

RichViewActions

RichViewActions © trichview.com

Table of Contents

Part I RichViewActions 14

1..Overview 14

Installing RichViewActions 14
Using RichViewActions 14
Style templates 17
History 18

2..Components 22

TRVAControlPanel 23
Properties 25
TRVAControlPanel.ActionsEnabled 25
TRVAControlPanel.AddColorNameToHints 25
TRVAControlPanel.AutoDeleteUnusedStyles 25
TRVAControlPanel.ClHTTP 25
TRVAControlPanel.ColorDialog 25
TRVAControlPanel.DefaultColor 26
TRVAControlPanel.DefaultControl 26
TRVAControlPanel.DefaultCustomFilterIndex 26
TRVAControlPanel.DefaultDocParameters 26
TRVAControlPanel.DefaultExt 27
TRVAControlPanel.DefaultFileFormat 27
TRVAControlPanel.DefaultFileName 27
TRVAControlPanel.DefaultMargin 27
TRVAControlPanel.DialogFontName 28
TRVAControlPanel.DialogFontSize 28
TRVAControlPanel.FirstPageNumber 28
TRVAControlPanel.Header, Footer 28
TRVAControlPanel.IdHTTP 29
TRVAControlPanel.Language 29
TRVAControlPanel.MetafileCompatibility 29
TRVAControlPanel.PixelBorders 29
TRVAControlPanel.RVAddictSpell3 30
TRVAControlPanel.RVAddictThesaurus3 30
TRVAControlPanel.RVFFilter 30
TRVAControlPanel.RVFLocalizable 30
TRVAControlPanel.RVFormatTitle 31
TRVAControlPanel.RVHTMLImporter 31
TRVAControlPanel.RVHTMLView Importer 31
TRVAControlPanel.RVPrint 31
TRVAControlPanel.RVStylesFilter, RVStylesExt 32
TRVAControlPanel.RVXML 32
TRVAControlPanel.SearchScope 32
TRVAControlPanel.Show SoftPageBreaks 32
TRVAControlPanel.TableGridStyle 33
TRVAControlPanel.UnitsDisplay 33
TRVAControlPanel.UnitsProgram 33
TRVAControlPanel.UseHelpFiles 34
TRVAControlPanel.UserInterface 34
TRVAControlPanel.UseTextCodePageDialog 34
TRVAControlPanel.UseXPThemes 35
TRVAControlPanel.XMLFilter 35

TRVAControlPanel.XMLLocalizable ... 35
Methods ... 36
TRVAControlPanel.Activate ... 36
Events ... 36
TRVAControlPanel.OnAddStyle ... 36
TRVAControlPanel.OnBackgroundChange ... 36
TRVAControlPanel.OnCreateTeForm ... 36
TRVAControlPanel.OnCustomFileOperation ... 37
TRVAControlPanel.OnDownload ... 38
TRVAControlPanel.OnGetActionControlCoords ... 38
TRVAControlPanel.OnMarginsChanged ... 39
TRVAControlPanel.OnStyleNeeded ... 39
TRVAPopupMenu ... 40
Properties ... 41
TRVAPopupMenu.ActionList ... 41
TRVAPopupMenu.MaxSuggestionsCount ... 41
Events ... 42
TRVAPopupMenu.OnLiveSpellAdd ... 42
TRVAPopupMenu.OnLiveSpellGetSuggestions ... 42
TRVAPopupMenu.OnLiveSpellIgnoreAll ... 43
TRVAPopupMenu.OnLiveSpellWordReplace ... 43
TCustomRVFontComboBox ... 43
Properties ... 44
TCustomRVFontComboBox.ActionFont ... 44
TCustomRVFontComboBox.Editor ... 44
TRVFontCharsetComboBox ... 44
Properties ... 45
TRVFontCharsetComboBox.AddDefaultCharset ... 45
TRVFontCharsetComboBox.DefaultCharsetCaption ... 45
TRVFontCharsetComboBox.FontName ... 45
TRVFontComboBox ... 46
TRVFontSizeComboBox ... 46
Properties ... 47
TRVFontSizeComboBox.FontName ... 47
TRVStyleTemplateComboBox, TRVStyleTemplateListBox ... 47
Properties ... 48
TRVStyleTemplateComboBox.*ImageIndex ... 48
TRVStyleTemplateComboBox.ControlPanel ... 48
TRVStyleTemplateComboBox.Editor ... 48
TRVStyleTemplateComboBox.Images ... 48
TRVStyleTemplateComboBox.ShowAllStyles ... 49
TRVStyleTemplateComboBox.ShowClearFormat ... 49
TRVStyleTemplateComboBox.SmartHeadings ... 49
Methods ... 49
TRVStyleTemplateComboBox.Localize ... 49
Events ... 50
TRVStyleTemplateComboBox.OnClickAllStyles ... 50
3 Actions ... 50
File ... 54
TrvActionCustomFileIO ... 54
Properties ... 54
TrvActionCustomFileIO.CustomFilter ... 54
TrvActionCustomIO ... 54
Properties ... 55
TrvActionCustomIO.AutoUpdateFileName ... 55
TrvActionCustomIO.DialogTitle ... 55
TrvActionCustomIO.FileName ... 55

TrvActionCustomIO.InitialDir ... 56
TrvActionCustomNew ... 56
Properties ... 57
TrvActionCustomNew.ActionSave ... 57
Events ... 57
TrvActionCustomNew.OnNew ... 57
TrvActionNew ... 57
Properties ... 58
TrvActionNew.StyleTemplates ... 58
Methods ... 59
TrvActionNew.Reset ... 59
TrvActionOpen ... 59
Properties ... 60
TrvActionOpen.ActionNew ... 60
TrvActionOpen.CustomFilter ... 60
TrvActionOpen.DialogTitle ... 60
TrvActionOpen.Filter ... 61
TrvActionOpen.InitialDir ... 61
TrvActionOpen.LastFilterIndex ... 61
Methods ... 61
TrvActionOpen.LoadFile ... 61
Events ... 62
TrvActionOpen.OnOpenFile ... 62
TrvActionExport ... 62
Properties ... 63
TrvActionExport.CreateDirectoryForHTMLImages ... 63
TrvActionExport.FileTitle ... 63
TrvActionExport.Filter ... 63
TrvActionExport.ImagePrefix ... 63
TrvActionExport.SaveOptions ... 64
Methods ... 64
TrvActionExport.ExportToFile ... 64
TrvActionPageSetup ... 64
Properties ... 65
TrvActionPageSetup.MarginsUnits ... 65
Events ... 65
TrvActionPageSetup.OnChange ... 65
TrvActionPrint ... 65
Properties ... 66
TrvActionPrint.Title ... 66
TrvActionPrintPreview ... 66
Properties ... 67
TrvActionPrintPreview.ActionPageSetup ... 67
TrvActionPrintPreview.ActionPrint ... 67
TrvActionPrintPreview.Maximized ... 67
Events ... 68
TrvActionPrintPreview.OnGetPreviewFormClass ... 68
TrvActionQuickPrint ... 68
Properties ... 68
TrvActionQuickPrint.Title ... 68
TrvActionSave ... 69
Properties ... 69
TrvActionSave.ActionSaveAs ... 69
TrvActionSave.CreateDirectoryForHTMLImages ... 69
TrvActionSave.DisableWhenUnmodified ... 70
TrvActionSave.Documents ... 70
TrvActionSave.FileTitle ... 70
TrvActionSave.ImagePrefix ... 70

TrvActionSave.LostFormatWarning ... 71
TrvActionSave.SaveOptions ... 71
TrvActionSave.SuppressNextErrorMessage ... 71
Methods ... 71
TrvActionSave.CanCloseDoc ... 71
TrvActionSave.FindDoc ... 72
Events ... 72
TrvActionSave.OnSaving ... 72
TrvActionSave.OnDocumentFileChange ... 72
TrvActionSaveAs ... 73
Properties ... 73
TrvActionSaveAs.ActionSave ... 73
TrvActionSaveAs.Filter ... 74
TrvCustomPrintAction ... 74
Edit ... 74
TrvActionCharCase ... 74
TrvActionCopy ... 75
TrvActionCut ... 75
TrvActionFind ... 76
Properties ... 76
TrvActionFind.ActionReplace ... 76
Methods ... 77
TrvActionFind.CloseDialog ... 77
TrvActionFindNext ... 77
Properties ... 77
TrvActionFindNext.ActionFind ... 77
TrvActionPaste ... 77
TrvActionPasteAsText ... 78
TrvActionPasteSpecial ... 78
Properties ... 79
TrvActionPasteSpecial.StoreFileName ... 79
TrvActionPasteSpecial.StoreFileNameInItemName ... 80
Methods ... 80
TrvActionPasteSpecial.AddFormat ... 80
Events ... 80
TrvActionPasteSpecial.OnCanPaste ... 80
TrvActionPasteSpecial.OnCustomPaste ... 81
TrvActionPasteSpecial.OnShow ing ... 81
TrvActionRedo ... 81
TrvActionReplace ... 81
Properties ... 82
TrvActionReplace.ActionFind ... 82
TrvActionReplace.Show ReplaceAllSummary ... 82
Methods ... 82
TrvActionReplace.CloseDialog ... 82
Events ... 82
TrvActionReplace.OnReplaceAllEnd ... 82
TrvActionReplace.OnReplaceAllStart ... 83
TrvActionReplace.OnReplacing ... 83
TrvActionSelectAll ... 83
TrvActionUndo ... 83
TrvCustomEditAction ... 84
Font ... 85
TrvActionFontAllCaps ... 85
TrvActionFontBackColor ... 85
TrvActionFontBold ... 86
TrvActionFontColor ... 86
TrvActionFontCustomColor ... 87

TrvActionFontEx ... 87
Properties ... 88
TrvActionFontEx.AutoApplySymbolCharset ... 88
TrvActionFontEx.BackColor ... 88
TrvActionFontEx.CharScale ... 89
TrvActionFontEx.CharSpacing ... 89
TrvActionFontEx.FontStyleEx ... 89
TrvActionFontEx.SubSuperScriptType ... 89
TrvActionFontEx.UnderlineColor ... 90
TrvActionFontEx.UnderlineType ... 90
TrvActionFontEx.ValidProperties ... 90
TrvActionFontEx.VShift ... 90
TrvActionFontEx.FontSizeDouble ... 91
TrvActionFontGrow ... 91
Properties ... 91
TrvActionFontGrow .MaxSize ... 91
TrvActionFontGrow OnePoint ... 92
Properties ... 92
TrvActionFontGrow OnePoint.MaxSize ... 92
TrvActionFontItalic ... 92
TrvActionFontOverline ... 93
TrvActionFonts ... 93
Properties ... 94
TrvActionFonts.Font ... 94
TrvActionFonts.UserInterface ... 94
TrvActionFontShrink ... 94
Properties ... 95
TrvActionFontShrink.MinSize ... 95
TrvActionFontShrinkGrow ... 95
Properties ... 96
TrvActionFontShrinkGrow .Percent ... 96
TrvActionFontShrinkOnePoint ... 96
Properties ... 96
TrvActionFontShrinkOnePoint.MinSize ... 96
TrvActionFontStrikeout ... 96
TrvActionFontStyle ... 97
TrvActionFontStyleEx ... 98
TrvActionFontUnderline ... 98
TrvActionSSScript ... 99
TrvActionSubscript ... 99
TrvActionSuperscript ... 100
TrvActionTextBiDi ... 100
TrvActionTextLTR ... 101
TrvActionTextRTL ... 102
TrvActionTextStyles ... 102
Paragraph ... 103
TrvActionAlignCenter ... 103
TrvActionAlignCustomJustify ... 104
Properties ... 104
TrvActionAlignCustomJustify.LastLineAlignment, UseLastLineAlignment ... 104
TrvActionAlignDistribute ... 104
TrvActionAlignJustify ... 105
TrvActionAlignLeft ... 106
TrvActionAlignment ... 106
TrvActionAlignRight ... 107
TrvActionClearBoth ... 107
TrvActionClearLeft ... 108
TrvActionClearNone ... 108

TrvActionClearRight ... 109
TrvActionClearTextFlow ... 109
TrvActionCustomParaListSwitcher ... 110
Properties ... 110
TrvActionCustomParaListSwitcher.IndentStep ... 110
TrvActionCustomParaListSwitcher.ListLevels ... 110
TrvActionIndent ... 111
Properties ... 111
TrvActionIndent.IndentStep ... 111
TrvActionIndentDec ... 112
TrvActionIndentInc ... 112
Properties ... 113
TrvActionIndentInc.IndentMax ... 113
TrvActionLineSpacing ... 113
TrvActionLineSpacing100 ... 113
TrvActionLineSpacing150 ... 114
TrvActionLineSpacing200 ... 114
TrvActionParaBiDi ... 115
TrvActionParaBorder ... 116
Properties ... 116
TrvActionParaBorder.Background ... 116
TrvActionParaBorder.Border ... 116
TrvActionParaBorder.UserInterface ... 117
TrvActionParaBorder.ValidProperties ... 117
TrvActionParaBullets ... 117
TrvActionParaColor ... 118
TrvActionParaColorAndPadding ... 118
Properties ... 119
TrvActionParaColorAndPadding.Padding ... 119
TrvActionParaColorAndPadding.UsePadding ... 119
TrvActionParaCustomColor ... 119
TrvActionParagraph ... 120
Properties ... 120
TrvActionParagraph.Alignment ... 120
TrvActionParagraph.DeleteAllTabs ... 121
TrvActionParagraph.FirstIndent ... 121
TrvActionParagraph.KeepLinesTogether ... 121
TrvActionParagraph.KeepWithNext ... 121
TrvActionParagraph.LastLineAlignment ... 122
TrvActionParagraph.LeftIndent ... 122
TrvActionParagraph.LineSpacing ... 122
TrvActionParagraph.LineSpacingType ... 122
TrvActionParagraph.OutlineLevel ... 123
TrvActionParagraph.RightIndent ... 123
TrvActionParagraph.SpaceAfter ... 123
TrvActionParagraph.SpaceBefore ... 123
TrvActionParagraph.Tabs ... 124
TrvActionParagraph.TabsToDelete ... 124
TrvActionParagraph.UserInterface ... 124
TrvActionParagraph.ValidProperties ... 124
TrvActionParaList ... 125
Properties ... 125
TrvActionParaList.IndentStep ... 125
TrvActionParaList.UpdateAllActionsOnForm ... 126
TrvActionParaList.ActionParaNumbering ... 126
TrvActionParaList.ActionParaBullets ... 126
TrvActionParaLTR ... 126
TrvActionParaNumbering ... 127

TrvActionParaRTL ... 128
TrvActionParaStyles ... 128
TrvActionWordWrap ... 129
Styles ... 129
TrvActionStyleTemplates ... 129
Properties ... 130
TrvActionStyleTemplates.Images ... 130
TrvActionStyleTemplates.StandardStyleTemplates ... 130
TrvActionStyleTemplates.*StyleImageIndex ... 131
TrvActionAddStyleTemplate ... 131
Properties ... 132
TrvActionAddStyleTemplate.ActionStyleTemplates ... 132
TrvActionStyleInspector ... 132
Properties ... 132
TrvActionStyleInspector.Images ... 132
TrvActionStyleInspector.FontImageIndex, ParaImageIndex ... 133
Methods ... 133
TrvActionStyleInspector.UpdateInfo ... 133
Events ... 133
TrvActionCustomInfoWindow .OnShow ing ... 133
TrvActionClearFormat ... 133
TrvActionClearTextFormat ... 134
Insert ... 134
TrvActionInsertCaption ... 134
Properties ... 135
TrvActionInsertCaption.ExcludeLabel ... 135
TrvActionInsertCaption.InsertAbove ... 135
TrvActionInsertCustomPageNumber ... 136
Properties ... 136
TrvActionInsertCustomPageNumber.NumberType ... 136
TrvActionInsertFile ... 136
Properties ... 137
TrvActionInsertFile.Filter ... 137
Methods ... 137
TrvActionInsertFile.InsertFile ... 137
TrvActionInsertHLine ... 137
Properties ... 138
TrvActionInsertHLine.Color ... 138
TrvActionInsertHLine.Style ... 138
TrvActionInsertHLine.Width ... 138
TrvActionInsertHyperlink ... 138
Properties ... 140
TrvActionInsertHyperlink.ActionStyleTemplates ... 140
TrvActionInsertHyperlink.AutoAddHyperlinkStyleTemplate ... 140
TrvActionInsertHyperlink.BackColor ... 140
TrvActionInsertHyperlink.Color ... 140
TrvActionInsertHyperlink.Cursor ... 140
TrvActionInsertHyperlink.HoverBackColor ... 141
TrvActionInsertHyperlink.HoverColor ... 141
TrvActionInsertHyperlink.HoverEffects ... 141
TrvActionInsertHyperlink.HoverUnderlineColor ... 141
TrvActionInsertHyperlink.SetToPictures ... 142
TrvActionInsertHyperlink.SpaceFiller ... 142
TrvActionInsertHyperlink.Style ... 142
TrvActionInsertHyperlink.StyleTemplateName ... 142
TrvActionInsertHyperlink.UnderlineColor ... 143
TrvActionInsertHyperlink.ValidProperties ... 143
Methods ... 143

TrvActionInsertHyperlink.DetectURL 143
TrvActionInsertHyperlink.EncodeTarget 144
TrvActionInsertHyperlink.GetHyperlinkStyleNo 144
TrvActionInsertHyperlink.GetNormalStyleNo 145
TrvActionInsertHyperlink.GoToLink 145
TrvActionInsertHyperlink.SetTags 145
TrvActionInsertHyperlink.TerminateHyperlink 146
Events 146
TrvActionInsertHyperlink.OnApplyHyperlinkToItem 146
TrvActionInsertHyperlink.OnGetHyperlinkTargetFromItem 146
TrvActionInsertHyperlink.OnHyperlinkForm 147
TrvActionInsertNumber 147
TrvActionInsertNumSequence 148
Properties 148
TrvActionInsertNumSequence.NumberType 148
TrvActionInsertNumSequence.SeqName 148
TrvActionInsertNumSequence.UseDefaults 149
TrvActionInsertPageBreak 149
TrvActionInsertPageCount 149
TrvActionInsertPageNumber 150
TrvActionInsertPicture 150
Properties 151
TrvActionInsertPicture.BackgroundColor 151
TrvActionInsertPicture.BorderWidth, BorderColor 151
TrvActionInsertPicture.DefaultExt 151
TrvActionInsertPicture.Filter 151
TrvActionInsertPicture.MaxImageSize 152
TrvActionInsertPicture.OuterHSpacing, OuterVSpacing 152
TrvActionInsertPicture.Spacing 152
TrvActionInsertPicture.StoreFileName 152
TrvActionInsertPicture.StoreFileNameInItemName 153
TrvActionInsertPicture.VAlign 153
TrvActionInsertSymbol 153
Properties 154
TrvActionInsertSymbol.AlwaysInsertUnicode 154
TrvActionInsertSymbol.DisplayUnicodeBlocks 154
TrvActionInsertSymbol.SymbolType 154
TrvActionInsertSymbol.UseCurrentFont 155
TrvActionInsertText 155
Events 155
TrvActionInsertText.OnInsertText 155
Notes 156
TrvActionCustomInsertSidenote 156
Properties 156
TrvActionCustomInsertSidenote.BoxPosition 156
TrvActionCustomInsertSidenote.BoxProperties 157
TrvActionCustomInsertSidenote.StyleTemplateName 157
TrvActionEditNote 157
Properties 158
TrvActionEditNote.JumpToNextNote 158
TrvActionInsertEndnote 159
TrvActionInsertFootnote 159
TrvActionInsertNote 160
Properties 161
TrvActionInsertNote.ActionEditNote 161
TrvActionInsertNote.Font 161
TrvActionInsertSidenote 161
Properties 162

TrvActionInsertSidenote.RefStyleTemplateName ... 162
TrvActionInsertTextBox ... 162
Tables ... 163
TrvActionInsertTable ... 163
Properties ... 164
TrvActionInsertTable.BackgroundPicture ... 164
TrvActionInsertTable.BackgroundStyle ... 164
TrvActionInsertTable.BestWidth ... 164
TrvActionInsertTable.BorderColor ... 164
TrvActionInsertTable.BorderHSpacing ... 164
TrvActionInsertTable.BorderLightColor ... 165
TrvActionInsertTable.BorderStyle ... 165
TrvActionInsertTable.BorderVSpacing ... 165
TrvActionInsertTable.BorderWidth ... 165
TrvActionInsertTable.CellBorderColor ... 165
TrvActionInsertTable.CellBorderLightColor ... 166
TrvActionInsertTable.CellBorderStyle ... 166
TrvActionInsertTable.CellBorderWidth ... 166
TrvActionInsertTable.CellHPadding,CellVPadding ... 166
TrvActionInsertTable.CellHSpacing ... 167
TrvActionInsertTable.CellVSpacing ... 167
TrvActionInsertTable.ColBandSize, Row BandSize ... 167
TrvActionInsertTable.ColCount ... 167
TrvActionInsertTable.Color ... 168
TrvActionInsertTable.HeadingRow Count ... 168
TrvActionInsertTable.HOutermostRule ... 168
TrvActionInsertTable.HRuleColor ... 168
TrvActionInsertTable.HRuleWidth ... 168
TrvActionInsertTable.ItemText ... 169
TrvActionInsertTable.Row Count ... 169
TrvActionInsertTable.Row sKeepTogether ... 169
TrvActionInsertTable.Row sVAlign ... 169
TrvActionInsertTable.TableOptions ... 169
TrvActionInsertTable.TablePrintOptions ... 170
TrvActionInsertTable.VisibleBorders ... 170
TrvActionInsertTable.VOutermostRule ... 170
TrvActionInsertTable.VRuleColor ... 170
TrvActionInsertTable.VRuleWidth ... 170
TrvActionInsertTable's row and column colors ... 171
Methods ... 171
TrvActionInsertTable.Show TableSizeDialog ... 171
Events ... 171
TrvActionInsertTable.OnCloseTableSizeDialog ... 171
TrvActionInsertTable.OnInserting ... 171
TrvActionTableCell ... 172
TrvActionTableCellAllBorders ... 172
TrvActionTableCellBorder ... 173
TrvActionTableCellBottomBorder ... 173
TrvActionTableCellLeftBorder ... 174
TrvActionTableCellNoBorders ... 175
TrvActionTableCellRightBorder ... 175
TrvActionTableCellRotation ... 176
TrvActionTableCellRotationNone ... 176
TrvActionTableCellRotation90 ... 177
TrvActionTableCellRotation180 ... 177
TrvActionTableCellRotation270 ... 178
TrvActionTableCellTopBorder ... 179
TrvActionTableCellVAlign ... 179

TrvActionTableCellVAlignBottom ... 180
TrvActionTableCellVAlignDefault ... 180
TrvActionTableCellVAlignMiddle ... 181
TrvActionTableCellVAlignTop ... 182
TrvActionTableDeleteCols ... 182
TrvActionTableDeleteRows ... 183
TrvActionTableDeleteTable ... 183
TrvActionTableGrid ... 184
TrvActionTableInsertColLeft ... 184
TrvActionTableInsertColRight ... 185
TrvActionTableInsertRowsAbove ... 185
TrvActionTableInsertRowsBelow ... 186
Properties ... 186
TrvActionTableInsertRowsBelow.AllowMultiple ... 186
TrvActionTableMergeCells ... 186
TrvActionTableMultiCellAttributes ... 187
TrvActionTableProperties ... 187
Properties ... 188
TrvActionTableProperties.ActionInsertTable ... 188
TrvActionTableProperties.BackgroundGraphicFilter ... 188
TrvActionTableProperties.StoreImageFileName ... 188
TrvActionTableRCBase ... 189
TrvActionTableSelectCell ... 189
TrvActionTableSelectCols ... 190
TrvActionTableSelectRows ... 190
TrvActionTableSelectTable ... 191
TrvActionTableSort ... 192
TrvActionTableSplit ... 192
TrvActionTableSplitCells ... 193
TrvActionTableToText ... 193
Addict 3&4 Support ... 194
TrvActionAddictSpell3 ... 194
TrvActionAddictThesaurus3 ... 194
Others ... 195
TrvAction ... 195
Properties ... 195
TrvAction.Control ... 195
TrvActionBackground ... 195
Properties ... 196
TrvActionBackground.CanChangeMargins ... 196
TrvActionBackground.ImageFileName ... 196
Events ... 197
TrvActionBackground.OnChange ... 197
TrvActionColor ... 197
TrvActionCustomColor ... 197
Properties ... 198
TrvActionCustomColor.CallerControl ... 198
TrvActionCustomColor.Color ... 198
TrvActionCustomColor.UserInterface ... 198
Events ... 199
TrvActionCustomColor.OnShowColorPicker ... 199
TrvActionCustomColor.OnHideColorPicker ... 199
TrvActionEvent ... 199
Events ... 199
TrvActionEvent.OnExecute ... 199
TrvActionEvent.OnUpdate ... 200
TrvActionFillColor ... 200
Properties ... 200

