³⁾
TsrvActionCustomZoom⁽²⁹⁸⁾

Description

This action assigns the following property of the target editor:

- ViewProperty⁽⁶⁸⁾.ZoomMode⁽¹⁶⁵⁾ := *rvzmFullPage*

See also:

- TsrvActionZoom⁽³¹²⁾
- TsrvActionZoomPageWidth⁽³¹³⁾

2.4.11 TsrvActionZoomPageWidth

TsrvActionZoomPageWidth zooms TSVRichViewEdit⁽⁴⁰⁾ control to fit the page width in the editor width.

Unit SRVActions;

Syntax

```
TsrvActionZoomPageWidth = class (TsrvActionCustomZoom(298))
```

▼ Hierarchy

TObject
TPersistent
TComponent
TBasicAction
TContainedAction
TCustomAction

TAction

TrvCustomAction

TsrvAction ⁽²⁸³⁾

TsrvActionCustomZoom ⁽²⁹⁸⁾

Description

This action assigns the following property of the target editor:


- `ViewProperty` ⁽⁶⁸⁾.`ZoomMode` ⁽¹⁶⁵⁾ := `rvzmPageWidth`.


See also:


- `TsrvActionZoom` ⁽³¹²⁾
- `TsrvActionZoomFullPage` ⁽³¹³⁾

2.5 Editing

Header and footer:

 `TsrvActionEditHeader` ⁽³¹⁴⁾ starts editing a page header.

 `TsrvActionEditFooter` ⁽³¹⁴⁾ starts editing a page footer.

 `TsrvActionEditMain` ⁽³¹⁵⁾ closes a header/footer editor and returns the main document.

2.5.1 TsrvActionEditHeader

Starts editing a page header.

Unit SRVActions;

Syntax

```
TsrvActionEditHeader = class (TsrvAction (283))
```

▼ **Hierarchy**

TObject

TPersistent

TComponent

TBasicAction

TContainedAction

TCustomAction

TAction

TrvCustomAction

TsrvAction ⁽²⁸³⁾

Description

This action calls `StartEditing` ⁽⁹⁸⁾ (`srvrveHeader`)

2.5.2 TsrvActionEditFooter

Starts editing a page footer.

Unit SRVActions;

Syntax

```
TsrvActionEditFooter = class (TsrvAction283)
```

▼ Hierarchy

TObject
TPersistent
TComponent
TBasicAction
TContainedAction
TCustomAction
TAction
TrvCustomAction
*TsrvAction*²⁸³

Description

This action calls StartEditing⁹⁸ (*srvrveFooter*)

2.5.3 TsrvActionEditMain

Closes header/footer editors and starts editing the main document.

Unit SRVActions;

Syntax

```
TsrvActionEditMain = class (TsrvAction283)
```

▼ Hierarchy





TObject
TPersistent
TComponent
TBasicAction
TContainedAction
TCustomAction
TAction
TrvCustomAction
*TsrvAction*²⁸³

Description


This action calls StartEditing⁹⁸ (*srvrveMain*)

2.6 Notes

Notes and text boxes:

-  TsrvActionInsertFootnote³¹⁹ inserts a footnote.
-  TsrvActionInsertEndnote³¹⁸ inserts an endnote.
-  TsrvActionInsertSidenote³²¹ inserts a sidenote (a note in a floating box).
-  TsrvActionInsertTextBox³²³ inserts a text box item.

 TsrvActionEditNote⁽³¹⁸⁾ starts editing a note or a text box.

 TsrvActionReturnToNote⁽³²⁴⁾ moves the caret to the parent note.

2.6.1 TsrvActionCustomInsertSidenote

TsrvActionCustomInsertSidenote is the base class for actions for inserting sidenotes and text box items.

Unit SRVActions;

Syntax

```
TsrvActionCustomInsertSidenote = class (TsrvActionInsertNote(320))
```

▼ Hierarchy

TObject
TPersistent
TComponent
TBasicAction
TContainedAction
TCustomAction
TAction
TrvCustomAction
TsrvAction⁽²⁸³⁾
TsrvActionInsertNote⁽³²⁰⁾

Description

This action introduces properties that are assigned to properties of an inserted sidenote (or a text box item):

- BoxProperties⁽³¹⁷⁾
- BoxPosition⁽³¹⁷⁾

Text in the note's Document is formatted either according to StyleTemplateName⁽³¹⁷⁾ (if style templates are used) or according to Font⁽³²¹⁾ (otherwise).

This class is not used directly. The following actions are inherited from TrvActionCustomInsertSidenote:

- TrvActionInsertSidenote⁽³²¹⁾
- TrvActionInsertTextBox⁽³²³⁾

2.6.1.1 Properties

In TsrvActionCustomInsertSidenote

- BoxPosition⁽³¹⁷⁾
- BoxProperties⁽³¹⁷⁾
- StyleTemplateName⁽³¹⁷⁾

Inherited from TsrvActionInsertNote⁽³²⁰⁾

- Font⁽³²¹⁾

Inherited from TsrvAction ⁽²⁸³⁾

■ Control ⁽²⁸⁴⁾

Inherited from TrvCustomAction

■ Disabled

2.6.1.1.1 TsrvActionCustomInsertSidenote.BoxPosition

Specifies position properties for the floating box.

property BoxPosition: TRVABoxPosition;

This property is assigned to BoxPosition property of the inserted sidenote / text box item.

Default values are different in inherited actions.

TsrvActionInsertSidenote ⁽³²¹⁾:

- HorizontalAnchor=*rvhanMainTextArea*;
- HorizontalPositionKind=*rvfpkAlignment*;
- HorizontalAlignment=*rvfphalRight*;
- VerticalAnchor=*rvvanParagraph*;
- VerticalPositionKind=*rvfpkAbsPosition*;
- VerticalOffset=0.

TsrvActionInsertTextBox ⁽³²³⁾:

- HorizontalAnchor=*rvhanMainTextArea*;
- HorizontalPositionKind=*rvfpkAlignment*;
- HorizontalAlignment=*rvfphalCenter*;
- VerticalAnchor=*rvvanLine*;
- VerticalPositionKind=*rvfpkAbsPosition*;
- VerticalOffset=20.

2.6.1.1.2 TsrvActionCustomInsertSidenote.BoxProperties

Specifies size, background, border, vertical alignment properties for the floating box.

property BoxProperties: TRVABoxProperties;

This property is assigned to BoxProperties property of the inserted sidenote / text box item.

Default values

- WidthType=*rvbwtRelPage*;
- Width=20000;

2.6.1.1.3 TsrvActionCustomInsertSidenote.StyleTemplateName

Specifies the name of a style template that will be applied to a paragraph of a sidenote's Document.

property StyleTemplateName: TRVStyleTemplateName;

This property is used only if the target editor's UseStyleTemplates ⁽⁶⁸⁾ = *True*. Otherwise, the 0-th paragraph style and Font ⁽³²¹⁾ property are used.

Default values are different in inherited actions:

TsrvActionInsertSidenote⁽³²¹⁾:

'Sidenote Text' (note: this style template is standard for RichViewActions, but it is not standard for Microsoft Word)

TsrvActionInsertTextBox⁽³²³⁾:

'Normal'

2.6.2 TsrvActionEditNote

TsrvActionEditNote starts editing the current or the next footnote/endnote/sidenote/text box item.

Unit SRVActions;

Syntax

```
TsrvActionEditNote = class (TsrvAction(283))
```

▼ Hierarchy

TObject
TPersistent
TComponent
TBasicAction
TContainedAction
TCustomAction
TAction
TrvCustomAction
TsrvAction⁽²⁸³⁾

Description

If the current item in TSRichViewEdit.RichViewEdit⁽⁵⁹⁾ is a note, this action starts editing it.

Otherwise, the action searches for the next note from the caret position. If found, this action starts editing it.

Use this action instead of TrvActionEditNote.

2.6.3 TsrvActionInsertEndnote

TsrvActionInsertEndnote inserts an endnote in the caret position.

Unit SRVActions;

Syntax

```
TsrvActionInsertEndnote = class (TsrvActionInsertNote(320))
```

▼ Hierarchy

TObject
TPersistent
TComponent
TBasicAction
TContainedAction

TCustomAction
TAction
TrvCustomAction
TsrvAction ⁽²⁸³⁾
TsrvActionInsertNote ⁽³²⁰⁾

Description

If *TSRichViewEdit.RichViewEdit* ⁽⁵⁹⁾ is active, this action inserts an endnote (*TRVEndnoteItemInfo* item).

If *TSRichViewEdit.RVNote* ⁽⁶⁴⁾ is active (editing a *endnote.Document*), this action inserts a reference to the parent endnote (*TRVNoteReferenceItemInfo* item).

Otherwise, this action is disabled.

When inserting an endnote, this action adds in its *Document* the following items:

- reference to this endnote;
- space character.

Attributes of inserted items depend on the target editor's *UseStyleTemplates* ⁽⁶⁸⁾ property.

If style templates are used, the endnote and the endnote reference characters are formatted using "endnote reference" style template. Otherwise, or if this style template does not exist, they are formatted as superscript text. In the both cases, *rvprDoNotAutoSwitch* is included in its *Protection*.

If style templates are used, endnote paragraph and text are formatted using "endnote text" style template. Otherwise, or if this style template does not exist, endnote text is formatted using *Font* ⁽³²¹⁾ property.

Note: "endnote reference" and "endnote text" style templates are added by default to *TrvActionNew.StyleTemplates*, so they are included in new documents.

Use this action instead of *TrvActionInsertEndnote*.

2.6.4 TsrvActionInsertFootnote

TsrvActionInsertFootnote inserts a footnote in the caret position.

Unit *SRVActions*;

Syntax

```
TsrvActionInsertFootnote = class (TsrvActionInsertNote (320))
```

▼ Hierarchy

TObject
TPersistent
TComponent
TBasicAction
TContainedAction
TCustomAction
TAction
TrvCustomAction
TsrvAction ⁽²⁸³⁾

TsrvActionInsertNote ⁽³²⁰⁾

Description

If *TSRichViewEdit.RichViewEdit* ⁽⁵⁹⁾ is active, this action inserts a footnote (*TRVFootnoteItemInfo* item).

If *TSRichViewEdit.RVNote* ⁽⁶⁴⁾ is active (editing a *footnote.Document*), this action inserts a reference to the parent footnote (*TRVNoteReferenceItemInfo* item).

Otherwise, this action is disabled.

When inserting a footnote, this action adds in its *Document* the following items:

- reference to this footnote;
- space character.

Attributes of inserted items depend on the target editor's *UseStyleTemplates* ⁽⁶⁸⁾ property.

If style templates are used, the footnote and the footnote reference characters are formatted using "footnote reference" style template. Otherwise, or if this style template does not exist, they are formatted as superscript text. In the both cases, *rvprDoNotAutoSwitch* is included in its *Protection*.

If style templates are used, footnote paragraph and text are formatted using "footnote text" style template. Otherwise, or if this style template does not exist, footnote text is formatted using *Font* ⁽³²¹⁾ property.

Note: "footnote reference" and "footnote text" style templates are added by default to *TrvActionNew.StyleTemplates*, so they are included in new documents.

Use this action instead of *TrvActionInsertFootnote*.

2.6.5 TsrvActionInsertNote

TsrvActionInsertNote is the base class for actions for inserting footnotes, endnotes, sidenotes and text box items.

Unit SRVActions;

Syntax

```
TsrvActionInsertNote = class (TsrvAction (283))
```

▼ Hierarchy

TObject
TPersistent
TComponent
TBasicAction
TContainedAction
TCustomAction
TAction
TrvCustomAction
TsrvAction ⁽²⁸³⁾

Description

This action is not used directly.

The following actions are inherited from it:

- TsrvActionInsertFootnote⁽³¹⁹⁾
- TsrvActionInsertEndnote⁽³¹⁸⁾
- TsrvActionInsertSidenote⁽³²¹⁾
- TsrvActionInsertTextBox⁽³²³⁾

2.6.5.1 Properties

In TsrvActionInsertNote

- Font⁽³²¹⁾

Inherited from TsrvAction⁽²⁸³⁾

- Control⁽²⁸⁴⁾

Inherited from TrvCustomAction

- Disabled

2.6.5.1.1 TsrvActionInsertNote.Font

Default font for text in a note's document.

type

```
TSRVFont = class (TFont)
```

```
property Font: TSRVFont;
```

This value is used if UseStyleTemplates⁽⁶⁸⁾ = *False* for the target editor, or the required style template does not exist in Editor.RichViewEdit⁽⁵⁹⁾.Style.StyleTemplates. Otherwise, note text is formatted according to a style template.

Style template names:

- "footnote text" for footnotes,
- "endnote text" for endnotes,
- StyleTemplateName for sidenotes and text boxes (default values: "Sidenote Text" for sidenotes, "Normal" for text boxes)

Default value:

'Arial', 10

2.6.6 TsrvActionInsertSidenote

TsrvActionInsertSidenote inserts a sidenote (a note in a floating box) in the caret position.

Unit SRVActions;

Syntax

```
TsrvActionInsertSidenote = class (TsrvActionCustomInsertSidenote(316))
```

▼ Hierarchy

TObject
TPersistent
TComponent
TBasicAction
TContainedAction
TCustomAction
TAction
TrvCustomAction
TsrvAction ⁽²⁸³⁾
TsrvActionInsertNote ⁽³²⁰⁾
TsrvActionCustomInsertSidenote ⁽³¹⁶⁾

Description

If *TSRichViewEdit.RichViewEdit* ⁽⁵⁹⁾, *RVHeader* ⁽⁶³⁾ or *RVFooter* ⁽⁶¹⁾ is active, this action inserts a sidenote (*TRVSidenoteItemInfo* item).

If *TSRichViewEdit.RVNote* ⁽⁶⁴⁾ is active (editing a *sidenote.Document*), this action inserts a reference to the parent sidenote (*TRVNoteReferenceItemInfo* item).

Otherwise, this action is disabled.

When inserting a sidenote, this action adds in its *Document* the following items:

- reference to this sidenote;
- space character.

Attributes of inserted items depend on the target editor's *UseStyleTemplates* ⁽⁶⁸⁾ property.

If style templates are used, the sidenote and the sidenote reference characters are formatted using *RefStyleTemplateName* ⁽³²³⁾ style template. Otherwise, or if this style template does not exist, they are formatted as superscript text. In the both cases, *rvprDoNotAutoSwitch* is included in its *Protection*.

If style templates are used, sidenote paragraph and text are formatted using *StyleTemplateName* ⁽³¹⁷⁾ style template. Otherwise, or if this style template does not exist, sidenote text is formatted using *Font* ⁽³²¹⁾ property.

The following properties are assigned to the inserted sidenote:

- *BoxProperties* ⁽³¹⁷⁾ (default: width: 20% of page width, height: auto)
- *BoxPosition* ⁽³¹⁷⁾ (default: horizontal - right aligned to the main text area, vertical - aligned to the top of the paragraph)

Use this action instead of *TrvActionInsertSidenote*.

2.6.6.1 Properties

In *TsrvActionInsertSidenote*

- *RefStyleTemplateName* ⁽³²³⁾

Inherited from *TsrvActionCustomInsertSidenote* ⁽³¹⁶⁾

- *BoxPosition* ⁽³¹⁷⁾
- *BoxProperties* ⁽³¹⁷⁾

■ StyleTemplateName⁽³¹⁷⁾

Inherited from TsrvActionInsertNote⁽³²⁰⁾

■ Font⁽³²¹⁾

Inherited from TsrvAction⁽²⁸³⁾

■ Control⁽²⁸⁴⁾

Inherited from TrvCustomAction

■ Disabled

2.6.6.1.1 TsrvActionInsertSidenote.RefStyleTemplateName

Specifies the name of a style template that will be applied to the sidenote character and to the reference of to the parent sidenote (inserted in the sidenote.Document)

property RefStyleTemplateName: TRVStyleTemplateName;

This property is used only if the target editor's UseStyleTemplates⁽⁶⁸⁾ = *True*. Otherwise, these characters are formatted as a superscript.

Default value

'Sidenote Reference' (note: this style template is standard for RichViewActions, but it is not standard for Microsoft Word)

2.6.7 TsrvActionInsertTextBox

TsrvActionInsertTextBox inserts a text box item in the caret position.

Unit SRVActions;

Syntax

TsrvActionInsertTextBox = **class**(TsrvActionCustomInsertSidenote⁽³¹⁶⁾)

▼ Hierarchy

TObject
TPersistent
TComponent
TBasicAction
TContainedAction
TCustomAction
TAction
TrvCustomAction
TsrvAction⁽²⁸³⁾
TsrvActionInsertNote⁽³²⁰⁾
TsrvActionCustomInsertSidenote⁽³¹⁶⁾

Description

If TSRichViewEdit.RichViewEdit⁽⁵⁹⁾, RVHeader⁽⁶³⁾ or RVFooter⁽⁶¹⁾ is active, this action inserts a text box item (TRVTextBoxItemInfo item).

Otherwise, this action is disabled.

When inserting a text box, this action adds a single empty text item in its Document .

Attributes of this text item depend on the target editor's UseStyleTemplates⁽⁶⁸⁾ property.

If style templates are used, this text item's paragraph and text are formatted using StyleTemplateName⁽³¹⁷⁾ style template. Otherwise, or if this style template does not exist, this text is formatted using Font⁽³²¹⁾ property.

The following properties are assigned to the inserted sidenote:

- BoxProperties⁽³¹⁷⁾ (default: width: 20% of page width, height: auto)
- BoxPosition⁽³¹⁷⁾ (default: horizontal - centered in the main text area, vertical - 20 pixels below the top of the line)

Use this action instead of TrvActionInsertSidenote.

2.6.8 TsrvActionReturnToNote

TsrvActionReturnToNote ends editing a footnote/endnote/sidenote/text box.

Unit SRVActions;

Syntax

```
TsrvActionReturnToNote = class (TsrvAction(283))
```

▼ Hierarchy

TObject
TPersistent
TComponent
TBasicAction
TContainedAction
TCustomAction
TAction
TrvCustomAction
TsrvAction⁽²⁸³⁾

Description

This action is enabled only if a note or a text box is being edited in TSRichViewEdit control.

The action activates the main document editor (RichViewEdit⁽⁵⁹⁾) and moves the caret to the note.

3 SRVControls

SRVControls is a set of controls designed for insertion in TSRichViewEdit⁽⁴⁰⁾.

Controls:



TSRVButton⁽³⁴²⁾ – button;





TSRVCheckBox⁽³⁴⁴⁾ – check box;



TSRVComboBox⁽³⁴⁵⁾ – combo box; hierarchical; with images, custom colors and fonts; with hints


(suggestions);


 **TSRVEdit** ⁽³⁴⁸⁾ – plain-text one-line editor with hints (suggestions);


 **TSRVGroupBox** ⁽³⁵¹⁾ – panel with a title;

 **TSRVImagesScroll** ⁽³⁵³⁾ – gallery of images;

 **TSRVLabel** ⁽³⁵⁷⁾ – text label;

 **TSRVListBox** ⁽³⁵⁸⁾ – scrollable list of items; hierarchical; with images, check boxes, custom colors and fonts;

 **TSRVMemo** ⁽³⁶¹⁾ – plain-text multi-line editor;

 **TSRVPaintBox** ⁽³⁶³⁾ – controls for custom drawing;

 **TSRVPanel** ⁽³⁶⁴⁾ – panel;

 **TSRVRadioButton** ⁽³⁶⁶⁾ – radio button.

See also:


 **TSRVScrollBar** ⁽²⁴²⁾;

 **TSRVTabSet** ⁽²⁴⁸⁾.


All the controls above can be skinned using **TSRVSkinManager** ⁽²²⁷⁾.

Data-aware controls:


 **TSRVDBCheckBox** ⁽³⁶⁸⁾ – data-aware check box;

 **TSRVDBComboBox** ⁽³⁴⁵⁾ – data-aware combo box; hierarchical; with images, custom colors and fonts; with hints (suggestions);

 **TSRVDBEdit** ⁽³⁷⁰⁾ – data-aware plain-text one-line editor with hints (suggestions);

 **TSRVDBListBox** ⁽³⁷¹⁾ – data-aware scrollable list of items; hierarchical; with images, check boxes, custom colors and fonts;

 **TSRVDBMemo** ⁽³⁷²⁾ – data-aware plain-text multi-line editor;

 **TSRVDBText** ⁽³⁷³⁾ – data-aware text label.

How to use

SRVControls may work both on a form (like any other Delphi controls) or inserted in **TSRichViewEdit** ⁽⁴⁰⁾ / **TDBSRichViewEdit** ⁽¹⁶⁸⁾.

While you can insert any control in **TSRichViewEdit**, SRVControls work better in this mode:

- they provide high-quality scaling when displaying in the editor;
- they provide high-quality printing;
- they are scaled properly if `Screen.PixelsPerInch <> 96` (but only in Delphi 10.1 and newer).

If you create SRVControls at run-time in code, it's highly recommended (instead of a direct assigning of its Parent) to use **RVSetControlParent** function (defined in **RVFuncs** unit). This recommendation is in effect both to controls that will be added to a form, and to controls that will be inserted in the editor (in the latter case, **RVSetControlParent** should be called before calling **InsertItem** or **AddItem** methods):

```
c := TSRVCheckBox344.Create(nil);
RVSetControlParent(c, SRichViewEdit1.ActiveEditor47);
c.Caption := 'Default';
SRichViewEdit1.ActiveEditor47.InsertControl(' ', c, rvvaMiddle);
```

GDI+ notes

SRVControls can use either standard Windows drawing functions (GDI), or advanced GDI+ drawing functions. GDI+ functions can be used if their SRVControlStyle = *srvcSimple*²⁷⁵.

GDI+ functions provide high-quality anti-aliased drawing, and SRVControls looks much better in this mode.

For Delphi: GDI+ is used automatically in Delphi XE2 or newer. It cannot be used on older versions of Delphi.

For Lazarus: GDI+ can be used optionally. To use it:

1. Download **GDI+ Library for Delphi and Lazarus** and compile lazgdiplus.lpk. Alternatively, you can install it using Lazarus Online package manager (menu "Package | Online Package Manager")
2. Compile <TRichView Dir>\TRichView\Source\rvgdipluslaz.lpk (Note: it is included in <TRichView Dir>\TRichViewLazarus_Optional.lpg project group)
3. Include a reference to rvgdipluslaz package in your project (Menu "Project | Project Inspector" to open Project Inspector window. Right click "Required Packages", select menu "Add...". Select RVGDIPlusLaz)
4. Include the unit RVGDIPlusGrInLaz in "uses" of the main form unit.

Note: there is an alternative GDI+ support package for Lazarus (*rvpgdipluslaz.lpk*, containing RVPGDIPlusGrInLaz unit). It uses GDI+ library by Progdi. It is not available now at its original location, but **can be downloaded from our website**. Still, I recommend using the package above.

3.1 Ancestor classes

Parent Classes for SRVControls:

- TCustomSRVEdit³²⁶
- TCustomSRVGraphicControl³²⁸
- TCustomSRVPanel³²⁹
- TSRVCanvasControl³³⁰
- TSRVCustomControl³³⁴
- TSRVGraphicControl³³⁸
- TSRVWinControl³³⁹
- TSRVCustomComboBox³³¹
- TSRVCustomListBox³³⁵
- TSRVEditControl³³⁸

3.1.1 TCustomSRVEdit

TSRVEditControl is the base class for plain-text edit controls.

Unit SRVCustomEdit;

Syntax

```
TCustomSRVEdit = class(TSRVEditControl338);
```

Hierarchy

TObject

TPersistent

TComponent

TControl


TWinControl

TSRVWinControl³³⁹

TSRVEditControl³³⁸

Properties and events

Public properties

Property	Type	Default value	Meaning
Alignment	TAlignment	<i>taLeftJustify</i>	Horizontal text alignment (left/right/center). RTL BiDiMode inverts left and right alignments.
 CanUndo	Boolean		Read-only. Indicates whether the edit control contains changes that can be backed out.
Modified	Boolean		Indicates whether the user edited the text of the edit control.
ReadOnly	Boolean	<i>False</i>	Determines whether the user can change the text of the edit control.
SelText	TRVUnicodeString		Specifies the selected portion of the edit control's text.
Transparent	Boolean	<i>False</i>	Indicates whether the background is transparent. This property works only for editors inserted in TSVRichViewEdit ⁴⁰ .

See also properties inherited from TSRVWinControl³³⁹.

Methods

Method	Description
Clear	Deletes all text from the edit control.
ClearSelection	Removes the selected text from the edit control.
ClearUndo	Clears the undo buffer so that no changes to the text can be backed out.
CopyToClipboard	Copies the selected text in the edit control to the Clipboard in CF_UNICODETEXT format.
CutToClipboard	Copies the selected text to the Clipboard in CF_UNICODETEXT format and then deletes the selection.

Method	Description
PasteFromClipboard	Pastes the contents of the Clipboard into edit control, replacing the current selection.
Undo	Backs out all changes in the undo buffer.

See also methods inherited from `TSRVWinControl` ⁽³³⁹⁾.

Inherited components



TSRVEdit ⁽³⁴⁸⁾



TSRVMemo ⁽³⁶¹⁾

3.1.2 TCustomSRVGraphicControl

TSRVGraphicControl is the base class for lightweight SRVControls.

Unit SRVControl;

Syntax

```
TCustomSRVGraphicControl = class (TGraphicControl) ;
```

Hierarchy

TObject

TPersistent

TComponent


TControl

TGraphicControl

Properties and events

Public properties:




Property	Type	Default value	Meaning
AutoDarkMode	Boolean	<i>True</i>	If <i>True</i> , and the control is inserted in a dark-mode editor (TSRichViewEdit or TRichView), luminance of all colors is inverted. Otherwise, DarkMode property is used.
DarkMode	Boolean	<i>False</i>	If <i>True</i> , luminance of all colors are inverted. This property is used only if the control is not inserted in an editor, or if AutoDarkMode = <i>False</i> .
DrawOnPrint	Boolean	<i>True</i>	if <i>True</i> , this control will be printed
▶ IsMouseDown	Boolean		Read-only. Returns <i>True</i> if the user pressed the left mouse button above the control.

Property	Type	Default value	Meaning
 MouseIn	Boolean		Read-only. Returns <i>True</i> , if the mouse pointer is above the control

Public events:

- **OnMouseEnter, OnMouseLeave** (TNotifyEvent) occur when the mouse pointer enter/leaves the control.

Protected properties:

Property	Type	Default value	Meaning
 Alignment	TAlignment	<i>taLeftJustify</i>	Text alignment. RTL BiDiMode inverts left and right alignments
 SkinManager	TSRVSkinManager ⁽²²⁷⁾		A link to a skin manager component.
 SkinSchemeIndex	Integer	0	Index in a scheme in SkinManager.CurrentSkin ⁽²²⁹⁾ (a scheme depends on the control type)

Additionally, for Delphi versions prior to 2009, this control introduces Unicode Caption (of TRVUnicodeString) type to use instead of ANSI string Caption (so Caption is Unicode in all versions of Delphi)

Inherited components

 TSRVLabel⁽³⁵⁷⁾

 TSRVPaintBox⁽³⁶³⁾

3.1.3 TCustomSRVPanel

TCustomSRVPanel is the base class for TSRVPanel⁽³⁶⁴⁾ and TSRVGroupBox⁽³⁵¹⁾.

Unit SRVPanel;

Syntax



```
TCustomSRVPanel = class (TSRVCustomControl)
```

Hierarchy



TObject
 TPersistent
 TComponent
 TControl
 TWinControl
 TCustomControl
 TSRVCustomControl⁽³³⁴⁾

Properties and events

Protected properties:

Property	Type	Default value	Meaning
 BorderWidth	Integer	1	Border width.
 CaptionOffset	TRVPixel96Length	10	Caption offset.

Protected properties used only if **SRVControlStyle** = *srvcClassic*:

Property	Type	Default value	Meaning
 BorderColor	TColor	\$00B8D8D7	Border color
 CornersOffset	Integer	10	Value in range 1..30. Defines how much border corners are rounded

Inherited components



TSRVGroupBox ⁽³⁵¹⁾



TSRVPanel ⁽³⁶⁴⁾

3.1.4 TSRVCanvasControl

TSRVCanvasControl is the base class for lightweight SRVControls.

Unit SRVControl;

Syntax

```
TSRVCanvasControl = class (TGraphicControl);
```

Hierarchy

TObject

TPersistent

TComponent

TControl

TGraphicControl

TCustomSRVGraphicControl ⁽³²⁸⁾

Properties and events

Protected events:

- **OnPaint** (TSRVCtrlPaintPage) occurs when the control must be painted.

type

```
TSRVCtrlPaintPage = procedure (Sender: TObject; R: TRect; Canvas: TCanvas) of ob
```

See also properties and events inherited from TCustomSRVGraphicControl ⁽³²⁸⁾.

Inherited components



TSRVPaintBox ⁽³⁶³⁾

3.1.5 TSRVCustomComboBox

TSRVCustomComboBox is the base class for combo box controls.

Unit SRVComboBox;

Syntax

```
TSRVCustomComboBox = class (TSRVEditControl338);
```

Hierarchy

TObject

TPersistent

TComponent

TControl

TWinControl


*TSRVWinControl*³³⁹

*TSRVEditControl*³³⁸

Properties and events

Public properties

Property	Type	Default value	Meaning
AutoComplete	Boolean	<i>True</i>	Specifies whether the combo box automatically completes words that the user types by selecting the first item that begins with the currently typed string.
AutoDropDown	Boolean	<i>False</i>	Specifies whether the drop-down list drops down automatically in response to user keystrokes.
AutoSize	Boolean	<i>True</i>	Determines whether the height of the edit region automatically resizes to accommodate the text height
CharCase	TSRVEditCharCase	<i>srvecNormal</i>	Determines the case of the text within the edit region: <i>srvecNormal</i> , <i>srvecUpperCase</i> , <i>srvecLowerCase</i>
Color	TColor	<i>clWindow</i>	Background color of the editing region.
DropDownCount	Integer	5	Specifies the maximum number of items displayed in the drop-down list.
DropDownHints	Integer	10	Defines how many hints (suggestions) are visible in a list.
HideSelection	Boolean	<i>True</i>	Determines whether the visual indication of the selected text remains

Property	Type	Default value	Meaning
			when focus shifts to another control.
HintsItemSkinSchemeIndex	Integer		An index in SkinManager.CurrentSkin ⁽²²⁹⁾ . BoxSchemes ⁽²³¹⁾ collection, for items in the list of hints (suggestions)
HintsScrollBarSkinSchemeIndex	Integer		An index in SkinManager.CurrentSkin ⁽²²⁹⁾ . VerticalScrollBarSchemes ⁽²³²⁾ collection, for scrollbars in the list of hints (suggestions)
HintsSkinSchemeIndex	Integer		An index in SkinManager.CurrentSkin ⁽²²⁹⁾ . BoxSchemes ⁽²³¹⁾ collection, for the list of hints (suggestions)
ImageList	TCustomImageList		Specifies the images that appear next to items in the list box.
ItemHeight	Integer	20	Specifies the height in pixels of an item in an owner-draw list. When the screen DPI is changed, this value is recalculated.
ItemIndex	Integer		Specifies the ordinal number of the selected item.
ItemWidth	TRVPixel96Length	0	If set to a positive value, defines a width of a drop down list.
Items	TSRVBoxItemCollection ⁽³⁷⁴⁾		Items, a collection of TSRVBoxItem ⁽³⁷⁴⁾ .
ListMarginHeight	TRVPixel96Length	2	Top and bottom margins in each item.
ListMarginWidth	TRVPixel96Length	3	Left and right margins of each item.
MaxLength	Integer	0	Specifies the maximum number of characters the user can type into the edit portion (0 for unlimited)
Modified	Boolean		Indicates whether the user edited the text of the edit control.
ReadOnly	Boolean	<i>False</i>	Determines whether the user can change the text of the edit region.
 ScrollBar	TSRVScrollBar ⁽²⁴²⁾		Read-only. Returns a scroll bar object.
Sorted	Boolean	<i>False</i>	Specifies whether the items in a list box are arranged alphabetically.
Style	TComboBoxStyle	<i>csDropDown</i>	Determines the display style of the combo box.

Property	Type	Default value	Meaning
Text	String		Text in the editing region.
UseItemFontNames	Boolean	<i>False</i>	Specifies whether the FontName properties of items are used to display their Captions .
UseSelectedImagesInHotState	Boolean	<i>False</i>	Specifies whether <code>Items[].SelectedImageIndex</code> is used instead of <code>Items[].ImageIndex</code> for highlighted items (below the mouse pointer).

Public properties working only if **SRVControlStyle** = *srvcClassic*:

Property	Type	Default value	Meaning
SelectedItemBackColor	TColor	<i>clHighlight</i>	Background color for the selected item.
SelectedItemBorderColor	TColor	<i>clSilver</i>	Border color for the selected item.
SelectedItemTextColor	TColor	<i>clHighlightText</i>	Text color for the selected item.

Public events

OnChange occurs when the user changes the text displayed in the edit region.

OnDropDown occurs when the user opens the drop-down list.

OnMeasureItem occurs when the application needs to redisplay an item in a variable height owner-draw list box.

OnDrawItem occurs when an item in an owner-draw list box needs to be redisplayed.

OnEditHints occurs when a list of hints (suggestions) is about to be displayed. See the section about "Hints (Suggestions)" for `TSRVEdit`⁽³⁴⁸⁾.

See also properties and events inherited from `TSRVEditControl`⁽³³⁸⁾.

Methods

BeginUpdate/EndUpdate locks/unlocks redrawing.

IndexOf returns the index of item by its Caption.

ItemHasParent returns *True* if the given item has a parent item.

GetItemParent returns the index of the parent item for the given item.

AppendItem adds an item with the specified properties to the end of the item list.

InsertItem inserts an item with the specified properties at the position Index.

DeleteItem deletes the specified item.

Inherited components

TSRVComboBox ⁽³⁴⁵⁾

3.1.6 TSRVCustomControl

TSRVCustomControl is the base class for several SRVControls.

Unit SRVControl;

Syntax

```
TSRVCustomControl = class (TCustomControl);
```

Hierarchy

TObject

TPersistent

TComponent

TControl


TWinControl

TCustomControl

Properties and events

Public properties:





Property	Type	Default value	Meaning
Alignment	TAlignment	<i>taLeftJustify</i>	
AutoDarkMode	Boolean	<i>True</i>	If <i>True</i> , and the control is inserted in a dark-mode editor (TSRichViewEdit or TRichView), luminance of all colors is inverted. Otherwise, DarkMode property is used. Supported only if SRVControlStyle = <i>srvcSimple</i> .
▶ CaretPos	TPoint		Read-only. Returns the caret position in the control. If the control does not have a caret, returns (-1, -1).
▶ CaretHeight	Integer		Read-only. Returns the caret height in the control. If the control does not have a caret, returns -1.
DrawOnPrint	Boolean	<i>True</i>	if <i>True</i> , this control will be printed
DarkMode	Boolean	<i>False</i>	If <i>True</i> , luminance of all colors are inverted. This property is used only if the control is not inserted in an editor, or if AutoDarkMode = <i>False</i> . Supported only if SRVControlStyle = <i>srvcSimple</i> .

Property	Type	Default value	Meaning
 MouseIn	Boolean		Read-only. Returns <i>True</i> , if the mouse pointer is above the control
SRVControlStyle	TSRVControlStyle ⁽²⁷⁵⁾	srvcSimple	Defines visual appearance

Public events:







- **OnMouseEnter, OnMouseLeave** (TNotifyEvent) occur when the mouse pointer enter/leaves the control.

Protected properties:

Property	Type	Default value	Meaning
 ButtonSkinSchemeIndex	Integer	0	Additional index in a scheme in SkinManager.CurrentSkin ⁽²²⁹⁾ (a scheme depends on the control type)
 SkinManager	TSRVSkinManager ⁽²²⁷⁾		A link to a skin manager component.
 SkinSchemeIndex	Integer	0	Index in a scheme in SkinManager.CurrentSkin ⁽²²⁹⁾ (a scheme depends on the control type)
 SkinFontIndex	Integer	0	Index in SkinManager.CurrentSkin ⁽²²⁹⁾ .Fonts ⁽²³¹⁾

Additionally, for Delphi versions prior to 2009, this control introduces Unicode Caption (of TRVUnicodeString) type to use instead of ANSI string Caption (so Caption is Unicode in all versions of Delphi)

Inherited components

-  TSRVButton⁽³⁴²⁾
-  TSRVCheckBox⁽³⁴⁴⁾
-  TSRVGroupBox⁽³⁵¹⁾
-  TSRVPanel⁽³⁶⁴⁾
-  TSRVRadioButton⁽³⁶⁶⁾
-  TSRVScrollBar⁽²⁴²⁾

3.1.7 TSRVCustomListBox

TSRVCustomListBox is the base class for list box controls.

Unit SRVListBox;

Syntax

```
TSRVCustomListBox = class (TSRVWinControl(339));
```

Hierarchy

TObject

TPersistent

TComponent

TControl

TWinControl

TSRVWinControl⁽³³⁹⁾

Properties and events

Public properties

Property	Type	Default value	Meaning
HideSelection	Boolean	<i>True</i>	Determines whether the visual indication of the selected item remains when focus shifts to another control.
HotItemBackColor		<i>clBackground</i>	Background color for the highlighted item (below the mouse pointer).
HotItemBorderColor		<i>clSilver</i>	Border color for the highlighted item.
HotItemTextColor		<i>clWhite</i>	Text color for the highlighted item.
ImageList	TCustomImageList		Specifies the images that appear next to items in the list box.
ItemHeight	Integer		Ignored. Item height is calculated automatically.
ItemIndex	Integer		Specifies the ordinal number of the selected item in the list box's item list. If the value of the MultiSelect property is <i>True</i> the user can select more than one item in the list box. In this case, the ItemIndex value is the index of the selected item that has focus.
Items	TSRVListBoxItemCollection ⁽³⁷⁴⁾		Items, a collection of TSRVListBoxItem ⁽³⁷⁴⁾ .
ListMarginHeight	TRVPixel96Length	2	Top and bottom margins in each item.
ListMarginWidth	TRVPixel96Length	3	Left and right margins of each item.
MultiSelect	Boolean	<i>False</i>	Determines whether the user can select more than one element at a time.
► ScrollBar	TSRVScrollBar ⁽²⁴²⁾		Read-only. Returns a scroll bar object.
ShowCheckBoxes	Boolean	<i>False</i>	Shows/hides check boxes in each item.

Property	Type	Default value	Meaning
Sorted	Boolean	<i>False</i>	Specifies whether the items in a list box are arranged alphabetically.
UseItemFontNames	Boolean	<i>False</i>	Specifies whether the FontName properties of items are used to display their Captions .
UseSelectedImagesInHot State	Boolean	<i>False</i>	Specifies whether <code>Items[].SelectedImageIndex</code> is used instead of <code>Items[].ImageIndex</code> for highlighted items (below the mouse pointer).

Public properties working only if **SRVControlStyle** = *srvcClassic*:

Property	Type	Default value	Meaning
SelectedItemBackColor	TColor	<i>clHighlight</i>	Background color for the selected item.
SelectedItemBorderColor	TColor	<i>clSilver</i>	Border color for the selected item.
SelectedItemTextColor	TColor	<i>clHighlightText</i>	Text color for the selected item.

Public events

OnMeasureItem occurs when the application needs to redisplay an item in a variable height owner-draw list box.

OnDrawItem occurs when an item in an owner-draw list box needs to be redisplayed.

See also properties and events inherited from `TSRVWinControl` ⁽³³⁹⁾.

Methods

BeginUpdate/EndUpdate locks/unlocks redrawing.

IndexOf returns the index of item by its Caption.

ItemHasParent returns *True* if the given item has a parent item.

GetItemParent returns the index of the parent item for the given item.

AppendItem adds an item with the specified properties to the end of the item list.

InsertItem inserts an item with the specified properties at the position Index.

DeleteItem deletes the specified item.

Inherited components



TSRVListBox ⁽³⁵⁸⁾

3.1.8 TSRVEditControl

TSRVEditControl is the base class for plain-text controls.

Unit SRVControl;

Syntax

```
TSRVEditControl = class (TSRVWinControl (339));
```

Hierarchy

TObject

TPersistent

TComponent

TControl

TWinControl

TSRVWinControl ⁽³³⁹⁾

Properties and events

Public properties

Property	Type	Default value	Meaning
SelLength	Integer		Specifies the number of selected characters.
SelStart	Integer		Specifies the position of the first selected character in the text, from 0

See also properties inherited from TSRVWinControl ⁽³³⁹⁾.

Methods

SelectAll selects all text in the edit control.

Inherited components



TSRVEdit ⁽³⁴⁸⁾



TSRVMemo ⁽³⁶¹⁾



TSRVComboBox ⁽³⁴⁵⁾

3.1.9 TSRVGraphicControl

TSRVGraphicControl is the base class for lightweight SRVControls.

Unit SRVControl;

Syntax

```
TSRVGraphicControl = class (TCustomSRVGraphicControl328) ;
```

Hierarchy

TObject

TPersistent

TComponent

TControl

TGraphicControl

*TCustomSRVGraphicControl*³²⁸

Properties and events

Public properties:

Property	Type	Default value	Meaning
AutoSize	Boolean	<i>True</i>	Determines whether the control automatically resizes itself to accommodate its content.

See also properties and events inherited from *TCustomSRVGraphicControl*³²⁸.

Inherited components

Abc *TSRVLabel*³⁵⁷

3.1.10 TSRVWinControl

TSRVWinControl is the base class for all *SRVControls* that are wrappers for Microsoft Windows screen objects.

Unit *SRVControl*;

Syntax

```
TSRVWinControl = class (TWinControl) ;
```

Hierarchy

TObject

TPersistent

TComponent

TControl

TWinControl

Properties and events

Public properties:

Property	Type	Default value	Meaning
AutoDarkMode	Boolean	<i>True</i>	If <i>True</i> , and the control is inserted in a dark-mode editor (<i>TSRichViewEdit</i> or <i>TRichView</i>),

Property	Type	Default value	Meaning
			luminance of all colors is inverted. Otherwise, DarkMode property is used. Supported only if SRVControlStyle = <i>svcsSimple</i> .
▶ CaretPos	TPoint		Read-only. Returns the caret position in the control. If the control does not have a caret, returns (-1, -1).
▶ CaretHeight	Integer		Read-only. Returns the caret height in the control. If the control does not have a caret, returns -1.
DarkMode	Boolean	<i>False</i>	If <i>True</i> , luminance of all colors are inverted. This property is used only if the control is not inserted in an editor, or if AutoDarkMode = <i>False</i> . Supported only if SRVControlStyle = <i>svcsSimple</i> .
DrawOnPrint	Boolean	<i>True</i>	if <i>True</i> , this control will be printed
DisabledBorderWidth	Integer	1	Border width in disabled state
EnabledBorderWidth	Integer	1	Border width in normal state
FocusedBorderWidth	Integer	1	Border width when the control is focused
▶ MouseIn	Boolean		Read-only. Returns True, if the mouse pointer is above the control
MouseInBorderWidth	Integer	1	Border width when the control is below the mouse pointer

Public properties working only if **SRVControlStyle** = *svcsClassic*

Property	Type	Default value	Meaning
DisabledBorderColor	TColor	<i>\$B8D8D7</i>	Border color in a disabled state
EnabledBorderColor	TColor	<i>\$BF9F7F</i>	Border color in a normal state
FocusedBorderColor	TColor	<i>clBlue</i>	Border color when the control is focused
MouseInBorderColor	TColor	<i>clRed</i>	Border color when the control is below the mouse pointer

Public events:

- **OnMouseEnter**, **OnMouseLeave** (TNotifyEvent) occur when the mouse pointer enter/leaves the control.

Protected properties:










Property	Type	Default value	Meaning
■ SkinManager	TSRVSkinManager ⁽²²⁷⁾		A link to a skin manager component.
■ SkinSchemeIndex	Integer	0	An index in a scheme in SkinManager.CurrentSkin ⁽²²⁹⁾ (a scheme depends on the control type)
■ ScrollBarSkinSchemeIndex	Integer	0	An index in SkinManager.CurrentSkin ⁽²²⁹⁾ .VerticalScrollBarSchemes ⁽²³²⁾ collection, for controls having scrollbars
■ ItemSkinSchemeIndex	Integer	0	An index in SkinManager.CurrentSkin ⁽²²⁹⁾ .BoxSchemes ⁽²³¹⁾ collection, for controls having items
■ EditSkinSchemeIndex	Integer	0	An index in SkinManager.CurrentSkin ⁽²²⁹⁾ .BoxSchemes ⁽²³¹⁾ collection, for controls having an embedded text editor
■ ButtonSkinSchemeIndex	Integer	0	An index in SkinManager.CurrentSkin ⁽²²⁹⁾ .BoxSchemes ⁽²³¹⁾ collection, for controls having an embedded button
■ SRVControlStyle	TSRVControlStyle ⁽²⁷⁵⁾	<i>srvcSimple</i>	Defines a visual appearance

Inherited classes

- TSRVEditControl⁽³³⁸⁾
- TSRVCustomListBox⁽³³⁵⁾

3.2 Main Controls

Controls:

-  TSRVButton⁽³⁴²⁾ – button;
-  TSRVCheckBox⁽³⁴⁴⁾ – check box;
-  TSRVComboBox⁽³⁴⁵⁾ – combo box; hierarchical; with images, custom colors and fonts; with hints (suggestions);
-  TSRVEdit⁽³⁴⁸⁾ – plain-text one-line editor with hints (suggestions);
-  TSRVGroupBox⁽³⁵¹⁾ – panel with a title;
-  TSRVImagesScroll⁽³⁵³⁾ – gallery of images;
-  TSRVLabel⁽³⁵⁷⁾ – text label;
-  TSRVListBox⁽³⁵⁸⁾ – scrollable list of items; hierarchical; with images, check boxes, custom colors and fonts;
-  TSRVMemo⁽³⁶¹⁾ – plain-text multi-line editor;



TSRVPaintBox⁽³⁶³⁾ – controls for custom drawing;



TSRVPanel⁽³⁶⁴⁾ – panel;



TSRVRadioButton⁽³⁶⁶⁾ – radio button.

See also:



TSRVScrollBar⁽²⁴²⁾;



TSRVTabSet⁽²⁴⁸⁾.

All the controls above can be skinned using TSRVSkinManager⁽²²⁷⁾.

3.2.1 TSRVButton

TSRVLabel is a control displaying text.

Unit SRVButton;

Syntax

```
TSRVButton = class (TSRVCustomControl(334))
```

Hierarchy

TObject

TPersistent

TComponent

TControl

TWinControl

TCustomControl

TSRVCustomControl⁽³³⁴⁾

Properties and events

This component publishes the following properties inherited from TSRVCustomControl⁽³³⁴⁾:

- **DrawOnPrint**;
- **SkinManager**;
- **SkinSchemeIndex**;
- **SRVControlStyle**;
- properties inherited from TCustomControl.

New published properties:

Property	Type	Default value	Meaning
■ Default	Boolean	<i>False</i>	Determines whether the button's OnClick event handler executes when the Enter key is pressed.
■ Glyph	TPicture	(empty)	Specified the picture that appears on the button. The picture is displayed to the left/right side of text, depending on BiDiMode.

Property	Type	Default value	Meaning
■ ModalResult	TModalResult	<i>mrNone</i>	Determines whether and how the button closes its (modal) parent form.
■ OffsetOnClick	TRVPixel96Length	0	Specifies how far Glyph and Caption are shifted when the button is pressed.

The following published properties work only if **SRVControlStyle** = *srvcsClassic*:

Property	Type	Default value	Meaning
■ MouseBorderWidth	TRVPixel96Length	2	Width of highlight frame under the mouse or when the button is focused
■ ShowFocusRect	Boolean	<i>True</i>	Shows/hides a dotted rectangle drawn on a focused button

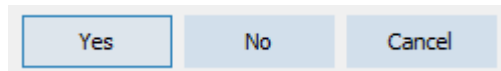
New published event:

- **OnClick**, occurs when the user clicks the control.

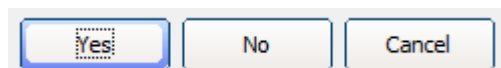
Painting

The default button appearance (if skins are not assigned) depends on **SRVControlStyle**:
 TSRVControlStyle⁽²⁷⁵⁾ property:

srvcsSimple (if RVControlsPainter.Theme = *rvctPaleBlue*):

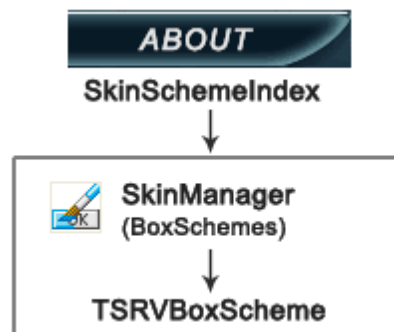


srvcsClassic:



If **SkinManager** is assigned, the button is drawn using **SkinManager.CurrentSkin**⁽²²⁹⁾.BoxSchemes⁽²³¹⁾ [**SkinSchemeIndex**].

Example:



You can also use the skin font: **SkinManager.CurrentSkin**⁽²²⁹⁾.Fonts⁽²³¹⁾ [**SkinFontIndex**].

3.2.2 TSRVCheckBox

TSRVCheckBox represents a check box that can be on (checked) or off (unchecked)

Unit SRVCheckBox;

Syntax

```
TSRVCheckBox = class (TSRVCustomControl334)
```

Hierarchy




TObject
 TPersistent
 TComponent
 TControl
 TWinControl
 TCustomControl
 TSRVCustomControl³³⁴

Properties and events

This component publishes the following properties inherited from TSRVCustomControl³³⁴:

- **Alignment**;
- **ButtonSkinSchemeIndex**;
- **DrawOnPrint**;
- **SkinManager, SkinSchemeIndex, SkinFontIndex**;
- **SRVControlStyle**;
- properties inherited from TCustomControl.

New published properties:

Property	Type	Default value	Meaning
 AllowGrayed	Boolean	<i>False</i>	Determines whether a check box can be in a “grayed” state. Not supported in a skinned mode.
 Checked	Boolean	<i>False</i>	Specifies whether the check box is checked. If the AllowGrayed property is <i>True</i> , you may find it more useful to use the State property.
 State	TSRVCheckBoxState	cbUnchecked	Indicates whether the check box is selected (<i>cbChecked</i>), deselected (<i>cbUnchecked</i>), or grayed (<i>cbGrayed</i>)

Painting

The default check-box appearance (if skins are not assigned) depends on **SRVControlStyle**:
 TSRVControlStyle²⁷⁵ property:

srvcSimple (if RVControlsPainter.Theme = *rvctPaleBlue*):

- ☒ Apple
☒ Banana
☐ Cherry

srvcsClassic:

- ☒ Apple
- ☒ Banana
- ☐ Cherry

Positions of a box and **Caption** depend on **Alignment**:

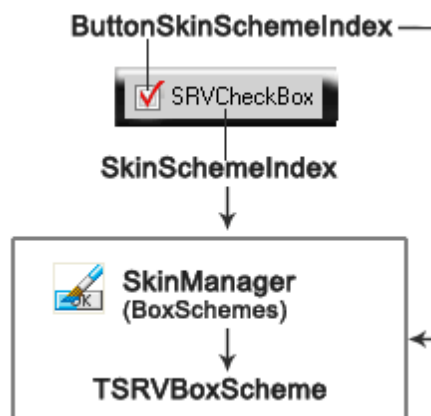
- *taLeftJustify*: the box and the caption are aligned to the left; the box is to the left of the caption;
- *taRightJustify*: the box and the caption are aligned to the right; the box is to the right of the caption;
- *taCenter*: the box and the caption are in the middle; the box is to the left/right of the caption depending in BiDiMode.