TrvActionFillColor.AllowApplyingTo ... 200
TrvActionHide ... 201
TrvActionItemProperties ... 201
Properties ... 202
TrvActionItemProperties.ActionInsertHLine ... 202
TrvActionItemProperties.ActionInsertTable ... 202
TrvActionItemProperties.BackgroundGraphicFilter ... 202
TrvActionItemProperties.GraphicFilter ... 202
TrvActionItemProperties.StoreImageFileName ... 203
Events ... 203
TrvActionItemProperties.OnCanApply ... 203
TrvActionItemProperties.OnCustomItemPropertiesDialog ... 203
TrvActionRemoveHyperlinks ... 204
Properties ... 204
TrvActionRemoveHyperlinks.ActionInsertHyperlink ... 204
TrvActionRemovePageBreak ... 204
TrvActionShowSpecialCharacters ... 205
TrvActionVAlign ... 205
TrvCustomAction ... 206
Properties ... 207
TrvCustomAction.Caption ... 207
TrvCustomAction.ControlPanel ... 207
TrvCustomAction.Disabled ... 207
TrvCustomAction.Hint ... 207
Methods ... 208
TrvCustomAction.GetControlPanel ... 208
TrvCustomEditorAction ... 208
Properties ... 208
TrvCustomEditorAction.Form ... 208
TrvCustomEditorAction.SubDocEditor ... 208
Events ... 209
TrvCustomEditorAction.OnFormCreate ... 209
TrvCustomEditorAction.OnShowing, OnHide ... 209
.4 Localization of RichViewActions ... 210
RVALocalize unit ... 212
Procedures and Functions ... 212
RVA_EnumLanguages procedure ... 212
RVA_FillLanguageList procedure ... 213
RVA_GetCharset function ... 213
RVA_GetHelpFile function ... 213
RVA_GetLanguageName function ... 214
RVA_GetS function ... 214
RVA_SwitchLanguage procedure ... 214
RVA_TranslateUnits procedure ... 215
Functions for progress messages ... 215
Types ... 216
TRVALanguageName ... 216
RichViewActions unit ... 217
Procedures and functions ... 217
RVA_ChooseLanguage function ... 217
RVA_GetColorName function ... 218
RVA_LocalizeForm procedure ... 218
Other units ... 218
GetAddictSpellLanguage, GetAddictThesLanguage functions ... 218
RVALocalizeRuler procedure ... 219
.5 Third-party tools ... 219
Components ... 219

Addict 3 and 4 ... 219
Indy ... 220
CleverComponents ... 221
RvHtmlImporter ... 221
RvHtmlView Importer ... 222
RichView XML ... 223
SpTBXLib ... 223
ThemeEngine ... 223
TNT Controls ... 224
TBX ... 224
Toolbar2000 ... 224
Toolbar Images ... 225
TRichView Icons ... 225
GlyFX toolbar images ... 225
Glyfz toolbar images ... 226
Fugue Icons toolbar images ... 227
FamFamFam Silk Icons toolbar images ... 227
.6 Procedures and Functions ... 228
RichViewActions unit ... 228
ConvertToUnicode procedure ... 228
GoToCheckpoint function ... 228
PasteHTML procedure ... 228
RVA_Addict3AutoCorrect function ... 228
RVA_ChooseStyle function ... 229
RVA_EditorControlFunctionDef function ... 229
RVA_ConvertToPixels, RVA_ConvertToTwips procedures ... 229
RvHtmlHelp function ... 229
RVARibbonUtils unit ... 230
FillActionClientContents procedure ... 230
RemoveEllipsis, RemoveEllipsisFromRibbon procedures ... 230
LocalizeRibbonPage procedure ... 230
SetRulerColors procedure ... 230
Other units ... 230
LoadCSV function ... 230
MarkSubString functions ... 231
NormalizeRichView procedure ... 231
RVChangeCharCase, RVGetCharCase procedures ... 232
.7 Types ... 232
TRVAEditEvent ... 232
TRVAHFInfo ... 233
TRVASpellString, TRVASpellStrings, TRVASpellStringList ... 233
TrvFileExportFilter, TrvFileSaveFilter ... 233
TrvFileImportFilter, TrvFileOpenFilter ... 234
TRVShowFormEvent ... 235
.8 Global Variables ... 235
MainRVAControlPanel ... 235
RVA_EditForceDefControl ... 235
RVA_EditorControlFunction ... 236
ShowUntranslatedControls ... 236
.9 How to ... 237
How to change RVF format name ... 237

Index 238

1 RichViewActions

RichViewActions is a set of actions for Delphi and C++Builder allowing you to create a user interface for the TRichView and ScaleRichView editors. Actions can be assigned to command buttons, menu items, toolbar buttons. You can use the standard Delphi components or your favorite third-party components if they support actions.

The actions require no programming, just add a new action in the action manager, assign it to a component, and it will do all the work automatically.

RichViewActions homepage: ***www.trichview.com/resources/actions/***.

Minimal supported version of Delphi: 5 (or Delphi 4 with some limitations)

Minimal supported version of C++Builder: 6.

This help file includes the descriptions of all actions, main components, and utility functions. It does not describe visual controls installed with the package (the most of them are self-explanatory).

See also:

- Overview 14
- What's new? 18

1.1 Overview

- Installing RichViewActions 14
- Using RichViewActions 14
- Style Templates 17
- What's new 18

1.1.1 Installing RichViewActions

Installing

RichViewActions is installed by the automatic installer together with TRichView.

If you want to use third-party tools 219, read the installation instructions in the topics about them.

Read **Install.txt** for additional information.

1.1.2 Using RichViewActions

Overview

All actions have Control 195 property of TCustomRVControl type. If this property is not assigned, actions work with the focused control (or GetControlPanel 208.DefaultControl 26 component).

All actions have Disabled 207 property (Boolean). If it is *True*, all actions are disabled. If it is *False* (default), actions update their Enabled property automatically.

TRVAControlPanel 23 component has a set of properties affecting all (or many) actions.

TRVAPopupMenu 40 is a popup menu for TCustomRichViewEdit that maintains items itself. Set its ActionList 41 property to the action list containing RichViewActions. Assign this component to the editor's PopupMenu. Do not add items in this menu manually – they all will be cleared (you can add them on each call of OnPopup, though). The menu supports live spelling, see below.

Preparing for the Actions

Right-click RichViewEdit in Delphi/C++Builder, choose Settings in the context menu, select "Allow adding styles

dynamically", check options for saving and loading background and layout.

Include *rvoCtrlJumps* in RichViewEdit.EditorOptions.

How to use

Put an ActionList component on the form. Link it with some image-list. Double click the ActionList. Click "New Standard Actions" on the component editor toolbar. Select the action and click OK. If linked image-list is 16x16, an action image will be added automatically.

Alternatively, you can use dmActions.pas unit. This is a datamodule containing a ready to use ActionList and ImageList. If you will not modify this unit, you can include it in your project directly. But it's highly recommended to create a copy of this unit (under a new name) and use it instead. Important: if you access actions in OnCreate event of some form, this form must be created after this datamodule (the form creation order may be specified in the project options).

Localization

Several languages are available for translations of user interface. Some operations must be performed even if you use only one language. See Localization of RichViewActions 210.

Ruler

RichViewActions include TRuler component by Pieter Zijlstra and TRVRuler component inherited from it.

Create a TRVRuler component, place on the form above editor, assign RVRuler.RichViewEdit to this editor.

In TRVAControlPanel.OnMarginsChanged 39 call RVRuler.UpdateRulerMargins.

The ruler supports units: millimeters, picas, pixels and points.

If you want to remove support for tab stops, set ruler's MaxTabs property to 0.

Actions Requiring a Special Attention

It's recommended to assign Control 195 property for the following actions:

- TrvActionStyleInspector 132;
- TrvActionEditNote 157.

Otherwise, it will be assigned automatically when the action is executed for the first time.

Some additional code may be required to implement UI for a note editor of TrvActionEditNote 157.

Files

File-related actions (TrvActionNew 56, TrvActionOpen 59, TrvActionSave 69, TrvActionSaveAs 73) are linked together. The most of them cannot work without others. These actions keep track of all opened files and can be used even for multiple RichViewEdits (for example, in MDI application) without any additional effort from the programmer. These actions cannot be used for TDBRichViewEdit.

TrvActionExport 62, TrvActionInsertFile 136 are independent – they are not linked with other actions, they just allow to export/insert a file.

Printing

TRVAControlPanel 23 has RVPrint 31: TRVPrint property. All printing actions (TrvActionPrintPreview 66, TrvActionQuickPrint 68, TrvActionPrint 65) use it.

It is recommended to assign it to some TRVPrint component.

If this property is not assigned, these actions search TRVPrint component on the same form as the target editor. If it is not found, they create a temporal TRVPrint component.

Besides, these actions use another property of TRVAControlPanel 23 – ShowSoftPageBreaks 32: Boolean. If

True (default), soft page breaks are shown when possible (after print preview or printing – until the next change in the document).

TrvActionPageSetup[64] is a special printing action. It does not require RichViewEdit, but requires RVPrint[31] assigned to TRVAControlPanel[23].

Color

All the main coloring actions (TrvActionFontColor[86], TrvActionFontBackColor[85], TrvActionParaColor[118], TrvActionColor[197]) can work in three modes, depending on UserInterface[198] property:

- *rvacAdvanced* (default) – on execution, a special non-modal color dialog pops up; it pops up at the position calculated by the coordinates of control that caused this action to execute (ActionComponent – button, for example) or by mouse coordinates;
- *rvacColorDialog* – on execution, a color is chosen with standard color dialog;
- *rvacNone* – no user interface, action just applies Color property.

This property gives you an ability to implement a color combo (you need two actions – one for applying the last chosen color (UserInterface[198] = *rvacNone*) and one for displaying a color dialog). Or you can implement your own interface for choosing colors.

All coloring actions have OnShowColorPicker[199] and OnHideColorPicker[199] events helping to implement a custom color combo.

Formatting

TrvActionFontGrow[91] and TrvActionFontShink[94] change the size of selected font by the given number of percents (Percent[96] property). If you change its value, do not forget to update hints for the actions.

TrvActionFontGrow[91], TrvActionFontGrowOnePoint[92] have MaxSize property, TrvActionFontShrink[94] TrvActionFontShrinkOnePoint[96] have MinSize property.

TrvActionParagraph[120] can work in two modes, depending on UserInterface[124]: Boolean property:

- *True* (default) – the action gets values of its properties from RichViewEdit, displays a dialog for modifying them, then apply them to the selection;
- *False* – the action applies its properties to the selection, without displaying a dialog.

TrvActionFontEx[87] is similar to TrvActionParagraph[120], it can also work in two modes.

Live spelling check

TRVAPopupMenu supports live spelling. If you use Addict 3-4[219], suggestions, "add to dictionary" and "ignore all" commands appear in the menu automatically. Otherwise, you need to process menu events: OnLiveSpellGetSuggestions[42], OnLiveSpellIgnoreAll[43], OnLiveSpellAdd[42].

If your spellchecker can store a user choice to generate better suggestions in future, you can also process OnLiveSpellWordReplace[43].

RichViewActions do not process OnSpellingCheck event, you still need to do it yourself, see the commented code in Unit3.pas.

Bidirected text

Use these actions only if you really need this feature in your application (bidirected text is not completely supported in all versions of Windows).

Before using these actions set RichViewEdit.BiDiMode either to *rvbdLeftToRight* or *rvbdRightToLeft* (do not use for BiDiMode=*rvbdUnspecified*!)

TrvActionParaLTR[126], TrvActionParaRTL[128] – set default bidi mode for the selected paragraphs.

TrvActionTextLTR[101], TrvActionTextRTL[102] – set bidi mode for the selected text.

1.1.3 Style templates

Several actions work differently depending on the value of UseStyleTemplates property of the target editor.

TrvActionNew [56], TrvActionOpen [59]

If Editor.UseStyleTemplates=*True*, TrvActionNew resets the Editor.Style's TextStyles and ParaStyles according to StyleTemplates [58], and clears its ListStyles.

If Editor.UseStyleTemplates=*False*, TrvActionNew does not reset styles; it may only delete unused styles (if GetControlPanel [208].AutoDeleteUnusedStyles [25]=*True*). If you want to reset styles, use OnNew [57] event.

TrvActionOpen uses its linked [60] TrvActionNew to perform the same operations before loading.

TrvActionPasteSpecial [78]

If Editor.UseStyleTemplates=*True*, the action displays additional options for pasting RTF and RVF documents:

- "apply styles of the target document"
- "keep styles and appearance"
- "keep appearance, ignore styles".

The action temporarily changes Editor.StyleTemplateInsertMode according to the chosen option.

TrvActionInsertHyperlink [138]

If Editor.UseStyleTemplates=*True*, the action applies StyleTemplateName [142] to hyperlinks (or clears a style template when it converts hyperlinks to a plain text). The dialog window allows choosing a style template to apply to a hyperlink.

If Editor.UseStyleTemplates=*False*, the action applies its properties listed in ValidProperties [143] (or applies properties of Editor.Style.TextStyles[0] when it converts hyperlinks to a plain text). The dialog window allows choosing colors and effects to apply to a hyperlink.

TrvActionRemoveHyperlinks [204] is affected as well.

TrvActionClearFormat [133], TrvActionClearTextFormat [134]

If Editor.UseStyleTemplates=*True*, the actions clear formatting according to style templates (applying "Normal" to paragraphs, clearing styles from normal text, applying "Hyperlink" to hyperlinks, resetting additional attributes).

If Editor.UseStyleTemplates=*False*, the actions apply Editor.Style.TextStyles[0] and Editor.Style.ParaStyles[0].

TrvActionStyleTemplates [129], TrvActionAddStyleTemplate [131]

These actions are enabled only if Editor.UseStyleTemplates=*True.*

TrvActionInsertFootnote [159], TrvActionInsertEndnote [159], TrvActionInsertSidenote [161]

If Editor.UseStyleTemplates=*True,* TrvActionInsertFootnote [159] inserts footnote characters using "footnote reference" style template, and formats footnote text using "footnote text" style template. TrvActionInsertEndnote [159] uses "endnote reference" and "endnote text" style templates, TrvActionInsertSidenote [161] uses RefStyleTemplateName [162] ("Sidenote Reference" by default) and StyleTemplateName [157] ("Sidenote Text" by default) style templates.

"footnote reference", "footnote text", "endnote reference", "endnote text" are included in TrvActionNew.StyleTemplates [58] by default. "Sidenote Reference" and "Sidenote Text" are not included.

ScaleRichView: TsrvActionInsertFootnote, TsrvActionInsertEndnote, TsrvActionInsertSidenote use the same style templates.

TrvActionInsertCaption 134

TrvActionInsertCaption uses "caption" style template. It is included in TrvActionNew.StyleTemplates 58 by default.

1.1.4 History

Changes after version 6 (between TRichView 16 and 17)

- TrvActionInsertPageBreak 149 and TrvActionRemovePageBreak 204, when called from a table cell, add and remove page breaks before the current table row.
- PageNumberType 136 property is added to TrvActionInsertPageCount 149. For inserted "page count" fields, a number type can be changed by TrvActionItemProperties 201.
- TrvActionInsertTable 163 has new properties to assign to table (HeadingRowColor and other colors 171, ColBandSize, RowBandSize 167) and to table rows (RowsVAlign 169, RowsKeepTogether 169).
- The table properties dialog (displayed by TrvActionItemProperties 201 and TrvActionTableProperties 187) allows defining colors of rows and columns.
- The "Default" chek box in the table properties affects not only the page "Table", but also pages "Rows" and "Columns".
- New untranslated features can be hidden using ShowUntranslatedControls 236 variable.

Changes in version 6 (between TRichView 15 and 16)

Compatibility issues:

TrvActionItemProperties 201 and TrvActionTableProperties 187 update linked TrvActionInsertTable 163 and TrvActionInsertHLine 137 actions when the user checks "Default" checkbox. Previously, they were always updated when assigned explicitly.

Installation and directory structure

Starting from this update, RichViewActions are installed automatically in Delphi and C++Builder IDE together with TRichView.

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-files are moved to "Source\Include" folder, demo projects are moved to "Demos" folder.

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

New actions

 TrvActionAlignDistribute 104 aligns the selected paragraph to the both left and right sides by adding space between all characters.

 TrvActionInsertPageCount 149 inserts a page count field.

Changes related to paragraph alignments

TrvActionAlignJustify 105 (as well as TrvActionAlignDistribute 104) has new properties: LastLineAlignment and UseLastLineAlignment 104.

TrvActionParagraph 120 has a new property: LastLineAlignment 122.

New properties:

- TRVAControlPanel[23].MetafileCompatibility[29] affects printing of TSRichViewEdit components;
- TrvActionSave[69].DisableWhenUnmodified[70] allows to disable the action for unmodified documents.

TrvActionRemovePageBreak[204] removes a page break from the current paragraph (not the current item as before).

TrvActionItemProperties[201] and TrvActionTableProperties[187] update linked TrvActionInsertTable[163] and TrvActionInsertHLine[137] actions when the user checks "Default" checkbox.

Changes in version 5 (between TRichView 14 and 15)

Compatibility issues:

- TrvActionFontEx[87].Font[94].Size is no longer used. Existing code must be changed to use TrvActionFontEx[87].FontSizeDouble[91] instead.
- 'Francais' cannot be used as a lanuage name; use 'French' or 'Français'.

New actions

- TrvActionInsertTextBox[162] inserts a floating box
- TrvActionInsertFootnote[159] inserts a footnote
- TrvActionInsertEndnote[159] inserts an endnote
- TrvActionInsertSidenote[161] inserts a note in a floating box
- TrvActionEditNote[157] displays a window for editing a note or a floating box
- TrvActionInsertNumber[147] inserts a "numbered sequence" item
- TrvActionInsertCaption[134] inserts a caption for an image or a table
- TrvActionInsertPageNumber[150] inserts a "page number" field

Changes

TrvActionExport[62] can export DocX files (without office converters). Make sure that *ffeDocX* is included in Filter[63].

Changes made by TrvActionColor[197], TrvActionBackground[195] can be undone.

New property: TrvActionFontEx[87].FontSizeDouble[91].

TRVFontSizeComboBox[46] supports fractional font sizes.

Changes in TrvActionItemProperties[201]:

- properties for sidenotes and text boxes
- properties for "numbered sequence" items
- properties for "page number" item
- redesigned picture properties pages
- redesigned table properties pages
- changes can be applied to ActionInsertTable[202] and ActionInsertHLine[202] (in addition to the current item)

Changes in TrvActionNew[57]:

- Reset[59] is made public.
- new item is included in StyleTemplates[58]: 'caption'.

New TrvActionInsertPicture[150] properties to assign to inserted pictures: BackgroundColor[151], BorderWidth, BorderColor[151], OuterHSpacing, OuterVSpacing[152].

New TrvActionInsertTable[163] properties to assign to inserted tables: CellHPadding and CellVPadding[166] (they replace CellPadding), VisibleBorders[170], BackgroundStyle[164], BackgroundPicture[164].

TRVAControlPanel[23].DialogFontName[28]'s default value is changed to 'Tahoma'.

New RemoveEllipsisFromRibbon[230] procedure replaces RemoveEllipsis.

Languages have two names specified: a name in English and a native name.

Changes added between TRichView v13 and v14

Compatibility issues:

TRVAControlPanel 23.TableGridStyle property is removed.

Redesigned control panel

TRVAControlPanel 23 is redesigned. It does not assign global variables any more. You can have more than one control panel in a single application. New properties of TRVAControlPanel: Header, Footer 28.

New actions

Styles: 17

 TrvActionStyleTemplates 129
 TrvActionAddStyleTemplate 131
 TrvActionClearFormat 133
 TrvActionClearTextFormat 134
 TrvActionStyleInspector 132

Cell rotation:

 TrvActionTableCellRotationNone 176
 TrvActionTableCellRotation90 177
 TrvActionTableCellRotation180 177
 TrvActionTableCellRotation270 178

Table operations:

 TrvActionTableSplit 192
 TrvActionTableSort 192
 TrvActionTableToText 193

New and improved components

New components for applying style templates 17:

 TRVStyleTemplateComboBox 47
 TRVStyleTemplateListBox 47

Improved components (can be linked to an editor, do not require additional code any more):

 TRVFontComboBox 46
 TRVFontSizeComboBox 46
 TRVFontCharsetComboBox 44

ScaleRichView support

You can assign not only TRichViewEdit, but also TSRichViewEdit to TRVAControlPanel.DefaultControl 26, or Control 195 properties of actions.

Simplification: since this version, when using ScaleRichView, you do not need to assign RVA_GetRichViewEditFromPopupComponent and RVA_GetRichViewEdit variables, you do not need to call SRichViewEdit.SetRVMargins in TRVAControlPanel.OnMarginsChanged 39.

Visual changes

Delphi XE2+'s visual styles are supported in components and dialogs. The function RVA_ChooseStyle 229 allows choosing and applying one of available styles.

Inverted (a light text on a dark background) color schemes are supported.

Text files

The file actions allow choosing a code page for text files (including UTF-8). This feature can be disabled by assigning TRVAControlPanel.UseTextCodePageDialog[34]=*False*.

New features, properties and events

A new event TRVAControlPanel.OnGetActionControlCoords[38] allows to specify coordinates for color-picker windows when actions are linked to non-visual components (for example, when using ExpressBars by Developer Express).

TrvActionNew[57] resets styles according to StyleTemplates[58], if style templates are used. TrvActionOpen[59] can use a linked[60] TrvActionNew to reset document before loading.

TrvActionPasteSpecial[78]:

- allows pasting URLs;
- respects RichViewEdit.AcceptPasteFormats property;
- if style templates[17] are used, displays style modes for pasting RTF and RVF.

TrvActionFontEx[87] can change a text background color (new BackColor[88] property).

TrvActionInsertHyperlink[138] can apply HoverUnderlineColor and HoverEffects properties (if style templates are not used). If style templates are used, it applies the chosen StyleTemplateName[142].

TrvActionTableGrid[184] shows/hides grid only in the target editor.

In the print preview form (TrvActionPrintPreview[66]), a mouse wheel changes pages, Ctrl+mouse wheel zooms in/out

New optional parameters in NormalizeRichView[231].

Localization

New localization functions: RVA_GetProgressMessage, RVA_GetPrintingMessage[215].

New languages:

- Portuguese (European)
- Catalan
- Hindi (in Delphi 4-2007, it is available only if TNT Controls[224] are used);
- Thai

Changes added between TRichView v12 and v13

New units of measurement

All dialogs display values measured according to new TRVAControlPanel[23].UnitsDisplay[33] and TRVAControlPanel.PixelBorders[29] properties.

Properties of several actions are measured in TRVAControlPanel[23].UnitsProgram[33].

New images

A new set of toolbar images[225] is available, created specially for RichViewActions. This set of images is used in this manual.

New component images are used in the Delphi/C++Builder's Component Palette for the components included in RichViewActions.

New properties of TRVAControlPanel

TRVAControlPanel.DefaultDocParameters[26] define a page layout for new documents (especially useful for ScaleRichView)

New TRVAControlPanel[23].DialogFontSize[28] property.

Changes in menus

New TRVAPopupActionBar 40 component (for Delphi 2006 or newer).

The type of ActionList 41 property for menus 40 was changed from TActionList to TCustomActionList, to allow using TActionManager.

Changes in actions

New action: TrvActionHide 201. TrvActionShowSpecialCharacters 205 shows/hides a hidden text as well as special characters.

New properties:

- TrvActionFontEx 87.AutoApplySymbolCharset 88
- TrvActionInsertHLine 137.Style 138 defines the style for inserted horizontal lines
- TrvActionBackground 195.CanChangeMargins 196 allows/disallows changing margins
- TrvActionParaBullets 117's and TrvActionParaNumbering 127's IndentStep 110 defines default indents for list levels.

TrvActionParaList 125.IndentStep 125 now affects indents of list style templates in non-HTML dialog too.

Other changes

New RVARibbonUtils 230 unit.

Ability to use CleverComponents 221 for downloading images.

Changes added between TRichView v11 and v12

New actions:

- TrvActionPasteAsText 78
- TrvActionVAlign 205
- TrvActionClearLeft 108
- TrvActionClearRight 109
- TrvActionClearBoth 107
- TrvActionClearNone 108
- TrvActionRemoveHyperlinks 204

New methods, properties and events

- TrvActionInsertFile.InsertFile 137 method
- TrvActionInsertPicture.Spacing 152 property
- TrvActionParagraph.OutlineLevel 123 property (paragraph dialog has a new combobox for specifying a paragraph outline level)
- TrvActionItemProperties.OnCanApply 203 event

New features:

- New types of vertical alignment for pictures in TrvActionItemProperties 201.
- Properties dialog for breaks (TrvActionItemProperties 201) has a new combobox for specifying a break style (line/rectangle/3d/dotted/dashed).
- Pasting HTML from the Clipboard using rvHtmlViewImporter 222.
- New datamodules with alternative toolbar images (GlyFX 225 and Fugue Icons 227)

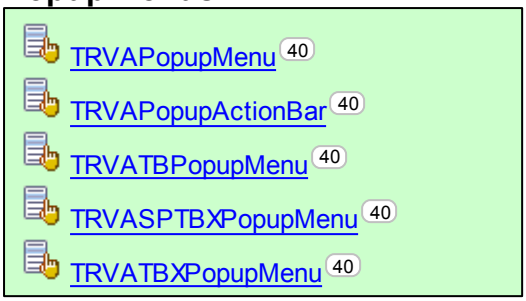
1.2 Components

Components

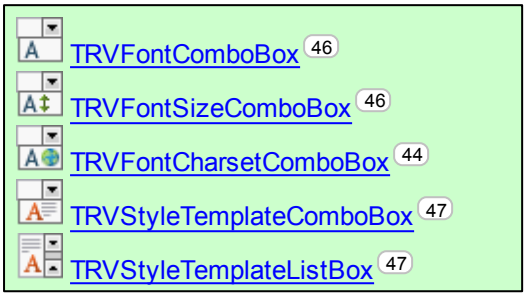
Settings

TRVAControlPanel 23

Popup Menus



Combo- and List-Boxes



1.2.1 TRVAControlPanel

TRVAControlPanel is the component allowing to change settings for RichViewActions

Unit RichViewActions;

Syntax

```
TRVAControlPanel = class(TComponent)
```

Hierarchy

TObject
TPersistent
TComponent

Description

This component contains properties affecting many RichViewActions.

Main control panel

If an application does not contain TRVAControlPanel on a form, a default control panel is used (accessible as MainRVAControlPanel 235). If an application contains a single TRVControlPanel on a form, it becomes the main control panel automatically (assigned to MainRVAControlPanel). Another control panel can be made a main one if you call its Activate 36 method.

Linking

All actions and many components in RichViewActions have links to a control panel, including:

- TrvCustomAction 206.ControlPanel 207
- TRVStyleTemplateCombo/ListBox 47.ControlPanel 48

If this link is not assigned (*nil*), the action/component uses the main control panel (see above).

Properties

The component has the following group of properties:

- links to components providing additional functionality:

  - ColorDialog 25
  - IdHTTP 29
  - ClHTTP 25
  - RVAddictSpell3 30
  - RVAddictThesaurus3 30
  - RVHTMLImporter 31
  - RVHTMLViewImporter 31
  - RVPrint 31
  - RVXML 32
- properties for customizing user interface:
  - Language 29
  - AddColorNameToHints 25
  - UserInterface 34
  - UseXPThemes 35
  - DialogFontName 28, DialogFontSize 28
  - UnitsDisplay 33, PixelBorders 29
  - UseHelpFiles 34
- properties of new documents:
  - DefaultColor 26
  - DefaultCustomFilterIndex 26
  - DefaultExt 27
  - DefaultFileFormat 27
  - DefaultFileName 27
  - DefaultMargin 27
  - DefaultDocParameters 26
- properties controlling all actions:
  - DefaultControl 26
  - ActionsEnabled 25
- properties for changing names of file formats:
  - RVFFilter 30
  - RVFLocalizable 30
  - RVFormatTitle 31
  - RVStylesFilter 32
  - RVStylesExt 32
  - XMLFilter 35
  - XMLLocalizable 35
- other properties:
  - AutoDeleteUnusedStyles 25
  - FirstPageNumber 28
  - SearchScope 32
  - ShowSoftPageBreaks 32
  - UseTextCodePageDialog 34
  - MetafileCompatibility 29

Events

If you use TRVRuler, process OnMarginsChanged 39.

If you want to implement opening, saving, exporting and inserting files in additional formats, use OnCustomFileOperation 37.

1.2.1.1 Properties

1.2.1.1.1 TRVAControlPanel.ActionsEnabled

Allows to disable all actions.

```
property ActionsEnabled: Boolean;
```

Set to *False* to disable all actions inherited from TrvAction[195], linked to this control panel.

Default value:

True

1.2.1.1.2 TRVAControlPanel.AddColorNameToHints

Allowing adding color names to hints of actions inherited from TrvActionCustomColor[197].

```
property AddColorNameToHints: Boolean;
```

A color name can be added to the action's hint, if its UserInterface[198]=*rvacNone* (i.e. the action applies a specific color without displaying a dialog).

Default value:

True

1.2.1.1.3 TRVAControlPanel.AutoDeleteUnusedStyles

Instructs to delete unused styles from documents when it's possible.

```
property AutoDeleteUnusedStyles: Boolean;
```

If *True*, all unused text, paragraph and list styles are deleted after executing TrvActionNew[56] and TrvActionOpen[59].

It's highly recommended to leave it equal to *True*, otherwise a number of unused styles will be constantly increased (as well as sizes of documents).

If style templates are used[17], this property is ignored: instead of deleting unused styles, "New" and "Open" actions resets styles according to StyleTemplates[58].

Default value:

True

1.2.1.1.4 TRVAControlPanel.ClHTTP

Points to TclHTTP component used for downloading external pictures referenced from HTML and RTF files.

```
property ClHTTP: TclHTTP;
```

This property is available only if CleverComponents[221] are used.

The following actions use this component when inserting or loading HTML and RTF files: TrvActionOpen[59], TrvActionInsertFile[136], TrvActionPaste[77], TrvActionPasteSpecial[78]. They do it by assigning a temporal handler for TCustomRichViewEdit.OnImportPicture event. This component allows downloading images from remote HTTP servers.

1.2.1.1.5 TRVAControlPanel.ColorDialog

Specifies a color dialog for using in RichViewActions.

```
property ColorDialog: TColorDialog;
```

This is a link to TColorDialog component. If it is not defined, actions create and use temporal color dialogs.

If the link is defined, the user can define a set of custom colors and use them in all actions. Otherwise, this link

if optional.

1.2.1.1.6 TRVAControlPanel.DefaultColor

Specifies the default background color.

```
property DefaultColor: TColor;
```

This property is assigned to TCustomRichViewEdit.Color property when executing TrvActionNew[56] or TrvActionOpen[59] (before loading).

See also:

- DefaultMargin[27]
- DefaultDocParameters[26]
- TrvActionNew[57].StyleTemplates[58]

1.2.1.1.7 TRVAControlPanel.DefaultControl

Specifies the editor component for using in RichViewActions.

```
property DefaultControl: TCustomRVControl;
```

If this link is defined, all actions inherited from TrvAction[195] (and linked with this control panel) work with this editor. This property can be overridden by Control[195] property of the specific action.

Controls of the following types can be assigned to this property:

- TRichViewEdit
- TDBRichViewEdit
- TSRichViewEdit (ScaleRichView)
- TDBSRichViewEdit (ScaleRichView)

If this link (and action.Control[195] property) is not assigned, the action works with the editor component having the input focus, or, if an editor component is not focused, with the first found editor component.

1.2.1.1.8 TRVAControlPanel.DefaultCustomFilterIndex

Specifies the default index of a custom format for new documents.

```
property DefaultCustomFilterIndex: Integer;
```

This property is used if DefaultFileFormat[27]=*rffeCustom*. This is an index of custom format (from 1); custom formats are listed in TrvActionSaveAs.CustomFilter[54] property.

This is a temporal format assigned to new documents after executing TrvActionNew[56]. This format will be offered as a default format on the first execution of TrvActionSave[69] or TrvActionSaveAs[73] for new documents.

OnCustomFileOperation[37] occurs when files of this format need to be saved.

Default value:

1

1.2.1.1.9 TRVAControlPanel.DefaultDocParameters

Specifies default page parameters for new documents (page size, margins, etc.)

```
property DefaultDocParameters: TRVDocParameters;
```

If *rvfoSaveDocProperties* is included in RVFOptions properties of TRichViewEdit control, this property is assigned to its DocParameters property in the following cases:

- when TrvActionNew[56] is executed;
- before TrvActionOpen[59] loads a file.

ZoomPercent and ZoomMode sub-properties are not assigned.

Units sub-property is not assigned too. All sizes are converted to RichViewEdit.DocParameters.Units.

ScaleRichView note: this property is especially important if RichViewActions are used with TSRichViewEdit control (ScaleRichView). DefaultDocParameters define a page layout for new documents, and DefaultMargin 27 property is ignored.

See also:

- DefaultColor 26
- TrvActionNew 57.StyleTemplates 58

1.2.1.1.10 TRVAControlPanel.DefaultExt

Specifies a default file extension for dialogs in TrvActionOpen 59, TrvActionSaveAs 73, and TrvActionExport 62 actions.

```
property DefaultExt: String;
```

This value is assigned to DefaultExt property of TOpenDialog and TSaveDialog components used in these actions.

Default value:

'rvf'

1.2.1.1.11 TRVAControlPanel.DefaultFileFormat

Specifies the default format for new documents.

```
property DefaultFileFormat: TrvFileSaveFilter 233;
```

If it equals to *ffeCustom*, DefaultCustomFilterIndex 26 property defines a subformat.

This is a temporal format assigned to new documents after executing TrvActionNew 56. This format will be offered as a default format on the first execution of TrvActionSave 69 or TrvActionSaveAs 73 for new documents.

Default value:

ffeRVF

See also:

- DefaultExt 27

1.2.1.1.12 TRVAControlPanel.DefaultFileName

Specifies a default file name for new documents.

```
property DefaultFileName: String;
```

This is a temporal file name assigned to new documents after executing TrvActionNew 56. This file name will be offered as a default file name on the first execution of TrvActionSave 69 or TrvActionSaveAs 73 for new documents.

Default value:

'Untitled.rvf'

See also:

- DefaultExt 27

1.2.1.1.13 TRVAControlPanel.DefaultMargin

Specifies default margins for new documents.

```
property DefaultMargin: Integer;
```

This property is assigned to LeftMargin, TopMargin, RightMargin, and BottomMargin properties of

TCustomRichViewEdit component in the following cases:

- when TrvActionNew 56 is executed;
- before TrvActionOpen 59 loads a file.

ScaleRichView note: for TSRichViewEdit , this property is ignored, see DefaultDocParameters 26.

Default value:

5

See also:

- DefaultColor 26
- DefaultDocParameters 26
- TrvActionNew 57.StyleTemplates 58

1.2.1.1.14 TRVAControlPanel.DialogFontName

Defines the default font name for all dialogs displayed by RichViewActions.

```
property DialogFontName: TFontName;
```

Default value:

'Tahoma'

See also:

- DialogFontSize 28

1.2.1.1.15 TRVAControlPanel.DialogFontSize

Defines the default font size for all dialogs displayed by RichViewActions.

```
property DialogFontSize: Integer;
```

Default value:

8

See also:

- DialogFontName 28

1.2.1.1.16 TRVAControlPanel.FirstPageNumber

Defines the page number for the first page.

```
property FirstPageNumber: Integer;
```

If page numbers are printed, this value specifies the number printed on the first page.

Default value:

1

See also properties:

- Header, Footer 28

1.2.1.1.17 TRVAControlPanel.Header, Footer

The properties define the page headers and footers.

```
property Header: TRVAHFInfo 233;
property Footer: TRVAHFInfo 233;
```

You can change properties of a header or a footer by assigning subproperties.

TrvActionPageSetup 64 changes these properties as well.

The following actions print headers and footers:

- TrvActionPrint 65
- TrvActionQuickPrint 68
- TrvActionPrintPreview 66

ScaleRichView note: these properties are ignored when printing TSRichViewEdit or TDBSRichViewEdit controls.

Default value:

Header: defaults for TRVAHFInfo, but Text := '- &p -';
Footer: defaults for TRVAHFInfo

See also properties:

- FirstPageNumber 28

1.2.1.1.18 TRVAControlPanel.IdHTTP

Points to TidHTTP component used for downloading external pictures referenced from HTML and RTF files.

```
property IdHTTP: TIdHTTP;
```

This property is available only if Indy 220 is used.

The following actions use this component when inserting or loading HTML and RTF files: TrvActionOpen 59, TrvActionInsertFile 136, TrvActionPaste 77, TrvActionPasteSpecial 78. They do it by assigning a temporal handler for TCustomRichViewEdit.OnImportPicture event. This component allows downloading images from remote HTTP servers.

1.2.1.1.19 TRVAControlPanel.Language

Defines the language for user interface.

```
property Language: TRVALanguageName 216;
```

This property returns a language name in English.

You can assign either English or native language name to this property.

See Localization of RichViewActions 210.

1.2.1.1.20 TRVAControlPanel.MetafileCompatibility

Specifies whether the printing output is compatible with metafiles.

```
property MetafileCompatibility: Boolean;
```

This property is ignored when printing TRichViewEdit, because TrvActionPrint 65, TrvActionQuickPrint 68, and TrvActionPrintPreview 66 use RVPrint 31.MetafileCompatibility property instead.

This property is used when printing ScaleRichView (see TsrvActionPrint and TsrvActionQuickPrint actions in the ScaleRichView help file).

Default value:

False

1.2.1.1.21 TRVAControlPanel.PixelBorders

Specifies pixels as the units of measurement for widths of lines in RichViewActions dialogs.

```
property PixelBorders: Boolean;
```

If this property is *True*, widths of lines in RichViewActions dialogs are measured in pixels, even if UnitsDisplay 33 <>*rvuPixels*.

This property is used by the following actions:

- TrvActionParaBorder 116 for border widths, including an internal width;
- TrvActionTableProperties 187 for widths of a table and cells borders;

- TrvActionItemProperties 201 for widths of a table and cells borders; for width of horizontal lines.

Default value:

False

1.2.1.1.22 TRVAControlPanel.RVAddictSpell3

Defines a link to TRVAddictSpell3 component.

```
property RVAddictSpell3: TRVAddictSpell3;
```

This property is available only if Addict 3 or 4 219 is used.

This component is used in TrvActionAddictSpell3 194 action and menus 40.

1.2.1.1.23 TRVAControlPanel.RVAddictThesaurus3

Defines a link to TRVThesaurus3 component.

```
property RVAddictThesaurus3: TRVThesaurus3;
```

This property is available only if Addict 3 or 4 219 is used.

This component is used in TrvActionAddictThesaurus3 194 action.

1.2.1.1.24 TRVAControlPanel.RVFFilter

Defines the file filter for RVF format.

```
property RVFFilter: String;
```

This value is used only if RVFLocalizable 30=*False*. This string will appear in file filters in dialogs used in TrvActionOpen 59, TrvActionSaveAs 73, TrvActionExport 62, TrvActionInsertFile 136, TrvActionStyleTemplates 129 (style import) actions.

If RVFLocalizable 30=*True*, this property is ignored (RVF file filter name is assigned automatically when a new language is applied).

Default value:

'RichView Files (*.rvf)|*.rvf'

See also properties:

- RVStylesFilter 32
- DefaultExt 27

1.2.1.1.25 TRVAControlPanel.RVFLocalizable

Specifies whether UI text related to RVF format is maintained automatically by the localization procedures.

```
property RVFLocalizable: Boolean;
```

If *True*, all text related to RVF format is maintained automatically (i.e. localized text is used when UI language is changed).

If *False*, RVFFilter 30, RVFormatTitle 31, RVStylesFilter, RVStylesExt 32 properties are used.

Default value:

True

See also:

- XMLLocalizable 35
- DefaultExt 27

1.2.1.1.26 TRVAControlPanel.RVFormatTitle

Specifies how RVF (RichView Format) appears in the "Paste Special" dialog.

```
property RVFormatTitle: String;
```

This value is used only if RVFLocalizable[30]=*False*. This string will appear in the dialog used in TrvActionPasteSpecial[78] action and in RVA_GetProgressMessage[215] function.

If RVFLocalizable[30]=*True*, this property is ignored (RVF title is assigned automatically when a new language is applied).

Default value:

'RichView Format'

1.2.1.1.27 TRVAControlPanel.RVHTMLImporter

Defines a link to a TRVHTMLImporter component.

```
property RVHTMLImporter: TRvHtmlImporter;
```

This property is available only if RvHtmlImporter[221] is used.

If this link is defined, HTML format can be used in TrvActionInsertFile[136], TrvActionPaste[77], and TrvActionPasteSpecial[78] actions.

If both this property and RVHTMLViewImporter[31] property are assigned, RVHTMLViewImporter[31] has a higher priority for TrvActionInsertFile[136] action.

1.2.1.1.28 TRVAControlPanel.RVHTMLView Importer

Defines a link to a TRVHTMLViewImporter component.

```
property RVHTMLViewImporter: TRVHTMLViewImporter;
```

This property is available only if RvHtmlViewImporter[222] is used.

This TRVHTMLViewImporter component must be, in its order, linked to (hidden) THTMLViewer component (via HTMLViewer property).

If this link is defined, HTML format can be used in TrvActionOpen[59], TrvActionSaveAs[73], and TrvActionInsertFile[136] actions.

If both this property and RVHTMLImporter[31] property are assigned, this property has a higher priority for TrvActionInsertFile[136] action.

1.2.1.1.29 TRVAControlPanel.RVPrint

Contains a link to TRVPrint component.

```
property RVPrint: TRVPrint;
```

This link (if defined) is used by TrvActionPrint[65], TrvActionQuickPrint[68], TrvActionPrintPreview[66], TrvActionPageSetup[64] actions.

For TrvActionPageSetup[64], this link *must* be defined, otherwise the action will be disabled (because this action reads/writes margins from/to properties of this component).

1.2.1.1.30 TRVAControlPanel.RVStylesFilter, RVStylesExt

RVStylesFilter defines the file filter for files containing style templates.

RVStylesExt defines the file extension for files containing style templates.

```
property RVStylesFilter: String;
property RVStylesExt: String;
```

These values are used only if RVFLocalizable 30 =*False*. They will appear in file filters in the style import and export dialogs used in TrvActionStyleTemplates 129, TrvActionAddStyleTemplate 131 actions.

If RVFLocalizable 30 =*True*, these properties are ignored (a file filter and extension are assigned automatically when a new language is applied).

Default values:

- RVStylesFilter: 'RichView Styles (*.rvst)|*.rvst'
- sRVStylesExtension: 'rvst'

See also properties:

- RVFFilter 30

1.2.1.1.31 TRVAControlPanel.RVXML

Defines a link to TRichViewXML component.

```
property RVXML: TRichViewXML;
```

This property is available only if RichViewXML 223 is used.

If this link is defined, RichView XML format can be used in TrvActionOpen 59, TrvActionSaveAs 73, TrvActionExport 62 and TrvActionInsertFile 136 actions.

1.2.1.1.32 TRVAControlPanel.SearchScope

Defines a search scope for TrvActionFind 76, TrvActionFindNext 77, and TrvActionReplace 81 actions.

```
type
  TRVASearchScope = (rvssFromCursor, rvssAskUser, rvssGlobal);

property SearchScope: TRVASearchScope;
```

Value	Meaning
rvssFromCursor	Searching from the caret position to the end/beginning of the document.
rvssAskUser	Searching from the caret position to the end/beginning of the document, then asking the user whether to continue from the beginning/end, then (if "yes") searching from the beginning/end.
rvssGlobal	Searching from the caret position to the end/beginning of the document, then searching from the beginning/end.

Default value:

rvssAskUser

1.2.1.1.33 TRVAControlPanel.ShowSoftPageBreaks

Specifies whether page breaks should be displayed in TCustomRichViewEdit component when possible.

```
property ShowSoftPageBreaks: Boolean;
```

Soft page breaks can be displayed after the document is formatted for printing (after executing TrvActionPrint 65, TrvActionQuickPrint 68, or TrvActionPrintPreview 66 actions) until it is changed.

Default value:

True

1.2.1.1.34 TRVAControlPanel.TableGridStyle

Defines the pen style for drawing grid lines in table.

```
property TableGridStyle: TPenStyle;
```

This property is used in TrvActionTableGrid[184] action.

Default value:

psDot

1.2.1.1.35 TRVAControlPanel.UnitsDisplay

Specifies units of measurement for using in RichViewActions dialogs.

```
property UnitsDisplay: TRVUnits;
```

These units are used for entering lengths in dialog windows. For pixels, only integer values are allowed. Values measured in other units can be fractional.

Small lengths, for example paragraph spacing, are measured either in pixels (if UnitsDisplay=*rvuPixels*) or in points (if UnitsDisplay has another value).

Line widths can be measured in pixels even if UnitsDisplay<>*rvuPixels*, if PixelBorders[29]=*True*.

Default value:

rvuPixels

1.2.1.1.36 TRVAControlPanel.UnitsProgram

Specifies units of measurement for properties of TRVStyleLength type of some actions.

```
property UnitsProgram: TRVStyleUnits;
```

These units are used for properties of the type TRVStyleLength of the following actions:

- TrvActionNew[57],
- TrvActionIndentInc[112],
- TrvActionIndentDec[112],
- TrvActionParaList[125],
- TrvActionParaBullets[117],
- TrvActionParaNumbering[127],
- TrvActionStyleTemplates[129],
- TrvActionInsertPicture[150],
- TrvActionInsertHLine[137],
- TrvActionInsertTable[163].

Not all actions' properties of the type TRVStyleLength are measured in these units. Generally, if the action takes values from the editor, modifies them and applies back to the editor, properties of this action are measured in Editor.Style.Units (where Editor is the target TRichViewEdit control). However, if the action applies some predefined values to the editor, these values are measured using this property.

At design time, values of all action's properties measured in these units are displayed with units ('px' or 'tw') in the Object Inspector (in Delphi 5 or newer). Values of properties measured in Editor.Style.Units are not displayed with units.

Note: assigning a new value to this property does not convert values of actions' properties. To convert properties to new units of measurement, RVA_ConvertToPixels/RVA_ConvertToTwips[229] must be called before changing value of this property.

Default value:

rvstuPixels

1.2.1.1.37 TRVAControlPanel.UseHelpFiles

Specifies whether help files should be used if available.

```
property UseHelpFiles: Boolean;
```

Help files can be provided together with translations. A help file name can be specified as the last (optional) parameter of RVA_RegisterLanguage procedure (see RVAL_*.pas files).

If a help file is available, additional "Help" button is shown in all RichViewActions dialogs.

Current version of RichViewActions does not include any help files.