RTL BiDiMode inverts left and right **Alignment**.

Note: **Alignment** is different from the standard TCheckBox.Alignment. In TCheckBox, there are only left and right alignments, and they define Caption position relative to a box. In TSRVCheckBox, there are three alignments, and they define alignment of a box and Caption in the control (so the meanings are almost opposite).

If **SkinManager** is assigned, the control is drawn using **SkinManager.CurrentSkin**⁽²²⁹⁾.
.BoxSchemes⁽²³¹⁾ [**SkinSchemeIndex**], and the box is drawn using **SkinManager.CurrentSkin**⁽²²⁹⁾.
.BoxSchemes⁽²³¹⁾ [**ButtonSkinSchemeIndex**]. In this mode, a grayed state is not supported.

Example:



3.2.3 TSRVComboBox

TSRVComboBox combines an edit box with a scrollable list. Items can have captions and images. A list can be hierarchical. Additionally, editing suggestions (hints) may be displayed.

Unit SRVComboBox;

Syntax

```
TSRVComboBox = class (TSRVCustomComboBox(331))
```

Hierarchy

TObject

TPersistent

TComponent

TControl

TWinControl

TSRVWinControl ⁽³³⁹⁾

TSRVEditControl ⁽³³⁸⁾

TSRVCustomComboBox ⁽³³¹⁾

Properties and events

This component publishes public properties inherited from *TSRVCustomComboBox* ⁽³³¹⁾.

Hints (Suggestions)

A list of suggestions can be displayed on editing.

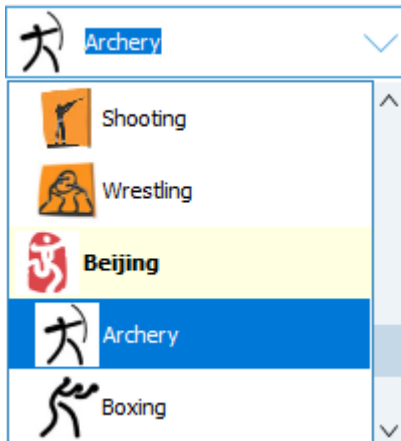
Suggestions are requested in the event **OnEditHints**. In this event, you can fill a list of suggestions (**EditHints**) for the **Text**, and specify the **Index** of item to highlight initially.

A height of this list is defined in **DropDownHints** property.

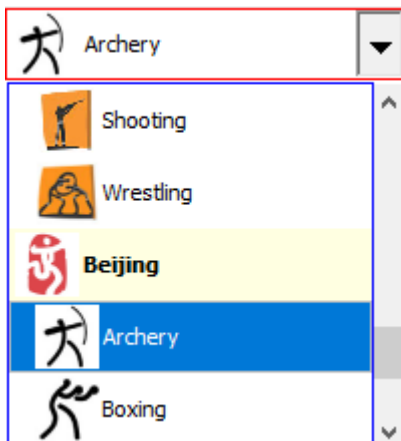
Painting

The default combo box appearance (if skins are not assigned) depends on **SRVControlStyle**: *TSRVControlStyle* ⁽²⁷⁵⁾ property:

srvcSimple (if *RVControlsPainter.Theme = rvctPaleBlue*):



srvcClassic (the scroll bar appearance depends on Windows, this screenshot is for Windows 10)::



The border around the box has width defined in **DisabledBorderWidth**, **EnabledBorderWidth**, **MouseInBorderWidth**, **FocusedBorderWidth** properties. When auto-sizing the control, it is assumed that the border width is equal to **EnabledBorderWidth**.

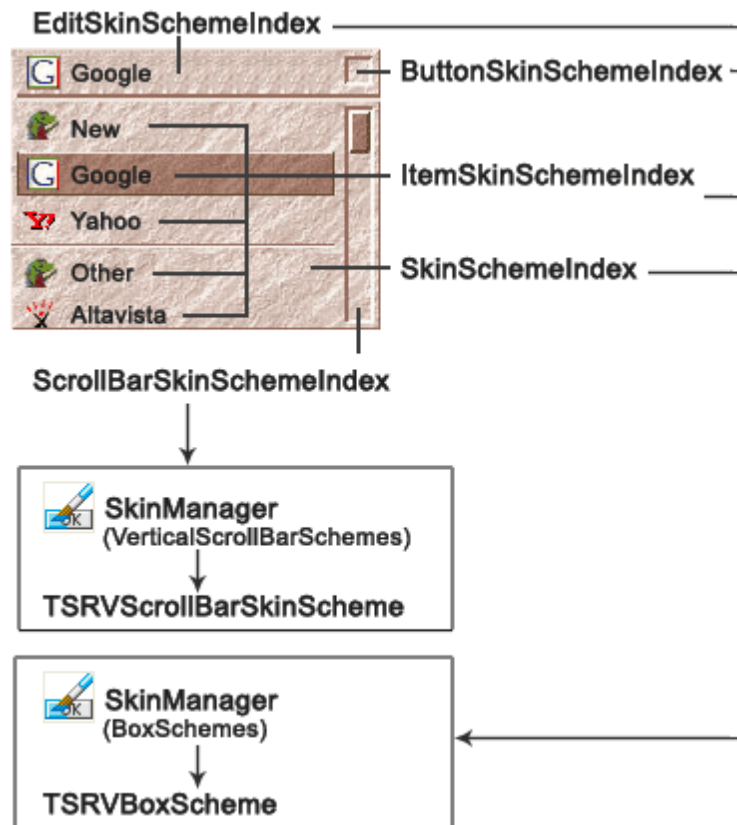
A border color depends on the combo box state (normal, disabled, under the mouse pointer, focused) and **SRVControlStyle**: `TSRVControlStyle`⁽²⁷⁵⁾ property. If it is equal to *srvcsSimple*, colors are defined by `RVControlsPainter` (see the `TRichView` manual). If it is equal to *srvcsClassic*, colors are defined in **EnabledBorderColor**, **DisabledBorderColor**, **MouseInBorderColor**, **FocusedBorderColor** properties. The same is for colors used for drawing selected items (**SelectedItemBackColor**, **SelectedItemBorderColor**, **SelectedItemTextColor**).

Item colors are defined in the items properties⁽³⁷⁴⁾.

If **SkinManager** is assigned:

- an editing region is drawn using **SkinManager**.CurrentSkin⁽²²⁹⁾.BoxSchemes⁽²³¹⁾ [EditSkinSchemeIndex];
- a drop down list is drawn using **SkinManager**.CurrentSkin⁽²²⁹⁾.BoxSchemes⁽²³¹⁾ [SkinSchemeIndex];
- each item in a drop down list is drawn using **SkinManager**.CurrentSkin⁽²²⁹⁾.BoxSchemes⁽²³¹⁾ [ItemSkinSchemeIndex];
- a scrollbar of a drop down list is drawn using **SkinManager**.CurrentSkin⁽²²⁹⁾.VerticalScrollBarSchemes⁽²³²⁾ [ScrollBarSkinSchemeIndex].
- a list of hints (suggestions) is drawn using **SkinManager**.CurrentSkin⁽²²⁹⁾.BoxSchemes⁽²³¹⁾ [HintsSkinSchemeIndex];
- each item in a list of hints is drawn using **SkinManager**.CurrentSkin⁽²²⁹⁾.BoxSchemes⁽²³¹⁾ [HintsItemSkinSchemeIndex];
- a scrollbar of a list hints is drawn using **SkinManager**.CurrentSkin⁽²²⁹⁾.VerticalScrollBarSchemes⁽²³²⁾ [HintsScrollBarSkinSchemeIndex].

Example:



3.2.4 TSRVEdit

TSRVEdit is a single-line plain-text edit control, with an optional popup list displaying hints (suggestions).

TSRVEdit edits Unicode text in all versions of Delphi.

Unit SRVEdit;

Syntax

```
TSRVEdit = class (TCustomSRVEdit326)
```

Hierarchy

TObject
TPersistent
TComponent
TControl
TWinControl
*TSRVWinControl*³³⁹
*TSRVEditControl*³³⁸
*TCustomSRVEdit*³²⁶

Properties and events

This component publishes the following new properties and properties inherited from TCustomSRVEdit³²⁶:

Property	Type	Default value	Meaning
■ Alignment	TAlignment	<i>taLeftJustify</i>	Horizontal text alignment (left/right/center). RTL BiDiMode inverts left and right alignments.
■ AutoSelect	Boolean	<i>True</i>	Determines whether all the text in the edit control is automatically selected when the control gets focus.
■ AutoShowHints	Boolean	<i>True</i>	Determines whether hints (suggestions) are shown when the control is focused and when its text is edited. They can also be displayed by DisplayHints method.
■ AutoSize	Boolean	<i>True</i>	Determines whether the height of the edit control automatically resizes to accommodate the text height.
■ CharCase	TSRVEditCharCase	<i>srvecNormal</i>	Determines the case of the text within the edit control: <i>srvecNormal, srvecUpperCase, srvecLowerCase</i>
■ DropDownHints	Integer	10	Defines how many hints (suggestions) are visible in a list.
■ MaxLength	Integer	0	Specifies the maximum number of characters the user can enter into the edit control (0 for unlimited)
■ NumbersOnly	Boolean	<i>False</i>	Allows only numbers to be typed into the text edit.
■ PasswordChar	Char	#0	Indicates the character, if any, to display in place of the actual characters typed in the control (#0 to disable this feature)
■ ReadOnly	Boolean	<i>False</i>	Determines whether the user can change the text of the edit control
■ SRVControlStyle	TSRVControlStyle ²⁷⁵	<i>srvcSimple</i>	Defines a visual appearance (mainly affects border colors)
■ Text	TRVUnicodeString		Contains a text displayed by the editor.

- **DrawOnPrint;**
- **SkinManager;**
- **EditSkinSchemeIndex, SkinSchemeIndex** (used in the hints window), **ScrollBarSkinSchemeIndex** (used in the hints window), **ItemSkinSchemeIndex** (used in the hints window);
- **SkinFontIndex;**

- **EnabledBorderColor, DisabledBorderColor, MouseInBorderColor, FocusedBorderColor** (all colors are used only if **SRVControlStyle** = *rvsvcsClassic*)
- **DisabledBorderWidth, EnabledBorderWidth, MouseInBorderWidth, FocusedBorderWidth;**
- properties inherited from **TRWinControl**.

New published events:

- **OnChange** occurs when the text for the edit control may have changed.
- **OnEditHints**: **TSRVEditHintsEvent** occurs when a list of hints (suggestions) is about to be displayed.
- **OnCloseHints**: **TFormHintsClose** occurs when a list of hints (suggestions) is closed.

type

```
TSRVEditHintsEvent = procedure (Sender: TSRVEditControl338;
  Text: TRVUnicodeString; EditHints: TRVStringList;
  var ItemIndex : Integer) of object;
TFormHintsClose = procedure (Sender: TObject;
  FormWidth, FormHeight: Integer;
  var NewText: TRVUnicodeString) of object;
```

where **TRVStringList** is a string list that contains Unicode string in all versions of Delphi (for Delphi 2009+, it contains **UnicodeString** items; for older versions of Delphi, it contains UTF-8 strings).

Methods

DisplayHints displays a list of hints (suggestions).

BeginHintsUpdate, EndHintsUpdate blocks/unblocks hints (suggestions) showing.

See also the methods inherited from **TCustomSRVEdit**³²⁶.

Hints (Suggestions)

A list of suggestions can be displayed on editing or by the method **ShowHints**. This feature can be temporary blocked using **BeginHintsUpdate/EndHintsUpdate**.

Suggestions are requested in the event **OnEditHints**. In this event, you can fill a list of suggestions (**EditHints**) for the **Text**, and specify the **Index** of item to highlight initially.

A height of this list is defined in **DropDownHints** property.

When a list of suggestions is closed, **OnCloseHints** event occurs. The parameter **NewText** contains the chosen suggestion (it is empty if canceled).

Painting

If skins are not defined:

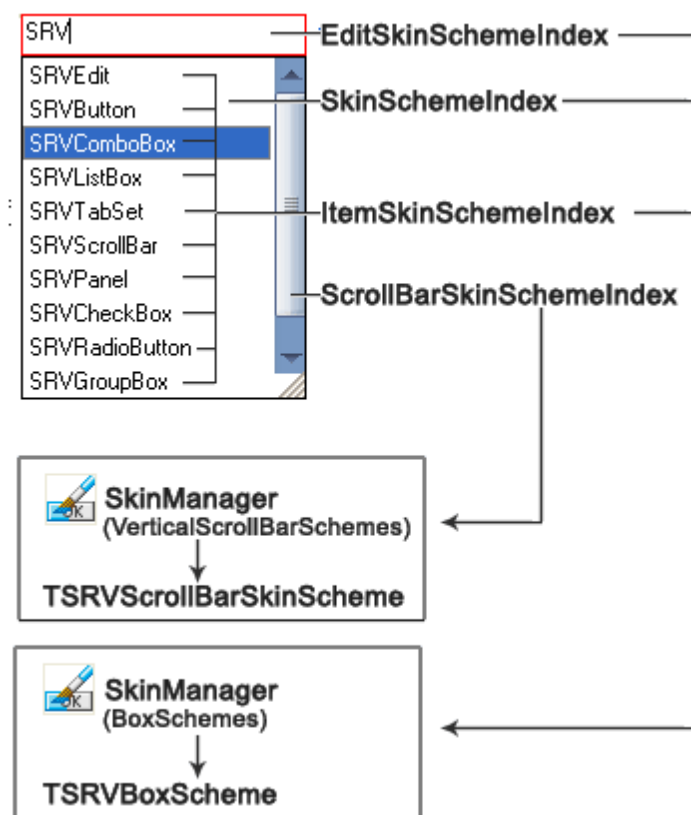
The border around the editor has width defined in **DisabledBorderWidth, EnabledBorderWidth, MouseInBorderWidth, FocusedBorderWidth** properties. When auto-sizing the editor, it is assumed that the border width is equal to **EnabledBorderWidth**.

A border color depends on the editors state (normal, disabled, under the mouse pointer, focused) and **SRVControlStyle**: **TSRVControlStyle**²⁷⁵ property. If it is equal to *rvsvcsSimple*, colors are defined by **RVControlsPainter** (see the **TRichView** manual). If it is equal to *rvsvcsClassic*, colors are defined in **EnabledBorderColor, DisabledBorderColor, MouseInBorderColor, FocusedBorderColor** properties.

If **SkinManager** is assigned:

- the control is drawn using **SkinManager.CurrentSkin**⁽²²⁹⁾.BoxSchemes⁽²³¹⁾[**EditSkinSchemeIndex**];
- a list of hints (suggestions) is drawn using **SkinManager.CurrentSkin**⁽²²⁹⁾.BoxSchemes⁽²³¹⁾[**SkinSchemeIndex**];
- each item in this list is drawn using **SkinManager.CurrentSkin**⁽²²⁹⁾.BoxSchemes⁽²³¹⁾[**ItemSkinSchemeIndex**];
- a scrollbar of this list is drawn using **SkinManager.CurrentSkin**⁽²²⁹⁾.VerticalScrollBarSchemes⁽²³²⁾[**ScrollBarSkinSchemeIndex**].
- font is defined in **SkinManager.CurrentSkin**⁽²²⁹⁾.Fonts⁽²³¹⁾[**SkinFontIndex**].

Example:



3.2.5 TSRVGroupBox

TSRVGroupBox component is used to group related controls. This component is similar to TSRVPanel⁽³⁶⁴⁾ (the difference is only in the caption position).

Unit SRVGroupBox;

Syntax

```
TSRVGroupBox = class (TCustomSRVPanel(329))
```

Hierarchy

TObject

TPersistent

TComponent

TControl

TWinControl

TCustomControl

TSRVCustomControl ³³⁴

TCustomSRVPanel ³²⁹

Properties and events

This component publishes the following new properties and properties inherited from *TCustomSRVPanel* ³²⁹:

Property	Type	Default value	Meaning
■ Alignment	TAlignment	<i>taLeftJustify</i>	Horizontal text alignment (left/right/center). RTL BiDiMode inverts left and right alignments.
■ BorderWidth	Integer	1	Border width.
■ CaptionOffset	TRVPixel96Length	10	Caption offset (Caption cannot be displayed closer to the left and right side of the control than specified in this property).
■ SRVControlStyle	TSRVControlStyle ²⁷⁵	<i>srvcSimple</i>	Defines visual appearance.

- **DrawOnPrint**;
- **SkinManager**;
- **SkinSchemeIndex**;
- **Color** (default value *cWhite*);
- properties inherited from *TCustomControl*.

Published properties used only if **SRVControlStyle** = *srvcClassic*:

Property	Type	Default value	Meaning
■ BorderColor	TColor	<i>\$00B8D8D7</i>	Border color
■ CornersOffset	Integer	10	Value in range 1..30. Defines how much border corners are rounded

Painting

Caption is aligned according to **Alignment**. **CaptionOffset** shifts the text (to the right, if **Alignment**=*taLeftJustify*, or to the left, if **Alignment**=*taRightJustify*). Caption is placed at the top. RTL **BiDiMode** inverts **Alignment**.

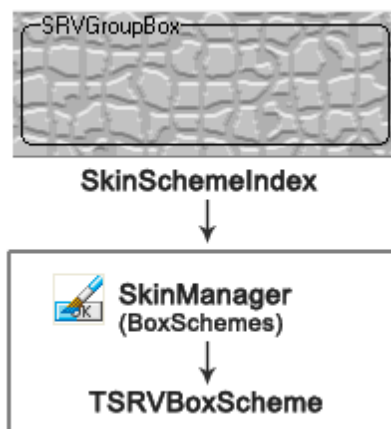
Background is drawn using **Color** property.

The Border has width defined in **BorderWidth** property (even in a skin mode). If **SRVControlStyle** = *srvcSimple*, the border color is defined by RVControlsPainter (see the TRichView manual). If it is equal to *srvcClassic*, the border color is defined in **BorderColor** property.

If **SRVControlStyle** = *srvcClassic*, the border can be rounded according to **CornersOffset** property.

If **SkinManager** is assigned, the control is drawn using **SkinManager.CurrentSkin**⁽²²⁹⁾.
.BoxSchemes⁽²³¹⁾ [**SkinSchemeIndex**].

Example:



3.2.6 TSRVImagesScroll

The TSRVImagesScroll component is used to display images in a scrollable row (or column)

Unit SRVImagesScroll;

Syntax

```
TSRVImagesScroll = class (TSRVCustomControl(334))
```

Hierarchy

TObject
 TPersistent
 TComponent
 TControl
 TWinControl
 TCustomControl
 TSRVCustomControl⁽³³⁴⁾

Properties

This component publishes the following properties inherited from TSRVCustomControl⁽³³⁴⁾:

- **SkinManager**;
- **SkinSchemeIndex**;
- **SRVControlStyle**;
- properties inherited from TCustomControl.

New published properties:

Property	Type	Default value	Meaning
■ CanMoveImages	Boolean	<i>True</i>	Determines whether items can be rearranged using the mouse.
■ CloseButton	TSRVCloseButton ⁽²⁶¹⁾		Properties for closing button in each item.
■ ImageList	TCustomImageList		A link to an image list to display in items
■ Indent	TRVPixel96Length	10	Indent before the first item.
■ ItemBorderStyle	<i>TPenStyle</i>	<i>psSolid</i>	Item border pen style.
■ ItemHeight	TRVPixel96Length	40	Item height
■ ItemIndex	Integer		Index of the selected item (in Items)
■ Items	TSRVImageScrollCollection ⁽³⁷⁴⁾		Items, a collection of TSRVImageScrollItem ⁽³⁷⁸⁾
■ ItemWidth	TRVPixel96Length	80	Item width
■ Kind	TSRVImageScrollKind	<i>srvbkHorizontal</i>	Determines how images are placed (<i>srvbkHorizontal</i> – in a row, <i>srvbkVertical</i> – in a column).
■ RestrictMove	Boolean	<i>False</i>	Determines whether the user can move items outside visible area of the control (if CanMoveImages = <i>True</i>)
■ Spacings	TRVPixel96Length	10	Spacing between items.

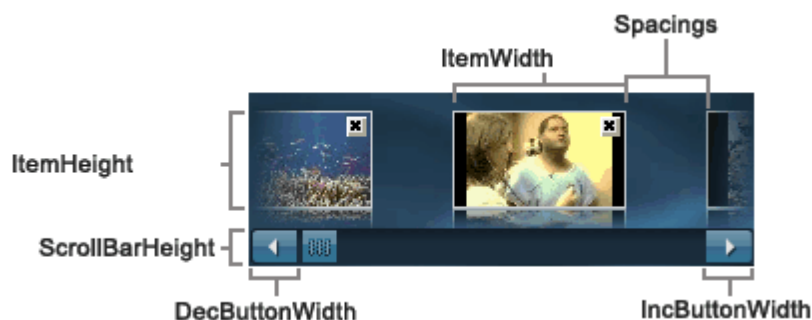
New published properties-colors (for non-skin mode):

Property	Type	Default value	Meaning
■ ItemBorderColor	TColor	<i>clBtnShadow</i>	Item border color.
■ ItemColor		<i>clBtnFace</i>	Item color in normal mode.
■ ItemDownColor		<i>clInactiveCaption</i>	Item color when pressed.
■ ItemHotColor		<i>clBtnShadow</i>	Item color below the mouse pointer.
■ ItemSelectedColor		<i>clHighlight</i>	Item color when selected.

New published properties related to scroll bar:

Property	Type	Default value	Meaning
■ DecButtonWidth	TRVPixel96Length	0	Width of up/left button in a skin mode. 0 for system default.
■ IncButtonWidth	TRVPixel96Length	0	Width of down/right button in a skin mode. 0 for system default.
■ LargeChange	Integer	10	Determines how much Position changes when the user clicks the scroll bar on

Property	Type	Default value	Meaning
			either side of the thumb tab.
■ ScrollerPosition	<i>TSRVScrollerPosition</i>	<i>srvpsEnd</i>	Scroll bar position (<i>srvpsBegin</i> – top/left, <i>srvpsEnd</i> – bottom/right)
■ ScrollingDelay	Integer	100	Defines a scrolling speed (interval between scrolling events)
■ SmallChange	Integer	1	Determines how much the scrollbar Position changes when the user clicks the arrow buttons on the scroll bar.



Published properties related to skins:

Property	Type	Default value	Meaning
■ SkinManager	<i>TSRVSkinManager</i> ⁽²²⁷⁾		Inherited. A link to a skin manager component.
■ SkinSchemeIndex	Integer	0	Inherited. Index in SkinManager.CurrentSkin ⁽²²⁹⁾ . BoxSchemes ⁽²³¹⁾ , defines the appearance of the whole control.
■ ScrollBarSkinSchemeIndex	Integer	0	Index in SkinManager.CurrentSkin ⁽²²⁹⁾ . HorizontalScrollBarSchemes ⁽²³¹⁾ (or VerticalScrollBarSchemes ⁽²³²⁾ in a vertical mode), defines the appearance of scrollbar.
■ ItemSkinSchemeIndex	Integer	0	Index in SkinManager.CurrentSkin ⁽²²⁹⁾ . BoxSchemes ⁽²³¹⁾ , defines the appearance of each item.
■ ButtonSkinSchemeIndex	Integer	0	Index in SkinManager.CurrentSkin ⁽²²⁹⁾ . BoxSchemes ⁽²³¹⁾ , defines the appearance of closing buttons.

Events

type

```

TSRVCloseImageScrollEvent = procedure(Sender: TSRVImagesScroll;
  ItemIndex : Integer; var CanClose : Boolean) of object;
TSRVImageScrollMoveEvent = procedure(Sender: TSRVImagesScroll;
  NewImageIndex, OldImageIndex : Integer) of object;
TSRVDDrawImageScrollEvent = procedure(Sender: TSRVImagesScroll;
  Canvas : TCanvas; ARect, PaintRect : TRect; State : TSRVDDrawStates;
  var DoDefault : Boolean) of object;
TSRVDDrawImageScrollItemEvent = procedure(Sender: TSRVImagesScroll;
  Canvas : TCanvas; ARect, PaintRect : TRect; ItemIndex : Integer;
  State : TSRVDDrawStates; var DoDefault : Boolean) of object;

```

OnChange: TNotifyEvent occurs after a new item is selected.

OnItemMove: TSRVImageScrollMoveEvent occurs when two items exchange their places as a result of item moving. Similar to OnMoveTab⁽²⁶⁰⁾ for TSRVTabSet⁽²⁴⁸⁾.

OnCloseItem : TSRVCloseImageScrollEvent occurs when the user attempts to close an item.

OnDrawBorder, OnDrawBackground, OnDrawItem, OnDrawCloseButton:

TSRVDDrawImageScrollEvent allow custom drawing for the corresponding objects. Canvas – a canvas where to draw. ARect – a rectangle defining the position of the object to draw. APaintRect – a rectangle that needs to be redrawn; you can use this parameter to optimize drawing. State – a set describing the current object state. Set DoDefault to *False* to prevent the default drawing of this object.