Default value:

True

1.2.1.1.38 TRVAControlPanel.UserInterface

Allows hiding UI options incompatible with some formats.

```
type
  TRVAUserInterface = (rvauiFull, rvauiHTML, rvauiRTF, rvauiText);

property UserInterface: TRVAUserInterface;
```

Value	Meaning
rvauiFull	All user interface controls are available. Optimal for RVF documents.
rvauiHTML	All user interface controls providing options incompatible with HTML are hidden. The most noticeable changes: • an alternative HTML-style dialog in TrvActionParaList[125] action; • a hidden "Tabs" page in the TrvActionParagraph[120] action's dialog.
rvauiRTF	All user interface controls providing options incompatible with HTML are hidden. The most noticeable changes: • TrvActionColor[197], TrvActionVAlign[205], TrvActionBackground[195], TrvActionTableCellRotation180[177] are disabled; • in the table properties dialog, table background is disabled, some options for table borders are not available; • in the image properties dialog, image alignment is not available; • in the paragraph borders dialog, some options are disabled
rvauiText	All user interface controls providing options incompatible with a plain text are hidden or disabled.

Default value:

rvauiFull

1.2.1.1.39 TRVAControlPanel.UseTextCodePageDialog

Specifies whether the file-related actions can display a dialog for choosing a text file codepage.

```
property UseTextCodePageDialog: Boolean;
```

This property affects the following actions:

- TrvActionOpen[59]
- TrvActionExport[62]
- TrvActionSave[69]
- TrvActionInsertFile[136]

It affects loading and saving text files in the editor containing Unicode text.

If *False*, non-Unicode text files are saved using the system-default code page; a code page for insertion is taken from the Charset of the current text.

If *True*, a dialog for choosing a code page is displayed (the saving and exporting actions display this dialog only if the document is completely in Unicode). In addition to ANSI code pages, it allows choosing UTF-8 and UTF-16.

Default value:

True

1.2.1.1.40 TRVAControlPanel.UseXPThemes

Specifies whether controls on RichViewActions dialog should use Windows XP/Vista/7 themes (visual styles) if available.

```
property UseXPThemes: Boolean;
```

This property affects the following controls placed on dialogs:

- TRichView
- TRVPrintPreview
- TRVOfficeRadioButton
- TRVOfficeRadioGroup
- TRVColorCombo
- TRVColorGrid
- color picker window used in TrvActionCustomColor 197

Other controls use themes if it is supported by Delphi (and a manifest is included). If you use Delphi 4-5, you can use a free ***TThemeManager*** component. Newer versions of Delphi have a built-in support of themes.

Default value:

True

1.2.1.1.41 TRVAControlPanel.XMLFilter

Defines the file filter for RichView XML format.

```
property XMLFilter: String;
```

This property is available only if RichViewXML 223 is used.

This value is used only if XMLLocalizable 35=*False*. This string will appear in file filters in dialogs used in TrvActionOpen 59, TrvActionSaveAs 73, TrvActionExport 62, TrvActionInsertFile 136 actions.

If XMLLocalizable 35=*True*, this property is ignored (XML file filter name is assigned automatically when a new language is applied).

Default value:

'XML Files (*.xml)|*.xml'

1.2.1.1.42 TRVAControlPanel.XMLLocalizable

Specifies whether UI text related to RichView XML format is maintained automatically by the localization procedures.

```
property RVFLocalizable: Boolean;
```

This property is available only if RichViewXML 223 is used.

If *True*, all text related to RichView XML format is maintained automatically (i.e. localized text is used when UI language is changed).

If *False*, XMLFilter 35 property is used.

Default value:

True

See also:

- RVFLocalizable 30

1.2.1.2 Methods

1.2.1.2.1 TRVAControlPanel.Activate

Makes this control panel the main control panel in the application.

```
procedure Activate;
```

This method is useful if you have more than one control panel in an application. The first created control panel becomes the main control panel automatically.

The main control panel is used for actions and components that do not have a link to the specific control panel.

The main control panel is returned in MainRVAControlPanel 235 variable.

See also:

- TrvCustomAction.ControlPanel 207
- TRVStyleTemplateCombo/ListBox.ControlPanel 48

1.2.1.3 Events

1.2.1.3.1 TRVAControlPanel.OnAddStyle

Occurs after some action added a style as a result of an editing operation.

```
type
  TRVAddStyleEvent = procedure (Sender: TrvAction 195; StyleInfo: TCustomRVInfo) of object;

property OnAddStyle: TRVAddStyleEvent;
```

Actions may add text, paragraph and list styles as a result of editing operations.

Depending on the style type, **StyleInfo** can be of TFontInfo, TParaInfo, or TRVListInfo class.

If you need to perform additional operations on new styles, use this event.

See also:

- OnStyleNeeded 39
- TCustomRichView.OnAddStyle

1.2.1.3.2 TRVAControlPanel.OnBackgroundChange

Occurs after TrvActionBackground 195 or TrvActionColor 197 actions change background properties.

```
property OnBackgroundChange: TRVAEditEvent 232;
```

1.2.1.3.3 TRVAControlPanel.OnCreateTeForm

Occurs before any RichViewAction displays a dialog window.

```
type
  TRVCreateTeFormEvent = procedure (Sender: TForm; teForm: TTeForm) of object;

property OnCreateTeForm: TRVCreateTeFormEvent;
```

This event is available only if ThemeEngine 223 is used. It allows setting properties to the form before it is displayed.

1.2.1.3.4 TRVAControlPanel.OnCustomFileOperation

Occurs when the application needs to save or load a file in a custom format.

```
type
  TRVAFileOperation = (rvafoOpen, rvafoSave, rvafoExport, rvafoInsert);
  TRVCustomFileOperationEvent = procedure (Sender: TrvAction[195]; Edit: TCustomRichViewEdit
    const FileName: String; Operation: TRVAFileOperation;
    var SaveFormat: TrvFileSaveFilter[233];
    var CustomFilterIndex: Integer; var Success: Boolean) of object;

property OnCustomFileOperation: TRVCustomFileOperationEvent;
```

This event occurs when the following actions are executed:

- TrvActionOpen[59] (**Operation**=*rvafoOpen*)
- TrvActionSave[69] (**Operation**=*rvafoSave*)
- TrvActionExport[62] (**Operation**=*rvafoExport*)
- TrvActionInsertFile[136] (**Operation**=*rvafoInsertFile*)

FileName is a file name (full path) for opening/saving.

CustomFilterIndex is an index of custom format (in the CustomFilter property of the corresponding action), from 1.

For an opening operation, you can define a saving format in **SaveFormat** parameter (this format will be used for subsequent saving of document loaded from this file); you can also change **CustomFilterIndex** (on input, it corresponds to TrvActionOpen.CustomFilter[60]; on output, it must correspond to TrvActionSaveAs.CustomFilter[54], if **SaveFormat**=*ffeCustom*). For all other operations, **SaveFormat** and output value of **CustomFilterIndex** are ignored.

Set **Success** to *True* if you loaded/saved file successfully.

See also:

- TrvActionSave.SuppressNextErrorMessage[71]

You can use LoadCSV[230] function to implement an insertion of CSV files.

Example

By default, RichViewActions allows opening and saving HTML files only with rvHtmlViewImporter[222].

This example allows using rvHtmlImporter[221] for this. Use it at your own risk, because a quality of rvHtmlImporter is not guaranteed.

First, assign CustomFilter property of TrvActionOpen[59] and TrvActionSaveAs[73]. You can do it at design time in the Object Inspector. But if you want to make it localizable, do it at run time in

the localization procedure (in the ActionTest project, do it in the procedure TForm3.Localize):

```
  rvActionsResource.rvActionOpen1.CustomFilter :=  RVA_GetS[214](rvam_flt_HTMLOpen);
  rvActionsResource.rvActionSaveAs1.CustomFilter :=  RVA_GetS[214](rvam_flt_HTMLOpen);
```

Make sure that *ffiCustom* is in TrvActionOpen.Filter[61], and *ffeCustom* is in TrvActionSaveAs.Filter[74].

Next, process RVAContolPanel.OnCustomFileOperation event:

```
procedure TForm3.RVAControlPanel1CustomFileOperation(Sender: TrvAction[195];
  Edit: TCustomRichViewEdit; const FileName: String;
  Operation: TRVAFileOperation; var SaveFormat: TrvFileSaveFilter[233];
  var CustomFilterIndex: Integer; var Success: Boolean);
var Stream: TFileStream;
    s: TRVRawByteString;
begin
  case Operation of
    rvafoOpen:
      begin
```

```
        try
          // This example assumes that there is only one custom open format,
          // so it does not check CustomFilterIndex (it must be 1)
          Stream := TFileStream.Create(FileName, fmOpenRead);
          try
            SetLength(s, Stream.Size);
            Stream.ReadBuffer(PRVAnsiChar(s)^, Length(s));
          finally
            Stream.Free;
          end;
          RvHtmlImporter1.RichView := Edit;
          RvHtmlImporter1.LoadHtml(s);
          // This example assumes that there is only one custom save format,
          // so it does not assign CustomFilterIndex and SaveFormat
          // (they are already 1 and ffeCustom correspondingly)
          Success := True;
        except;
          Success := False;
        end;
      end;
    rvafoSave:
      begin
          // This example assumes that there is only one custom save format,
          // so it does not check CustomFilterIndex (it must be 1)
          Success := Edit.SaveHTML(FileName, rvActionsResource.rvActionExport1.FileTitle[63],
            rvActionsResource.rvActionExport1.ImagePrefix[63],
            rvActionsResource.rvActionExport1.SaveOptions[64]+[rvsoOverrideImages]);
      end;
  end;
end;
```

1.2.1.3.5 TRVAControlPanel.OnDownload

Occurs when IdHTTP[29] downloads an image.

```
type
  TRVADownloadEvent = procedure (Sender: TrvAction; const Source: String) of object;

property OnDownload: TRVADownloadEvent;
```

For each image, this event occurs twice:

- before downloading, **Source** is the image file name (HTTP location)
- after downloading, **Source** is an empty string.

See also:

how to display a localized text message about downloading[215]

1.2.1.3.6 TRVAControlPanel.OnGetActionControlCoords

The event requests coordinates of the control which called the given action.

```
type
  TRVGetActionControlCoordsEvent = procedure (Sender: TrvAction; var R: TRect) of object;
property OnGetActionControlCoords: TRVGetActionControlCoordsEvent;
```

This event may occur when one of the following actions needs to position a color-picker window relative to the control which called this action:

- TrvActionFontColor[86]
- TrvActionFontBackColor[85]

- TrvActionParaColor[118]
- TrvActionColor[197]

(if UserInterface[198]=*rvacAdvanced*)

This event occurs only when the component linked to the action is not a TControl (if a control executes the action, coordinates are calculated automatically).

Input parameters:

Sender – the action that is being executed.

R = Rect(0,0,0,0).

Output parameter:

R – screen coordinates of the control (if non-empty).

If this action is not processed, or if empty **R** is returned, a color-picker window will be displayed at the mouse pointer position.

Example:

This example shows how to work with TdxBarManager by ***Developer Express***:

```
procedure TMyForm.RVAControlPanel1GetActionControlCoords(
  Sender: TrvAction; var R: TRect);
begin
  if (Sender.ActionComponent<>nil) and
     (Sender.ActionComponent is TdxBarItem) then
     with TdxBarItem(Sender.ActionComponent).ClickItemLink do begin
       R := BarControl.GetItemRect (Control);
       R.TopLeft := BarControl.ClientToScreen (R.TopLeft);
       R.BottomRight := BarControl.ClientToScreen (R.BottomRight);
     end;
end;
```

1.2.1.3.7 TRVAControlPanel.OnMarginsChanged

Occurs after some action changes TCustomRichViewEdit control's margins (LeftMargin, TopMargin, RightMargin, and BottomMargin properties).

```
property OnMarginsChanged: TRVAEditEvent[232];
```

Margins can be changed by TrvActionOpen[59] and TrvActionBackground[195] actions.

If you use TRVRuler, you must update it in this event:

```
procedure TForm1.RVAControlPanel1MarginsChanged(Sender: TrvAction[195];
  Edit: TCustomRichViewEdit);
begin
  RVRuler1.UpdateRulerMargins;
end;
```

1.2.1.3.8 TRVAControlPanel.OnStyleNeeded

Occurs when some action needs a style for an editing operation.

```
type
  TRVStyleNeededEvent = procedure (Sender: TrvAction[195]; RVStyle: TRVStyle;
    StyleInfo: TCustomRVInfo; var StyleNo: Integer) of object;

property OnStyleNeeded: TRVStyleNeededEvent;
```

Actions may need text, paragraph and list styles.

StyleInfo is a sample of the required style. **Sender** needs a style with properties like in **StyleInfo**.

Depending on the style type, **StyleInfo** can be of TFontInfo, TParaInfo, or TRVListInfo class.

You need to return **StyleNo**, the index of style in the proper collection (TextStyles, ParaStyles, ListStyles) of **RVStyle** component. If you return -1, the action will perform a default processing (reusing existing styles if possible, adding new styles otherwise).

This event is rarely used. You can use it, for example, to disallow adding new styles (always return an index of an existing style the most similar to **StyleInfo**). Or you can prevent some style attributes from appearing in document as a result of editing operation (by changing **StyleInfo** properties and returning -1 in **StyleNo**).

1.2.2 TRVAPopupMenu

TRVAPopupMenu encapsulates the properties, methods, and events of a pop-up menu for TCustomRichViewEdit.

Unit RichViewActions, RVAPopupActionBar;

Syntax, if TNT Controls[224] are not used:

```
TRVAPopupMenu = class(TPopupMenu, IRVAPopupMenu)
```

Syntax, if TNT Controls[224] are used:

```
TRVAPopupMenu = class(TTntPopupMenu, IRVAPopupMenu)
```

Syntax:

```
TRVATBPopupMenu = class (TTBPopupMenu, IRVAPopupMenu)
TRVATBXPopupMenu = class (TTBXPopupMenu, IRVAPopupMenu)
TRVASPTBXPopupMenu = class (TSPTBXPopupMenu, IRVAPopupMenu)
TRVAPopupActionBar = class (TPopupActionBar, IRVAPopupMenu)
```

Hierarchy

(Hierarchy of TRVAPopupMenu, if TNT Controls[224] are not used)

TObject
TPersistent
TComponent
TMenu
TPopupMenu

Description

Menus

There are several similar components representing pop-up menu:

- TRVAPopupMenu is inherited from the standard TPopupMenu, or, if TNT Controls[224] are used, from TTntPopupMenu.
- TRVAPopupActionBar is inherited from TPopupActionBar, available for Delphi 2006 or newer, defined in RVAPopupActionBar unit.
- TRVATBPopupMenu is available if Toolbar2000[224] is used.
- TRVATBXPopupMenu is available if TBX[224] is used.
- TRVASPTBXPopupMenu is available if SpTBXLib[223] is used.

Standard actions

This menu is automatically filled with RichViewActions from ActionList[41].

The following actions are always added (if available):

- TrvActionCut[75]
- TrvActionCopy[75]
- TrvActionPaste[77]
- TrvActionFontEx[87]

- TrvActionParagraph 120
- TrvActionParaList 125

The following action is added if an image or a *break* is at the position of the caret:

- TrvActionItemProperties 201

The following actions are added if there is a table with multicell selection:

- TrvActionTableInsertRowsAbove 185
- TrvActionTableInsertRowsBelow 186
- TrvActionTableInsertColLeft 184
- TrvActionTableInsertColRight 185

The following actions are added if the table row(s)/column(s) are selected completely:

- TrvActionTableDeleteRows 183
- TrvActionTableDeleteCols 182

The following action is added if there is a table without edited cell:

- TrvActionFillColor 200

The following action is added if there is a table with selection, and the selected cells can be merged:

- TrvActionTableMergeCells 186

Othewise, the following action is added:

- TrvActionTableSplitCells 193

The following action is added if the current item is a table:

- TrvActionTableProperties 187

Live spelling

This menu supports live spelling. If Addict 3 or 4 219 is used, the live spelling commands appear automatically (suggestions for the current misspelled word, "Add to Dictionary" and "Ignore" commands). For other spellcheckers, you need to process the events:

- OnLiveSpellAdd 42
- OnLiveSpellGetSuggestions 42
- OnLiveSpellIgnoreAll 43

Adding custom commands

This menu is recreated each time before it is displayed, before OnPopup event occur. All menu items are deleted and new items are added.

If you need adding another commands (or delete some existing commands), do it in OnPopup event.

1.2.2.1 Properties

1.2.2.1.1 TRVAPopupMenu.ActionList

Defines a link to TActionList or TActionManager component.

```
property ActionList: TCustomActionList;
```

The menu can be filled with actions only if this link is defined.

1.2.2.1.2 TRVAPopupMenu.MaxSuggestionsCount

Specifies the maximal possible count of suggestions displayed for changing the current misspelled word.

```
property MaxSuggestionsCount: Integer;
```

If this property is set to positive value, it limits the count of suggestions displayed in the menu.

Other suggestions returned by Addict 3 219 or OnLiveSpellGetSuggestions 42 event are ignored.

Default value:

0 (all suggestions are displayed)

1.2.2.2 Events

1.2.2.2.1 TRVAPopupMenu.OnLiveSpellAdd

Occurs when the user clicks "Add to Dictionary" command in the menu.

```
type
  TRVALiveSpellAddEvent = procedure (Sender: TRVAPopupMenu[40];
    Edit: TCustomRichViewEdit; const Word: TRVASpellString[233]; StyleNo: Integer) of object;
  TRVA2LiveSpellAddEvent = procedure (Sender: TObject;
    Edit: TCustomRichViewEdit; const Word: TRVASpellString[233]; StyleNo: Integer) of object;
```

For TRVAPopupMenu:

```
property OnLiveSpellAdd: TRVALiveSpellAddEvent;
```

For TRVATBPopupMenu, TRVATBXPopupMenu, TRVASPTBXPopupMenu:

```
property OnLiveSpellAdd: TRVA2LiveSpellAddEvent;
```

If TNT Controls[224] are used, the Word parameter's type is TRVUnicodeString.

This event is used for live spelling check.

Word is a word for adding to the custom dictionary.

StyleNo is an index of text style for this word in **Edit**.Style.TextStyles.

If Addict 3[219] is used, this event is not called, this command is processed automatically.

Otherwise, use this event for adding word to the user dictionary (this command appear in the menu only if this event is assigned).

1.2.2.2.2 TRVAPopupMenu.OnLiveSpellGetSuggestions

Occurs when the menu needs to add items containing suggestions for the current misspelled word.

```
type
  TRVALiveSpellGetSuggestionsEvent = procedure (Sender: TRVAPopupMenu[40];
    Edit: TCustomRichViewEdit; const Word: TRVASpellString[233]; StyleNo: Integer;
    Suggestions: TRVASpellStrings[233]) of object;
  TRVA2LiveSpellGetSuggestionsEvent = procedure (Sender: TObject;
    Edit: TCustomRichViewEdit; const Word: TRVASpellString[233]; StyleNo: Integer;
    Suggestions: TRVASpellStrings[233]) of object;
```

For TRVAPopupMenu:

```
property OnLiveSpellGetSuggestions: TRVALiveSpellGetSuggestionsEvent
```

For TRVATBPopupMenu, TRVATBXPopupMenu, TRVASPTBXPopupMenu:

```
property OnLiveSpellGetSuggestions: TRVA2LiveSpellGetSuggestionsEvent
```

This event is used for live spelling check.

Word is a misspelled word.

StyleNo is an index of text style for this word in **Edit**.Style.TextStyles.

Add suggestions to **Suggestions**.

If Addict 3[219] is used, this event is not called, this command is processed automatically.

Otherwise, use this event for generating suggestions.

See also:

- MaxSuggestionsCount[41]

1.2.2.2.3 TRVAPopupMenu.OnLiveSpellIgnoreAll

Occurs when the user clicks "Ignore All" command in the menu.

```
type
  TRVALiveSpellIgnoreAllEvent = procedure (Sender: TRVAPopupMenu[40];
    Edit: TCustomRichViewEdit; const Word: String; StyleNo: Integer) of object;
  TRVA2LiveSpellIgnoreAllEvent = procedure (Sender: TObject;
    Edit: TCustomRichViewEdit; const Word: String; StyleNo: Integer) of object;
```

For TRVAPopupMenu:

```
property OnLiveSpellIgnoreAll: TRVALiveSpellIgnoreAllEvent;
```

For TRVATBPopupMenu, TRVATBXPopupMenu, TRVASPTBXPopupMenu:

```
property OnLiveSpellIgnoreAll: TRVA2LiveSpellIgnoreAllEvent;
```

If TNT Controls[224] are used, the Word parameter's type is TRVUnicodeString.

This event is used for live spelling check.

Word is a word for adding to the ignore list.

StyleNo is an index of text style for this word in **Edit**.Style.TextStyles.

If Addict 3[219] is used, this event is not called, this command is processed automatically.

Otherwise, use this event for adding word to the ignore list (this command appear in the menu only if this event is assigned).

1.2.2.2.4 TRVAPopupMenu.OnLiveSpellWordReplace

Occurs before **Word** is changed to **Correction** in **Edit**.

```
type
  TRVALiveSpellReplaceEvent = procedure (Sender: TRVAPopupMenu;
    Edit: TCustomRichViewEdit; const Word, Correction: String; StyleNo: Integer) of objec
  TRVA2LiveSpellReplaceEvent = procedure (Sender: TObject;
    Edit: TCustomRichViewEdit; const Word, Correction: String; StyleNo: Integer) of objec
```

For TRVAPopupMenu:

```
property OnLiveSpellWordReplace: TRVALiveSpellReplaceEvent;
```

For TRVATBPopupMenu, TRVATBXPopupMenu, TRVASPTBXPopupMenu:

```
property OnLiveSpellWordReplace: TRVA2LiveSpellReplaceEvent;
```

If TNT Controls[224] are used, the Word and Corrections parameters' type is TRVUnicodeString.

This event is used for live spelling check. It occurs when the user clicks on one of suggestions for the misspelled word in the menu.

StyleNo is an index of text style for this word in **Edit**.Style.TextStyles.

Processing this event is not necessary. It may be used if a spellchecker can use a history of replacements to generate better suggestions in future.

1.2.3 TCustomRVFontComboBox

TCustomRVFontComboBox is the base class for font combo-boxes.

Unit RVFontCombos;

Syntax, if TNT Controls[224] are not used:

```
TCustomRVFontComboBox = class(TComboBox)
```

Syntax, if TNT Controls[224] are used:

```
TCustomRVFontComboBox = class(TTntComboBox)
```

Hierarchy

(Hierarchy of TCustomRVFontComboBox, if TNT Controls[224] are not used)

TObject
TPersistent
TComponent
TControl
TWinControl
TCustomComboBox
TComboBox

Description

This component is not used directly.

The following components are inherited from TCustomRVFontComboBox:

- TRVFontComboBox[46]
- TRVFontSizeComboBox[46]
- TRVFontCharsetComboBox[44]

1.2.3.1 Properties

1.2.3.1.1 TCustomRVFontComboBox.ActionFont

Links this component to TrvActionFontEx[87] action.

```
property ActionFont: TrvActionFontEx[87];
```

This action is required to apply the selection to Editor[44].

1.2.3.1.2 TCustomRVFontComboBox.Editor

Links this combo-box with an editor.

```
property Editor: TCustomRVControl;
```

Controls of the following types can be assigned to this property:

- TRichViewEdit
- TDBRichViewEdit
- TSRichViewEdit (ScaleRichView)
- TDBSRichViewEdit (ScaleRichView)

When the combo-box is linked with **Editor**:

- the combo-box is updated automatically according to changes in **Editor**
- the combo-box applies the user choice to the selected fragment in **Editor** (ActionFont[44] must be defined as well)

1.2.4 TRVFontCharsetComboBox

TRVFontCharsetComboBox is a combo-box for choosing a font character set.

Unit RVFontCombos;

```
TRVFontCharsetComboBox = class(TCustomRVFontComboBox[43])
```

Hierarchy

(Hierarchy of TRVFontCharsetComboBox, if TNT Controls[224] are not used)

TObject
TPersistent
TComponent
TControl

TWinControl
TCustomComboBox
TComboBox
TCustomRVFontComboBox 43

Description

This combo-box allows changing a font Charset for a selected text in Editor 44.

This component works automatically: place it on a form, assign Editor 44 and ActionFont 44 properties; no additional code is required.

Unicode note: for Unicode text, Charset does not affect appearance (except for SYMBOL_CHARSET).