Painting

The default appearance (if skins are not assigned) depends on **SRVControlStyle**:

TSRVControlStyle⁽²⁷⁵⁾ property:

srvcsSimple (if RVControlsPainter.Theme = *rvctPaleBlue*):



srvcsClassic (the scroll bar appearance depends on Windows, this screenshot is for Windows 10):

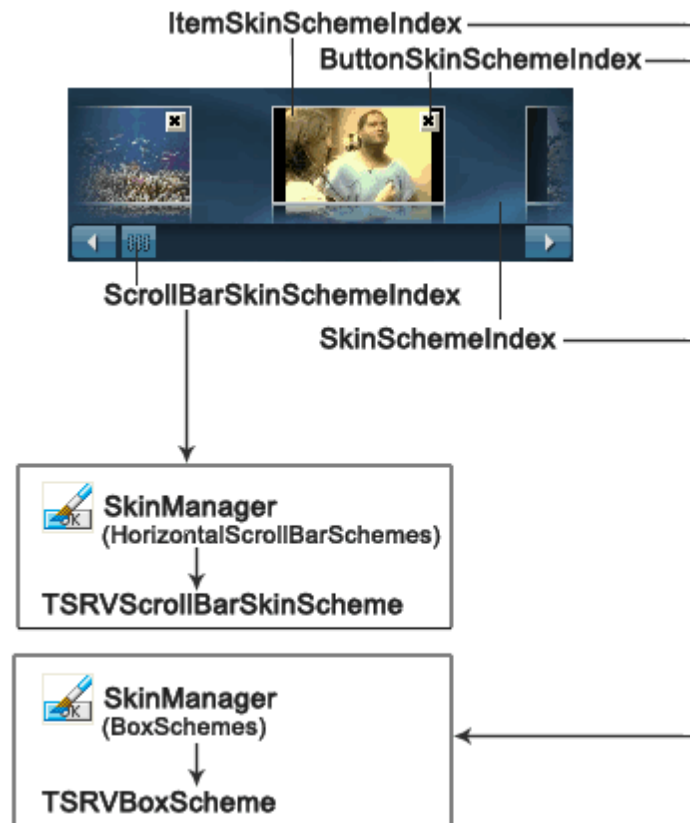


If **SRVControlStyle** = *srvcsSimple*, all colors are defined in RVControlsPainter (see the TRichView manual for details).

If **SRVControlStyle** = *srvcsClassic*, the control uses **Color**, **ItemBorderColor**, **ItemColor**, **ItemDownColor**, **ItemHotColor**, **ItemSelectedColor** properties.

If **SkinManager** is assigned, the control is drawn using **SkinManager.CurrentSkin**⁽²²⁹⁾. **BoxSchemes**⁽²³¹⁾ [**SkinSchemeIndex**], the scroll bar is drawn using **SkinManager.CurrentSkin**⁽²²⁹⁾. **HorizontalScrollBarSchemes**⁽²³¹⁾ [**ScrollBarSkinSchemeIndex**] (or **VerticalScrollBarSchemes**⁽²³²⁾, if a scroll bar is vertical).

Example:



3.2.7 TSRVLabel

TSRVLabel is a control displaying text.

Unit SRVLabel;

Syntax

```
TSRVLabel = class (TSRVGraphicControl338);
```

Hierarchy




TObject
TPersistent
TComponent
TControl
TGraphicControl
*TCustomSRVGraphicControl*³²⁸
*TSRVGraphicControl*³³⁸

Properties and events

This component publishes the following properties inherited from *TSRVGraphicControl*³³⁸:

- Alignment
- AutoSize
- DrawOnPrint;

- SkinManager;
- SkinSchemeIndex;
- properties inherited from TGraphicControl.
-

Property	Type	Default value	Meaning
 BackgroundColor	TColor	\$881C10	Text shadow color
 ForegroundColor	TColor	\$FFFFFF	Text color (overrides Font.Color)
 Offset	TRVPixel96Length	1	Shadow offset (0 disables shadow)

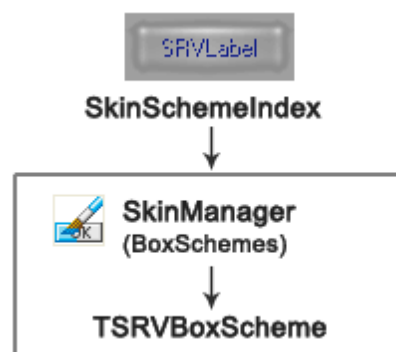
Painting

This component displays text specified in **Caption**. This component is transparent.

ForegroundColor is a text color, **BackgroundColor** is a text shadow color. A shadow is shifted by **Offset**. The caption is aligned according to **Alignment** (RTL **BiDiMode** inverts alignments). The control size is set automatically, if **AutoSize=True**.

If **SkinManager** is assigned, a border is drawn using **SkinManager.CurrentSkin**⁽²²⁹⁾.BoxSchemes⁽²³¹⁾ [**SkinSchemeIndex**].

Example:



3.2.8 TSRVListBox

Use TSRVListBox to display a scrollable list of items. Items can have captions, images, check boxes. A list can be hierarchical.

Unit SRVListBox;

Syntax

```
TSRVListBox = class (TSRVCustomListBox(335))
```

Hierarchy

TObject

TPersistent

TComponent

TControl

TWinControl

TSRVWinControl ⁽³³⁹⁾

TSRVCustomListBox ⁽³³⁵⁾

Properties and events

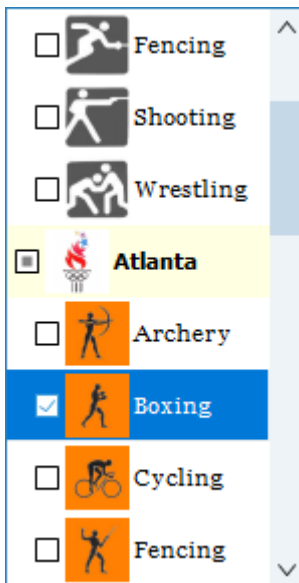
This component publishes public properties inherited from *TSRVCustomListBox* ⁽³³⁵⁾.

Painting

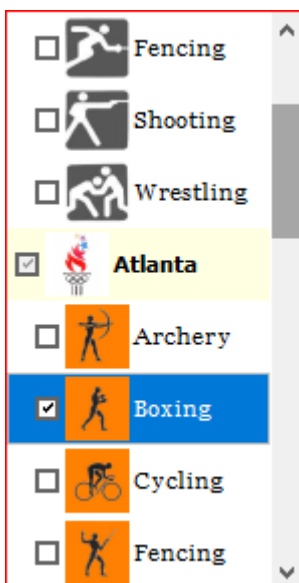
The default list box appearance (if skins are not assigned) depends on **SRVControlStyle**:

TSRVControlStyle ⁽²⁷⁵⁾ property:

srvcSimple (if *RVControlsPainter.Theme = rvctPaleBlue*):



srvcClassic (the scroll bar appearance depends on Windows, this screenshot is for Windows 10)::



The border around the list box has width defined in **DisabledBorderWidth**, **EnabledBorderWidth**, **MouseInBorderWidth**, **FocusedBorderWidth** properties. When auto-sizing the control, it is assumed that the border width is equal to **EnabledBorderWidth**.

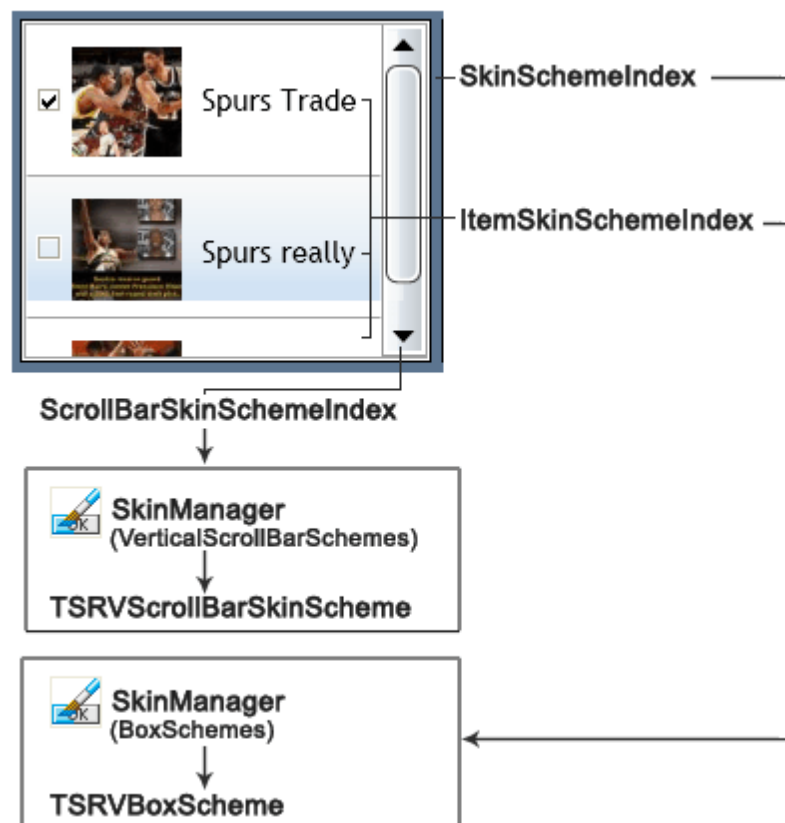
A border color depends on the list box state (normal, disabled, under the mouse pointer, focused) and **SRVControlStyle**: **TSRVControlStyle**⁽²⁷⁵⁾ property. If it is equal to *srvcsSimple*, colors are defined by **RVControlsPainter** (see the **TRichView** manual). If it is equal to *srvcsClassic*, colors are defined in **EnabledBorderColor**, **DisabledBorderColor**, **MouseInBorderColor**, **FocusedBorderColor** properties. The same is for colors used for drawing selected items (**SelectedItemBackColor**, **SelectedItemBorderColor**, **SelectedItemTextColor**).

Item colors are defined in the items properties⁽³⁷⁴⁾.

If **SkinManager** is assigned:

- the control is drawn using **SkinManager**.CurrentSkin⁽²²⁹⁾.BoxSchemes⁽²³¹⁾[**SkinSchemeIndex**];
- each item is drawn using **SkinManager**.CurrentSkin⁽²²⁹⁾.BoxSchemes⁽²³¹⁾[**ItemSkinSchemeIndex**];
- a scrollbar is drawn using **SkinManager**.CurrentSkin⁽²²⁹⁾.VerticalScrollBarSchemes⁽²³²⁾[**ScrollBarSkinSchemeIndex**].

Example:



3.2.9 TSRVMemo

TSRVMemo is a multiline plain-text edit control.

TSRVMemo edits Unicode text in all versions of Delphi.

Unit SRVMemo;

Syntax

```
TSRVMemo = class (TCustomSRVEdit326)
```

Hierarchy

TObject

TPersistent

TComponent

TControl

TWinControl

*TSRVWinControl*³³⁹

*TSRVEditControl*³³⁸

*TCustomSRVEdit*³²⁶

Properties and events

This component publishes the following new properties and properties inherited from TCustomSRVEdit³²⁶:

Property	Type	Default value	Meaning
■ Alignment	TAlignment	<i>taLeftJustify</i>	Horizontal text alignment (left/right/center). RTL BiDiMode inverts left and right alignments.
■ CharCase	TSRVEditCharCase	<i>srvecNormal</i>	Determines the case of the text within the edit control: <i>srvecNormal</i> , <i>srvecUpperCase</i> , <i>srvecLowerCase</i>
■ HScrollBarSchemeIndex	Integer	0	An index in SkinManager .CurrentSkin ²²⁹ .HorizontalScrollBarSchemes ²³¹ collection, allows applying skins to a horizontal scrollbar
■ Lines	TSRVMemoStrings		Contains the individual lines of text in the memo control. Lines contain Unicode strings for all versions of Delphi. WordWrap affects how Text is split into Lines .
■ MaxLength	Integer	0	Specifies the maximum number of characters the user can enter into the edit control (0 for unlimited)
■ ReadOnly	Boolean	<i>False</i>	Determines whether the user can change the text of the edit control.

Property	Type	Default value	Meaning
■ ScrollBars	TScrollStyle	<i>ssNone</i>	Determines whether the memo control has scroll bars.
■ SRVControlStyle	TSRVControlStyle ⁽²⁷⁵⁾	<i>srvcsSimple</i>	Defines a visual appearance (affects borders and scroll bars).
■ Text			Contains a text displayed by the editor (the same text as accessible by Lines).
■ VScrollBarSchemeIndex	Integer	0	An index in SkinManager.CurrentSkin ⁽²²⁹⁾ .VerticalScrollBarSchemes ⁽²³²⁾ collection, allows applying skins to a vertical scrollbar
■ WantReturns	Integer	<i>True</i>	Determines whether the user can insert return characters into the text.
■ WantTabs	Boolean	<i>False</i>	Determines whether the user can insert tab characters into the text.
■ WordWrap	Boolean	<i>True</i>	Determines whether the edit control automatically breaks lines so that the text wraps to fit the control's window.

- **DrawOnPrint**;
- **SkinManager**;
- **EditSkinSchemeIndex**;
- **SkinFontIndex**;
- **EnabledBorderColor**, **DisabledBorderColor**, **MouseInBorderColor**, **FocusedBorderColor** (all colors are used only if **SRVControlStyle** = *rvsrvcsClassic*)
- **DisabledBorderWidth**, **EnabledBorderWidth**, **MouseInBorderWidth**, **FocusedBorderWidth**;
- properties inherited from **TWinControl**.

Painting

If skins are not defined:

The border around the editor has width defined in **DisabledBorderWidth**, **EnabledBorderWidth**, **MouseInBorderWidth**, **FocusedBorderWidth** properties. When auto-sizing the editor, it is assumed that the border width is equal to **EnabledBorderWidth**.

A border color depends on the editors state (normal, disabled, under the mouse pointer, focused) and **SRVControlStyle**: **TSRVControlStyle**⁽²⁷⁵⁾ property. If it is equal to *srvcsSimple*, colors are defined by **RVControlsPainter** (see the **TRichView** manual). If it is equal to *srvcsClassic*, colors are defined in **EnabledBorderColor**, **DisabledBorderColor**, **MouseInBorderColor**, **FocusedBorderColor** properties.

If **SkinManager** is assigned:

- the control is drawn using **SkinManager.CurrentSkin**⁽²²⁹⁾.BoxSchemes⁽²³¹⁾[**EditSkinSchemeIndex**];
- the horizontal scroll bar is drawn using **SkinManager.CurrentSkin**⁽²²⁹⁾.HorizontalScrollBarSchemes⁽²³¹⁾[**HScrollBarSkinSchemeIndex**];

- the vertical scroll bar is drawn using **SkinManager**.CurrentSkin⁽²²⁹⁾.VerticalScrollBarSchemes⁽²³²⁾ [**VScrollBarSkinSchemeIndex**];
- font is defined in **SkinManager**.CurrentSkin⁽²²⁹⁾.Fonts⁽²³¹⁾ [**SkinFontIndex**].

3.2.10 TSRVPaintBox

TSRVPaintBox provides a canvas that applications can use for rendering an image.

Unit SRVPaintBox;

Syntax

```
TSRVPaintBox = class (TSRVCanvasControl(330));
```

Hierarchy

TObject

TPersistent

TComponent

TControl

TGraphicControl

TCustomSRVGraphicControl⁽³²⁸⁾

TSRVCanvasControl⁽³³⁰⁾

Properties and events

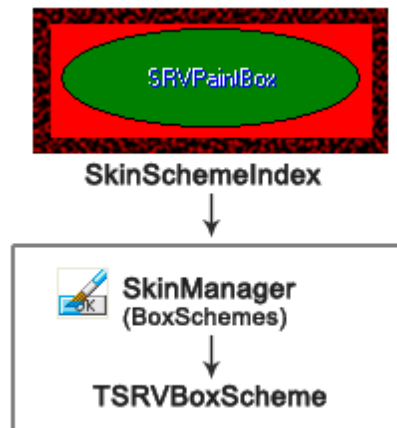
This component publishes the following properties and events inherited from TSRVCanvasControl⁽³³⁰⁾:

- **SkinManager**;
- **SkinSchemeIndex**;
- **DrawOnPrint**;
- **OnPaint**;
- **OnMouseEnter**, **OnMouseLeave**;
- properties inherited from TGraphicControl.

Painting

- 1) **Caption** is drawn at the position **Alignment** using Font (RTL **BiDiMode** inverts right and left alignments).
- 2) **OnPaint** event occurs (see TSRVCanvasControl⁽³³⁰⁾). Create an **OnPaint** event handler to draw the image of the paint box.
- 3) If **SkinManager** is assigned, a border is drawn using **SkinManager**.CurrentSkin⁽²²⁹⁾.BoxSchemes⁽²³¹⁾ [**SkinSchemeIndex**].

Example (Caption is empty; the text, the ellipse, the red rectangle are drawn in OnPaint):



3.2.11 TSRVPanel

TSRVPanel implements a generic panel control. You can also use panels to group controls together. This component is similar to TSRVGroupBox⁽³⁵¹⁾ (the difference is only in the caption position).

Unit SRVPanel;

Syntax

```
TSRVPanel = class(TCustomSRVPanel(329))
```

Hierarchy

TObject
 TPersistent
 TComponent
 TControl
 TWinControl
 TCustomControl
 TSRVCustomControl⁽³³⁴⁾
 TCustomSRVPanel⁽³²⁹⁾

Properties and events

This component publishes the following new properties and properties inherited from TCustomSRVPanel⁽³²⁹⁾:

Property	Type	Default value	Meaning
■ Alignment	TAlignment	taCenter	Horizontal text alignment (left/right/center). RTL BiDiMode inverts left and right alignments.
■ BorderWidth	Integer	1	Border width.
■ CaptionOffset	TRVPixel96Length	10	Caption offset (Caption cannot be displayed closer to the left and right side of the control than specified in this property).

Property	Type	Default value	Meaning
■ SRVControlStyle	TSRVControlStyle ⁽²⁷⁵⁾	<i>srvcSimple</i>	Defines visual appearance.

- **DrawOnPrint;**
- **SkinManager;**
- **SkinSchemeIndex;**
- **Color** (default value *clWhite*);
- properties inherited from TCustomControl.

Published properties used only if **SRVControlStyle** = *srvcClassic*:

Property	Type	Default value	Meaning
■ BorderColor	TColor	<i>\$00B8D8D7</i>	Border color
■ CornersOffset	Integer	10	Value in range 1..30. Defines how much border corners are rounded

Painting

Caption is aligned horizontally according to **Alignment**. **CaptionOffset** shifts the text (to the right, if **Alignment**=*taLeftJustify*, or to the left, if **Alignment**=*taRightJustify*). Caption is placed vertically in the middle. RTL **BiDiMode** inverts **Alignment**.

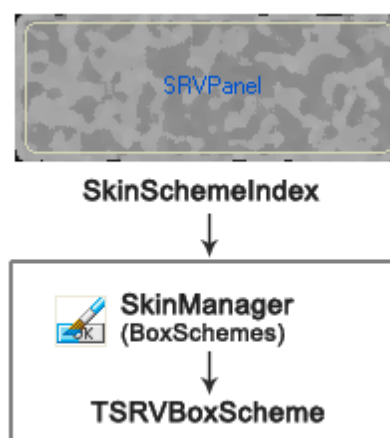
Background is drawn using **Color** property.

The Border has width defined in **BorderWidth** property (even in a skin mode). If **SRVControlStyle** = *srvcSimple*, the border color is defined by RVControlsPainter (see the TRichView manual). If it is equal to *srvcClassic*, the border color is defined in **BorderColor** property.

If **SRVControlStyle** = *srvcClassic*, the border can be rounded according to **CornersOffset** property.

If **SkinManager** is assigned, the control is drawn using **SkinManager.CurrentSkin**⁽²²⁹⁾.
BoxSchemes⁽²³¹⁾ [**SkinSchemeIndex**].

Example:



3.2.12 TSRVRadioButton

Radio buttons present a set of mutually exclusive options to the user; that is, only one radio button in a set can be selected at a time.

Unit SRVRadioButton;

Syntax

```
TSRVRadioButton = class (TSRVCustomControl334)
```

Hierarchy


TObject
TPersistent
TComponent
TControl
TWinControl
TCustomControl
*TSRVCustomControl*³³⁴

Properties and events

This component publishes the following properties inherited from *TSRVCustomControl*³³⁴:

- Alignment;
- ButtonSkinSchemeIndex;
- DrawOnPrint;
- SkinManager, SkinSchemeIndex, SkinFontIndex;
- SRVControlStyle;
- properties inherited from *TCustomControl*.

New published properties:

Property	Type	Default value	Meaning
 Checked	Boolean	<i>False</i>	Determines whether the option represented by the radio button is selected.

Painting

Painting

The default radio-button appearance (if skins are not assigned) depends on **SRVControlStyle**: *TSRVControlStyle*²⁷⁵ property:

srvcsSimple (if *RVControlsPainter.Theme = rvctPaleBlue*):

- ☒ Red
- ☐ Green
- ☐ Blue

srvcsClassic:



Positions of a circle and **Caption** depend on **Alignment**:

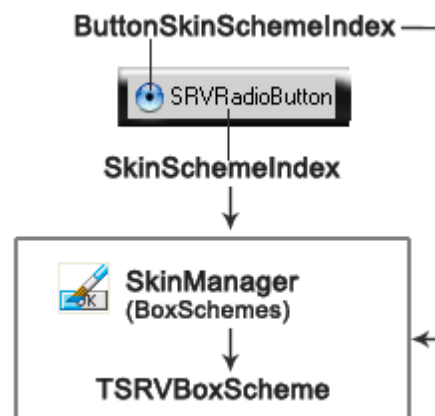
- *taLeftJustify*: the circle and the caption are aligned to the left; the circle is to the left of the caption;
- *taRightJustify*: the circle and the caption are aligned to the right; the circle is to the right of the caption;
- *taCenter*: the circle and the caption are in the middle; the circle is to the left/right of the caption depending in BiDiMode.

RTL BiDiMode inverts left and right **Alignment**.

Note: **Alignment** is different from the standard TRadioButton.Alignment. In TRadioButton, there are only left and right alignments, and they define Caption position relative to a circle. In TSRVRadioButton, there are three alignments, and they define alignment of a circle and Caption in the control (so the meanings are almost opposite).

If **SkinManager** is assigned, the control is drawn using **SkinManager.CurrentSkin**⁽²²⁹⁾.
.BoxSchemes⁽²³¹⁾ [**SkinSchemeIndex**], and the circle is drawn using **SkinManager.CurrentSkin**⁽²²⁹⁾.
.BoxSchemes⁽²³¹⁾ [**ButtonSkinSchemeIndex**].

Example:



3.3 Data-Aware Controls

Data-aware controls:



TSRVDBCheckBox⁽³⁶⁸⁾ – data-aware check box;



TSRVDBComboBox⁽³⁴⁵⁾ – data-aware combo box; hierarchical; with images, custom colors and fonts; with hints (suggestions);



TSRVDBEdit⁽³⁷⁰⁾ – data-aware plain-text one-line editor with hints (suggestions);



TSRVDBListBox⁽³⁷¹⁾ – data-aware scrollable list of items; hierarchical; with images, check boxes, custom colors and fonts;



TSRVDBMemo⁽³⁷²⁾ – data-aware plain-text multi-line editor;



TSRVDBText⁽³⁷³⁾ – data-aware text label.

3.3.1 TSRVDBCheckBox

TSRVDBCheckBox is a data-aware control that allows the user to select or deselect a single value.

Unit SRVDBCheckBox;

Syntax

```
TSRVDBCheckBox = class (TSRVCheckbox(344))
```

Hierarchy

TObject

TPersistent

TComponent

TControl

TWinControl

TCustomControl

TSRVCustomControl⁽³³⁴⁾

TSRVCheckbox⁽³⁴⁴⁾

Description



Use **TSRVDBCheckBox** to insert a data-aware check box in an editor. A check box presents an option to the user; the user can check it to select the option or uncheck it to deselect the option. A database check box (**TSRVDBCheckBox**) is much like an ordinary check box (**TSRVCheckbox**), except that it is aware of the data in a particular field of a dataset. Because check boxes can represent only two values (checked and unchecked), database check boxes are most appropriate for boolean fields. They can, however, be used to group the values of any field into two sets.

For applications that don't require the data-aware capabilities of **TSRVDBCheckBox**, use **TSRVCheckbox**⁽³⁴⁴⁾ instead to conserve system resources.

Properties and events

New properties:

Property	Type	Default value	Meaning
■ DataField	String	" (empty string)	Identifies the field from which the control displays data.
■ DataSource	TDataSource	nil	Links the control to a dataset.
Field	TField		Returns the TField object whose current value the control represents.
■ ReadOnly	Boolean	False	Determines whether the user can change the value of the field.

Property	Type	Default value	Meaning
 ValueChecked	String	'True'	Specifies the field value that corresponds to the checked state of the check box. ValueChecked can represent more than one value in a semicolon-delimited list.
 ValueUnchecked	String	'False'	Specifies the field value that corresponds to the unchecked state of the check box. ValueUnchecked can represent more than one value in a semicolon-delimited list.

3.3.2 TSRVDBComboBox

TSRVDBComboBox represents a data-aware combo box control.

Unit SRVDBComboBox;

Syntax

```
TSRVDBComboBox = class (TSRVComboBox345)
```

Hierarchy

TObject
TPersistent
TComponent
TControl
TWinControl
*TSRVWinControl*³³⁹
*TSRVEditControl*³³⁸
*TSRVCustomComboBox*³³¹
*TSRVComboBox*³⁴⁵

Description

Use **TSRVDBComboBox** to allow users to change the value of a field on the current record in a dataset either by selecting an item from a list or by typing in the edit box part of the control. The selected item or entered text becomes the new value of the field if the database combo box's **ReadOnly** property is *False*. The combo box can be customized to enable or disable typing in the edit region of the control, to display the list as a drop down or as a permanently displayed list, to sort the items in the list, and so on.