In addition to character sets available for the current font, the combo-box may contain DEFAULT_CHARSET item, if AddDefaultCharset 45 = *True*.

See also:

- TRVFontComboBox 46
- TRVFontSizeComboBox 46

1.2.4.1 Properties

1.2.4.1.1 TRVFontCharsetComboBox.AddDefaultCharset

Specifies whether the combo-box must contain DEFAULT_CHARSET item.

```
property AddDefaultCharset: Boolean;
```

The caption for this item is defined in DefaultCharsetCaption 45 property.

Default value:

False

1.2.4.1.2 TRVFontCharsetComboBox.DefaultCharsetCaption

Specifies the caption for DEFAULT_CHARSET item.

```
property DefaultCharsetCaption: String;
```

This item is added if AddDefaultCharset 45 = *True*.

The recommended value for this property: RVA_GetS 214(*rvam_font_DefaultCharset*).

Default value:

'(Default)'

1.2.4.1.3 TRVFontCharsetComboBox.FontName

Contains a font name.

```
property FontName: TFontName;
```

This property may be used when the combo-box is not linked with Editor 44. If linked, the combo-box items are maintained automatically.

When this property is assigned, the combo-box displays character sets available for this font.

1.2.5 TRVFontComboBox

TRVFontComboBox is a combo-box for choosing a font name.

Unit RVFontCombos;

```
TRVFontComboBox = class(TCustomRVFontComboBox 43)
```

Hierarchy

(Hierarchy of TRVFontComboBox, if TNT Controls 224 are not used)

TObject
TPersistent
TComponent
TControl
TWinControl
TCustomComboBox
TComboBox
TCustomRVFontComboBox 43

Description

This combo-box allows changing a font name for a selected text in Editor 44.

This component works automatically: place it on a form, assign Editor 44 and ActionFont 44 properties; no additional code is required.

See also:

- TRVFontSizeComboBox 46
- TRVFontCharsetComboBox 44

1.2.6 TRVFontSizeComboBox

TRVFontSizeComboBox is a combo-box for choosing a font size (with the precision up to 0.5 point).

Unit RVFontCombos;

```
TRVFontSizeComboBox = class(TCustomRVFontComboBox 43)
```

Hierarchy

(Hierarchy of TRVFontSizeComboBox, if TNT Controls 224 are not used)

TObject
TPersistent
TComponent
TControl
TWinControl
TCustomComboBox
TComboBox
TCustomRVFontComboBox 43

Description

This combo-box allows changing a font size for a selected text in Editor 44.

This component works automatically: place it on a form, assign Editor 44 and ActionFont 44 properties; no additional code is required.

See also:

- TRVFontComboBox 46
- TRVFontCharsetComboBox 44

1.2.6.1 Properties

1.2.6.1.1 TRVFontSizeComboBox.FontName

Contains a font name.

```
property FontName: TFontName;
```

This property may be used when the combo-box is not linked with Editor[44]. If linked, the combo-box items are maintained automatically.

When this property is assigned, the combo-box displays font sizes available for this font. For True Type and Open Type fonts, it displays a predefined set of font sizes (8, 9, 10, 11, 12, 14, 16, 18, 20, 22, 24, 26, 28, 36, 48, 72).

1.2.7 TRVStyleTemplateComboBox, TRVStyleTemplateListBox

TRVStyleTemplateComboBox is a combo-box for choosing a style template.

TRVStyleTemplateListBox is a list-box for choosing a style template.

Unit RVStyleFuncs;

Syntax, if TNT Controls[224] are not used:

```
TRVStyleTemplateComboBox = class(TCustomComboBox)
TRVStyleTemplateListBox = class(TCustomListBox)
```

Syntax, if TNT Controls[224] are used:

```
TRVStyleTemplateComboBox = class(TTntCustomComboBox)
TRVStyleTemplateListBox = class(TTntCustomListBox)
```

Hierarchy

(Hierarchy of TRVStyleTemplateComboBox/TRVStyleTemplateListBox, if TNT Controls[224] are not used:)

TObject
TPersistent
TComponent
TControl
TWinControl
TCustomComboBox
TCustomComboBox/TCustomListBox

Description

These component allow applying style templates to Editor[48]. The components require Editor[48] .UseStyleTemplates=*True*, otherwise their Items are empty.

The components work automatically: place them on a form, assign Editor[44] property; no additional code is required, unless you want to process "All Styles" command in OnClickAllStyles[50] event.

By default, the components show only style templates in use, "quick access" style templates, and heading[49] style templates. You can display all style templates by assigning ShowAllStyles[49]=*True*.

If ShowClearFormat[49]=*True*, the components display "Clear Format" command.

You can display icons in items by assigning Images[48] and image indices[48].

When the application language is changed, call Localize[49].

1.2.7.1 Properties

1.2.7.1.1 TRVStyleTemplateComboBox.*ImageIndex

The properties contain indexes in Images [48].

```
property AllStylesImageIndex: Integer;
property ClearFormatImageIndex: Integer;
property TextStyleImageIndex: Integer;
property ParaStyleImageIndex: Integer;
property ParaTextStyleImageIndex: Integer;
```

ClearFormatImageIndex – index of an image for "Clear Format" item, shown if ShowClearFormat [49]=*True*.

AllStylesImageIndex – index of an image for "All Styles" item, shown if ShowAllStyles [49]=*False*, and not all styles are displayed.

TextStyleImageIndex – index of an image for style templates having Kind=*rvstkText*

ParaStyleImageIndex – index of an image for style templates having Kind=*rvstkPara*

ParaTextStyleImageIndex – index of an image for style templates having Kind=*rvstkParaText*

Default value:

-1

1.2.7.1.2 TRVStyleTemplateComboBox.ControlPanel

Links this component with a control panel.

```
property ControlPanel: TComponent;
```

You can assign a TRVAControlPanel [23] component to this property. If this property is not assigned (*nil*), the component uses MainRVAControlPanel [235].

ControlPanel is used to display localized captions according to its Language [29].

1.2.7.1.3 TRVStyleTemplateComboBox.Editor

Links this combo-box with an editor.

```
property Editor: TCustomRVControl;
```

Controls of the following types can be assigned to this property:

- TRichViewEdit
- TDBRichViewEdit
- TSRichViewEdit (ScaleRichView)
- TDBSRichViewEdit (ScaleRichView)

When the combo-box is linked with **Editor**:

- the combo-box is updated automatically according to changes in **Editor**
- the combo-box applies the user choice to the selected fragment in **Editor**

1.2.7.1.4 TRVStyleTemplateComboBox.Images

A link to TImageList component containing images for items.

```
property Images: TCustomImageList;
```

See also:

- AllStylesImageIndex, ClearFormatImageIndex, ParaStyleImageIndex, ParaTextStyleImageIndex, TextStyleImageIndex [48]

1.2.7.1.5 TRVStyleTemplateComboBox.Show AllStyles

Specifies whether the component displays all available style templates.

```
property ShowAllStyles: Boolean;
```

If *True*, the component displays all style templates from Editor[48].Style.StyleTemplates.

If *False*, the component displays:

- style templates in use (i.e. style templates referred by Editor[48].Style.TextStyles and ParaStyles);
- style templates having QuickAccess=*True*
- "All styles" item (if not all style templates are displayed)

If SmartHeadings[49]=*True*, the component uses a special procedure to calculate a visibility of "heading 1".."heading 9" style templates.

When "All Styles" item is clicked:

- if OnClickAllStyles[50] is assigned, it is called
- otherwise, the combo-/list-box temporarily displays all style templates.

Default value:

False

See also:

- AllStylesImageIndex[48]

1.2.7.1.6 TRVStyleTemplateComboBox.Show ClearFormat

Shows/hides "Clear Format" item.

```
property ShowClearFormat: Boolean;
```

If *True*, "Clear Format" is added as the first item.

Default value:

True

See also:

- ClearFormatImageIndex[48]

1.2.7.1.7 TRVStyleTemplateComboBox.SmartHeadings

Specifies whether a special procedure should be used to calculate a visibility of heading style templates.

```
property SmartHeadings: Boolean;
```

If ShowAllStyles[49]=*False* and **SmartHeadings**=*True*, the component uses the following procedure to display heading style templates ("heading 1".."heading 9"):

- if headings are not used, or only "heading 1" is used, the component displays headings 1, 2, and 3
- otherwise, if headings up to level N are used (inclusive), the component displays headings from 1 to N+1

Default value:

True

1.2.7.2 Methods

1.2.7.2.1 TRVStyleTemplateComboBox.Localize

Rebuilds the content of the component.

```
procedure Localize;
```

Call this method when Language[29] of ControlPanel[48] is changed.

1.2.7.3 Events

1.2.7.3.1 TRVStyleTemplateComboBox.OnClickAllStyles

Occurs when "All Styles" item is clicked.

```
property OnClickAllStyles: TNotifyEvent;
```

See ShowAllStyles 49 for details.

1.3 Actions

Actions

Actions in the order as they appear in the menu of the ActionTest demo:

File

- TrvActionNew 56
- TrvActionOpen 59
- TrvActionSave 69
- TrvActionSaveAs 73
- TrvActionExport 62
- TrvActionPageSetup 64
- TrvActionPrintPreview 66
- TrvActionPrint 65

Edit

- TrvActionUndo 83
- TrvActionRedo 81
- TrvActionCut 75
- TrvActionCopy 75
- TrvActionPaste 77
- TrvActionPasteAsText 78
- TrvActionPasteSpecial 78
- TrvActionFind 76
- TrvActionFindNext 77
- TrvActionReplace 81
- TrvActionCharCase 74
- TrvActionInsertPageBreak 149
- TrvActionRemovePageBreak 204
- TrvActionSelectAll 83

Font

- TrvActionFonts 93
- TrvActionFontEx 87

TrvActionFontBold 86
TrvActionFontItalic 92
TrvActionFontUnderline 98
TrvActionFontStrikeout 96

TrvActionFontAllCaps 85
TrvActionFontOverline 93

TrvActionSubscript 99
TrvActionSuperscript 100

TrvActionFontShrink 94
TrvActionFontGrow 91

TrvActionFontShrinkOnePoint 96
TrvActionFontGrowOnePoint 92

TrvActionFontColor 86
TrvActionFontBackColor 85

Paragraph

TrvActionParagraph 120
TrvActionParaBorder 116

TrvActionAlignLeft 106
TrvActionAlignCenter 103
TrvActionAlignRight 107
TrvActionAlignJustify 105
TrvActionAlignDistribute 104

TrvActionParaList 125
TrvActionParaBullets 117
TrvActionParaNumbering 127

TrvActionWordWrap 129

TrvActionIndentInc 112
TrvActionIndentDec 112

TrvActionLineSpacing100 113
TrvActionLineSpacing150 114
TrvActionLineSpacing200 114

TrvActionClearLeft 108
TrvActionClearRight 109
TrvActionClearBoth 107
TrvActionClearNone 108

TrvActionParaColor 118

Format

TrvActionStyleTemplates 129
TrvActionAddStyleTemplate 131
TrvActionClearFormat 133
TrvActionClearTextFormat 134
TrvActionStyleInspector 132

TrvActionBackground 195
TrvActionColor 197

TrvActionFillColor 200

TrvActionHide 201
TrvActionRemoveHyperlinks 204

TrvActionItemProperties 201
TrvActionInsertCaption 134
TrvActionVAlign 205

Insert

TrvActionInsertFile 136
TrvActionInsertPicture 150
TrvActionInsertHLine 137
TrvActionInsertHyperlink 138
TrvActionInsertSymbol 153
TrvActionInsertNumber 147
TrvActionInsertTextBox 162
TrvActionInsertPageNumber 150
TrvActionInsertPageCount 149

TrvActionInsertFootnote 159
TrvActionInsertEndnote 159
TrvActionInsertSidenote 161
TrvActionEditNote 157

Table

TrvActionInsertTable 163
TrvActionTableInsertColLeft 184
TrvActionTableInsertColRight 185

TrvActionTableInsertRowsAbove 185
TrvActionTableInsertRowsBelow 186

TrvActionTableDeleteCols 182
TrvActionTableDeleteRows 183
TrvActionTableDeleteTable 183
TrvActionTableToText 193

TrvActionTableSelectTable 191
TrvActionTableSelectCols 190
TrvActionTableSelectRows 190
TrvActionTableSelectCell 189

TrvActionTableCellVAlignTop 182
TrvActionTableCellVAlignMiddle 181
TrvActionTableCellVAlignBottom 180
TrvActionTableCellVAlignDefault 180

TrvActionTableCellRotationNone 176
TrvActionTableCellRotation90 177
TrvActionTableCellRotation180 177
TrvActionTableCellRotation270 178

TrvActionTableCellLeftBorder 174
TrvActionTableCellTopBorder 179
TrvActionTableCellRightBorder 175
TrvActionTableCellBottomBorder 173
TrvActionTableCellAllBorders 172
TrvActionTableCellNoBorders 175

TrvActionTableSplit 192
TrvActionTableSplitCells 193
TrvActionTableMergeCells 186

TrvActionTableGrid 184

TrvActionTableSort 192
TrvActionTableProperties 187

Toolbar only

TrvActionQuickPrint 68
TrvActionShowSpecialCharacters 205

Not used in ActionTest

TrvActionTextLTR 101
TrvActionTextRTL 102

TrvActionParaLTR 126
TrvActionParaRTL 128

TrvActionAddictSpell3 194
TrvActionAddictThesaurus3 194

Customized

TrvActionEvent 199
TrvActionInsertText 155

1.3.1 File

1.3.1.1 TrvActionCustomFileIO

TrvActionCustomFileIO is a base class for "Save As", "Export" and "Insert File" actions.

Unit RichViewActions 50;

Syntax

```
TrvActionCustomFileIO = class(TrvActionCustomIO 54)
```

Hierarchy

TObject
TPersistent
TComponent
TBasicAction
TContainedAction
TCustomAction
TAction
TrvCustomAction 206
TrvAction 195
TrvActionCustomIO 54

Description

This action is not used directly.

Descendants:

- TrvActionSaveAs 73
- TrvActionExport 62
- TrvActionInsertFile 136

1.3.1.1.1 Properties

1.3.1.1.1.1 TrvActionCustomFileIO.CustomFilter

Specifies the file filter for custom file formats.

```
property CustomFilter: String;
```

This property allows opening, saving and inserting files in custom formats (in addition to the standard formats provided by RichViewActions).

Value of this property is appended to the Filter property of the file opening/saving dialog, if the action supports custom formats.

Default value:

" (empty string)

See also:

- TRVActionSaveAs.Filter 74
- TRVActionExport.Filter 63
- TRVActionInsertFile.Filter 137
- TrvActionOpen.CustomFilter 60
- TRVAControlPanel.OnCustomFileOperation 37

1.3.1.2 TrvActionCustomIO

TrvActionCustomIO is a base class for the actions displaying dialogs for opening or saving files.

Unit RichViewActions 50;

Syntax

```
TrvActionCustomIO = class(TrvAction 195)
```

Hierarchy

TObject
TPersistent
TComponent
TBasicAction
TContainedAction
TCustomAction
TAction
TrvCustomAction 206
TrvAction 195

Description

This action is not used directly.

Direct descendants:

- TrvActionCustomFileIO 54
- TrvActionInsertPicture 150

1.3.1.2.1 Properties

1.3.1.2.1.1 TrvActionCustomIO.AutoUpdateFileName

Specifies whether FileName 55 is automatically updated by the action.

```
property AutoUpdateFileName: Boolean;
```

If this property is *True*, FileName 55 property is updated when the action is successfully executed (the user chose a new file from dialog).

This property is used and published in the following actions:

- TrvActionExport 62
- TrvActionInsertFile 136

Default value:
True

1.3.1.2.1.2 TrvActionCustomIO.DialogTitle

Defines a custom title (caption) for the file dialog.

```
property DialogTitle: string;
```

If this property is empty, default (localized) dialog titles are used.

Default value:

" (empty string)

See also:

- TrvActionOpen.DialogTitle 60

1.3.1.2.1.3 TrvActionCustomIO.FileName

Specifies the default file name.

```
property FileName: String;
```

This property is assigned to FileName property of the file open/save dialog before showing it.

This property is used and public in the following actions:

- TrvActionExport 62
- TrvActionInsertFile 136
- TrvActionInsertPicture 150

Default value:
" (empty string)

See also properties:

- InitialDir 56
- AutoUpdateFileName 55

1.3.1.2.1.4 TrvActionCustomIO.InitialDir

Specifies the initial directory for the file dialog.

```
property InitialDir: String;
```

This property is used only if FileName 55 is empty.

It is assigned to InitialDir property of the file open/save dialog before showing it.

Default value:

" (empty string)

1.3.1.3 TrvActionCustomNew

TrvActionCustomNew is a base class for "New" and "Open" actions.

Unit RichViewActions 50;

Syntax

```
TrvActionCustomNew = class(TrvAction 195)
```

Hierarchy

TObject
TPersistent
TComponent
TBasicAction
TContainedAction
TCustomAction
TAction
TrvCustomAction 206
TrvAction 195

Description

This action is not used directly.

Descendants:

- TrvActionNew 57
- TrvActionOpen 59

1.3.1.3.1 Properties

1.3.1.3.1.1 TrvActionCustomNew .ActionSave

Links this action to TrvActionSave [69] action.

```
property ActionSave: TrvActionSave [69];
```

If this link is not defined, the first found TrvActionSave [69] action on the same form/datamodule is used. If no TrvActionSave [69] action found, an exception occurs on executing.

1.3.1.3.2 Events

1.3.1.3.2.1 TrvActionCustomNew .OnNew

Occurs when the action is executed.

```
property OnNew: TNotifyEvent;
```

This event occurs after the document is cleared.

This event is called by TrvActionNew [56].

TrvActionOpen [59] has this event as well, it may be called before the loading occurs. However, it calls this event only if it is not linked (either explicitly [60] or implicitly) with TrvActionNew. If linked, TrvActionOpen calls the event of TrvActionNew instead.

If style templates are not used [17], this is a good place to define default values for collections of styles. For example:

```
procedure TForm1.rvActionNew1New(Sender: TObject);
begin
  RVStyle1.ParaStyles.Clear;
  RVStyle1.ParaStyles.Add; // one default paragraph style

  RVStyle1.ListStyles.Clear; // no default list styles

  RVStyle1.TextStyles.Clear;
  with RVStyle1.TextStyles.Add do begin // one default text style
    FontName := 'Times New Roman';
    Size     := 12;
  end;
end;
```

If style templates are used, styles are reset automatically.

1.3.1.4 TrvActionNew

TrvActionNew is the action for "File | New" command.

Unit RichViewActions [50];

Syntax

```
TrvActionNew = class(TrvActionCustomNew [56])
```

Hierarchy

TObject
TPersistent
TComponent
TBasicAction
TContainedAction
TCustomAction
TAction
TrvCustomAction [206]

TrvAction 195
TrvActionCustomNew 56

Description

If you use this action, you must also use TrvActionSave 69 and TrvActionSaveAs 73 actions.

This action must be linked to TrvActionSave 69 action, because TrvActionSave 69 contains a list of all documents opened in TCustomRichViewEdit controls. This link may be defined explicitly (via ActionSave 57 property) or implicitly (this action finds and uses TrvActionSave 69 action placed on the same form/datamodule).

This action performs the following tasks, if style templates are not used 17:

1. If the document was modified, asks for saving and saves it (using TrvActionSave 69).
2. Clears the document (calls TCustomRichViewEdit.Clear); clears the background image; if *rvfoSaveDocProperties* is included in TCustomRichViewEdit.RVFOptions, clears its DocProperties and resets DocParameters properties.
3. Applies the default values specified in the properties of GetControlPanel 208: DefaultColor 26, DefaultMargin 27, DefaultDocParameters 26.
4. Calls OnNew 57 event.
5. Calls OnMarginsChanged 39 event of GetControlPanel 208 component.
6. Formats the document (calls TCustomRichViewEdit.Format).
7. If GetControlPanel 208.AutoDeleteUnusedStyles 25=*True*, deletes unused styles (calls TCustomRichViewEdit.DeleteUnusedStyles(True, True, True)).
8. Stores the following values for this TCustomRichViewEdit using the properties of GetControlPanel 208: temporal file name = DefaultFileName 27, file format = DefaultFileFormat 27, custom filter index (used if file format is *ffeCustom*) = DefaultCustomFilterIndex 26. These values are stored in the list of documents maintained by TrvActionSave 69.
9. Calls OnDocumentFileChange 72 event of TrvActionSave 69 action.

If style templates are used 17:

The sequence of operations is the same, but the step #7 is different. The action resets text and paragraph styles according to StyleTemplates 58.

1.3.1.4.1 Properties

1.3.1.4.1.1 TrvActionNew .StyleTemplates

A collection of style templates for assigning to new documents.

```
property StyleTemplates: TRVStyleTemplateCollection;
```

This property is used:

- when TrvActionNew is executed
- when TrvActionOpen 59 (linked to this action) clears a target editor before loading a file.

Values in this collection are measured in GetControlPanel 208.UnitsProgram 33.

If style templates are used 17 (i.e. the target editor's UseStyleTemplates=*True*), the action:

1. deletes all items in TextStyles, ParaStyles, ListStyles;
2. adds one paragraph style; if StyleTemplates contain "Normal" style template, it is applied to this paragraph style;
3. adds one text style; if StyleTemplates contain "Normal" style template, this text style is formatted accordingly.

ScaleRichView note: for TSRichViewEdit, this the action performs similar operations with its header and footer. However, instead of "Normal", the action uses "header" style template for RVHeader, and "footer" style template for RVFooter properties.

Default value:

By default, the property contains the following style templates: "Normal", "heading 1", "heading 2", "heading 3", "Hyperlink", "header", "footer", "footnote reference", "endnote reference", "footnote text", "endnote text",

"caption".

"header" and "footer" are not used by TRichViewEdit (used only in ScaleRichView), so you can delete these items.

See also properties of GetControlPanel 208:

- DefaultColor 26
- DefaultMargin 27
- DefaultDocParameters 26

See also:

- TrvActionStyleTemplates 129.StandardStyleTemplates 130

1.3.1.4.2 Methods

1.3.1.4.2.1 TrvActionNew .Reset

Resets properties of rve.

```
procedure Reset(rve: TCustomRichViewEdit);
```

This method performs almost the same operation as this action when it is executed: it assigns default values to properties of **rve**; if style templates are used, it applies StyleTemplates 58.

This method:

- does not clear **rve**;
- does not delete unused styles (if style templates are not used, otherwise all old styles are deleted when StyleTemplates 58 are applied);
- (the most important) does not associate any file with **rve**.

Normally, you do not need to call this method, because all necessary operations are performed when the action is executed.

However, this method is useful if you work with TDBRichViewEdit or TDBSRichViewEdit. For db-controls, you cannot use file-based actions (TrvActionNew 57, TrvActionSave 69, TrvActionSaveAs 73, TrvActionOpen 59) directly. However, you can place TrvActionNew of a form, clear its Shortcut (to prevent execution on Ctrl+N), and call rvActionNew1.Reset(DBRichViewEdit1) in DBRichViewEdit1.OnNewDocument event.

1.3.1.5 TrvActionOpen

TrvActionOpen is the action for "File | Open..." command.

Unit RichViewActions 50;

Syntax

```
TrvActionOpen = class(TrvActionCustomNew 56)
```

Hierarchy

TObject
TPersistent
TComponent
TBasicAction
TContainedAction
TCustomAction
TAction
TrvCustomAction 206
TrvAction 195
TrvActionCustomNew 56

Description

The action displays a file opening dialog and loads the chosen file in TCustomRichViewEdit component. After opening, this editor is associated with the chosen file name and format, so subsequent executions of

TrvActionSave[69] action save content of this TCustomRichViewEdit component in this file in this format.

The following formats are supported:

- RichView Format (RVF)
- Rich Text Format (RTF)
- Text files (TXT, Unicode and ANSI)
- HTML (only if RvHtmlViewImporter[222] is used)
- RichViewXML Format (XML, only if RichViewXML[223] is used)
- custom formats, supported in GetControlPanel[208].OnCustomFileOperation[37] event.

If you use this action, you must also use TrvActionSave[69] and TrvActionSaveAs[73] actions.