Unlike the standard **TDBComboBox**, **TSRVDBComboBox** can be used in **TDBCtrlGrid** even if its **Style** = *csSimple*.

Properties and events

New properties:

Property	Type	Default value	Meaning
■ DataField	String	" (empty string)	Identifies the field from which the control displays data.
■ DataSource	TDataSource	nil	Links the control to a dataset.
Field	TField		Returns the TField object whose current value the control represents.
■ ReadOnly	Boolean	False	Determines whether the user can change the value of the field.

3.3.3 TSRVDBEdit

TSRVDBEdit represents a single-line edit control that can display and edit a field in a dataset.

Unit SRVDBEdit;

Syntax

```
TSRVEdit = class (TSRVEdit(348))
```

Hierarchy

TObject
 TPersistent
 TComponent
 TControl
 TWinControl
 TSRVWinControl⁽³³⁹⁾
 TSRVEditControl⁽³³⁸⁾
 TCustomSRVEdit⁽³²⁶⁾
 TSRVEdit⁽³⁴⁸⁾

Description

Use TSRVDBEdit to enable users to edit a database field. TSRVDBEdit uses the Text property to represent the contents of the field.

TSRVDBEdit permits only a single line of text. If the field may contain lengthy data that would require multiple lines, consider using a TSRVDBMemo⁽³⁷²⁾ object.

If the application does not require the data-aware capabilities of TSRVDBEdit, use an edit control (TSRVEdit⁽³⁴⁸⁾) instead, to conserve system resources.

Properties and events

New properties:

Property	Type	Default value	Meaning
■ DataField	String	" (empty string)	Identifies the field from which the control displays data.
■ DataSource	TDataSource	nil	Links the control to a dataset.
Field	TField		Returns the TField object whose current value the control represents.
■ ReadOnly	Boolean	False	Determines whether the user can change the value of the field.

3.3.4 TSRVDBListBox

TSRVDBListBox represents a data-aware list box that allows users to change field values by selecting an item from a list.

Unit SRVDBListBox;

Syntax

```
TSRVListBox = class (TSRVListBox358)
```

Hierarchy

TObject
 TPersistent
 TComponent
 TControl
 TWinControl
 TSRVWinControl³³⁹
 TSRVCustomListBox³³⁵
 TSRVListBox³⁵⁸

Description

Use TSRVDBListBox to add a list box to a document that permits users to change the value of a field on the current record to one of a fixed set of choices.

If the application doesn't require the data-aware capabilities of TSRVDBListBox, use a list box (TSRVListBox³⁵⁸) instead to conserve system resources.

Properties and events

New properties:

Property	Type	Default value	Meaning
■ DataField	String	" (empty string)	Identifies the field from which the control displays data.
■ DataSource	TDataSource	nil	Links the control to a dataset.

Property	Type	Default value	Meaning
Field	TField		Returns the TField object whose current value the control represents.
■ ReadOnly	Boolean	<i>False</i>	Determines whether the user can change the value of the field.

3.3.5 TSRVDBMemo

TSRVDBMemo represents a multiline edit control that can display and edit a field in a dataset.

Unit SRVDBMemo;

Syntax

```
TSRVMemo = class (TSRVMemo361)
```

Hierarchy

TObject
 TPersistent
 TComponent
 TControl
 TWinControl
 TSRVWinControl³³⁹
 TSRVEditControl³³⁸
 TCustomSRVEdit³²⁶
 TSRVMemo³⁶¹

Description

Use TSRVDBMemo to let users edit a field that may contain lengthy textual data or to simply display the contents of such a field. TSRVDBMemo uses the Text property to represent the contents of the field.

TSRVDBMemo permits multiple lines of text. Thus, TSRVDBMemo is appropriate for long alphanumeric fields or text binary large objects (BLOBs). For short alphanumeric fields, consider using a TSRVDBEdit³⁷⁰ component instead.

If the application doesn't require the data-aware capabilities of TSRVDBMemo, use a memo control (TSRVMemo³⁶¹) instead, to conserve system resources.

Properties and events

New properties:

Property	Type	Default value	Meaning
■ DataField	String	<i>" (empty string)</i>	Identifies the field from which the control displays data.

Property	Type	Default value	Meaning
■ DataSource	TDataSource	<i>nil</i>	Links the control to a dataset.
Field	TField		Returns the TField object whose current value the control represents.
■ ReadOnly	Boolean	<i>False</i>	Determines whether the user can change the value of the field.

3.3.6 TSRVDBText

TSRVDBText represents a data-aware control that displays the value of a field in a document.

Unit SRVDBLabel;

Syntax

```
TSRVLabel = class (TSRVLabel(357));
```

Hierarchy

TObject
 TPersistent
 TComponent
 TControl
 TGraphicControl
 TCustomSRVGraphicControl⁽³²⁸⁾
 TSRVGraphicControl⁽³³⁸⁾
 TSRVLabel⁽³⁵⁷⁾

Description

Use TSRVDBText to display the contents of a field in the current record of a dataset on a form. Field values displayed by database text controls cannot be modified by the user using the text control. To allow the user to edit the field value, use TSRVDBEdit⁽³⁷⁰⁾ or TSRVDBMemo⁽³⁷²⁾ instead.

If the application does not require the data-aware capabilities of TSRVDBText, use the label component (TSRVLabel⁽³⁵⁷⁾) instead to conserve system resources.

Properties and events

New properties:

Property	Type	Default value	Meaning
■ DataField	String	<i>" (empty string)</i>	Identifies the field from which the control displays data.
■ DataSource	TDataSource	<i>nil</i>	Links the control to a dataset.
Field	TField		Returns the TField object whose current value the control represents.

3.4 Classes of properties

- `TSRVComboBoxItemCollection`⁽³⁷⁴⁾ – collection of `TSRVComboBoxItem`⁽³⁷⁴⁾ items (used in `TSRVComboBox`⁽³⁴⁵⁾)
- `TSRVListBoxItemCollection`⁽³⁷⁴⁾ – collection of `TSRVListBoxItem`⁽³⁷⁴⁾ items (used in `TSRVListBox`⁽³⁵⁸⁾)
- `TSRVImageScrollCollection`⁽³⁷⁴⁾ – collection of `TSRVImageScrollItem`⁽³⁷⁸⁾ items (used in `TSRVImageScroll`⁽³⁵³⁾)

3.4.1 TSRVComboBoxItemCollection, TSRVListBoxItemCollection

`TSRVComboBoxItemCollection` is a type of collection of items of `TSRVComboBox`⁽³⁴⁵⁾. Item type is `TSRVComboBoxItem`⁽³⁷⁴⁾.

`TSRVListBoxItemCollection` is a type of collection of items of `TSRVListBox`⁽³⁵⁸⁾. Item type is `TSRVListBoxItem`⁽³⁷⁴⁾.

Unit `SRVTypeItems`;

Syntax

```
TSRVComboBoxItemCollection = class (TCollection) ;
TSRVListBoxItemCollection = class (TSRVComboBoxItemCollection) ;
```

Hierarchy

TObject
TPersistent
TCollection

3.4.2 TSRVImageScrollCollection

`TSRVImageScrollCollection` is a type of collection of items of `TSRVImageScroll`⁽³⁵³⁾. Item type is `TSRVImageScrollItem`⁽³⁷⁸⁾.

Unit `SRVImageScroll`;

Syntax

```
TSRVImageScrollCollection = class (TCollection) ;
```

Hierarchy

TObject
TPersistent
TCollection

3.4.3 TSRVComboBoxItem, TSRVListBoxItem

`TSRVComboBoxItem` is a type of item for collection of items⁽³⁷⁴⁾ of `TSRVComboBox`⁽³⁴⁵⁾.

`TSRVListBoxItem` is a type of item for collection of items⁽³⁷⁴⁾ of `TSRVListBox`⁽³⁵⁸⁾.

Unit `SRVTypeItems`;

Syntax

```
TSRVComboBoxItem = class (TCollectionItem) ;
```

```
TSRVLBoxItem = class (TSRVBoxItem);
```

Hierarchy

TObject

TPersistent

TCollectionItem

Properties of TSRVCBoxItem

Published properties

Property	Type	Default value	Meaning
■ AlignImage	TLRAlignment	<i>tlrLeftJustify</i>	Specifies the position of image relative to Caption . RTL BiDiMode inverts value of this property.
■ Alignment	TAlignment	<i>taLeftJustify</i>	Specifies the horizontal alignment of content inside the item. RTL BiDiMode inverts value of this property.
■ BackColor	TColor	<i>clWindow</i>	Background color.
■ Caption	TRVUnicodeString		Specifies the text value of the item. Can be hidden, see Options .
■ Color	TColor	<i>clWindowText</i>	Text color for Caption .
■ Enabled	Boolean	<i>True</i>	Controls whether the item can be selected.
■ Expanded	Boolean	<i>True</i>	Specifies whether the item is expanded (children are visible)
■ FontName	TFontName	'Tahoma'	Font name for displaying Caption . Used only if UseItemFontNames = <i>True</i> for the parent control.
■ Hint	String		Reserved
■ ImageIndex	Integer	-1	Identifies the image associated with this item. ImageIndex is an index into the ImageList property of the parent control. When the item is selected, the item displays the image specified by the SelectedImageIndex property instead, if that property is assigned. Can be hidden, see Options .
■ Indent	TRVPixel96Length	0	Defines both the indent from the left side to the item content, and the level of the item in the items hierarchy. Items with greater Indent are children of the preceding item with smaller Indent .

Property	Type	Default value	Meaning
■ Options	TSRVItemOptions	[<i>srvioShowCheckBox</i> , <i>srvioShowImage</i> , <i>srvioShowText</i> , <i>srvioCanSelect</i> , <i>srvioCanHaveChildren</i>]	Item options (see below)
■ SkinSchemeIndex	Integer	-1	Index in the SkinManager.CurrentSkin ⁽²²⁹⁾ . BoxSchemes ⁽²³¹⁾ (SkinManager is a property of the parent control). Defines a visual appearance of the item. If equals to -1, ItemSkinSchemeIndex property of the parent control is used.
■ SelectedImageIndex	Integer	-1	Identifies the image associated with this item when it is selected. SelectedImageIndex is an index into the ImageList property of the parent control. If it equals to -1, ImageIndex property is used instead. Can be hidden, see Options . Can be also used for highlighted (under the mouse pointer) items, if see UseSelectedImagesInHotState of TSRVListBox ⁽³⁵⁸⁾ and TSRVComboBox ⁽³⁴⁵⁾ .
■ Style	TFontStyles	[]	Font styles to display Caption .
■ Tag	Integer	0	Stores an integer value as part of the item.

type

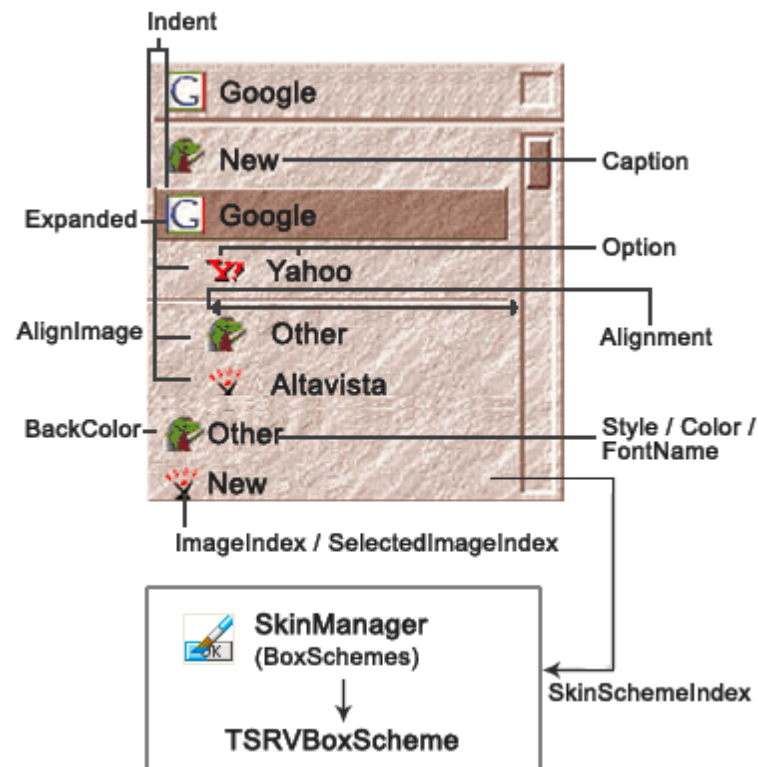
```

TLRAlignment = (tlraLeftJustify, tlraRightJustify);
TSRVItemOption = (srvioShowCheckBox, srvioShowImage, srvioShowText,
    srvioCanSelect, srvioCanHaveChildren);
TSRVItemOptions = set of TSRVItemOption;

```

Options can include the following values:

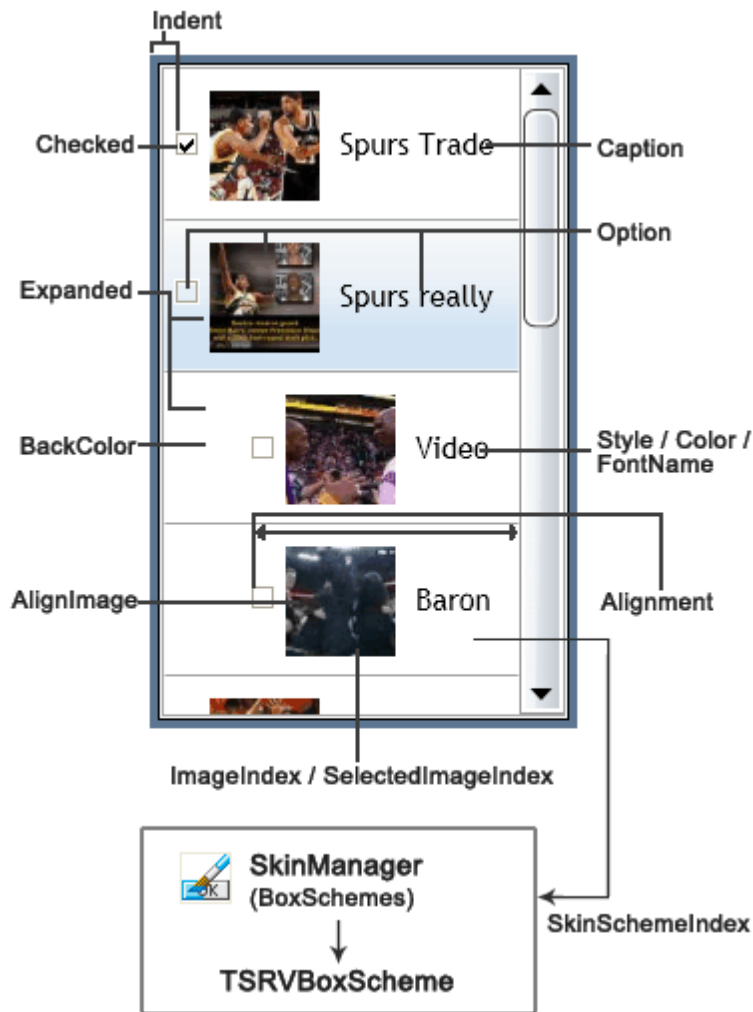
- *srvioShowCheckBox*: if excluded, a check box for this item is not shown (check boxes are shown only if **ShowCheckBoxes** property of the parent control is *True*); this option is ignored for combo boxes, it is used only in list boxes;
- *srvioShowImage*: shows the image specified in **ImageIndex** and **SelectedImageIndex**;
- *srvioShowText*: shows **Caption**;
- *srvioCanSelect*: allows this item to be selected; similar to **Enabled**, but does not provide a visual indication;
- *srvioCanHaveChildren*: allows this item to have children; if excluded, subsequent items are not treated as its children, even if they have greater **Indents**;



Properties of TSRVLBoxItem

In addition to the properties inherited from **TSRVBoxItem**, **TSRVLBoxItem** has the following published properties:

Property	Type	Default value	Meaning
■ AlignCheckBox	TLRAlignment	<i>tlrLeftJustify</i>	Specifies the position of checkbox relative to Caption and image. RTL BiDiMode inverts value of this property.
■ Checked	Boolean	<i>False</i>	Specifies whether the check box is checked. See also State property.
■ State	TCheckBoxState	<i>cbUnchecked</i>	Indicates whether the check box is selected (<i>cbChecked</i>), deselected (<i>cbUnchecked</i>), or grayed (<i>cbGrayed</i>)



3.4.4 TSRVImageScrollItem

TSRVImageScrollItem is a type of item for collection of items⁽³⁷⁴⁾ of TSRVImagesScroll⁽³⁵³⁾.

Unit SRVImagesScroll;

Syntax

```
TSRVImageScrollItem = class (TCollectionItem);
```

Hierarchy

TObject

TPersistent

TCollectionItem

Properties

Published properties

Property	Type	Default value	Meaning
■ Enabled	Boolean	<i>True</i>	Determines whether the item can be selected.
■ Hint	String		Reserved
■ ImageIndex	Integer	-1	Identifies the image associated with this item. ImageIndex is an index into the ImageList property of the parent control. When the item is selected, the item displays the image specified by the SelectedImageIndex property instead, if that property is assigned.
■ Link	TRVUnicodeString		Reserved
■ Name	TRVUnicodeString		Item name, not displayed
■ SkinSchemeIndex	Integer	-1	Index in the SkinManager.CurrentSkin ⁽²²⁹⁾ . BoxSchemes ⁽²³¹⁾ (SkinManager is a property of the parent control). Defines a visual appearance of the item. If equals to -1, ItemSkinSchemeIndex of the parent control is used.
■ SelectedImageIndex	Integer	-1	Identifies the image associated with this item when it is selected. SelectedImageIndex is an index into the ImageList property of the parent control. If it equals to -1, ImageIndex property is used instead.
■ ShowHint	Boolean	<i>True</i>	Reserved
■ Tag	LongInt	0	Stores an integer value as part of the item.
■ Visible	Boolean	<i>True</i>	Determines whether the item is visible

Index

- A -

AcceptDragDropFormats 47
 TSRichViewEdit 47
 AcceptPasteFormats 47
 TSRichViewEdit 47
 ActionSave 285
 TsrvCustomActionThatNeedsDocProps 285
 Active 151
 TSRVReadModeProperty 151
 ActiveEditor 47
 TSRichViewEdit 47
 Align 148, 149
 TSRVPropertyTBH 148
 TSRVPropertyTBV 149
 AlignButton 128, 190
 TSRVCustomPropertyTB 128
 TSRVToolBar 190
 AlignPageH 144
 TSRVPagePositionProperty 144
 AlignPageV 144
 TSRVPagePositionProperty 144
 AlignText 175, 190, 197
 TSRVPageScroll 175
 TSRVToolBar 190
 TSRVToolWindow 197
 AllowAllUp 272
 TSRVToolButton 272
 AllowCloseLastTab 262
 TSRVCloseButton 262
 AnimationMode 47
 TSRichViewEdit 47
 AutoDarkMode 252
 TSRVTabSet 252
 AutoDisplay 172
 TDBSRichViewEdit 172
 AutoScrolled 176
 TSRVPageScroll 176
 AutoSize 191
 TSRVToolBar 191
 AutoTabWidth 251
 TSRVTabSet 251
 AutoUpdate 176, 206
 TSRVPageScroll 176
 TSRVPrint 206

AutoVScrollBarPosition 155
 TSRVViewProperty 155
 AutoWidth 135
 TSRVPageProperty 135

- B -

BackgroundProperty 48
 TSRichViewEdit 48
 BackgroundTextColor 176
 TSRVPageScroll 176
 BeginComponentsUpdate 72
 TSRichViewEdit 72
 BeginUpdate 182, 217
 TSRVPageScroll 182
 TSRVPrint 217
 BiDiMode 48
 TSRichViewEdit 48
 BorderColor 224
 TSRVPreview 224
 BorderPen 135
 TSRVPageProperty 135
 BorderWidth 128, 191, 198, 224
 TSRVCustomPropertyTB 128
 TSRVPreview 224
 TSRVToolBar 191
 TSRVToolWindow 198
 BottomMargin 135
 TSRVPageProperty 135
 BottomMargin100Pix 48
 TSRichViewEdit 48
 BottomMarginPix 48
 TSRichViewEdit 48
 BoxPosition 317
 TsrvActionCustomInsertSidenote 317
 BoxProperties 317
 TsrvActionCustomInsertSidenote 317
 BoxSchemes 231
 TSRVSkin 231
 ButtonCol 191, 198
 TSRVToolBar 191
 TSRVToolWindow 198
 ButtonHeight 128, 191, 198
 TSRVCustomPropertyTB 128
 TSRVToolBar 191
 TSRVToolWindow 198
 Buttons 128, 191, 198
 TSRVCustomPropertyTB 128
 TSRVToolBar 191
 TSRVToolWindow 198

ButtonWidth 128, 192, 198
 TSRVCustomPropertyTB 128
 TSRVToolBar 192
 TSRVToolWindow 198

- C -

CachedPagesCount 177
 TSRVPageScroll 177
 CacheScrollLimit 177
 TSRVPageScroll 177
 CalculateAllPagePositions 72
 TSRichViewEdit 72
 CalculatePageCount 72
 TSRichViewEdit 72
 CalculatePagePosition 72
 TSRichViewEdit 72
 CanMoveTabs 252
 TSRVTabSet 252
 CanScroll 48
 CanUpdate 48
 TSRichViewEdit 48
 CanUpdateMargin 49
 TSRichViewEdit 49
 CanUpdatePosition 48
 TSRichViewEdit 48
 Caption 272
 TSRVToolButton 272
 CaretBlinkTime 49
 TSRichViewEdit 49
 CaretPen 135
 TSRVPageProperty 135
 CaretPos 49
 TSRichViewEdit 49
 CaretVisible 49
 TSRichViewEdit 49
 Center 206
 TSRVPrint 206
 ChangeDelay 177
 TSRVPageScroll 177
 CheckMargin 217
 TSRVPrint 217
 CheckMargins 187
 TSclRVRule 187
 Clear 73
 TSRichViewEdit 73
 ClipMargins 207
 TSRVPrint 207
 CloseButton 252
 TSRVTabSet 252

Color 129, 192, 199, 302, 305, 308
 TsrvActionLayoutPrint 302
 TsrvActionLayoutRead 305
 TsrvActionLayoutSideToSide 308
 TSRVCustomPropertyTB 129
 TSRVToolBar 192
 TSRVToolWindow 199
 ColorBegin 124
 TSRVBackgroundProperty 124
 ColorDisable 129, 192, 199
 TSRVCustomPropertyTB 129
 TSRVToolBar 192
 TSRVToolWindow 199
 ColorDown 129, 192, 199
 TSRVCustomPropertyTB 129
 TSRVToolBar 192
 TSRVToolWindow 199
 ColorEnd 124
 TSRVBackgroundProperty 124
 ColorMiddle 124
 TSRVBackgroundProperty 124
 ColorSelect 129, 193, 200
 TSRVCustomPropertyTB 129
 TSRVToolBar 193
 TSRVToolWindow 200
 Columns 177
 TSRVPageScroll 177
 Control 284
 TsrvAction 284
 ConvertPageFormatToPageSize 142
 TSRVPageProperty 142
 ConvertPageSizeToPageFormat 142
 TSRVPageProperty 142
 ConvertRVtoSRV 73
 TSRichViewEdit 73
 ConvertSRVtoRV 73
 TSRichViewEdit 73
 ConvertToDifferentUnits 74
 TSRichViewEdit 74
 ConvertToEMU 74
 TSRichViewEdit 74
 ConvertToPixels 74
 TSRichViewEdit 74
 ConvertToTwips 74
 TSRichViewEdit 74
 Copy 75
 TSRichViewEdit 75
 CPEventKind 49
 TSRichViewEdit 49
 Create 75

Create 75
 TSRichViewEdit 75
 CurControl 50
 TSRichViewEdit 50
 CurrentNote 50
 TSRichViewEdit 50
 CurrentNoteParentEditor 50
 TSRichViewEdit 50
 CurrentPage 50
 TSRichViewEdit 50
 CurrentSkin 229
 TSRVSkinManager 229
 Cut 75
 TSRichViewEdit 75

- D -

DarkMode 156, 252
 TSRVTabSet 252
 TSRVViewProperty 156
 DarkModeUI 156
 TSRVViewProperty 156
 DataField 172
 TDBSRichViewEdit 172
 DataSource 172
 TDBSRichViewEdit 172
 DecButtonWidth 244
 TSRVScrollBar 244
 DeletePage 75
 TSRichViewEdit 75
 Destroy 76
 TSRichViewEdit 76
 DistanceFromText 131
 TSRVLineNumberProperty 131
 Document 51
 TSRichViewEdit 51
 Document parts in TSRichViewEdit 36
 Down 272
 TSRVToolButton 272
 DrawMetafile 76
 TSRichViewEdit 76
 DrawPage 76
 TSRichViewEdit 76

- E -

Elements 231
 TSRVSkin 231
 Enabled 264, 272
 TSRVTabItem 264

TSRVToolButton 272
 EndComponentsUpdate 77
 TSRichViewEdit 77
 EndUpdate 182, 218
 TSRVPageScroll 182
 TSRVPrint 218
 Execute 202
 TSRVToolWindow 202
 ExternalRV 173
 TDBSRichViewEdit 173
 ExternalRVStyle 51
 TSRichViewEdit 51
 ExternalRVStyleFooter 51
 TSRichViewEdit 51
 ExternalRVStyleHeader 51
 TSRichViewEdit 51