This action must be linked to TrvActionSave[69] action, because TrvActionSave[69] contains a list of all documents opened in TCustomRichViewEdit controls. This link may be defined explicitly (via ActionSave[57] property) or implicitly (this action finds and uses TrvActionSave[69] action placed on the same form/datamodule).

This action may be linked to TrvActionNew[57] action. This link may be defined explicitly (via ActionNew[60] property) or implicitly (this action finds and uses TrvActionNew[57] action placed on the same form/datamodule). If linked, TrvActionNew[57] is used to reset editor before loading (it is especially important, if style templates are used[17]).

This action performs the following tasks:

1. If the document was modified, asks for saving and saves it (using TrvActionSave[69]).
2. Shows a file opening dialog allowing the user to choose the file for opening.
3. If the user chose a file, calls LoadFile[61] method. The file format is determined not by the file extension but by the filter index chosen in the file opening dialog.

1.3.1.5.1 Properties

1.3.1.5.1.1 TrvActionOpen.ActionNew

Links this action to TrvActionNew[56] action.

```
property ActionNew: TrvActionNew[56];
```

If this link is not defined, the first found TrvActionNew[56] action on the same form/datamodule is used.

1.3.1.5.1.2 TrvActionOpen.CustomFilter

Specifies the file filter for custom file formats.

```
property CustomFilter: String;
```

This property allows opening files in custom formats (in addition to the standard formats listed in the Filter[61] property of this action).

Value of this property is appended to the Filter property of the file opening dialog, if the Filter[61] property includes *ffiCustom*.

Default value:

" (empty string)

See also:

- GetControlPanel[208].OnCustomFileOperation[37]
- TrvActionCustomFileIO.CustomFilter[54]

1.3.1.5.1.3 TrvActionOpen.DialogTitle

Defines a custom title (caption) for the file opening dialog.

```
property DialogTitle: string;
```

If this property is empty, default (localized) dialog title is used.

Default value:

" (empty string)

See also:

- TrvActionCustomIO.DialogTitle 55

1.3.1.5.1.4 TrvActionOpen.Filter

Defines a set of file formats appearing in the file opening dialog.

```
property Filter: TrvFileOpenFilterSet 234;
```

If *ffiCustom* is included, formats listed in the CustomFilter 60 property are used.

Name and extension of RichView Format (RVF) may be customized, see GetControlPanel 208.RVFFilter 30.

Default value:

all formats

1.3.1.5.1.5 TrvActionOpen.InitialDir

Specifies the initial directory for the file dialog.

```
property InitialDir: String;
```

Value of this property is assigned to InitialDir property of the file opening dialog before showing it.

Default value:

" (empty string)

1.3.1.5.1.6 TrvActionOpen.LastFilterIndex

Defines file format selected in the file opening dialog by default.

```
property LastFilterIndex: Integer;
```

Value of this property is assigned to FilterIndex property of the file opening dialog before showing it.

Default value:

1

1.3.1.5.2 Methods

1.3.1.5.2.1 TrvActionOpen.LoadFile

Loads the specified file in **rve**.

```
procedure LoadFile(rve: TCustomRichViewEdit; const FileName: String;
  FileFormat: TrvFileOpenFilter 234; CustomFilterIndex: Integer=0);
```

Parameters:

rve – editor where to load file.

FileName – name of the file to load.

FileFormat – format of this file.

CustomFilterIndex used only if **FileFormat** is *ffiCustom.*

If **FileFormat** = *ffiCustom*, **CustomFilterIndex** identifies a custom file format. Custom formats are numbered from 1 in the order they are listed in the CustomFilter 60 property.

This procedure performs the following tasks:

1. Clears the document (calls TCustomRichViewEdit.Clear); clears the background image; if *rvfoSaveDocProperties* is included in TCustomRichViewEdit.RVFOptions, clears its DocProperties and resets DocParameters properties.
2. Applies the default values specified in the properties of GetControlPanel 208: DefaultColor 26, DefaultMargin

27.
3. Calls OnNew 57 event.
4. Loads the file in **rve**.
5. Associates **rve** with this file and file format (in the list of documents maintained by TrvActionSave 69).
6. Calls OnDocumentFileChange 72 event of TrvActionSave 69 action.
7. Calls OnMarginsChanged 39 event of GetControlPanel 208 component.
8. Formats the document (calls TCustomRichViewEdit.Format).
9. If GetControlPanel 208.AutoDeleteUnusedStyles 25=*True*, deletes unused styles (calls TCustomRichViewEdit.DeleteUnusedStyles(True, True, True)).
10. Calls OnOpenFile 62 event.

Subsequent executions of TrvActionSave 69 will save **rve** content in this file in this format.

1.3.1.5.3 Events

1.3.1.5.3.1 TrvActionOpen.OnOpenFile

Occurs when the action opened a file.

```
type
  TRVOpenFileEvent = procedure (Sender: TObject;
    Editor: TCustomRichViewEdit; const FileName: String;
    FileFormat: TrvFileOpenFilter 234; CustomFilterIndex: Integer) of object;

property OnOpenFile: TRVOpenFileEvent;
```

Parameters are the same as in LoadFile 61 method.

1.3.1.6 TrvActionExport

TrvActionExport is the action for "File | Export..." command.

Unit RichViewActions 50;

Syntax

```
TrvActionExport = class(TrvActionCustomFileIO 54)
```

Hierarchy

TObject
TPersistent
TComponent
TBasicAction
TContainedAction
TCustomAction
TAction
TrvCustomAction 206
TrvAction 195
TrvActionCustomIO 54
TrvActionCustomFileIO 54

Description

This action displays a file saving dialog and saves the content of TRichViewEdit component in the chosen file.

In addition to formats supported by TrvActionSaveAs 73 action, this action supports:

- HTML (even if rvHtmlViewImporter 222 is not used)
- simplified HTML (without CSS)
- formats supported by Microsoft Office file export converters.

After the execution, this TCustomRichViewEdit component is still associated with the file name and format assigned by the last execution of TrvActionNew 56, TrvActionOpen 59 or TrvActionSaveAs 73 action. TrvActionExport does not change this association.

This action performs the following tasks:

1. Shows a file saving dialog allowing the user to choose the file name and format for exporting.
2. If the user chose a file, calls ExportToFile 64 method. The file format is determined not by the file extension but by the filter index chosen in the file saving dialog.

1.3.1.6.1 Properties

1.3.1.6.1.1 TrvActionExport.CreateDirectoryForHTMLImages

Specifies whether a subdirectory for saving HTML images is created when exporting HTML file.

```
property CreateDirectoryForHTMLImages: Boolean;
```

This property is used when exporting to HTML formats (*ffeHTMLCSS or ffeHTML* 233).

If *True*, images from HTML files are saved in subdirectory having the same name as HTML file but '.files' extension.

If *False*, all images are saved in the same directory as the HTML file.

This option is used for images with undefined (empty) file names, or when *rvsoUseItemImageFileNames* is not included in SaveOptions 64 property.

Default value:

True

See also:

- TrvActionSave.CreateDirectoryForHTMLImages 69

1.3.1.6.1.2 TrvActionExport.FileTitle

Defines title for HTML files.

```
property FileTitle: String;
```

This property is used when exporting to HTML formats (*ffeHTMLCSS or ffeHTML* 233). This property is used as the Title parameter of TCustomRichViewEdit.SaveHTML and SaveHTMLEx methods.

See also:

- TrvActionSave.FileTitle 70

1.3.1.6.1.3 TrvActionExport.Filter

Defines a set of file formats appearing in the file saving dialog.

```
property Filter: TrvFileExportFilterSet 233;
```

If *ffeCustom* is included, formats listed in the CustomFilter 54 property are used.

Name and extension of RichView Format (RVF) may be customized, see GetControlPanel 208.RVFFilter 30.

Default value:

all formats

1.3.1.6.1.4 TrvActionExport.ImagePrefix

Prefix of image files saved together with HTML file.

```
property ImagePrefix: String;
```

This property is used when exporting to HTML formats (*ffeHTMLCSS or ffeHTML* 233). This property (probably with subdirectory name) is used as the ImagesPrefix parameter of TCustomRichViewEdit.SaveHTML and SaveHTMLEx methods.

This property is used for images with undefined (empty) file names, or when *rvsoUseItemImageFileNames* is not included in SaveOptions 64 property.

See also properties:

- CreateDirectoryForHTMLImages 63

See also:

- TrvActionSave.ImagePrefix 70

1.3.1.6.1.5 TrvActionExport.SaveOptions

Defines options for saving HTML files.

```
property SaveOptions: TRVSaveOptions;
```

This property is used when exporting to HTML formats (*ffeHTMLCSS or ffeHTML* 233). This property is used as the SaveOptions parameter of TCustomRichViewEdit.SaveHTML and SaveHTMLEx methods.

Default value:

[]

See also:

- TrvActionSave.SaveOptions 71

1.3.1.6.2 Methods

1.3.1.6.2.1 TrvActionExport.ExportToFile

Saves **Edit** in the specified file in the specified format.

```
function ExportToFile(Edit: TCustomRichViewEdit; const FileName: String;
  ExportFormat: TrvFileExportFilter 233; ConverterOrCustomIndex: Integer;
  rvc: TRVOfficeConverter): Boolean;
```

Parameters:

Edit – editor for saving.

FileName – file name for saving.

FileFormat – file format for saving.

CustomFilterIndex used only if **FileFormat** is *ffeCustom* or *ffeOfficeConverters*.

If **FileFormat** = *ffeCustom*, **CustomFilterIndex** identifies a custom file format. Custom formats are numbered from 1 in the order they are listed in the CustomFilter 54 property.

If **FileFormat** = *ffeOfficeConverters*, **CustomFilterIndex** identifies a format supported by Microsoft Office file export converters. Formats are numbered from 0. A list of these formats and their order are different on different computers. This is the index in **rvc**.ExportConverters collection.

If saving is unsuccessful, an error message is displayed.

Return value:

True if the saving was successful.

1.3.1.7 TrvActionPageSetup

TrvActionPageSetup is the action for "File | Page Setup" command.

Unit RichViewActions 50;

Syntax

```
TrvActionPageSetup = class(TrvCustomAction 206)
```

Hierarchy

TObject
TPersistent

TComponent
TBasicAction
TContainedAction
TCustomAction
TAction
TrvCustomAction 206

Description

This action displays a page setup dialog.

This action does not work with the specific editor. It defines global page properties for the application.

This action is enabled in GetControlPanel 208.RVPrint 31 is assigned, and there is at least one printer is installed in the system (Printer.Printers.Count>0).

The settings are stored:

- margins are stored in GetControlPanel 208.RVPrint 31 properties;
- headers and footers are stored in GetControlPanel 208.Header and Footer 28 properties;
- other properties (paper size, orientation, source) are stored in the application's printing settings (in the object returned by Printer function).

1.3.1.7.1 Properties

1.3.1.7.1.1 TrvActionPageSetup.MarginsUnits

Defines measuring units used for margins in the page setup dialog .

```
type
  TrvPaperMarginsUnits = (rvpmuMillimeters, rvpmuInches);

property MarginsUnits: TrvPaperMarginsUnits;
```

Internally, millimeters are used.

Default value:

rvpmuMillimeters

1.3.1.7.2 Events

1.3.1.7.2.1 TrvActionPageSetup.OnChange

Occurs after the changes made in the page setup dialog are applied.

```
property OnChange: TNotifyEvent;
```

1.3.1.8 TrvActionPrint

TrvActionPrint is the action for "File | Print..." command.

Unit RichViewActions 50;

Syntax

```
TrvActionPrint = class(TrvCustomPrintAction 74)
```

Hierarchy

TObject
TPersistent
TComponent
TBasicAction
TContainedAction
TCustomAction
TAction
TrvCustomAction 206
TrvAction 195
TrvCustomPrintAction 74

Description

This action displays a printing dialog (TPrintDialog) and prints the document from TCustomRichViewEdit component.

This action uses settings defined in TrvActionPageSetup 64. If GetControlPanel 208.RVPrint 31 is not assigned, the action uses the first TRVPrint component found on the form owning the TCustomRichViewEdit component. If not found, it creates a temporal TRVPrint with default property settings.

The action can print the whole document, the range of pages, or the selected fragment, depending on the user's choice.

This action is enabled only if at least one printer is installed in the system (Printer.Printers.Count>0).

See also:

- TrvActionQuickPrint 68
- TrvActionPrintPreview 66

1.3.1.8.1 Properties

1.3.1.8.1.1 TrvActionPrint.Title

Determines the text that is listed in the Print Manager.

```
property Title: string;
```

See also:

TrvActionQuickPrint.Title 68

1.3.1.9 TrvActionPrintPreview

TrvActionPrintPreview is the action for "File | Print Preview" command.

Unit RichViewActions 50;

Syntax

```
TrvActionPrintPreview = class(TrvCustomPrintAction 74)
```

Hierarchy

TObject
TPersistent
TComponent
TBasicAction
TContainedAction
TCustomAction
TAction
TrvCustomAction 206
TrvAction 195

TrvCustomPrintAction 74

Description

This action displays a printing preview window. This window has "Page Setup" and "Print" buttons where the user can print the document. For the "Page Setup" button, TrvActionPageSetup 64 action is used. If it is not linked (via ActionPageSetup 67 property), a temporal TrvActionPageSetup 64 action is used. For the "Print" button, TrvActionPrint 65 action is used. If it is not linked (via ActionPrint 67 property), the first TrvActionPrint 65 action on the same form/datamodule is used; if not found, printing in this action is not possible.

This action uses settings defined in TrvActionPageSetup 64. If GetControlPanel 208.RVPrint 31 is not assigned, the action uses the first TRVPrint component found on the form owning the TCustomRichViewEdit component. If not found, it creates a temporal TRVPrint with default property settings.

This action is enabled only if at least one printer is installed in the system (Printer.Printers.Count>0).

See also:

- TrvActionQuickPrint 68
- TrvActionPrint 65

1.3.1.9.1 Properties

1.3.1.9.1.1 TrvActionPrintPreview.ActionPageSetup

A link to TrvActionPageSetup 64 action.

```
property ActionPageSetup: TrvActionPageSetup 64;
```

This link is used for the "Page Setup" button on the print preview dialog.

If this link is not assigned, a temporal action is used.

1.3.1.9.1.2 TrvActionPrintPreview.ActionPrint

A link to TrvActionPrint 65 action.

```
property ActionPrint: TrvActionPrint 65;
```

This link is used for the "Print" button on the print preview dialog.

If this link is not assigned, the first found TrvActionPrint 65 action on the same form is used. If not found, printing is not possible.

1.3.1.9.1.3 TrvActionPrintPreview.Maximized

Defines the initial state of the print preview window.

```
property Maximized: Boolean;
```

Assign *True* to this property to display the print preview window maximized.

Default value:

False

1.3.1.9.2 Events

1.3.1.9.2.1 TrvActionPrintPreview .OnGetPreview FormClass

Allows defining your own window for a print preview.

```
type
  TRVGetFormClassEvent = procedure (Sender: TObject;
    var FormClass: TFormClass) of object;

property OnGetPreviewFormClass: TRVGetFormClassEvent;
```

You can return your own form class for a print preview window. This class must be inherited from TfrmRVPreview, otherwise an exception occurs.

1.3.1.10 TrvActionQuickPrint

TrvActionQuickPrint is the action for the "quick print" command (usually assigned to a toolbar button).

Unit RichViewActions 50;

Syntax

```
TrvActionQuickPrint = class(TrvCustomPrintAction 74)
```

Hierarchy

TObject
TPersistent
TComponent
TBasicAction
TContainedAction
TCustomAction
TAction
TrvCustomAction 206
TrvAction 195
TrvCustomPrintAction 74

Description

This action prints the document from TCustomRichViewEdit component without displaying a printing dialog.

This action uses settings defined in TrvActionPageSetup 64 action. If GetControlPanel 208.RVPrint 31 is not assigned, the action uses the first TRVPrint component found on the form owning the TCustomRichViewEdit component. If not found, it creates a temporal TRVPrint with default property settings.

This action is enabled only if at least one printer is installed in the system (Printer.Printers.Count>0).

See also:

- TrvActionPrint 65
- TrvActionPrintPreview 66

1.3.1.10.1 Properties

1.3.1.10.1.1 TrvActionQuickPrint.Title

Determines the text that is listed in the Print Manager.

```
property Title: string;
```

See also:

TrvActionPrint.Title 66

1.3.1.11 TrvActionSave

TrvActionSave is the action for "File | Save" command.

Unit RichViewActions[50];

Syntax

```
TrvActionSave = class(TrvAction[195])
```

Hierarchy

TObject
TPersistent
TComponent
TBasicAction
TContainedAction
TCustomAction
TAction
TrvCustomAction[206]
TrvAction[195]

Description

This action saves the document from TCustomRichViewEdit to a file. In addition, this action contains list of all documents (TCustomRichViewEdits associated with their file names and formats).

If the document has a temporal name (assigned after executing TrvActionNew[56] action), this action executes TrvActionSaveAs[73] action. So if you use this action, you must also use TrvActionSaveAs[73] action. The link between the actions may be defined explicitly (via ActionSaveAs[69] property) or implicitly (this action finds and uses TrvActionSaveAs[73] action placed on the same form/datamodule).

If the document has a permanent name and format (assigned after executing TrvActionOpen[59] or TrvActionSaveAs[73] action), the action just saves the file.

If the file cannot be saved, an error message is displayed.

CanCloseDoc[71] method should be used in Form.OnCloseQuery event.

1.3.1.11.1 Properties

1.3.1.11.1.1 TrvActionSave.ActionSaveAs

Links this action to TrvActionSaveAs[73] action.

```
property ActionSaveAs: TrvActionSaveAs[73];
```

TrvActionSaveAs[73] action is used when the document has a temporal name (after the execution of TrvActionNew[56] action). In this case, instead of saving, the action executes TrvActionSaveAs[73] action.

If this link is not defined, the first found TrvActionSaveAs[73] action on the same form/datamodule is used. If no TrvActionSaveAs[73] action found, an exception occurs on executing.

1.3.1.11.1.2 TrvActionSave.CreateDirectoryForHTMLImages

Specifies whether a subdirectory for saving HTML images is created when saving HTML file.

```
property CreateDirectoryForHTMLImages: Boolean;
```

This property is defined only if RvHTMLViewImporter[222] is used.

This property is used when saving to HTML format (*ffeHTMLCSS*[233]).

If *True*, images from HTML files are saved in subdirectory having the same name as HTML file but '.files' extension.

If *False*, all images are saved in the same directory as the HTML file.

This option is used for images with undefined (empty) file names, or when *rvsoUseItemImageFileNames* is not included in SaveOptions 71 property.

Default value:

True

See also:

TrvActionExport.CreateDirectoryForHTMLImages 63

1.3.1.11.1.3 TrvActionSave.DisableWhenUnmodified

Allows disabling the action if the editor is unmodified.

```
property DisableWhenUnmodified: Boolean;
```

If *False*, the action is always enabled.

If *True*, the action is enabled only when a document in the target editor is changed.

Default value

False

1.3.1.11.1.4 TrvActionSave.Documents

A read-only list of opened documents.

```
property Documents: TRVList;
```

This is a list of TrvaDocumentInfo objects.

The main properties of TrvaDocumentInfo are:

- rve: TCustomRichViewEdit – editor;
- FileName: String – file name (full path) for the document in rve.
- FileFormat: TrvFileSaveFilter 233 – format for saving this file.
- CustomFilterIndex: Integer, used only if FileFormat=*ffeCustom*. Custom formats are numbered from 1 in the order they are listed in the CustomFilter 54 property of the linked TrvActionSaveAs 73 action.
- Defined – *False* if FileName and FileFormat contain temporal values assigned for new document.

This list is maintained automatically.

See also methods:

- FindDoc 72

1.3.1.11.1.5 TrvActionSave.FileTitle

Defines title for HTML files.

```
property FileTitle: String;
```

This property is defined only if RvHTMLViewImporter 222 is used.

This property is used when saving to HTML format (*ffeHTMLCSS* 233). This property is used as the Title parameter of TCustomRichViewEdit.SaveHTMLEx method.

See also:

- TrvActionExport.FileTitle 63

1.3.1.11.1.6 TrvActionSave.ImagePrefix

Prefix of image files saved together with HTML file.

```
property ImagePrefix: String;
```

This property is defined only if RvHTMLViewImporter 222 is used.

This property is used when saving to HTML format (*ffeHTMLCSS* 233). This property (probably with subdirectory name) is used as the ImagesPrefix parameter of TCustomRichViewEdit.SaveHTMLEx method.

This property is used for images with undefined (empty) file names, or when *rvsoUseItemImageFileNames* is not included in SaveOptions 71 property.

See also properties:

- CreateDirectoryForHTMLImages 69

See also:

- TrvActionExport.ImagePrefix 63

1.3.1.11.1.7 TrvActionSave.LostFormatWarning

Lists "lossy" file formats.

```
property LostFormatWarning: TrvFileSaveFilterSet 233;
```

When saving to file in one of these formats, a warning/confirmation is displayed:

"<FileName> may contain features that are not compatible with the chosen saving format. Do you want to save the document in this format?". This message is displayed only when saving the document in this format for the first time.

Default value:

all formats except for RVF (RichView Format)

1.3.1.11.1.8 TrvActionSave.SaveOptions

Defines options for saving HTML files.

```
property SaveOptions: TRVSaveOptions;
```

This property is defined only if RvHTMLViewImporter 222 is used.

This property is used when saving to HTML format (*ffeHTMLCSS* 233). This property is used as the SaveOptions parameter of TCustomRichViewEdit.SaveHTMLEx method.

Default value:

[]

See also:

- TrvActionExport.SaveOptions 64

1.3.1.11.1.9 TrvActionSave.SuppressNextErrorMessage

Turns off the default error message when saving to the custom format (*ffeCustom* 233).

```
property SuppressNextErrorMessage: Boolean;
```

This property is set to *False* before calling GetControlPanel 208.OnCustomFileOperation 37.

If you assign *True* to this property in this event, an error message will not be displayed even if *False* is returned in the Success parameter of this event.

1.3.1.11.2 Methods

1.3.1.11.2.1 TrvActionSave.CanCloseDoc

Performs necessary operations when closing the form containing **rve**.

```
function CanCloseDoc(rve: TCustomRichViewEdit): Boolean;
```

If document in **rve** is not modified, the method returns *True*.

Otherwise, the method asks the user: *"Save changes to <FileName>?"* (Yes/No/Cancel). If "Yes", the method

saves and returns *True*. If "No", the method does not save and returns *True*. If "Cancel", the method returns *False*.

This method should be used in OnCloseQuery method of the form containing **rve**.

Example:

```
procedure TForm1.FormCloseQuery(Sender: TObject; var CanClose: Boolean);
begin
  CanClose := rvActionSave1.CanCloseDoc(RichViewEdit1);
end;
```

1.3.1.11.2.2 TrvActionSave.FindDoc

Returns the index of item in Documents [70] for the given editor **rve**. Returns -1 if not found.

```
function FindDoc(rve: TCustomRichViewEdit): Integer;
```

1.3.1.11.3 Events

1.3.1.11.3.1 TrvActionSave.OnSaving

Occurs before saving.

```
type
  TRVSaveFileEvent = procedure (Sender: TObject;
    Editor: TCustomRichViewEdit; const FileName: String;
    FileFormat: TrvFileSaveFilter[233]; CustomFilterIndex: Integer) of object;

property OnSaving: TRVSaveFileEvent;
```

Parameters:

Editor – editor for saving.

FileName – file name (full path) to save.

FileFormat – format to save.

CustomFilterIndex used only if **FileFormat** is *ffeCustom.*

If **FileFormat** = *ffeCustom*, **CustomFilterIndex** identifies a custom file format. Custom formats are numbered from 1 in the order they are listed in the CustomFilter [54] property of the linked TrvActionSaveAs [73] action.