- F -

FacingPages 136
 TSRVPageProperty 136
 FieldFormat 173
 TDBSRichViewEdit 173
 FillType 124
 TSRVBackgroundProperty 124
 FindNextCheckpoint 77
 TSRichViewEdit 77
 FindNextHeading 78
 TSRichViewEdit 78
 FindNextHyperlink 78
 TSRichViewEdit 78
 FindNextItem 79
 TSRichViewEdit 79
 FindPriorCheckpoint 80
 TSRichViewEdit 80
 FindPriorHeading 80
 TSRichViewEdit 80
 FindPriorHyperlink 81
 TSRichViewEdit 81
 FindPriorItem 81
 TSRichViewEdit 81
 First 183, 226
 TSRVPageScroll 183
 TSRVPreview 226
 FirstCurPage 82
 TSRichViewEdit 82
 FirstPageNo 51
 TSRichViewEdit 51
 FirstTabIndex 252
 TSRVTabSet 252

FixMarginsMode 52
 TSRichViewEdit 52
 Flat 244
 TSRVScrollBar 244
 FntTab 240
 TSRVTabSetSkinScheme 240
 FocusBlinkDelay 244
 TSRVScrollBar 244
 Font 132, 321
 TsrvActionInsertNote 321
 TSRVLineNumberProperty 132
 Fonts 231
 TSRVSkin 231
 FooterPen 156
 TSRVViewProperty 156
 FooterTitleColor 156
 TSRVViewProperty 156
 FooterTitleFont 157
 TSRVViewProperty 157
 FooterVisible 136, 302, 308
 TsrvActionLayoutPrint 302
 TsrvActionLayoutSideToSide 308
 TSRVPageProperty 136
 FooterY 136
 TSRVPageProperty 136
 Format 82
 TSRichViewEdit 82
 FrameBorderColor 224
 TSRVPreview 224
 FrameBorderWidth 224
 TSRVPreview 224
 FrameCountX 207
 TSRVPrint 207
 FrameCountY 203, 208
 TSRVPrint 208
 FrameHeightPix 208
 TSRVPrint 208
 FrameWidthPix 209
 TSRVPrint 209
 FreePosPage 157
 TSRVViewProperty 157

- G -

GetCaretPosInUnits 83
 TSRichViewEdit 83
 GetCurrentLineCol 83
 TSRichViewEdit 83
 GetFirstItemVisibleRV 83
 TSRichViewEdit 83
 GetFirstVisiblePage 83
 TSRichViewEdit 83
 GetItemAt 84
 TSRichViewEdit 84
 GetItemBounds 84
 TSRichViewEdit 84
 GetItemBounds100 84
 TSRichViewEdit 84
 GetItemCoords100 85
 TSRichViewEdit 85
 GetItemPages 86
 TSRichViewEdit 86
 GetLastItemVisibleRV 86
 TSRichViewEdit 86
 GetLastVisiblePage 86
 TSRichViewEdit 86
 GetLeftMargin100Pix 54
 TSRichViewEdit 54
 GetLeftMarginPix 54
 TSRichViewEdit 54
 GetPageAt 86, 183
 TSRichViewEdit 86
 TSRVPageScroll 183
 GetPageClientRect 87
 TSRichViewEdit 87
 GetPageLastItemNo 87
 TSRichViewEdit 87
 GetPageNo 87
 TSRichViewEdit 87
 GetPageProperties 187
 TScIRVRuler 187
 GetPageStartItemNo 88
 TSRichViewEdit 88
 GetPageStartTableItem 88
 TSRichViewEdit 88
 GetPageStartTableRow 88
 TSRichViewEdit 88
 GetRightMargin100Pix 60
 TSRichViewEdit 60
 GetRightMarginPix 60
 TSRichViewEdit 60
 GetTableIconItem 89
 TSRichViewEdit 89
 GetTableIconRVData 89
 TSRichViewEdit 89
 GlobalPageBackgroundColor 125
 TSRVBackgroundProperty 125
 GroupIndex 273
 TSRVToolButton 273

- H -

- HeaderFooterShade 157
 - TSRVViewProperty 157
- HeaderPen 158
 - TSRVViewProperty 158
- HeaderTitleColor 158
 - TSRVViewProperty 158
- HeaderTitleFont 158
 - TSRVViewProperty 158
- HeaderVisible 137, 302, 308
 - TsrvActionLayoutPrint 302
 - TsrvActionLayoutSideToSide 308
 - TSRVPageProperty 137
- HeaderY 137
 - TSRVPageProperty 137
- Height 262
 - TSRVCloseButton 262
- HiddenDistances 145
 - TSRVPagePositionProperty 145
- HidePageBackgroundImage 151, 305
 - TsrvActionLayoutRead 305
 - TSRVReadModeProperty 151
- Hint 272
 - TSRVToolButton 272
- HintCancel 193, 200
 - TSRVToolBar 193
 - TSRVToolWindow 200
- HintFont 159, 193, 200
 - TSRVToolBar 193
 - TSRVToolWindow 200
 - TSRVViewProperty 159
- HintHeight 193, 200
 - TSRVToolBar 193
 - TSRVToolWindow 200
- HintPrefixText 159
 - TSRVViewProperty 159
- HMaxScrollPos 52
 - TSRichViewEdit 52
- HorizontalAlign 262
 - TSRVCloseButton 262
- HorizontalScrollBarSchemes 231
 - TSRVSkin 231
- HorizontalTabSetSchemes 231
 - TSRVSkin 231
- HPadding 145, 300, 303, 305, 308
 - TsrvActionLayoutDraft 300
 - TsrvActionLayoutPrint 303
 - TsrvActionLayoutRead 305
- TsrvActionLayoutSideToSide 308
- TSRVPagePositionProperty 145
- HScrollBar 52
 - TSRichViewEdit 52
- HScrollBarSchemeIndex 53
 - TSRichViewEdit 53
- HScrollPos 53
 - TSRichViewEdit 53
- HTMLReadProperties 53
 - TSRichViewEdit 53
- HTMLSaveProperties 53
 - TSRichViewEdit 53

- I -

- IconStyle 159
 - TSRVViewProperty 159
- IgnoreEscape 173
 - TDBSRichViewEdit 173
- ImageIndex 264
 - TSRVTabItem 264
- ImageIndexDisable 273
 - TSRVToolButton 273
- ImageIndexDown 273
 - TSRVToolButton 273
- ImageIndexMove 273
 - TSRVToolButton 273
- ImageIndexNormal 274
 - TSRVToolButton 274
- ImageList 130, 194, 201, 252
 - TSRVCustomPropertyTB 130
 - TSRVTabSet 252
 - TSRVToolBar 194
 - TSRVToolWindow 201
- ImgAddTabButton 240
 - TSRVTabSetSkinScheme 240
- ImgBackgroundBegin 240
 - TSRVTabSetSkinScheme 240
- ImgBackgroundCenter 240
 - TSRVTabSetSkinScheme 240
- ImgBackgroundEnd 240
 - TSRVTabSetSkinScheme 240
- ImgBorderBackground 240
 - TSRVTabSetSkinScheme 240
- ImgBorderBegin 240
 - TSRVTabSetSkinScheme 240
- ImgBorderEnd 240
 - TSRVTabSetSkinScheme 240
- ImgCloseButton 240
 - TSRVTabSetSkinScheme 240

ImgDecButton 240
 TSRVTabSetSkinScheme 240
 ImgIncButton 240
 TSRVTabSetSkinScheme 240
 ImgTabBackground 240
 TSRVTabSetSkinScheme 240
 ImgTabBegin 240
 TSRVTabSetSkinScheme 240
 ImgTabEnd 240
 TSRVTabSetSkinScheme 240
 IncButtonWidth 245
 TSRVScrollBar 245
 Indent 130, 194, 253
 TSRVCustomPropertyTB 130
 TSRVTabSet 253
 TSRVToolBar 194
 IndentX 262
 TSRVCloseButton 262
 IndentY 262
 TSRVCloseButton 262
 InsertPageBreak 89
 TSRRichViewEdit 89
 IntegerCount 209
 TSRVPrint 209
 InZoomRect 89
 TSRRichViewEdit 89

- K -

Kind 245, 253
 TSRVScrollBar 245
 TSRVTabSet 253

- L -

LargeChange 246
 TSRVScrollBar 246
 Last 183, 226
 TSRVPageScroll 183
 TSRVPreview 226
 LastCurPage 90
 TSRRichViewEdit 90
 LastPageNo 53
 TSRRichViewEdit 53
 LastTabIndex 253
 TSRVTabSet 253
 LeftMargin 137
 TSRVPageProperty 137
 LeftMargin100Pix 54
 TSRRichViewEdit 54

LeftMarginPix 54
 TSRRichViewEdit 54
 LineNumberProperty 54
 TSRRichViewEdit 54
 LoadRVF 90
 TSRRichViewEdit 90
 LoadRVFFromStream 90
 TSRRichViewEdit 90

- M -

MainDocumentShade 159
 TSRVViewProperty 159
 MainPen 160
 TSRVViewProperty 160
 MainTitleColor 160
 TSRVViewProperty 160
 MainTitleFont 160
 TSRVViewProperty 160
 MarginHorizontal 178
 TSRVPageScroll 178
 Margins 253
 TSRVTabSet 253
 MarginsPen 161
 TSRVViewProperty 161
 MarginsRectVisible 161
 TSRVViewProperty 161
 MarginVertical 178
 TSRVPageScroll 178
 MarkdownProperties 54
 TSRRichViewEdit 54
 MaxPageColCount 146, 151
 TSRVPagePositionProperty 146
 TSRVReadModeProperty 151
 MaxPrintedItemNo 54
 TSRRichViewEdit 54
 MaxPrintedOffsInItem 54
 TSRRichViewEdit 54
 MaxTabSize 254
 TSRVTabSet 254
 MenuHButtons 55
 TSRRichViewEdit 55
 MenuHorizontal 55
 TSRRichViewEdit 55
 MenuVButtons 55
 TSRRichViewEdit 55
 MenuVertical 56
 TSRRichViewEdit 56
 MetafileCompatibility 210
 TSRVPrint 210

MinPageColCount 151
 TSRVReadModeProperty 151

MinPrintedItemNo 56
 TSRichViewEdit 56

MinPrintedOffsInItem 56
 TSRichViewEdit 56

MinTabSize 254
 TSRVTabSet 254

MirrorMargins 138
 TSRVPageProperty 138

MirrorText 254
 TSRVTabSet 254

MouseWheelZoom 161
 TSRVViewProperty 161

MouseWheelZoomMax 280

MouseWheelZoomMin 280

MouseWheelZoomSpeed 280

- N -

Name 232, 240
 TSRVSkin 232
 TSRVTabSetSkinScheme 240

Next 183, 226
 TSRVPageScroll 183
 TSRVPreview 226

NextCurCheckpoint 90
 TSRichViewEdit 90

NextCurHeading 90
 TSRichViewEdit 90

NextCurHyperlink 91
 TSRichViewEdit 91

NextCurItem 91
 TSRichViewEdit 91

NextCurPage 91
 TSRichViewEdit 91

NoMetafiles 210
 TSRVPrint 210

NoPageSpacing 308
 TsrActionLayoutSideToSide 308

Notes and text boxes in TSRichViewEdit 38

- O -

OffsetX 56
 TSRichViewEdit 56

OffsetY 57
 TSRichViewEdit 57

OffXPix 210
 TSRVPrint 210

OffYPix 211
 TSRVPrint 211

OnAfterMarginChanged 188
 TScLRuler 188

OnAfterOleDrop 103
 TSRichViewEdit 103

OnAfterPaint 227
 TSRVPreview 227

OnAssignImageFileName 103
 TSRichViewEdit 103

OnBeforeOleDrop 103
 TSRichViewEdit 103

OnBeforePaint 227
 TSRVPreview 227

OnCaretGetOut 104
 TSRichViewEdit 104

OnCaretMove 104
 TSRichViewEdit 104

OnCaretMoving 104
 TSRichViewEdit 104

OnChange 104, 259, 292
 TSRichViewEdit 104
 TsrActionPageSetup 292
 TSRVTabSet 259

OnChangeActiveEditor 104
 TSRichViewEdit 104

OnChangeCurrentControl 104
 TSRichViewEdit 104

OnChangeViewModeAfter 104
 TSRichViewEdit 104

OnChangeViewModeBefore 105
 TSRichViewEdit 105

OnChanging 105
 TSRichViewEdit 105

OnCheckControl 105
 TSRichViewEdit 105

OnCheckpointVisible 105
 TSRichViewEdit 105

OnCheckStickingItems 106
 TSRichViewEdit 106

OnClickButton 195, 202
 TSRVToolBar 195
 TSRVToolWindow 202

OnClickPage 106
 TSRichViewEdit 106

OnCloseTab 259
 TSRVTabSet 259

OnContextPopup 106
 TSRichViewEdit 106

OnControlAction 106

- OnControlAction 106
 - TSRichViewEdit 106
- OnCopy 106
 - TSRichViewEdit 106
- OnCurParaStyleChanged 106
 - TSRichViewEdit 106
- OnCurrentPageChange 107
 - TSRichViewEdit 107
- OnCurTextStyleChanged 107
 - TSRichViewEdit 107
- OnDrawArea 248
 - TSRVScrollBar 248
- OnDrawBackground 259
 - TSRVTabSet 259
- OnDrawBorder 248, 259
 - TSRVScrollBar 248
 - TSRVTabSet 259
- OnDrawButtonBackground 195
 - TSRVToolBar 195
- OnDrawCloseButton 259
 - TSRVTabSet 259
- OnDrawDecButton 248, 259
 - TSRVScrollBar 248
 - TSRVTabSet 259
- OnDrawHyperlink 107
 - TSRichViewEdit 107
- OnDrawIncButton 248, 259
 - TSRVScrollBar 248
 - TSRVTabSet 259
- OnDrawPageBorder 185
 - TSRVPageScroll 185
- OnDrawPageNumber 185
 - TSRVPageScroll 185
- OnDrawSelectedPage 186
 - TSRVPageScroll 186
- OnDrawTab 259
 - TSRVTabSet 259
- OnDrawThumbTab 248
 - TSRVScrollBar 248
- OnDropFile 107
 - TSRichViewEdit 107
- OnDropFiles 107
 - TSRichViewEdit 107
- OnEnter 195, 202
 - TSRVToolBar 195
 - TSRVToolWindow 202
- OnEnterButton 196, 202
 - TSRVToolBar 196
 - TSRVToolWindow 202
- OnExecuted 297, 311
 - TsrvActionCustomLayout 297
 - TsrvActionThumbnails 311
- OnFullRedraw 108
 - TSRichViewEdit 108
- OnGetPagePos 108
 - TSRichViewEdit 108
- OnHMenuClickButton 109
 - TSRichViewEdit 109
- OnHMenuEnterButton 109
 - TSRichViewEdit 109
- OnHScrolled 109
 - TSRichViewEdit 109
- OnHTMLSaveImage 109
 - TSRichViewEdit 109
- OnImportPicture 110
 - TSRichViewEdit 110
- OnItemAction 110
 - TSRichViewEdit 110
- OnItemHint 110
 - TSRichViewEdit 110
- OnItemResize 110
 - TSRichViewEdit 110
- OnItemTextEdit 110
 - TSRichViewEdit 110
- OnJump 110
 - TSRichViewEdit 110
- OnLeave 196, 203
 - TSRVToolBar 196
 - TSRVToolWindow 203
- OnLoadCustomFormat 111
 - TSRichViewEdit 111
- OnLoadDocument 111
 - TSRichViewEdit 111
- OnMarginsChanged 111
 - TSRichViewEdit 111
- OnMessageControl 112
 - TSRichViewEdit 112
- OnMoveTab 260
 - TSRVTabSet 260
- OnNewDocument 112
 - TSRichViewEdit 112
- OnOleDragEnter 113
 - TSRichViewEdit 113
- OnOleDragLeave 113
 - TSRichViewEdit 113
- OnOleDragOver 113
 - TSRichViewEdit 113
- OnOleDrop 113
 - TSRichViewEdit 113
- OnPageCountChanged 113

OnPageCountChanged 113	OnSaveCustomFormat 119
TSRichViewEdit 113	TDBSRichViewEdit 119
OnPageFormatChanged 114	OnSaveDocXExtra 119
TSRichViewEdit 114	TSRichViewEdit 119
OnPageScrolled 114	OnSaveHTMLExtra 119
TSRichViewEdit 114	TSRichViewEdit 119
OnPaint 114	OnSaveImage2 119
TSRichViewEdit 114	TSRichViewEdit 119
OnPaintComponent 114	OnSaveItemToFile 120
TSRichViewEdit 114	TSRichViewEdit 120
OnPaintPage 115	OnSaveRTFExtra 120
TSRichViewEdit 115	TSRichViewEdit 120
OnParaStyleConversion 116	OnSelect 120
TSRichViewEdit 116	TSRichViewEdit 120
OnPaste 116	OnSelectEvent 203
TSRichViewEdit 116	TSRVToolWindow 203
OnPrinting 116	OnSendingToPrinter 221
TSRichViewEdit 116	TSRVPrint 221
OnProgress 116	OnSmartPopupClick 120
TSRichViewEdit 116	TSRichViewEdit 120
OnReadField 116	OnSpellingCheck 120
TSRichViewEdit 116	TSRichViewEdit 120
OnReadHyperlink 117	OnStyleConversion 120
TSRichViewEdit 117	TSRichViewEdit 120
OnReadMergeField 117	OnTableIconClick 121
TSRichViewEdit 117	TSRichViewEdit 121
OnResize 117	OnTextFound 121
TSRichViewEdit 117	TSRichViewEdit 121
OnResizeDoc 117	OnUpdateData 221
TSRichViewEdit 117	TSRVPrint 221
OnResizing 117	OnURLNeeded 121
TSRichViewEdit 117	TSRichViewEdit 121
OnRVDbClick 117	OnVMenuClickButton 121
TSRichViewEdit 117	TSRichViewEdit 121
OnRVFControlNeeded 117	OnVMenuEnterButton 121
TSRichViewEdit 117	TSRichViewEdit 121
OnRVFImageListNeeded 118	OnVScrolled 122
TSRichViewEdit 118	TSRichViewEdit 122
OnRVFPictureNeeded 118	OnWriteHyperlink 122
TSRichViewEdit 118	TSRichViewEdit 122
OnRVMouseDown 118	OnWriteObjectProperties 122
TSRichViewEdit 118	TSRichViewEdit 122
OnRVMouseMove 118	OnZoomChanged 122
TSRichViewEdit 118	TSRichViewEdit 122
OnRVMouseUp 118	OnZoomViewChanged 122
TSRichViewEdit 118	TSRichViewEdit 122
OnRVRightClick 119	OppositeTabPosition 255
TSRichViewEdit 119	TSRVTabSet 255
OnSaveComponentToFile 119	OptimalOrientation 218
TSRichViewEdit 119	TSRVPrint 218

Orientation 138, 211
 TSRVPageProperty 138
 TSRVPrint 211
 OutZoomRect 92
 TSRRichViewEdit 92

- P -

PageBorderColor 178, 225
 TSRVPageScroll 178
 TSRVPreview 225
 PageBorderStyle 225
 TSRVPreview 225
 PageBorderWidth 225
 TSRVPreview 225
 PageBreakHeight 178
 TSRVPageScroll 178
 PageBreakWidth 179
 TSRVPageScroll 179
 PageColCount 212
 TSRVPrint 212
 PageColor 152, 305
 TsrvActionLayoutRead 305
 TSRVReadModeProperty 152
 PageCount 57
 TSRRichViewEdit 57
 PageFlipEffect 152, 305, 309
 TsrvActionLayoutRead 305
 TsrvActionLayoutSideToSide 309
 TSRVReadModeProperty 152
 PageFormat 138, 212, 289
 TsrvActionPageFormat 289
 TSRVPageProperty 138
 TSRVPrint 212
 PageFormats1 291
 TsrvActionPageSetup 291
 PageFormats2 291
 TsrvActionPageSetup 291
 PageFormats3 291
 TsrvActionPageSetup 291
 PageFormats4 291
 TsrvActionPageSetup 291
 PageFormats5 291
 TsrvActionPageSetup 291
 PageFormats6 291
 TsrvActionPageSetup 291
 PageHeight 139, 161, 179, 289
 TsrvActionPageFormat 289
 TSRVPageProperty 139
 TSRVPageScroll 179
 TSRVViewProperty 161
 PageHeight100Pix 57
 TSRRichViewEdit 57
 PageHeightPix 57, 212
 TSRRichViewEdit 57
 TSRVPrint 212
 PageHSpacing 146
 TSRVPagePositionProperty 146
 PageNo 225
 TSRVPreview 225
 PageNoFirst 139
 TSRVPageProperty 139
 PageNoFont 139, 179
 TSRVPageProperty 139
 TSRVPageScroll 179
 PageNoFromNumber 139
 TSRVPageProperty 139
 PageNoHAlign 139
 TSRVPageProperty 139
 PageNoMouseDown 57
 TSRRichViewEdit 57
 PageNoMouseMove 57
 TSRRichViewEdit 57
 PageNoVAlign 140
 TSRVPageProperty 140
 PageNoVisible 140, 180
 TSRVPageProperty 140
 TSRVPageScroll 180
 PagePosProperty 58
 TSRRichViewEdit 58
 PageProperty 58
 TSRRichViewEdit 58
 PageProportions 180
 TSRVPageScroll 180
 PageRowCount 213
 TSRVPrint 213
 PageScroll 311
 TsrvActionThumbnails 311
 PageSize 245
 TSRVScrollBar 245
 PageViewMode 140
 TSRVPageProperty 140
 PageVSpacing 146
 TSRVPagePositionProperty 146
 PageWidth 141, 161, 180, 289
 TsrvActionPageFormat 289
 TSRVPageProperty 141
 TSRVPageScroll 180
 TSRVViewProperty 161
 PageWidth100Pix 58

PageWidth100Pix 58
 TSRichViewEdit 58
 PageWidthPix 58, 213
 TSRichViewEdit 58
 TSRVPrint 213
 PaperCodeToPageFormat 218
 TSRVPrint 218
 Paste 92
 TSRichViewEdit 92
 PenFrame 130, 194, 201
 TSRVCustomPropertyTB 130
 TSRVToolBar 194
 TSRVToolWindow 201
 PercentMiddle 125
 TSRVBackgroundProperty 125
 Picture 125
 TSRVBackgroundProperty 125
 PicturePosition 126
 TSRVBackgroundProperty 126
 Position 245
 TSRVScrollBar 245
 Prev 184, 226
 TSRVPageScroll 184
 TSRVPreview 226
 Print 218
 TSRVPrint 218
 PrintableAreaPen 141
 TSRVPageProperty 141
 PrintAll 92
 TSRichViewEdit 92
 PrintCurrent 93
 TSRichViewEdit 93
 PrintFrame 219
 TSRVPrint 219
 PrintFrames 219
 TSRVPrint 219
 PrintFramesEx 220
 TSRVPrint 220
 PrintMode 213
 TSRVPrint 213
 PrintPages 220
 TSRVPrint 220
 PrintRange 93
 TSRichViewEdit 93
 PriorCurCheckpoint 94
 TSRichViewEdit 94
 PriorCurHeading 94
 TSRichViewEdit 94
 PriorCurHyperlink 94
 TSRichViewEdit 94

PriorCurlItem 94
 TSRichViewEdit 94
 PriorCurPage 95
 TSRichViewEdit 95
 ProcessControls 95
 TSRichViewEdit 95

- R -

RangeSearch 58
 TSRichViewEdit 58
 RangeSearchFirstY 59
 TSRichViewEdit 59
 RangeSearchLastY 59
 TSRichViewEdit 59
 ReadModeProperty 59
 TSRichViewEdit 59
 ReadOnly 59
 TSRichViewEdit 59
 RealHeight 214
 TSRVPrint 214
 RealWidth 215
 TSRVPrint 215
 Reformat 82
 TSRichViewEdit 82
 RefStyleTemplateName 323
 TsrvActionInsertSidenote 323
 RestoreCanvasZoom 100
 TSRichViewEdit 100
 ReturnToNote 95
 TSRichViewEdit 95
 RichViewEdit 59, 173
 TDBSRichViewEdit 173
 TSRichViewEdit 59
 RightMargin 141
 TSRVPageProperty 141
 RightMargin100Pix 60
 TSRichViewEdit 60
 RightMarginPix 60
 TSRichViewEdit 60
 RTFOptions 60
 TSRichViewEdit 60
 RTFReadProperties 60
 TSRichViewEdit 60
 RVBackgroundBitmap 61
 TSRichViewEdit 61
 RVBackgroundStyle 61
 TSRichViewEdit 61
 RVColor 61
 TSRichViewEdit 61

RVEditorOptions 61
 TSRichViewEdit 61
 RVFooter 61
 TSRichViewEdit 61
 RVFOptions 62
 TSRichViewEdit 62
 RVFParaStylesReadMode 63
 TSRichViewEdit 63
 RVFTextStylesReadMode 63
 TSRichViewEdit 63
 RVHeader 63
 TSRichViewEdit 63
 RVNote 64
 TSRichViewEdit 64
 RVOptions 64
 TSRichViewEdit 64

- S -

ScaleImage 215
 TSRVPrint 215
 ScaleImagesForDPI 130, 194, 201
 TSRVCustomPropertyTB 130
 TSRVToolBar 194
 TSRVToolWindow 201
 ScaleRichView
 actions 281
 Document parts 36
 New after v11.0 20
 New after v12.0 20
 New in v10.0 22
 New in v11.0 21
 New in v2.0 34
 New in v3.0 32
 New in v4.0 30
 New in v5.0 28
 New in v6.0 27
 New in v7.0 26
 New in v8.0 25
 New in v9.0 23
 Notes and text boxes 38
 ScaleRichView overview 18, 36
 ScrollBarControlStyle 69
 TSRichViewEdit 69
 ScrollButtonWidth 255
 TSRVTabSet 255
 ScrollCaretToCenter 95
 TSRichViewEdit 95
 Scrolled 188
 TScIRVRuler 188
 ScrolledPage 65
 TSRichViewEdit 65
 ScrollerMenu 255
 TSRVTabSet 255
 ScrollerPosition 255
 TSRVTabSet 255
 ScrollingDelay 246, 256
 TSRVScrollBar 246
 TSRVTabSet 256
 ScrollToCaret 95
 TSRichViewEdit 95
 ScrollToItem 96
 TSRichViewEdit 96
 ScrollToPage 184
 TSRVPageScroll 184
 ScrollToTab 258
 TSRVTabSet 258
 ScrollType 180
 TSRVPageScroll 180
 SelectAll 96
 TSRichViewEdit 96
 SelectColor 181
 TSRVPageScroll 181
 SetFloatPropertyEd 96
 TSRichViewEdit 96
 SetIntPropertyEd 96
 TSRichViewEdit 96
 SetMargin 97
 TSRichViewEdit 97
 SetMarginMM 97
 TSRichViewEdit 97
 SetMarginUnit 98
 TSRichViewEdit 98
 SetPageProperties 188
 TScIRVRuler 188
 SetRVMargins 98
 TSRichViewEdit 98
 ShadowColor 181
 TSRVPageScroll 181
 ShadowWidth 181
 TSRVPageScroll 181
 ShowHint 274
 TSRVToolButton 274
 ShowScrollHint 162
 TSRVViewProperty 162
 SkinFontIndex 264
 TSRVTabItem 264
 SkinIndex 229
 TSRVSkinManager 229
 SkinManager 65, 246, 256

SkinManager 65, 246, 256
 TSRichViewEdit 65
 TSRVScrollBar 246
 TSRVTabSet 256
 Skins 229
 TSRVSkinManager 229
 SkinSchemeIndex 246, 256, 265
 TSRVScrollBar 246
 TSRVTabItem 265
 TSRVTabSet 256
 SmallChange 246
 TSRVScrollBar 246
 SmartPopupProperties 65
 TSRichViewEdit 65
 SmartPopupVisible 65
 TSRichViewEdit 65
 SmoothScroll 181
 TSRVPageScroll 181
 SmoothThumbnails 182
 TSRVPageScroll 182
 Spacer 130, 195, 201
 TSRVCustomPropertyTB 130
 TSRVToolBar 195
 TSRVToolWindow 201
 Spacings 256
 TSRVTabSet 256
 SRichViewEdit 182, 187, 216
 TScIRVRuler 187
 TSRVPageScroll 182
 TSRVPrint 216
 SRVA_GetPC 282
 SRVA_GetS 282
 SRVA_GetUnitsName 282
 SRVA_LocalizeSRichViewEdit 282
 SRVA_LocalizeToolWindow 282
 SRVControlStyle 247, 257
 TSRVScrollBar 247
 TSRVTabSet 257
 SRVPrint 226
 TSRVPreview 226
 SRVToolBar 131
 TSRVCustomPropertyTB 131
 StartEditing 98
 TSRichViewEdit 98
 StartEditNote 99
 TSRichViewEdit 99
 StartFrom 132
 TSRVLineNumberProperty 132
 Step 132
 TSRVLineNumberProperty 132

StyleTemplateInsertMode 65
 TSRichViewEdit 65
 StyleTemplateName 317
 TsrvActionCustomInsertSidenote 317
 Subdocuments 66
 TSRichViewEdit 66

- T -

TabHeight 257
 TSRVTabSet 257
 TabIndex 257
 TSRVTabSet 257
 TableIconDelay 162
 TSRVViewProperty 162
 TableIconPopupMenu 162
 TSRVViewProperty 162
 TablesAsLines 132
 TSRVLineNumberProperty 132
 TabNavigation 67
 TSRichViewEdit 67
 Tabs 258
 TSRVTabSet 258
 Tag 265, 274
 TSRVTabItem 265
 TSRVToolButton 274
 TCustomSRVEdit 326
 TCustomSRVGraphicControl 328
 TCustomSRVPanel 329
 TDBSRichViewEdit 168
 TDBSRichViewEdit.AutoDisplay 172
 TDBSRichViewEdit.DataField 172
 TDBSRichViewEdit.DataSource 172
 TDBSRichViewEdit.ExternalRV 173
 TDBSRichViewEdit.FieldFormat 173
 TDBSRichViewEdit.IgnoreEscape 173
 TDBSRichViewEdit.RichViewEdit 173
 Text 265
 TSRVTabItem 265
 TextAlignH 258
 TSRVTabSet 258
 TextAlignV 258
 TSRVTabSet 258
 TextEngine 67
 TSRichViewEdit 67
 Texts 162
 TSRVViewProperty 162
 TFormHintsClose 348
 Ticks 132
 TSRVLineNumberProperty 132

- Title 293, 294
 - TsrvActionPrint 293
 - TsrvActionQuickPrint 294
- TitlePage 141
 - TSRVPageProperty 141
- TLRAlignment 374
- TopMargin 142
 - TSRVPageProperty 142
- TopMargin100Pix 67
 - TSRichViewEdit 67
- TopMarginPix 67
 - TSRichViewEdit 67
- TotalFrameHPix 216
 - TSRVPrint 216
- TotalFrameWPix 216
 - TSRVPrint 216
- TPSAlignText 175
- TPSDrawPageBorderEvent 185
- TPSDrawPageNumberEvent 185
- TPSDrawSelectedPageEvent 186
- TPSScrollType 180
- TransparentColor 126
 - TSRVBackgroundProperty 126
- TScIRVRuler 186
- TScIRVRuler.CheckMargins 187
- TScIRVRuler.GetPageProperties 187
- TScIRVRuler.OnAfterMarginChanged 188
- TScIRVRuler.Scrolled 188
- TScIRVRuler.SetPageProperties 188
- TScIRVRuler.SRichViewEdit 187
- TSRichViewEdit 40
 - classes of properties 123
- TSRichViewEdit.AcceptDragDropFormats 47
- TSRichViewEdit.AcceptPasteFormats 47
- TSRichViewEdit.ActiveEditor 47
- TSRichViewEdit.AnimationMode 47
- TSRichViewEdit.BackgroundProperty 48
- TSRichViewEdit.BeginComponentsUpdate 72
- TSRichViewEdit.BiDiMode 48
- TSRichViewEdit.BottomMargin100Pix 48
- TSRichViewEdit.BottomMarginPix 48
- TSRichViewEdit.CalculateAllPagePositions 72
- TSRichViewEdit.CalculatePageCount 72
- TSRichViewEdit.CalculatePagePosition 72
- TSRichViewEdit.CanScroll 48
- TSRichViewEdit.CanUpdate 48
- TSRichViewEdit.CanUpdateMargin 49
- TSRichViewEdit.CanUpdatePosition 48
- TSRichViewEdit.CaretBlinkTime 49
- TSRichViewEdit.CaretPos 49
- TSRichViewEdit.CaretVisible 49
- TSRichViewEdit.Clear 73
- TSRichViewEdit.ConvertRVtoSRV 73
- TSRichViewEdit.ConvertSRVtoRV 73
- TSRichViewEdit.ConvertToDifferentUnits 74
- TSRichViewEdit.ConvertToEMU 74
- TSRichViewEdit.ConvertToPixels 74
- TSRichViewEdit.ConvertToTwips 74
- TSRichViewEdit.Copy 75
- TSRichViewEdit.CPEventKind 49
- TSRichViewEdit.Create 75
- TSRichViewEdit.CurControl 50
- TSRichViewEdit.CurrentNote 50
- TSRichViewEdit.CurrentNoteParentEditor 50
- TSRichViewEdit.CurrentPage 50
- TSRichViewEdit.Cut 75
- TSRichViewEdit.DeletePage 75
- TSRichViewEdit.Destroy 76
- TSRichViewEdit.Document 51
- TSRichViewEdit.DrawMetafile 76
- TSRichViewEdit.DrawPage 76
- TSRichViewEdit.EndComponentsUpdate 77
- TSRichViewEdit.ExternalRVStyle 51
- TSRichViewEdit.ExternalRVStyleFooter 51
- TSRichViewEdit.ExternalRVStyleHeader 51
- TSRichViewEdit.FindNextCheckpoint 77
- TSRichViewEdit.FindNextHeading 78
- TSRichViewEdit.FindNextHyperlink 78
- TSRichViewEdit.FindNextItem 79
- TSRichViewEdit.FindPriorCheckpoint 80
- TSRichViewEdit.FindPriorHeading 80
- TSRichViewEdit.FindPriorHyperlink 81
- TSRichViewEdit.FindPriorItem 81
- TSRichViewEdit.FirstCurPage 82
- TSRichViewEdit.FirstPageNo 51
- TSRichViewEdit.FixMarginsMode 52
- TSRichViewEdit.Format 82
- TSRichViewEdit.GetCaretPosInUnits 83
- TSRichViewEdit.GetCurrentLineCol 83
- TSRichViewEdit.GetFirstItemVisibleRV 83
- TSRichViewEdit.GetFirstVisiblePage 83
- TSRichViewEdit.GetItemAt 84
- TSRichViewEdit.GetItemBounds 84
- TSRichViewEdit.GetItemBounds100 84
- TSRichViewEdit.GetItemCoords100 85
- TSRichViewEdit.GetItemPages 86
- TSRichViewEdit.GetLastItemVisibleRV 86
- TSRichViewEdit.GetLastVisiblePage 86

TSRichViewEdit.GetLeftMargin100Pix	54	TSRichViewEdit.OnCaretMoving	104
TSRichViewEdit.GetLeftMarginPix	54	TSRichViewEdit.OnChange	104
TSRichViewEdit.GetPageAt	86	TSRichViewEdit.OnChangeActiveEditor	104
TSRichViewEdit.GetPageClientRect	87	TSRichViewEdit.OnChangeCurrentControl	104
TSRichViewEdit.GetPageLastItemNo	87	TSRichViewEdit.OnChangeViewModeAfter	104
TSRichViewEdit.GetPageNo	87	TSRichViewEdit.OnChangeViewModeBefore	105
TSRichViewEdit.GetPageStartItemNo	88	TSRichViewEdit.OnChanging	105
TSRichViewEdit.GetPageStartTableItem	88	TSRichViewEdit.OnCheckControl	105
TSRichViewEdit.GetPageStartTableRow	88	TSRichViewEdit.OnCheckpointVisible	105
TSRichViewEdit.GetRightMargin100Pix	60	TSRichViewEdit.OnCheckStickingItems	106
TSRichViewEdit.GetRightMarginPix	60	TSRichViewEdit.OnClickPage	106
TSRichViewEdit.GetTableconItem	89	TSRichViewEdit.OnContextPopup	106
TSRichViewEdit.GetTableconRVData	89	TSRichViewEdit.OnControlAction	106
TSRichViewEdit.HMaxScrollPos	52	TSRichViewEdit.OnCopy	106
TSRichViewEdit.HScrollBar	52	TSRichViewEdit.OnCurParaStyleChanged	106
TSRichViewEdit.HScrollBarSchemeIndex	53	TSRichViewEdit.OnCurrentPageChange	107
TSRichViewEdit.HScrollPos	53	TSRichViewEdit.OnCurTextStyleChanged	107
TSRichViewEdit.HTMLReadProperties	53	TSRichViewEdit.OnDrawHyperlink	107
TSRichViewEdit.HTMLSaveProperties	53	TSRichViewEdit.OnDropFile	107
TSRichViewEdit.InsertPageBreak	89	TSRichViewEdit.OnDropFiles	107
TSRichViewEdit.InZoomRect	89	TSRichViewEdit.OnFullRedraw	108
TSRichViewEdit.LastCurPage	90	TSRichViewEdit.OnGetPagePos	108
TSRichViewEdit.LastPageNo	53	TSRichViewEdit.OnHMenuClickButton	109
TSRichViewEdit.LeftMargin100Pix	54	TSRichViewEdit.OnHMenuEnterButton	109
TSRichViewEdit.LeftMarginPix	54	TSRichViewEdit.OnHScrolled	109
TSRichViewEdit.LineNumberProperty	54	TSRichViewEdit.OnHTMLSaveImage	109
TSRichViewEdit.LoadRVF	90	TSRichViewEdit.OnImportPicture	110
TSRichViewEdit.LoadRVFFromStream	90	TSRichViewEdit.OnItemAction	110
TSRichViewEdit.MarkdownProperties	54	TSRichViewEdit.OnItemHint	110
TSRichViewEdit.MaxPrintedItemNo	54	TSRichViewEdit.OnItemResize	110
TSRichViewEdit.MaxPrintedOffsInItem	54	TSRichViewEdit.OnItemTextEdit	110
TSRichViewEdit.MenuHButtons	55	TSRichViewEdit.OnJump	110
TSRichViewEdit.MenuHorizontal	55	TSRichViewEdit.OnLoadCustomFormat	111
TSRichViewEdit.MenuVButtons	55	TSRichViewEdit.OnLoadDocument	111
TSRichViewEdit.MenuVertical	56	TSRichViewEdit.OnMarginsChanged	111
TSRichViewEdit.MinPrintedItemNo	56	TSRichViewEdit.OnMessageControl	112
TSRichViewEdit.MinPrintedOffsInItem	56	TSRichViewEdit.OnNewDocument	112
TSRichViewEdit.NextCurCheckpoint	90	TSRichViewEdit.OnOleDragEnter	113
TSRichViewEdit.NextCurHeading	90	TSRichViewEdit.OnOleDragLeave	113
TSRichViewEdit.NextCurHyperlink	91	TSRichViewEdit.OnOleDragOver	113
TSRichViewEdit.NextCurItem	91	TSRichViewEdit.OnOleDrop	113
TSRichViewEdit.NextCurPage	91	TSRichViewEdit.OnPageCountChanged	113
TSRichViewEdit.OffsetX	56	TSRichViewEdit.OnPageFormatChanged	114
TSRichViewEdit.OffsetY	57	TSRichViewEdit.OnPageScrolled	114
TSRichViewEdit.OnAfterOleDrop	103	TSRichViewEdit.OnPaint	114
TSRichViewEdit.OnAssignImageFileName	103	TSRichViewEdit.OnPaintComponent	114
TSRichViewEdit.OnBeforeOleDrop	103	TSRichViewEdit.OnPaintPage	115
TSRichViewEdit.OnCaretGetOut	104	TSRichViewEdit.OnParaStyleConversion	116
TSRichViewEdit.OnCaretMove	104	TSRichViewEdit.OnPaste	116

- TSRichViewEdit.OnProgress 116
- TSRichViewEdit.OnReadField 116
- TSRichViewEdit.OnReadHyperlink 117
- TSRichViewEdit.OnReadMergeField 117
- TSRichViewEdit.OnResize 117
- TSRichViewEdit.OnResizeDoc 117
- TSRichViewEdit.OnResizing 117
- TSRichViewEdit.OnRVDbIClick 117
- TSRichViewEdit.OnRVFControlNeeded 117
- TSRichViewEdit.OnRVFImageListNeeded 118
- TSRichViewEdit.OnRVFPictureNeeded 118
- TSRichViewEdit.OnRVMouseDown 118
- TSRichViewEdit.OnRVMouseMove 118
- TSRichViewEdit.OnRVMouseUp 118
- TSRichViewEdit.OnRVRightClick 119
- TSRichViewEdit.OnSaveComponentToFile 119
- TSRichViewEdit.OnSaveCustomFormat 119
- TSRichViewEdit.OnSaveDocXExtra 119
- TSRichViewEdit.OnSaveHTMLExtra 119
- TSRichViewEdit.OnSaveImage2 119
- TSRichViewEdit.OnSaveItemToFile 120
- TSRichViewEdit.OnSaveRTFExtra 120
- TSRichViewEdit.OnSelect 120
- TSRichViewEdit.OnSmartPopupClick 120
- TSRichViewEdit.OnSpellingCheck 120
- TSRichViewEdit.OnStyleConversion 120
- TSRichViewEdit.OnTableIconClick 121
- TSRichViewEdit.OnTextFound 121
- TSRichViewEdit.OnURLNeeded 121
- TSRichViewEdit.OnVMenuClickButton 121
- TSRichViewEdit.OnVMenuEnterButton 121
- TSRichViewEdit.OnVScrolled 122
- TSRichViewEdit.OnWriteHyperlink 122
- TSRichViewEdit.OnWriteObjectProperties 122
- TSRichViewEdit.OnZoomChanged 122
- TSRichViewEdit.OnZoomViewChanged 122
- TSRichViewEdit.OutZoomRect 92
- TSRichViewEdit.PageCount 57
- TSRichViewEdit.PageHeight100Pix 57
- TSRichViewEdit.PageHeightPix 57
- TSRichViewEdit.PageNoMouseDown 57
- TSRichViewEdit.PageNoMouseMove 57
- TSRichViewEdit.PagePosProperty 58
- TSRichViewEdit.PageProperty 58
- TSRichViewEdit.PageWidth100Pix 58
- TSRichViewEdit.PageWidthPix 58
- TSRichViewEdit.Paste 92
- TSRichViewEdit.PrintAll 92
- TSRichViewEdit.PrintCurrent 93
- TSRichViewEdit.Printing 116
- TSRichViewEdit.PrintRange 93
- TSRichViewEdit.PriorCurCheckpoint 94
- TSRichViewEdit.PriorCurHeading 94
- TSRichViewEdit.PriorCurHyperlink 94
- TSRichViewEdit.PriorCurlItem 94
- TSRichViewEdit.PriorCurPage 95
- TSRichViewEdit.ProcessControls 95
- TSRichViewEdit.RangeSearch 58
- TSRichViewEdit.RangeSearchFirstY 59
- TSRichViewEdit.RangeSearchLastY 59
- TSRichViewEdit.ReadModeProperty 59
- TSRichViewEdit.ReadOnly 59
- TSRichViewEdit.Reformat 82
- TSRichViewEdit.RestoreCanvasZoom 100
- TSRichViewEdit.ReturnToNote 95
- TSRichViewEdit.RichViewEdit 59
- TSRichViewEdit.RightMargin100Pix 60
- TSRichViewEdit.RightMarginPix 60
- TSRichViewEdit.RTFOptions 60
- TSRichViewEdit.RTFReadProperties 60
- TSRichViewEdit.RVBackgroundBitmap 61
- TSRichViewEdit.RVBackgroundStyle 61
- TSRichViewEdit.RVColor 61
- TSRichViewEdit.RVEditorOptions 61
- TSRichViewEdit.RVFooter 61
- TSRichViewEdit.RVFOptions 62
- TSRichViewEdit.RVFPaStylesReadMode 63
- TSRichViewEdit.RVFTextStylesReadMode 63
- TSRichViewEdit.RVHeader 63
- TSRichViewEdit.RVNote 64
- TSRichViewEdit.RVOptions 64
- TSRichViewEdit.ScrollBarControlStyle 69
- TSRichViewEdit.ScrollCaretToCenter 95
- TSRichViewEdit.ScrolledPage 65
- TSRichViewEdit.ScrollToCaret 95
- TSRichViewEdit.ScrollToItem 96
- TSRichViewEdit.SelectAll 96
- TSRichViewEdit.SetFloatPropertyEd 96
- TSRichViewEdit.SetIntPropertyEd 96
- TSRichViewEdit.SetMargin 97
- TSRichViewEdit.SetMarginMM 97
- TSRichViewEdit.SetMarginUnit 98
- TSRichViewEdit.SetRVMargins 98
- TSRichViewEdit.SkinManager 65
- TSRichViewEdit.SmartPopupProperties 65
- TSRichViewEdit.SmartPopupVisible 65
- TSRichViewEdit.StartEditing 98
- TSRichViewEdit.StartEditNote 99

TSRichViewEdit.StyleTemplateInsertMode	65	TsrvActionLayoutPrint.FooterVisible	302
TSRichViewEdit.Subdocuments	66	TsrvActionLayoutPrint.HeaderVisible	302
TSRichViewEdit.TabNavigation	67	TsrvActionLayoutPrint.HPadding	303
TSRichViewEdit.TextEngine	67	TsrvActionLayoutRead	303
TSRichViewEdit.TopMargin100Pix	67	TsrvActionLayoutRead.Color	305
TSRichViewEdit.TopMarginPix	67	TsrvActionLayoutRead.HidePageBackgroundImage	305
TSRichViewEdit.UnitsPerInchH	99	TsrvActionLayoutRead.HPadding	305
TSRichViewEdit.UnitsPerInchV	100	TsrvActionLayoutRead.PageColor	305
TSRichViewEdit.UnitsProgram	68	TsrvActionLayoutRead.PageFlipEffect	305
TSRichViewEdit.UpdateBuffer	100	TsrvActionLayoutSideToSide	306
TSRichViewEdit.UpdateBufferAt	100	TsrvActionLayoutSideToSide.Color	308
TSRichViewEdit.UseDrawHyperlinksEvent	68	TsrvActionLayoutSideToSide.FooterVisible	308
TSRichViewEdit.UseStyleTemplates	68	TsrvActionLayoutSideToSide.HeaderVisible	308
TSRichViewEdit.Version	68	TsrvActionLayoutSideToSide.HPadding	308
TSRichViewEdit.ViewProperty	68	TsrvActionLayoutSideToSide.NoPageSpacing	308
TSRichViewEdit.VMaxScrollPos	68	TsrvActionLayoutSideToSide.PageFlipEffect	309
TSRichViewEdit.VScrollBar	69	TsrvActionLayoutWeb	309
TSRichViewEdit.VScrollBarSchemeIndex	69	TsrvActionLineNumbers	286
TSRichViewEdit.VScrollPos	69	TsrvActionOrientationLandscape	286
TSRichViewEdit.ZoomCanvas	100	TsrvActionOrientationPortrait	287
TSRichViewEdit.ZoomRectIn	100	TsrvActionPageFormat	288
TSRichViewEdit.ZoomRectOut	100	TsrvActionPageFormat.PageFormat	289
TSRichViewEdit.ZoomToFullPages	101	TsrvActionPageFormat.PageHeight	289
TsrvAction	283	TsrvActionPageFormat.PageWidth	289
TsrvAction.Control	284	TsrvActionPageFormat.Units	289
TsrvActionCustomInsertSidenote	316	TsrvActionPageFormat.Units	289
TsrvActionCustomInsertSidenote.BoxPosition	317	TsrvActionPageSetup	290
TsrvActionCustomInsertSidenote.BoxProperties	317	TsrvActionPageSetup.OnChange	292
TsrvActionCustomInsertSidenote.StyleTemplateName	317	TsrvActionPageSetup.PageFormats1	291
TsrvActionCustomLayout	295	TsrvActionPageSetup.PageFormats2	291
TsrvActionCustomLayout.OnExecuted	297	TsrvActionPageSetup.PageFormats3	291
TsrvActionCustomOrientation	285	TsrvActionPageSetup.PageFormats4	291
TsrvActionCustomZoom	298	TsrvActionPageSetup.PageFormats5	291
TsrvActionEditFooter	314	TsrvActionPageSetup.PageFormats6	291
TsrvActionEditHeader	314	TsrvActionPreview	292
TsrvActionEditMain	315	TsrvActionPrint	292
TsrvActionEditNote	318	TsrvActionPrint.Title	293
TsrvActionInsertEndnote	318	TsrvActionQuickPrint	293
TsrvActionInsertFootnote	319	TsrvActionQuickPrint.Title	294
TsrvActionInsertNote	320	TsrvActionReturnToNote	324
TsrvActionInsertNote.Font	321	TsrvActionThumbnails	310
TsrvActionInsertSidenote	321	TsrvActionThumbnails.OnExecuted	311
TsrvActionInsertSidenote.RefStyleTemplateName	323	TsrvActionThumbnails.PageScroll	311
TsrvActionInsertTextBox	323	TsrvActionZoom	312
TsrvActionLayoutDraft	298	TsrvActionZoom.ZoomPercent	312
TsrvActionLayoutDraft.HPadding	300	TsrvActionZoomFullPage	313
TsrvActionLayoutPrint	300	TsrvActionZoomPageWidth	313
TsrvActionLayoutPrint.Color	302	TSRVALignPageH	144
		TSRVALignPageV	144
		TSRVALignText	270