1.3.1.11.3.2 TrvActionSave.OnDocumentFileChange

Occurs when file for document in **Editor** is changed.

```
type
  TRVFileChangeEvent = procedure (Sender: TObject;
    Editor: TCustomRichViewEdit; const FileName: String;
    FileFormat: TrvFileSaveFilter[233]; IsNew: Boolean) of object;

property OnDocumentFileChange: TRVFileChangeEvent;
```

This event occurs in the following cases:

- TrvActionNew [56] action is executed;
- TrvActionOpen [59] action is executed, or TrvActionOpen.LoadFile [61] is called;
- TrvActionSaveAs [73] action is executed.

FileName is a file name (full path) assigned to the document in **Editor**.

FileFormat is a file format assigned to the document in **Editor**.

IsNew is *True* after the execution of TrvActionNew [56] action.

If **FileFormat**=*ffeCustom*, this event does not allow to identify the specific custom format. You can get it using Documents [70] property.

1.3.1.12 TrvActionSaveAs

TrvActionSaveAs is the action for "File | Save As..." command.

Unit RichViewActions [50];

Syntax

```
TrvActionSaveAs = class(TrvActionCustomFileIO [54])
```

Hierarchy

TObject
TPersistent
TComponent
TBasicAction
TContainedAction
TCustomAction
TAction
TrvCustomAction [206]
TrvAction [195]
TrvActionCustomIO [54]
TrvActionCustomFileIO [54]

Description

This action displays a file saving dialog and saves the content of TRichViewEdit component in the chosen file. The editor remains associated with this file name and file format, so subsequent executions of TrvActionSave [69] action save its content in this file in this format.

The following formats are supported:

- RichView Format (RVF)
- Rich Text Format (RTF)
- Text files (TXT, Unicode and ANSI)
- HTML (only if RvHtmlViewImporter [222] is used)
- RichViewXML Format (XML, only if RichViewXML [223] is used)
- custom formats, supported in GetControlPanel [208].OnCustomFileOperation [37] event.

If you use this action, you must also use TrvActionSave [69] action.

This action must be linked to TrvActionSave [69] action, because TrvActionSave [69] contains a list of all documents opened in TCustomRichViewEdit controls. Besides, this action uses TrvActionSave [69] to save the file when file name and format are chosen. This link may be defined explicitly (via ActionSave [73] property) or implicitly (this action finds and uses TrvActionSave [69] action placed on the same form/datamodule).

This action performs the following tasks:

1. Shows a file saving dialog allowing the user to choose the file name and format for saving.
2. If the chosen format is in LostFormatWarning [71] of TrvActionSave [69], displays a warning about lossy saving.
3. Saves the file, associates TCustomRichViewEdit control with this file name and file format.
4. Calls OnDocumentFileChange [72] event of TrvActionSave [69] action.

1.3.1.12.1 Properties

1.3.1.12.1.1 TrvActionSaveAs.ActionSave

Links this action to TrvActionSave [69] action.

```
property ActionSave: TrvActionSave [69];
```

If this link is not defined, the first found TrvActionSave [69] action on the same form/datamodule is used. If no TrvActionSave [69] action found, an exception occurs on executing.

1.3.1.12.1.2 TrvActionSaveAs.Filter

Defines a set of file formats appearing in the file saving dialog.

```
property Filter: TrvFileSaveFilterSet[233];
```

If *ffeCustom* is included, formats listed in the CustomFilter[54] property are used.

Name and extension of RichView Format (RVF) may be customized, see GetControlPanel[208].RVFFilter[30].

Default value:

all formats

1.3.1.13 TrvCustomPrintAction

TrvCustomPrintAction is a base class for the printing actions.

Unit RichViewActions[50];

Syntax

```
TrvCustomPrintActions = class(TrvAction[195])
```

Hierarchy

TObject
TPersistent
TComponent
TBasicAction
TContainedAction
TCustomAction
TAction
TrvCustomAction[206]
TrvAction[195]

Description

This action does not introduce any new properties in addition to properties of TrvAction.

This action is not used directly, this is the base class for the following actions:

- TrvActionPrintPreview[66]
- TrvActionQuickPrint[68]
- TrvActionPrint[65]

All these actions are enabled only if at least one printer installed in the system (Printer.Printers.Count>0).

1.3.2 Edit

1.3.2.1 TrvActionCharCase

TrvActionCharCase is the action for "Character Case" command.

Unit RichViewActions[50];

Syntax

```
TrvActionCharCase = class(TrvAction[195])
```

Hierarchy

TObject
TPersistent
TComponent
TBasicAction
TContainedAction
TCustomAction
TAction

TrvCustomAction 206
TrvAction 195

Description

The action changes character case in the selected fragment: "all lower case" → "all upper case" → "first letter in each word upper case" → "all lower case" → ...

This action does not introduce any new properties in addition to properties of TrvAction.

Limitation: this action cannot be applied to multicell selection in table.

See also:

- RVChangeCharCase, RVGetCharCase 232 procedures.
- TrvActionFontAllCaps 85

1.3.2.2 TrvActionCopy

TrvActionCopy is the action for "Edit | Copy" command.

Unit RichViewActions 50;

Syntax

```
TrvActionCopy = class(TrvCustomEditAction 84)
```

Hierarchy

TObject
TPersistent
TComponent
TBasicAction
TContainedAction
TCustomAction
TAction
TrvCustomAction 206
TrvAction 195
TrvCustomEditAction 84

Description

This action copies the selection to the Clipboard (calls TCustomRichViewEdit.CopyDef).

This action does not introduce any new properties in addition to properties of TrvAction.

This action can work with several types of editors, see the comments in the TrvCustomEditAction 84 topic.

1.3.2.3 TrvActionCut

TrvActionCut is the action for "Edit | Cut" command.

Unit RichViewActions 50;

Syntax

```
TrvActionCut = class(TrvCustomEditAction 84)
```

Hierarchy

TObject
TPersistent
TComponent
TBasicAction
TContainedAction
TCustomAction
TAction

TrvCustomAction 206
TrvAction 195
TrvCustomEditAction 84

Description

This action cuts the selection to the Clipboard (calls TCustomRichViewEdit.CutDef).

This action does not introduce any new properties in addition to properties of TrvAction.

This action can work with several types of editors, see the comments in the TrvCustomEditAction 84 topic.

1.3.2.4 TrvActionFind

TrvActionFind is the action for "Edit | Find" command.

Unit RichViewActions 50;

Syntax

```
TrvActionFind = class(TrvAction 195)
```

Hierarchy

TObject
TPersistent
TComponent
TBasicAction
TContainedAction
TCustomAction
TAction
TrvCustomAction 206
TrvAction 195

Description

This action displays a TFindDialog for searching a substring in the TCustomRichViewEdit component. If at the moment of execution a single line of text is selected, this text is copied to the find dialog. If at the moment of execution a replace dialog is shown, it is closed.

The search starts from the caret position to the end or to the beginning of the document. If the text is found, it is selected. If not found, the action's behavior depends on GetControlPanel 208.SearchScope 32 property:

- *rvssFromCursor*, the search stops;
- *rvssAskUser*, the action asks user whether to continue from the beginning/end;
- *rvssGlobal*, the search continues from the beginning/end.

If nothing found, a message is displayed.

In Delphi 2009+, or if TNT Controls 224 are used, the action searches for Unicode string. Otherwise, it searches for ANSI string.

See also:

TrvActionFindNext 77
TrvActionReplace 81

1.3.2.4.1 Properties

1.3.2.4.1.1 TrvActionFind.ActionReplace

Links this action to TrvActionReplace 81 action.

```
property ActionReplace: TrvActionReplace 81;
```

If this link is not defined, the first TrvActionReplace 81 action found in the same form/datamodule is used.

When this action is executed, it calls TrvActionReplace.CloseDialog 82.

1.3.2.4.2 Methods

1.3.2.4.2.1 TrvActionFind.CloseDialog

Closes the find dialog (if it is shown).

```
procedure CloseDialog;
```

1.3.2.5 TrvActionFindNext

TrvActionFindNext is the action for "Edit | Find Next" action.

Unit RichViewActions[50];

Syntax

```
TrvActionFindNext = class(TrvAction[195])
```

Hierarchy

TObject
TPersistent
TComponent
TBasicAction
TContainedAction
TCustomAction
TAction
TrvCustomAction[206]
TrvAction[195]

Description

This action continues the search started by TrvActionFind[76] action. The link between the actions may be defined explicitly (via ActionFind[77] property) or implicitly (this action finds and uses TrvActionFind[76] action placed on the same form/datamodule).

1.3.2.5.1 Properties

1.3.2.5.1.1 TrvActionFindNext.ActionFind

Links this action to TrvActionFind[76] action.

```
property ActionFind: TrvActionFind[76];
```

If this link is not defined, the first TrvActionFind[76] action found in the same form/datamodule is used. If it cannot be found, an exception occurs.

When executed, TrvActionFindNext action continues the search started by TrvActionFind[76] action. If search was not started, the action executes TrvActionFind[76] action instead.

1.3.2.6 TrvActionPaste

TrvActionPaste is the action for "Edit | Paste" command.

Unit RichViewActions[50];

Syntax

```
TrvActionPaste = class(TrvCustomEditAction[84])
```

Hierarchy

TObject
TPersistent
TComponent
TBasicAction
TContainedAction

TCustomAction
TAction
TrvCustomAction 206
TrvAction 195
TrvCustomEditAction 84

Description

This action pastes data from the Clipboard (calls TCustomRichViewEdit.Paste).

If RvHtmlViewImporter 222 or rvHtmlImporter 221 is used, the action can paste also in HTML format. This format is pasted only if:

- the Clipboard does not contain data in RVF format, and
- the Clipboard does not contain data in RTF format (or it contains empty RTF data; it happens when MS Internet Explorer copies a complex fragment with nested tables).

This action does not introduce any new properties in addition to properties of TrvAction.

This action can work with several types of editors, see the comments in the TrvCustomEditAction 84 topic.

1.3.2.7 TrvActionPasteAsText

TrvActionPasteAsText is the action for "Edit | Paste as Text" command.

Unit RichViewActions 50;

Syntax

```
TrvActionPasteAsText = class(TrvCustomEditAction 84)
```

Hierarchy

TObject
TPersistent
TComponent
TBasicAction
TContainedAction
TCustomAction
TAction
TrvCustomAction 206
TrvAction 195
TrvCustomEditAction 84

Description

This action pastes text from the Clipboard. If the current text in the editor is Unicode, it pastes Unicode text.

This action does not introduce any new properties in addition to properties of TrvAction.

This action can work with several types of editors, see the comments in the TrvCustomEditAction 84 topic.

1.3.2.8 TrvActionPasteSpecial

TrvActionPasteSpecial is the action for "Edit | Paste Special" command.

Unit RichViewActions 50;

Syntax

```
TrvActionPasteSpecial = class(TrvActionPaste 77)
```

Hierarchy

TObject
TPersistent
TComponent

TBasicAction
TContainedAction
TCustomAction
TAction
TrvCustomAction 206
TrvAction 195
TrvCustomEditAction 84
TrvActionPaste 77

Description

This action displays a dialog containing a list of data formats available in the Clipboard. When the user chooses a format, data are pasted into TCustomRichViewEdit/TSRichViewEdit component in this format.

Supported formats:

- ANSI text (CF_TEXT)
- Unicode text (CF_UNICODETEXT)
- Windows bitmap (CF_BITMAP)
- Windows metafile (CF_METAFILEPICT)
- URL ("UniformResourceLocator")
- RTF ("Rich Text Format")
- RVF ("RichView Format")
- HTML ("HTML Format") (only if RvHtmlViewImporter 222 or RvHtmlImporter 221 is used)
- Graphic files (CF_HDROP)
- custom formats (see OnShowing 81 and OnCustomPaste 81 events).

The action displays only formats listed in in TCustomRichViewEdit.AcceptPasteFormats. **Note:** HTML and URL formats are not included in AcceptPasteFormats by default.

If GetControlPanel 208.RVFLocalizable 30=*True*, GetControlPanel 208.RVFFormatTitle 31 is used in the list for RVF Format. You can change it to something like "My Application's Format". But in this case you need to provide a localization of this property yourself.

When pasting graphic files, the properties StoreFileName 79 and StoreFileNameInItemName 80 are used.

If style templates are used (RichViewEdit.UseStyleTemplates=*True*), the action displays options for insertion of RTF and RVF.

Unlike all other actions inherited from TrvCustomEditAction 84, this action cannot work with editor types other than TCustomRichViewEdit and TSRichViewEdit.

1.3.2.8.1 Properties

1.3.2.8.1.1 TrvActionPasteSpecial.StoreFileName

This property is used when pasting graphic files.

```
property StoreFileName: Boolean;
```

If this property is *True*, path to the graphic file is stored in the document in the *rvespImageFileName* item property (see the RichView help file, TRVExtraItemStrProperty type).This is a recommended way to store path to graphic files. You can use these file names when saving HTML files (Include *rvsoUseItemImageFileNames* in TrvActionSave.SaveOptions 71 and TrvActionExport.SaveOptions 64).

Paths cannot be stored for bitmaps and metafiles pasted in CF_BITMAP and CF_METAFILEPICT formats.

Default value

False

See also:

- TrvActionInsertPicture.StoreFileName 152
- TrvActionItemProperties.StoreImageFileName 203
- TrvActionTableProperties.StoreImageFileName 188

- TCustomRichView.RTFReadProperties.StoreImagesFileNames
- HTML Import (for images inserted with HTML, paths to graphic files are always stored in *rvespImageFileName*)

1.3.2.8.1.2 TrvActionPasteSpecial.StoreFileNameInItemName

This property is used when pasting graphic files (CF_HDROP).

```
property StoreFileNameInItemName: Boolean;
```

If this property is *True*, path to the graphic file is stored in the document in the item name (item text).

It's not recommended to use this option. Use StoreFileName[79] instead.

Paths cannot be stored for bitmaps and metafiles pasted in CF_BITMAP and CF_METAFILEPICT formats.

Default value

False

See also:

- TrvActionInsertPicture.StoreFileNameInItemName[153]
- TrvActionItemProperties[201] (does not support storing paths in item names)
- RTF reading (does not support storing paths in item names)
- TRvHtmlViewImporter[222].StoreFileNameInItemName
- TRvHtmlImporter[221].Options (*rvhtmloImgSrcToImageName* option)

1.3.2.8.2 Methods

1.3.2.8.2.1 TrvActionPasteSpecial.AddFormat

Adds a new format in the paste-special dialog.

```
procedure AddFormat(const FormatName: String; Format: Word);
```

This method must be called only from OnShowing[81] event.

Parameters:

FormatName – format name, how it appears in the dialog.

Format – Clipboard format. This parameter can be a registered format or any of the standard Clipboard formats.

Data in the format specified in this method must be available in the Clipboard. If the user will choose this format, OnCustomPaste[81] event will occur.

1.3.2.8.3 Events

1.3.2.8.3.1 TrvActionPasteSpecial.OnCanPaste

This event is used to update Enabled property of the action.

```
type
  TRVACanPasteEvent = procedure (Sender: TrvActionPasteSpecial;
    var CanPaste: Boolean) of object;

property OnCanPaste: TRVACanPasteEvent;
```

On input, **CanPaste** parameter is the value returned by TCustomRichViewEdit.CanPaste.

If you implement pasting in additional formats, assign *True* to **CanPaste** if data in these formats are available in the Clipboard.

See also events:

- OnShowing[81]
- OnCustomPaste[81]

1.3.2.8.3.2 TrvActionPasteSpecial.OnCustomPaste

Occurs when the user chose a Clipboard format added with AddFormat [80] method.

```
type
  TRVACustomPasteEvent = procedure (Sender: TrvActionPasteSpecial;
    Editor: TCustomRichViewEdit; Format: Word) of object;

property OnCustomPaste: TRVACustomPasteEvent;
```

Paste data in the format **Format** into the editor **Editor**.

1.3.2.8.3.3 TrvActionPasteSpecial.OnShowing

Occurs before the paste-special dialog is shown.

```
property OnShowing: TNotifyEvent;
```

In this event, use AddFormat [80] method to add additional formats in the dialog.

1.3.2.9 TrvActionRedo

TrvActionRedo is the action for "Edit | Redo" command.

Unit RichViewActions [50];

Syntax

```
TrvActionRedo = class(TrvAction [195])
```

Hierarchy

TObject
TPersistent
TComponent
TBasicAction
TContainedAction
TCustomAction
TAction
TrvCustomAction [206]
TrvAction [195]

Description

This action redoes the last undone editing operation (calls TCustomRichViewEdit.Redo).

This action does not introduce any new properties in addition to properties of TrvAction.

1.3.2.10 TrvActionReplace

TrvActionReplace is the action for "Edit | Replace" command

Unit RichViewActions [50];

Syntax

```
TrvActionReplace = class(TrvAction [195])
```

Hierarchy

TObject
TPersistent
TComponent
TBasicAction
TContainedAction
TCustomAction
TAction

TrvCustomAction 206
TrvAction 195

Description

This action displays a TReplaceDialog for searching and replacing a substring in the TCustomRichViewEdit component. If at the moment of execution a single line of text is selected, this text is copied to the replace dialog (as a text to find). If at the moment of execution a find dialog is shown, it is closed.

The search starts from the caret position to the end or to the beginning of the document. If the text is found, it is replaced. If not found, the action's behavior depends on GetControlPanel 208.SearchScope 32 property:

- *rvssFromCursor*, the search stops;
- *rvssAskUser*, the action asks user whether to continue from the beginning/end;
- *rvssGlobal*, the search continues from the beginning/end.

If nothing found, a message is displayed.

In Delphi 2009+, or if TNT Controls 224 are used, the action searches for Unicode string and replaces it with Unicode string. Otherwise, it works with ANSI strings.

See also:

TrvActionFindNext 77
TrvActionFind 76

1.3.2.10.1 Properties

1.3.2.10.1.1 TrvActionReplace.ActionFind

Links this action to TrvActionFind 76 action.

```
property ActionFind: TrvActionFind 76;
```

If this link is not defined, the first TrvActionFind 76 action found in the same form/datamodule is used.

When this action is executed, it calls TrvActionFind.CloseDialog 77.

1.3.2.10.1.2 TrvActionReplace.Show ReplaceAllSummary

Specifies whether the action should display a summary message after completing a "Replace All" command.

```
property ShowReplaceAllSummary: Boolean;
```

If *True*, a message *"N strings replaced."* is shown.

Default value:

True

1.3.2.10.2 Methods

1.3.2.10.2.1 TrvActionReplace.CloseDialog

Closes the replace dialog (if it is shown).

```
procedure CloseDialog;
```

1.3.2.10.3 Events

1.3.2.10.3.1 TrvActionReplace.OnReplaceAllEnd

Occurs when "replace all" operation finishes.

```
property OnReplaceAllEnd: TRVAEditEvent 232;
```

If GetControlPanel 208.SearchScope 32 property is equal to *rvssAskUser* or *rvssGlobal*, this event may occur twice: when the end/beginning of the document is reached, and when all replacements are completed.

The event occurs before showing a replace-all summary message.

1.3.2.10.3.2 TrvActionReplace.OnReplaceAllStart

Occurs when "replace all" operation starts.

```
property OnReplaceAllStart: TRVAEditEvent[232];
```

If GetControlPanel[208].SearchScope[32] property is equal to *rvssAskUser* or *rvssGlobal*, this event may occur twice: when the search starts, and when the search continues from the beginning/end of the document.

1.3.2.10.3.3 TrvActionReplace.OnReplacing

Occurs before the action replaces one occurrence of the found text with **NewText**.

```
type
  TRVAReplacingEvent = procedure (Sender: TObject; Editor: TCustomRichViewEdit;
    const NewText: String) of object;

property OnReplacing: TRVAReplacingEvent;
```

When this event occurs, the found text is selected in **Editor**.

1.3.2.11 TrvActionSelectAll

TrvActionSelectAll is the action for "Edit | Select All" command.

Unit RichViewActions[50];

Syntax

```
TrvActionSelectAll = class(TrvAction[195])
```

Hierarchy

TObject
TPersistent
TComponent
TBasicAction
TContainedAction
TCustomAction
TAction
TrvCustomAction[206]
TrvAction[195]

Description

This action selects the whole document (calls TCustomRichViewEdit.SelectAll).

This action does not introduce any new properties in addition to properties of TrvAction.

1.3.2.12 TrvActionUndo

TrvActionUndo is the action for "Edit | Undo" command.

Unit RichViewActions[50];

Syntax

```
TrvActionUndo = class(TrvCustomEditAction[84])
```

Hierarchy

TObject
TPersistent
TComponent

TBasicAction
TContainedAction
TCustomAction
TAction
TrvCustomAction 206
TrvAction 195
TrvCustomEditAction 84

Description

This action undoes the last editing operation (calls TCustomRichViewEdit.Undo).

This action does not introduce any new properties in addition to properties of TrvAction.

This action can work with several types of editors, see the comments in the TrvCustomEditAction 84 topic.

1.3.2.13 TrvCustomEditAction

TrvCustomEditAction is a base class for several editing actions.

Unit RichViewActions 50;

Syntax

```
TrvCustomEditAction = class(TrvAction 195)
```

Hierarchy

TObject
TPersistent
TComponent
TBasicAction
TContainedAction
TCustomAction
TAction
TrvCustomAction 206
TrvAction 195

Description

This action does not introduce any new properties in addition to properties of TrvAction.

This action is not used directly. The following actions are inherited from it:

- TrvActionUndo 83
- TrvActionCut 75
- TrvActionCopy 75
- TrvActionPaste 77

All these actions can work not only with TCustomRichViewEdit components, but also with another editors.

By default, the list of these editors includes:

- TCustomEdit (TEdit, TMemo, TRichEdit and others);
- TComboBox (having *csDropDown* or *csSimple* style).

You can implement working with additional editor types by assigning your own function to RVA_EditorControlFunction 236 variable.

See also RVA_EditForceDefControl 235 global variable.

1.3.3 Font

1.3.3.1 TrvActionFontAllCaps

TrvActionFontAllCaps is the action for "All Capitals" command.

Unit RichViewActions [50];

Syntax

```
TrvActionFontAllCaps = class(TrvActionFontStyleEx [98])
```

Hierarchy

TObject
TPersistent
TComponent
TBasicAction
TContainedAction
TCustomAction
TAction
TrvCustomAction [206]
TrvAction [195]
TrvActionTextStyles [102]
TrvActionFontStyleEx [98]

Description

This action does not introduce any new properties in addition to properties of TrvAction.

This action converts the selected fragment to upper case characters (toggle). It adds/removes *rvfsAllCaps* in StyleEx property for styles of all selected text items, without changing the actual text.

This is a checkbox-like action, its Checked property is updated depending on the state of selection (or the current text style).

See also:

- TrvActionCharCase [74]

1.3.3.2 TrvActionFontBackColor

TrvActionFontBackColor is the action for changing background color of the selected text.

Unit RichViewActions [50];

Syntax

```
TrvActionFontBackColor = class(TrvActionFontCustomColor [87])
```

Hierarchy

TObject
TPersistent
TComponent
TBasicAction
TContainedAction
TCustomAction
TAction
TrvCustomAction [206]
TrvAction [195]
TrvActionCustomColor [197]
TrvActionFontCustomColor [87]

Description

This action does not introduce any new properties and events in addition to properties and events of

TrvActionCustomColor[197].

On execution, this action changes BackColor property of styles of the selected text (RVStyle.TextStyles[].BackColor).

See also the list of color changing actions in the topic about TrvActionCustomColor[197].

1.3.3.3 TrvActionFontBold

TrvActionFontBold is the action for "Bold" command.

Unit RichViewActions[50];

Syntax

```
TrvActionFontBold = class(TrvActionFontStyle[97])
```

Hierarchy

TObject
TPersistent
TComponent
TBasicAction
TContainedAction
TCustomAction
TAction
TrvCustomAction[206]
TrvAction[195]
TrvActionTextStyles[102]
TrvActionFontStyle[97]

Description

This action does not introduce any new properties in addition to properties of TrvAction.

This action makes the selected text bold (or not bold if already bold). It adds/removes *fsBold* in Style property for styles of all selected text items.

This is a checkbox-like action, its Checked property is updated depending on the state of selection (or the current text style).

1.3.3.4 TrvActionFontColor

TrvActionFontColor is the action for changing color of the selected text.