- TSRVBackgroundProperty 123
- TSRVBackgroundProperty.ColorBegin 124
- TSRVBackgroundProperty.ColorEnd 124
- TSRVBackgroundProperty.ColorMiddle 124
- TSRVBackgroundProperty.FillType 124
- TSRVBackgroundProperty.GlobalPageBackgroundColor 125
- TSRVBackgroundProperty.PercentMiddle 125
- TSRVBackgroundProperty.Picture 125
- TSRVBackgroundProperty.PicturePosition 126
- TSRVBackgroundProperty.TransparentColor 126
- TSRVBackgroundProperty.Visible 126
- TSRVBorderPen 135
- TSRVBoxScheme 232
- TSRVBoxSchemeCollection 233
- TSRVButton 342
- TSRVButtonCollection 270
- TSRVCanvasControl 330
- TSRVCaretPen 135
- TSRVCBoxItem 374
- TSRVCBoxItemCollection 374
- TSRVChangeActiveEditorEvent 104
- TSRVChangeCurrentControlEvents 104
- TSRVChangeViewModeEvent 104, 105
- TSRVCheckBox 344
- TSRVCheckBoxState 344
- TSRVCheckControl 105
- TSRVCheckpointVisibleEvent 105
- TSRVClickButtonEvent 271
- TSRVCloseButton 261
- TSRVCloseButton.AllowCloseLastTab 262
- TSRVCloseButton.Height 262
- TSRVCloseButton.HorizontalAlign 262
- TSRVCloseButton.IndentX 262
- TSRVCloseButton.IndentY 262
- TSRVCloseButton.VerticalAlign 263
- TSRVCloseButton.Visibility 263
- TSRVCloseButton.Visible 263
- TSRVCloseButton.Width 263
- TSRVCloseTabEvent 259
- TSRVComboBox 345
- TSRVComponentEvent 114
- TSRVControlStyle 275
- TSRVCtrlPaintPage 330
- TsrvCustomActionThatNeedsDocProps 284
- TsrvCustomActionThatNeedsDocProps.ActionSave 285
- TSRVCustomComboBox 331
- TSRVCustomControl 334
- TSRVCustomFormatEvent 111, 119
- TSRVCustomListBox 335
- TsrvCustomPrintAction 295
- TSRVCustomPropertyTB 126
- TSRVCustomPropertyTB.AlignButton 128
- TSRVCustomPropertyTB.BorderWidth 128
- TSRVCustomPropertyTB.ButtonHeight 128
- TSRVCustomPropertyTB.Buttons 128
- TSRVCustomPropertyTB.ButtonWidth 128
- TSRVCustomPropertyTB.Color 129
- TSRVCustomPropertyTB.ColorDisable 129
- TSRVCustomPropertyTB.ColorDown 129
- TSRVCustomPropertyTB.ColorSelect 129
- TSRVCustomPropertyTB.ImageList 130
- TSRVCustomPropertyTB.Indent 130
- TSRVCustomPropertyTB.PenFrame 130
- TSRVCustomPropertyTB.ScaleImagesForDPI 130
- TSRVCustomPropertyTB.Spacer 130
- TSRVCustomPropertyTB.SRVToolBar 131
- TSRVDataDrawHyperlinkEvent 107
- TSRVDBCheckBox 368
- TSRVDBComboBox 369
- TSRVDBEdit 370
- TSRVDBListBox 371
- TSRVDBMemo 372
- TSRVDBText 373
- TSRVDrawButtonEvent 195
- TSRVDrawScrollBarEvent 267
- TSRVDrawState 268
- TSRVDrawStates 268
- TSRVDrawTabEvent 268
- TSRVDrawTabSetEvent 269
- TSRVEdit 348
- TSRVEditCharCase 348
- TSRVEditControl 338
- TSRVEditHintsEvent 348
- TSRVFillType 124
- TSRVFixMarginsMode 52
- TSRVFloatProperty 275
- TSRVGetPagePosEvent 108
- TSRVGraphicControl 338
- TSRVGroupBox 351
- TSRVHAlign 139
- TSRVHeaderFooterFont 157, 158, 160
- TSRVHeaderFooterPen 156, 158
- TSRVHeaderFooterType 66
- TSRVIconStyle 159
- TSRVImageScrollCollection 374
- TSRVImageScrollItem 378
- TSRVImageScrollKind 353

TSRVImagesScroll	353	TSRVPagePositionProperty.PageVSpacing	146
TSRVIntProperty	276	TSRVPagePositionProperty.VPadding	145
TSRVItemOption	374	TSRVPageProperty	133
TSRVItemOptions	374	TSRVPageProperty.AutoWidth	135
TSRVLabel	357	TSRVPageProperty.BorderPen	135
TSRVListBoxItem	374	TSRVPageProperty.BottomMargin	135
TSRVListBoxItemCollection	374	TSRVPageProperty.CaretPen	135
TSRVLineNumberFont	132	TSRVPageProperty.ConvertPageFormatToPageSize	142
TSRVLineNumberProperty	131	TSRVPageProperty.ConvertPageSizeToPageFormat	142
TSRVLineNumberProperty.DistanceFromText	131	TSRVPageProperty.FacingPages	136
TSRVLineNumberProperty.Font	132	TSRVPageProperty.FooterVisible	136
TSRVLineNumberProperty.StartFrom	132	TSRVPageProperty.FooterY	136
TSRVLineNumberProperty.Step	132	TSRVPageProperty.HeaderVisible	137
TSRVLineNumberProperty.TablesAsLines	132	TSRVPageProperty.HeaderY	137
TSRVLineNumberProperty.Ticks	132	TSRVPageProperty.LeftMargin	137
TSRVLineNumberProperty.Visible	133	TSRVPageProperty.MirrorMargins	138
TSRVListBox	358	TSRVPageProperty.Orientation	138
TSRVMainPen	160	TSRVPageProperty.PageFormat	138
TSRVMargins	253	TSRVPageProperty.PageHeight	139
TSRVMarginsChanged	111	TSRVPageProperty.PageNoFirst	139
TSRVMarginsPen	161	TSRVPageProperty.PageNoFont	139
TSRVMemo	361	TSRVPageProperty.PageNoFromNumber	139
TSRVMessageControl	112	TSRVPageProperty.PageNoHAlign	139
TSRVMouseEvent	276	TSRVPageProperty.PageNoVAlign	140
TSRVPageFlipEffect	152	TSRVPageProperty.PageNoVisible	140
TSRVPageFormat	277	TSRVPageProperty.PageViewMode	140
TSRVPageFormatAB	291	TSRVPageProperty.PageWidth	141
TSRVPageFormatANSIArch	291	TSRVPageProperty.PrintableAreaPen	141
TSRVPageFormatC	291	TSRVPageProperty.RightMargin	141
TSRVPageFormatCardTrad	291	TSRVPageProperty.TitlePage	141
TSRVPageFormatPAOtherJISB	291	TSRVPageProperty.TopMargin	142
TSRVPageFormats	291	TSRVPageProperty.UsePageOrders	142
TSRVPageFormatsAB	291	TSRVPageScroll	173
TSRVPageFormatsANSIArch	291	TSRVPageScroll.AlignText	175
TSRVPageFormatsC	291	TSRVPageScroll.AutoScrolled	176
TSRVPageFormatsCardTrad	291	TSRVPageScroll.AutoUpdate	176
TSRVPageFormatsPAOtherJISB	291	TSRVPageScroll.BackgroundColor	176
TSRVPageFormatsUSA	291	TSRVPageScroll.BeginUpdate	182
TSRVPageFormatUSA	291	TSRVPageScroll.CachedPagesCount	177
TSRVPageNoFont	139	TSRVPageScroll.CacheScrollLimit	177
TSRVPagePositionDistance	145	TSRVPageScroll.ChangeDelay	177
TSRVPagePositionDistances	145	TSRVPageScroll.Columns	177
TSRVPagePositionProperty	143	TSRVPageScroll.EndUpdate	182
TSRVPagePositionProperty.AlignPageH	144	TSRVPageScroll.First	183
TSRVPagePositionProperty.AlignPageV	144	TSRVPageScroll.GetPageAt	183
TSRVPagePositionProperty.HiddenDistances	145	TSRVPageScroll.Last	183
TSRVPagePositionProperty.HPadding	145	TSRVPageScroll.MarginHorizontal	178
TSRVPagePositionProperty.MaxPageColCount	146	TSRVPageScroll.MarginVertical	178
TSRVPagePositionProperty.PageHSpacing	146	TSRVPageScroll.Next	183

- TSRVPageScroll.OnDrawPageBorder 185
- TSRVPageScroll.OnDrawPageNumber 185
- TSRVPageScroll.OnDrawSelectedPage 186
- TSRVPageScroll.PageBorderColor 178
- TSRVPageScroll.PageBreakHeight 178
- TSRVPageScroll.PageBreakWidth 179
- TSRVPageScroll.PageHeight 179
- TSRVPageScroll.PageNoFont 179
- TSRVPageScroll.PageNoVisible 180
- TSRVPageScroll.PageProportions 180
- TSRVPageScroll.PageWidth 180
- TSRVPageScroll.Prev 184
- TSRVPageScroll.ScrollToPage 184
- TSRVPageScroll.ScrollType 180
- TSRVPageScroll.SelectColor 181
- TSRVPageScroll.ShadowColor 181
- TSRVPageScroll.ShadowWidth 181
- TSRVPageScroll.SmoothScroll 181
- TSRVPageScroll.SmoothThumbnails 182
- TSRVPageScroll.SRichViewEdit 182
- TSRVPageScroll.UpdateCache 184
- TSRVPaintBox 363
- TSRVPaintEvent 114
- TSRVPaintPage 115
- TSRVPanel 364
- TSRVAPen 141
- TSRVPicturePosition 126
- TSRVPOnPaint 269
- TSRVPreview 222
- TSRVPreview.BorderColor 224
- TSRVPreview.BorderWidth 224
- TSRVPreview.First 226
- TSRVPreview.FrameBorderColor 224
- TSRVPreview.FrameBorderWidth 224
- TSRVPreview.Last 226
- TSRVPreview.Next 226
- TSRVPreview.OnAfterPaint 227
- TSRVPreview.OnBeforePaint 227
- TSRVPreview.PageBorderColor 225
- TSRVPreview.PageBorderStyle 225
- TSRVPreview.PageBorderWidth 225
- TSRVPreview.PageNo 225
- TSRVPreview.Prev 226
- TSRVPreview.SRVPrint 226
- TSRVPrint 203
- TSRVPrint.AutoUpdate 206
- TSRVPrint.BeginUpdate 217
- TSRVPrint.Center 206
- TSRVPrint.CheckMargin 217
- TSRVPrint.ClipMargins 207
- TSRVPrint.EndUpdate 218
- TSRVPrint.FrameCountX 207
- TSRVPrint.FrameCountY 208
- TSRVPrint.FrameHeightPix 208
- TSRVPrint.FrameWidthPix 209
- TSRVPrint.IntegerCount 209
- TSRVPrint.MetafileCompatibility 210
- TSRVPrint.NoMetafiles 210
- TSRVPrint.OffXPix 210
- TSRVPrint.OffYPix 211
- TSRVPrint.OnSendingToPrinter 221
- TSRVPrint.OnUpdateData 221
- TSRVPrint.OptimalOrientation 218
- TSRVPrint.Orientation 211
- TSRVPrint.PageColCount 212
- TSRVPrint.PageFormat 212
- TSRVPrint.PageHeightPix 212
- TSRVPrint.PageRowCount 213
- TSRVPrint.PageWidthPix 213
- TSRVPrint.PaperCodeToPageFormat 218
- TSRVPrint.Print 218
- TSRVPrint.PrintFrame 219
- TSRVPrint.PrintFrames 219
- TSRVPrint.PrintFramesEx 220
- TSRVPrint.PrintMode 213
- TSRVPrint.PrintPages 220
- TSRVPrint.RealHeight 214
- TSRVPrint.RealWidth 215
- TSRVPrint.ScaleImage 215
- TSRVPrint.SRichViewEdit 216
- TSRVPrint.TotalFrameHPix 216
- TSRVPrint.TotalFrameWPix 216
- TSRVPrint.Update 221
- TSRVPrint.UsePhysicalOffsets 216
- TSRVPrintingEvent 221
- TSRVPrintingProgressEvent 116
- TSRVPrintMode 213
- TSRVPropertyTBH 147
- TSRVPropertyTBH.Align 148
- TSRVPropertyTBH.Visible 148
- TSRVPropertyTBV 148
- TSRVPropertyTBV.Align 149
- TSRVPropertyTBV.Visible 150
- TSRVRadioButton 366
- TSRVReadModeProperty 150
- TSRVReadModeProperty.Active 151
- TSRVReadModeProperty.HidePageBackgroundImage 151

TSRVReadModeProperty.MaxPageColCount	151	TSRVTabItem	263
TSRVReadModeProperty.MinPageColCount	151	TSRVTabItem.Enabled	264
TSRVReadModeProperty.PageColor	152	TSRVTabItem.ImageIndex	264
TSRVReadModeProperty.PageFlipEffect	152	TSRVTabItem.SkinFontIndex	264
TSRVRichViewEditType	98	TSRVTabItem.SkinSchemeIndex	265
TSRVScrollBar	242	TSRVTabItem.Tag	265
TSRVScrollBar.DecButtonWidth	244	TSRVTabItem.Text	265
TSRVScrollBar.Flat	244	TSRVTabItem.Visible	265
TSRVScrollBar.FocusBlinkDelay	244	TSRVTabMoveEvent	260
TSRVScrollBar.IncButtonWidth	245	TSRVTabSet	248
TSRVScrollBar.Kind	245	classes of properties	260
TSRVScrollBar.LargeChange	246	TSRVTabSet.AutoDarkMode	252
TSRVScrollBar.OnDrawArea	248	TSRVTabSet.AutoTabWidth	251
TSRVScrollBar.OnDrawBorder	248	TSRVTabSet.CanMoveTabs	252
TSRVScrollBar.OnDrawDecButton	248	TSRVTabSet.CloseButton	252
TSRVScrollBar.OnDrawIncButton	248	TSRVTabSet.DarkMode	252
TSRVScrollBar.OnDrawThumbTab	248	TSRVTabSet.FirstTabIndex	252
TSRVScrollBar.PageSize	245	TSRVTabSet.ImageList	252
TSRVScrollBar.Position	245	TSRVTabSet.Indent	253
TSRVScrollBar.ScrollingDelay	246	TSRVTabSet.Kind	253
TSRVScrollBar.SkinManager	246	TSRVTabSet.LastTabIndex	253
TSRVScrollBar.SkinSchemeIndex	246	TSRVTabSet.Margins	253
TSRVScrollBar.SmallChange	246	TSRVTabSet.MaxTabSize	254
TSRVScrollBar.SRVControlStyle	247	TSRVTabSet.MinTabSize	254
TSRVScrollBar.UseXPThemes	247	TSRVTabSet.MirrorText	254
TSRVScrollBarSkinScheme	237	TSRVTabSet.OnChange	259
TSRVScrollBarSkinSchemeCollection	239	TSRVTabSet.OnCloseTab	259
TSRVScrollerPosition	255, 353	TSRVTabSet.OnDrawBackground	259
TSRVSkin	230	TSRVTabSet.OnDrawBorder	259
TSRVSkin.BoxSchemes	231	TSRVTabSet.OnDrawCloseButton	259
TSRVSkin.Elements	231	TSRVTabSet.OnDrawDecButton	259
TSRVSkin.Fonts	231	TSRVTabSet.OnDrawIncButton	259
TSRVSkin.HorizontalScrollBarSchemes	231	TSRVTabSet.OnDrawTab	259
TSRVSkin.HorizontalTabSetSchemes	231	TSRVTabSet.OnMoveTab	260
TSRVSkin.Name	232	TSRVTabSet.OppositeTabPosition	255
TSRVSkin.VerticalScrollBarSchemes	232	TSRVTabSet.ScrollButtonWidth	255
TSRVSkin.VerticalTabSetSchemes	232	TSRVTabSet.ScrrollerMenu	255
TSRVSkinCollection	234	TSRVTabSet.ScrrollerPosition	255
TSRVSkinElement	236	TSRVTabSet.ScrollingDelay	256
TSRVSkinElementCollection	237	TSRVTabSet.ScrollToTab	258
TSRVSkinFont	234	TSRVTabSet.SkinManager	256
TSRVSkinFontCollection	235	TSRVTabSet.SkinSchemeIndex	256
TSRVSkinManager	227	TSRVTabSet.Spacings	256
Classes of properties	229	TSRVTabSet.SRVControlStyle	257
TSRVSkinManager.CurrentSkin	229	TSRVTabSet.TabHeight	257
TSRVSkinManager.SkinIndex	229	TSRVTabSet.TabIndex	257
TSRVSkinManager.Skins	229	TSRVTabSet.Tabs	258
TSRVSubDocuments	66	TSRVTabSet.TextAlignH	258
TSRVTabCollection	266	TSRVTabSet.TextAlignV	258

- TSRVTabSetKind 253
- TSRVTabSetSkinScheme 240
- TSRVTabSetSkinScheme.FntTab 240
- TSRVTabSetSkinScheme.ImageAddTabButton 240
- TSRVTabSetSkinScheme.ImageBackgroundBegin 240
- TSRVTabSetSkinScheme.ImageBackgroundCenter 240
- TSRVTabSetSkinScheme.ImageBackgroundEnd 240
- TSRVTabSetSkinScheme.ImageBorderBackground 240
- TSRVTabSetSkinScheme.ImageBorderBegin 240
- TSRVTabSetSkinScheme.ImageBorderEnd 240
- TSRVTabSetSkinScheme.ImageCloseButton 240
- TSRVTabSetSkinScheme.ImageDecButton 240
- TSRVTabSetSkinScheme.ImageIncButton 240
- TSRVTabSetSkinScheme.ImageTabBackground 240
- TSRVTabSetSkinScheme.ImageTabBegin 240
- TSRVTabSetSkinScheme.ImageTabEnd 240
- TSRVTabSetSkinScheme.Name 240
- TSRVTabSetSkinSchemeCollection 241
- TSRVTBAlignH 148
- TSRVTBAlignV 149
- TSRVTexts 162
- TSRVTHAlign 280
- TSRVToolBar 188
- TSRVToolBar.AlignButton 190
- TSRVToolBar.AlignText 190
- TSRVToolBar.AutoSize 191
- TSRVToolBar.BorderWidth 191
- TSRVToolBar.ButtonCol 191
- TSRVToolBar.ButtonHeight 191
- TSRVToolBar.Buttons 191
- TSRVToolBar.ButtonWidth 192
- TSRVToolBar.Color 192
- TSRVToolBar.ColorDisable 192
- TSRVToolBar.ColorDown 192
- TSRVToolBar.ColorSelect 193
- TSRVToolBar.HintCancel 193
- TSRVToolBar.HintFont 193
- TSRVToolBar.HintHeight 193
- TSRVToolBar.ImageList 194
- TSRVToolBar.Indent 194
- TSRVToolBar.OnClickButton 195
- TSRVToolBar.OnDrawButtonBackground 195
- TSRVToolBar.OnEnter 195
- TSRVToolBar.OnEnterButton 196
- TSRVToolBar.OnLeave 196
- TSRVToolBar.PenFrame 194
- TSRVToolBar.ScaleImagesForDPI 194
- TSRVToolBar.Spacer 195
- TSRVToolButton 271
- TSRVToolButton.AllowAllUp 272
- TSRVToolButton.Caption 272
- TSRVToolButton.Down 272
- TSRVToolButton.Enabled 272
- TSRVToolButton.GroupIndex 273
- TSRVToolButton.Hint 272
- TSRVToolButton.ImageIndexDisable 273
- TSRVToolButton.ImageIndexDown 273
- TSRVToolButton.ImageIndexMove 273
- TSRVToolButton.ImageIndexNormal 274
- TSRVToolButton.ShowHint 274
- TSRVToolButton.Tag 274
- TSRVToolButton.Visible 274
- TSRVToolWindow 196
- TSRVToolWindow.AlignText 197
- TSRVToolWindow.BorderWidth 198
- TSRVToolWindow.ButtonCol 198
- TSRVToolWindow.ButtonHeight 198
- TSRVToolWindow.Buttons 198
- TSRVToolWindow.ButtonWidth 198
- TSRVToolWindow.Color 199
- TSRVToolWindow.ColorDisable 199
- TSRVToolWindow.ColorDown 199
- TSRVToolWindow.ColorSelect 200
- TSRVToolWindow.Execute 202
- TSRVToolWindow.HintCancel 200
- TSRVToolWindow.HintFont 200
- TSRVToolWindow.HintHeight 200
- TSRVToolWindow.ImageList 201
- TSRVToolWindow.OnClickButton 202
- TSRVToolWindow.OnEnter 202
- TSRVToolWindow.OnEnterButton 202
- TSRVToolWindow.OnLeave 203
- TSRVToolWindow.OnSelectEvent 203
- TSRVToolWindow.PenFrame 201
- TSRVToolWindow.ScaleImagesForDPI 201
- TSRVToolWindow.Spacer 201
- TSRVTSCloseButtonVisibility 263
- TSRVTVAlign 280
- TSRVVAlign 140
- TSRVViewMode 164
- TSRVViewProperty 153
- TSRVViewProperty.AutoVScrollBarPosition 155
- TSRVViewProperty.DarkMode 156
- TSRVViewProperty.DarkModeUI 156
- TSRVViewProperty.FooterPen 156
- TSRVViewProperty.FooterTitleColor 156
- TSRVViewProperty.FooterTitleFont 157
- TSRVViewProperty.FreePosPage 157

TSRVViewProperty.HeaderFooterShade 157
 TSRVViewProperty.HeaderPen 158
 TSRVViewProperty.HeaderTitleColor 158
 TSRVViewProperty.HeaderTitleFont 158
 TSRVViewProperty.HintFont 159
 TSRVViewProperty.HintPrefixText 159
 TSRVViewProperty.IconStyle 159
 TSRVViewProperty.MainDocumentShade 159
 TSRVViewProperty.MainPen 160
 TSRVViewProperty.MainTitleColor 160
 TSRVViewProperty.MainTitleFont 160
 TSRVViewProperty.MarginsPen 161
 TSRVViewProperty.MarginsRectVisible 161
 TSRVViewProperty.MouseWheelZoom 161
 TSRVViewProperty.PageHeight 161
 TSRVViewProperty.PageWidth 161
 TSRVViewProperty.ShowScrollHint 162
 TSRVViewProperty.TableIconDelay 162
 TSRVViewProperty.TableIconPopupMenu 162
 TSRVViewProperty.Texts 162
 TSRVViewProperty.UseTableIcons 163
 TSRVViewProperty.UseVCLThemes 164
 TSRVViewProperty.ViewMode 164
 TSRVViewProperty.WheelStep 164
 TSRVViewProperty.WorkArea 164
 TSRVViewProperty.ZoomFont 165
 TSRVViewProperty.ZoomMax 165
 TSRVViewProperty.ZoomMenu 165
 TSRVViewProperty.ZoomMin 165
 TSRVViewProperty.ZoomMode 165
 TSRVViewProperty.ZoomModeEdit 166
 TSRVViewProperty.ZoomModeIN 166
 TSRVViewProperty.ZoomModeOUT 166
 TSRVViewProperty.ZoomPanelFont 166
 TSRVViewProperty.ZoomPanelUsesGradient 167
 TSRVViewProperty.ZoomPanelVisible 167
 TSRVViewProperty.ZoomPercent 167
 TSRVViewProperty.ZoomPercentEdit 168
 TSRVViewProperty.ZoomPercentIN 168
 TSRVViewProperty.ZoomPercentOUT 168
 TSRVWinControl 339
 TSRVWorkArea 164
 TSRVZoomInfo 100

- U -

Units 289
 TsrvActionPageFormat 289
 UnitsPerInchH 99

TSRichViewEdit 99
 UnitsPerInchV 100
 TSRichViewEdit 100
 UnitsProgram 68
 TSRichViewEdit 68
 Update 221
 TSRVPrint 221
 UpdateBuffer 100
 TSRichViewEdit 100
 UpdateBufferAt 100
 TSRichViewEdit 100
 UpdateCache 184
 TSRVPageScroll 184
 UseDrawHyperlinksEvent 68
 TSRichViewEdit 68
 UsePageOrders 142
 TSRVPageProperty 142
 UsePhysicalOffsets 216
 TSRVPrint 216
 UseStyleTemplates 68
 TSRichViewEdit 68
 UseTableIcons 163
 TSRVViewProperty 163
 UseVCLThemes 164
 TSRVViewProperty 164
 UseXPThemes 247
 TSRVScrollBar 247

- V -

Version 68
 TSRichViewEdit 68
 VerticalAlign 263
 TSRVCloseButton 263
 VerticalScrollBarSchemes 232
 TSRVSkin 232
 VerticalTabSetSchemes 232
 TSRVSkin 232
 ViewMode 164
 TSRVViewProperty 164
 ViewProperty 68
 TSRichViewEdit 68
 Visibility 263
 TSRVCloseButton 263
 Visible 126, 133, 148, 150, 263, 265, 274
 TSRVBackgroundProperty 126
 TSRVCloseButton 263
 TSRVLineNumberProperty 133
 TSRVPropertyTBH 148
 TSRVPropertyTBV 150

Visible 126, 133, 148, 150, 263, 265, 274
 TSRVTabItem 265
 TSRVToolButton 274
 VMaxScrollPos 68
 TSRichViewEdit 68
 VPadding 145
 TSRVPagePositionProperty 145
 VScrollBar 69
 TSRichViewEdit 69
 VScrollBarSchemeIndex 69
 TSRichViewEdit 69
 VScrollPos 69
 TSRichViewEdit 69

- W -

WheelStep 164
 TSRVViewProperty 164
 Width 263
 TSRVCloseButton 263
 WorkArea 164
 TSRVViewProperty 164

- Z -

ZoomCanvas 100
 TSRichViewEdit 100
 ZoomFont 165
 TSRVViewProperty 165
 ZoomMax 165
 TSRVViewProperty 165
 ZoomMenu 165
 TSRVViewProperty 165
 ZoomMin 165
 TSRVViewProperty 165
 ZoomMode 165
 TSRVViewProperty 165
 ZoomModeEdit 166
 TSRVViewProperty 166
 ZoomModelN 166
 TSRVViewProperty 166
 ZoomModeOUT 166
 TSRVViewProperty 166
 ZoomPanelFont 166
 TSRVViewProperty 166
 ZoomPanelUsesGradient 167
 TSRVViewProperty 167
 ZoomPanelVisible 167
 TSRVViewProperty 167
 ZoomPercent 167, 312

TsrvActionZoom 312
 TSRVViewProperty 167
 ZoomPercentEdit 168
 TSRVViewProperty 168
 ZoomPercentIN 168
 TSRVViewProperty 168
 ZoomPercentOUT 168
 TSRVViewProperty 168
 ZoomRectIn 100
 TSRichViewEdit 100
 ZoomRectOut 100
 TSRichViewEdit 100
 ZoomToFullPages 101
 TSRichViewEdit 101