Unit RichViewActions[50];

Syntax

```
TrvActionFontColor = class(TrvActionFontCustomColor[87])
```

Hierarchy

Hierarchy

TObject
TPersistent
TComponent
TBasicAction
TContainedAction
TCustomAction
TAction
TrvCustomAction[206]
TrvAction[195]
TrvActionCustomColor[197]

TrvActionFontCustomColor 87

Description

This action does not introduce any new properties and events in addition to properties and events of TrvActionCustomColor 197. However, it changes the default value of Color 198 property to *clWindowText*.

On execution, this action changes Color property of styles of the selected text (RVStyle.TextStyles[].Color).

See also the list of color changing actions in the topic about TrvActionCustomColor 197.

1.3.3.5 TrvActionFontCustomColor

TrvActionFontCustomColor is a base class for the actions changing color properties of the selected text.

Unit RichViewActions 50;

Syntax

```
TrvActionFontCustomColor = class(TrvActionCustomColor 197)
```

Hierarchy

Hierarchy

TObject
TPersistent
TComponent
TBasicAction
TContainedAction
TCustomAction
TAction
TrvCustomAction 206
TrvAction 195
TrvActionCustomColor 197

Description

This action does not introduce any new properties and events in addition to properties and events of TrvActionCustomColor 197.

This action is not used directly. The following actions are inherited from it:

- TrvActionFontColor 86
- TrvActionFontBackColor 85

1.3.3.6 TrvActionFontEx

TrvActionFontEx is the action for "Font" command. It uses an advanced font dialog.

Unit RichViewActions 50;

Syntax

```
TrvActionFontEx = class(TrvActionFonts 93)
```

Hierarchy

TObject
TPersistent
TComponent
TBasicAction
TContainedAction
TCustomAction
TAction
TrvCustomAction 206
TrvAction 195

TrvActionTextStyles 102
TrvActionFonts 93

Description

If UserInterface 94=*False*, the action applies its properties to the selected text. The list of properties to apply is specified in ValidProperties 90.

If UserInterface 94=*True*, the action displays a font dialog (initialized with font properties of the selected text), assigns the chosen font to the action's properties, and then applies these properties to the selected text.

For example, if you want to apply *{"Arial", 12, not bold, dotted underline}* to the selected text in RichViewEdit1:

```
  with rvActionFontEx1 do begin
    UserInterface := False;
    ValidProperties := [rvfimFontName, rvfimSize, rvfimBold, rvfimUnderline, rvfimUnder
    Font.Name := FontName;
    Font.Size := 12;
    Font.Style := [fsUnderline];
    UnderlineType := rvutDotted;
    ExecuteTarget(RichViewEdit1);
    UserInterface := True;
  end;
```

See also:

- TrvActionFonts 93

1.3.3.6.1 Properties

1.3.3.6.1.1 TrvActionFontEx.AutoApplySymbolCharset

Allows to correct charset for symbol fonts.

```
property AutoApplySymbolCharset: Boolean;
```

This property works if UserInterface 94=*False*.

If *True*, then when the action applies a font name without charset (i.e *rvfimFontName* in ValidProperties 90, *rvfimCharset* not in ValidProperties 90), if this is a symbol font (i.e. if it supports only SYMBOL_CHARSET), the action applies SYMBOL_CHARSET to the selection.

Default value:

True

1.3.3.6.1.2 TrvActionFontEx.BackColor

Specifies the background color for applying to the selected text.

```
property BackColor: TColor;
```

If UserInterface 94=*False* and *rvfimBackColor* in ValidProperties 90, the action applies this property to the selected text.

If UserInterface 94=*True*, the action displays a font dialog (initialized with font properties of the selected text); if the user changed value of this property in the dialog, *rvfimBackColor* is included in ValidProperties 90, and the action applies this property to the selected text.

This property is applied to the text style's (RVStyle.TextStyles[]) property of the same name.

1.3.3.6.1.3 TrvActionFontEx.CharScale

Specifies the character scale ratio for applying to the selected text.

```
property CharScale: Integer;
```

If UserInterface[94]=*False* and *rvfimCharScale* in ValidProperties[90], the action applies this property to the selected text.

If UserInterface[94]=*True*, the action displays a font dialog (initialized with font properties of the selected text); if the user changed value of this property in the dialog, *rvfimCharScale* is included in ValidProperties[90], and the action applies this property to the selected text.

This property is applied to the text style's (RVStyle.TextStyles[]) property of the same name.

1.3.3.6.1.4 TrvActionFontEx.CharSpacing

Specifies the character spacing for applying to the selected text.

```
property CharSpacing: TRVStyleLength;
```

If UserInterface[94]=*False* and *rvfimCharSpacing* in ValidProperties[90], the action applies this property to the selected text.

If UserInterface[94]=*True*, the action displays a font dialog (initialized with font properties of the selected text); if the user changed value of this property in the dialog, *rvfimCharSpacing* is included in ValidProperties[90], and the action applies this property to the selected text.

This property is applied to the text style's (RVStyle.TextStyles[]) property of the same name.

This value is measured in the same units (pixels or twips) as properties of the target editor (i.e. in RVStyle.Units).

1.3.3.6.1.5 TrvActionFontEx.FontStyleEx

Specifies the additional text styles (overline and/or all-capitals) for applying to the selected text.

```
property FontStyleEx: TRVFontStyles;
```

If UserInterface[94]=*False* and *rvfimOverline* or *rvfimAllCaps* in ValidProperties[90], the action applies this property to the selected text.

If UserInterface[94]=*True*, the action displays a font dialog (initialized with font properties of the selected text); if the user changed value of this property in the dialog, *rvfimOverline* and/or *rvfimAllCaps* are included in ValidProperties[90], and the action applies this property to the selected text.

This property is applied to the text style's (RVStyle.TextStyles[]) StyleEx property.

1.3.3.6.1.6 TrvActionFontEx.SubSuperScriptType

Specifies the sub/superscript mode for applying to the selected text.

```
property SubSuperScriptType: TRVSubSuperScriptType;
```

If UserInterface[94]=*False* and *rvfimSubSuperScriptType* in ValidProperties[90], the action applies this property to the selected text.

If UserInterface[94]=*True*, the action displays a font dialog (initialized with font properties of the selected text); if the user changed value of this property in the dialog, *rvfimSubSuperScriptType* is included in ValidProperties[90], and the action applies this property to the selected text.

This property is applied to the text style's (RVStyle.TextStyles[]) property of the same name.

1.3.3.6.1.7 TrvActionFontEx.UnderlineColor

Specifies the underline color for applying to the selected text.

```
property UnderlineColor: TColor;
```

If UserInterface[94]=*False* and *rvfimUnderlineColor* in ValidProperties[90], the action applies this property to the selected text.

If UserInterface[94]=*True*, the action displays a font dialog (initialized with font properties of the selected text); if the user changed value of this property in the dialog, *rvfimUnderlineColor* is included in ValidProperties[90], and the action applies this property to the selected text.

This property is applied to the text style's (RVStyle.TextStyles[]) property of the same name.

1.3.3.6.1.8 TrvActionFontEx.UnderlineType

Specifies the underline type for applying to the selected text.

```
property UnderlineType: TRVUnderlineType;
```

If UserInterface[94]=*False* and *rvfimUnderlineType* in ValidProperties[90], the action applies this property to the selected text.

If UserInterface[94]=*True*, the action displays a font dialog (initialized with font properties of the selected text); if the user changed value of this property in the dialog, *rvfimUnderlineType* is included in ValidProperties[90], and the action applies this property to the selected text.

This property is applied to the text style's (RVStyle.TextStyles[]) property of the same name.

1.3.3.6.1.9 TrvActionFontEx.ValidProperties

Defines a set of properties for applying to the selected text.

```
type
  TRVFontInfoMainProperty = (rvfimFontName, rvfimSize, rvfimCharset,
    rvfimBold, rvfimItalic, rvfimUnderline, rvfimStrikeout,
    rvfimOverline, rvfimAllCaps, rvfimVShift, rvfimColor, rvfimBackColor,
    rvfimCharScale, rvfimCharSpacing, rvfimSubSuperScriptType,
    rvfimUnderlineType, rvfimUnderlineColor);
  TRVFontInfoMainProperties = set of TRVFontInfoMainProperty;

property ValidProperties: TRVFontInfoMainProperties;
```

If UserInterface[94]=*False,* assign a subset of properties (for applying) to ValidProperties.

If UserInterface[94]=*True*, this property is maintained automatically. After displaying a font dialog, this property contains a set of properties corresponding to non-blank fields in the dialog.

For properties-sets (Font[94].Style and FontStyleEx[89]) each value in the set is represented by one value in this property. For example:

- if *rvfimOverline* in ValidProperties, *rvfsOverline* in FontStyleEx[89], the selected text will be overlined;
- if *rvfimOverline* in ValidProperties, *rvfsOverline* not in FontStyleEx[89], overlines will be removed from the selected text;
- if *rvfimOverline* not in ValidProperties, overlines in the selected text will not be affected.

1.3.3.6.1.10 TrvActionFontEx.VShift

Specifies the vertical shift for applying to the selected text.

```
property VShift: Integer;
```

If UserInterface[94]=*False* and *rvfimVShift* in ValidProperties[90], the action applies this property to the selected text.

If UserInterface 94=*True*, the action displays a font dialog (initialized with font properties of the selected text); if the user changed value of this property in the dialog, *rvfimVShift* is included in ValidProperties 90, and the action applies this property to the selected text.

This property is applied to the text style's (RVStyle.TextStyles[]) property of the same name.

1.3.3.6.1.11 TrvActionFontEx.FontSizeDouble

Specifies the font size (in half-points) for applying to the selected text.

```
property FontSizeDouble: Integer;
```

If UserInterface 94=*False* and *rvfimSize* in ValidProperties 90, the action applies this property to the selected text.

If UserInterface 94=*True*, the action displays a font dialog (initialized with font properties of the selected text); if the user changed value of this property in the dialog, *rvfimSize* is included in ValidProperties 90, and the action applies this property to the selected text.

This property is applied to the text style's (RVStyle.TextStyles[]) property SizeDouble.

This property overrides Font 94.Size.

1.3.3.7 TrvActionFontGrow

TrvActionFontGrow is the action for "Grow Font" command.

Unit RichViewActions 50;

Syntax

```
TrvActionFontGrow = class(TrvActionFontShrinkGrow 95)
```

Hierarchy

TObject
TPersistent
TComponent
TBasicAction
TContainedAction
TCustomAction
TAction
TrvCustomAction 206
TrvAction 195
TrvActionTextStyles 102
TrvActionFontShrinkGrow 95

Description

The action increases font size of the selected text by Percent 96 (until it reaches MaxSize 91).

See the list of font size changing actions in the topic about TrvActionFontShrinkGrow 95.

1.3.3.7.1 Properties

1.3.3.7.1.1 TrvActionFontGrow .MaxSize

Defines maximal font size.

```
property MaxSize: Integer;
```

The action does not increase font size above this value.

Default value:

100

1.3.3.8 TrvActionFontGrowOnePoint

TrvActionFontGrowOnePoint is the action for "Grow Font by One Point" command.

Unit RichViewActions 50;

Syntax

```
TrvActionFontGrowOnePoint = class(TrvActionTextStyles 102)
```

Hierarchy

TObject
TPersistent
TComponent
TBasicAction
TContainedAction
TCustomAction
TAction
TrvCustomAction 206
TrvAction 195
TrvActionTextStyles 102

Description

The action increases font size of the selected text by 1 (until it reaches MaxSize 92).

See the list of font size changing actions in the topic about TrvActionFontShrinkGrow 95.

1.3.3.8.1 Properties

1.3.3.8.1.1 TrvActionFontGrow OnePoint.MaxSize

Defines maximal font size.

```
property MaxSize: Integer;
```

The action does not increase font size above this value.

Default value:

100

1.3.3.9 TrvActionFontItalic

TrvActionFontItalic is the action for "Italic" command.

Unit RichViewActions 50;

Syntax

```
TrvActionFontItalic = class(TrvActionFontStyle 97)
```

Hierarchy

TObject
TPersistent
TComponent
TBasicAction
TContainedAction
TCustomAction
TAction
TrvCustomAction 206
TrvAction 195
TrvActionTextStyles 102
TrvActionFontStyle 97

Description

This action does not introduce any new properties in addition to properties of TrvAction.

This action makes the selected text italic (or not italic if already italic). It adds/removes *fsItalic* in Style property for styles of all selected text items.

This is a checkbox-like action, its Checked property is updated depending on the state of selection (or the current text style).

1.3.3.10 TrvActionFontOverline

TrvActionFontOverline is the action for "Overline" command.

Unit RichViewActions 50;

Syntax

```
TrvActionFontOverline = class(TrvActionFontStyleEx 98)
```

Hierarchy

TObject
TPersistent
TComponent
TBasicAction
TContainedAction
TCustomAction
TAction
TrvCustomAction 206
TrvAction 195
TrvActionTextStyles 102
TrvActionFontStyleEx 98

Description

This action does not introduce any new properties in addition to properties of TrvAction.

This action overlines the selected text (or removes overline, if already overlined). It adds/removes *rvfsOverline* in StyleEx property for styles of all selected text items.

This is a checkbox-like action, its Checked property is updated depending on the state of selection (or the current text style).

1.3.3.11 TrvActionFonts

TrvActionFonts is the action for "Font" command. It uses a standard font dialog (TFontDialog).

Unit RichViewActions 50;

Syntax

```
TrvActionFonts = class(TrvActionTextStyles 102)
```

Hierarchy

TObject
TPersistent
TComponent
TBasicAction
TContainedAction
TCustomAction
TAction
TrvCustomAction 206
TrvAction 195

TrvActionTextStyles 102

Description

If UserInterface 94=*False*, the action applies Font 94 to the selected text.

If UserInterface 94=*True*, the action displays a TFontDialog (initialized with font properties of the current text style), assigns the chosen font to Font 94, and then applies Font 94 to the selected text.

It's recommended to use TrvActionFontEx 87 instead of this action because of two reasons:

- TrvActionFonts applies all font properties, it's not possible to leave some fields blank in TFontDialog. In TrvActionFontEx 87, if you open a dialog and press OK without changing anything, nothing will be changed in the selected text; this is not so for this action;
- TrvActionFontEx 87 allows defining much more text properties.

1.3.3.11.1 Properties

1.3.3.11.1.1 TrvActionFonts.Font

Specifies the font for applying to the selected text.

```
type
  TRVAFont = class(TFont);
property Font: TRVAFont;
```

If UserInterface 94=*False*, the action applies this property to the selected text. For TrvActionFontEx 87, only properties specified in ValidProperties 90 are applied (a subset of *rvfimFontName*, *rvfimSize*, *rvfimCharset*, *rvfimBold*, *rvfimItalic*, *rvfimUnderline*, *rvfimStrikeout*, *rvfimColor*).

If UserInterface 94=*True*, the action displays a font dialog, assigns the chosen font to this property, and then applies this property to the selected text.

This property is applied to the following properties of text style (RVStyle.TextStyles[]):

- FontName
- Size (in TrvActionFontEx 87, it is overridden by FontSizeDouble 91)
- Charset
- Style
- Color

1.3.3.11.1.2 TrvActionFonts.UserInterface

Allows working with or without the user interaction.

```
property UserInterface: Boolean;
```

If UserInterface=*False*, the action applies its properties to the selected text.

If UserInterface=*True*, the action displays a font dialog, assigns changes to the action's properties, and then applies these properties to the selected text.

Default value:

True

1.3.3.12 TrvActionFontShrink

TrvActionFontShrink is the action for "Shrink Font" command.

Unit RichViewActions 50;

Syntax

```
TrvActionFontShrink = class(TrvActionFontShrinkGrow 95)
```

Hierarchy

TObject

TPersistent
TComponent
TBasicAction
TContainedAction
TCustomAction
TAction
TrvCustomAction 206
TrvAction 195
TrvActionTextStyles 102
TrvActionFontShrinkGrow 95

Description

The action decreases font size of the selected text by Percent 96 (until it reaches MinSize 95).

See the list of font size changing actions in the topic about TrvActionFontShrinkGrow 95.

1.3.3.12.1 Properties

1.3.3.12.1.1 TrvActionFontShrink.MinSize

Defines minimal font size.

```
property MinSize: Integer;
```

The action does not decrease font size below this value.

Default value:

1

1.3.3.13 TrvActionFontShrinkGrow

TrvActionFontShrinkGrow is a base class for the actions changing font size of the selected text by the specified percent.

Unit RichViewActions 50;

Syntax

```
TrvActionFontShrinkGrow = class(TrvActionTextStyles 102)
```

Hierarchy

TObject
TPersistent
TComponent
TBasicAction
TContainedAction
TCustomAction
TAction
TrvCustomAction 206
TrvAction 195
TrvActionTextStyles 102

Description

The action changes Size property of text styles for the selected text items (RVStyle.TextStyles[].Size).

This action is not used directly. The following actions are inherited from it:

- TrvActionFontShrink 94
- TrvActionFontGrow 91

See also:

- TrvActionFontShrinkOnePoint 96
- TrvActionFontGrowOnePoint 92

1.3.3.13.1 Properties

1.3.3.13.1.1 TrvActionFontShrinkGrow .Percent

Defines the percent for increasing or decreasing font size.

```
property Percent: Integer;
```

Default value:

10

1.3.3.14 TrvActionFontShrinkOnePoint

TrvActionFontShrinkOnePoint is the action for "Shrink Font by One Point" command.

Unit RichViewActions [50];

Syntax

```
TrvActionFontShrinkOnePoint = class(TrvActionTextStyles [102])
```

Hierarchy

TObject
TPersistent
TComponent
TBasicAction
TContainedAction
TCustomAction
TAction
TrvCustomAction [206]
TrvAction [195]
TrvActionTextStyles [102]

Description

The action decreases font size of the selected text by 1 (until it reaches MinSize [96]).

See the list of font size changing actions in the topic about TrvActionFontShrinkGrow [95].

1.3.3.14.1 Properties

1.3.3.14.1.1 TrvActionFontShrinkOnePoint.MinSize

Defines minimal font size.

```
property MinSize: Integer;
```

The action does not decrease font size below this value.

Default value:

1

1.3.3.15 TrvActionFontStrikeout

TrvActionFontStrikeout is the action for "Strikeout" command.

Unit RichViewActions [50];

Syntax

```
TrvActionFontStrikeout = class(TrvActionFontStyle [97])
```

Hierarchy

TObject
TPersistent
TComponent

TBasicAction
TContainedAction
TCustomAction
TAction
TrvCustomAction 206
TrvAction 195
TrvActionTextStyles 102
TrvActionFontStyle 97

Description

This action does not introduce any new properties in addition to properties of TrvAction.

This action strikes through the selected text (toggle). It adds/removes *fsStrikeOut* in Style property for styles of all selected text items.

This is a checkbox-like action, its Checked property is updated depending on the state of selection (or the current text style).

1.3.3.16 TrvActionFontStyle

TrvActionFontStyle is a base class for the actions changing Style property of text styles for the selected text (RVStyle.TextStyles[].Style)

Unit RichViewActions 50;

Syntax

```
TrvActionFontStyle = class(TrvActionTextStyles 102)
```

Hierarchy

TObject
TPersistent
TComponent
TBasicAction
TContainedAction
TCustomAction
TAction
TrvCustomAction 206
TrvAction 195
TrvActionTextStyles 102

Description

This action does not introduce any new properties in addition to properties of TrvAction.

This action is not used directly. The following actions are inherited from it:

- TrvActionFontBold 86
- TrvActionFontItalic 92
- TrvActionFontUnderline 98
- TrvActionFontStrikeout 96

These actions change one option in the Style property of styles of text in the selected fragment of TCustomRichViewEdit (for example, *fsBold*)

These actions are checked (Checked=*True*) in the following cases:

- if several table cells are selected, all text items in these cells have the required option included in Style (for example, all text is bold);
- otherwise, if the current text style (TCustomRichViewEdit.CurTextStyleNo) has this option included in Style (for example, it is bold),

1.3.3.17 TrvActionFontStyleEx

TrvActionFontStyle is a base class for the actions changing StyleEx property of text styles for the selected text (RVStyle.TextStyles[].StyleEx)

Unit RichViewActions 50;

Syntax

```
TrvActionFontStyleEx = class(TrvActionTextStyles 102)
```

Hierarchy

TObject
TPersistent
TComponent
TBasicAction
TContainedAction
TCustomAction
TAction
TrvCustomAction 206
TrvAction 195
TrvActionTextStyles 102

Description

This action does not introduce any new properties in addition to properties of TrvAction.

This action is not used directly. The following actions are inherited from it:

- TrvActionFontAllCaps 85
- TrvActionFontOverline 93

These actions change one option in the StyleEx property of styles of text in the selected fragment of TCustomRichViewEdit (for example, *rvfsOverline*)

These actions are checked (Checked=*True*) in the following cases:

- if several table cells are selected, all text items in these cells have the required option included in StyleEx (for example, all text is overlined);
- otherwise, if the current text style (TCustomRichViewEdit.CurTextStyleNo) has this option included in StyleEx (for example, it is overlined),

1.3.3.18 TrvActionFontUnderline

TrvActionFontUnderline is the action for "Underline" command.

Unit RichViewActions 50;

Syntax

```
TrvActionFontUnderline = class(TrvActionFontStyle 97)
```

Hierarchy

TObject
TPersistent
TComponent
TBasicAction
TContainedAction

TCustomAction
TAction
TrvCustomAction 206
TrvAction 195
TrvActionTextStyles 102
TrvActionFontStyle 97

Description

This action does not introduce any new properties in addition to properties of TrvAction.

This action underlines the selected text (or removes underline, if already underlined). It adds/removes *fsUnderline* in Style property for styles of all selected text items. This action changes UnderlineType property of text styles to *rvutNormal*. This action does not change underline color.

This is a checkbox-like action, its Checked property is updated depending on the state of selection (or the current text style).

1.3.3.19 TrvActionSSScript

TrvActionSSScript is a base class for the subscript and the superscript actions.

Unit RichViewActions 50;

Syntax

```
TrvActionTextStyles = class(TrvActionTextStyles 102)
```

Hierarchy

TObject
TPersistent
TComponent
TBasicAction
TContainedAction
TCustomAction
TAction
TrvCustomAction 206
TrvAction 195
TrvActionTextStyles 102

Description

This action does not introduce any new properties in addition to properties of TrvAction.

This action changes SubSuperScriptType property for styles of the selected text (RVStyle.TextStyles[].SubSuperScriptType).

This action is not used directly. The following actions are inherited from it:

- TrvActionSubscript 99
- TrvActionSuperscript 100

1.3.3.20 TrvActionSubscript

TrvActionSubscript is the action for "Subscript" command.

Unit RichViewActions 50;

Syntax

```
TrvActionSubscript = class(TrvActionSSScript 99)
```

Hierarchy

TObject
TPersistent

TComponent
TBasicAction
TContainedAction
TCustomAction
TAction
TrvCustomAction 206
TrvAction 195
TrvActionTextStyles 102
TrvActionSSScript 99

Description

This action does not introduce any new properties in addition to properties of TrvAction.

The action makes the selected text subscript (or normal, if already subscript).

This is a checkbox-like action, its Checked property is updated depending on the state of selection (or the current text style).

1.3.3.21 TrvActionSuperscript

TrvActionSuperscript is the action for "Superscript" command.

Unit RichViewActions 50;

Syntax

```
TrvActionSuperscript = class(TrvActionSSScript 99)
```

Hierarchy

TObject
TPersistent
TComponent
TBasicAction
TContainedAction
TCustomAction
TAction
TrvCustomAction 206
TrvAction 195
TrvActionTextStyles 102
TrvActionSSScript 99

Description

This action does not introduce any new properties in addition to properties of TrvAction.

The action makes the selected text superscript (or normal, if already superscript).

This is a checkbox-like action, its Checked property is updated depending on the state of selection (or the current text style).

1.3.3.22 TrvActionTextBiDi

TrvActionTextBiDi is a base class for the actions changing the default text flow direction in the selected text.

Unit RichViewActions 50;

Syntax

```
TrvActionTextBiDi = class(TrvActionTextStyles 102)
```

Hierarchy

TObject
TPersistent

TComponent
TBasicAction
TContainedAction
TCustomAction
TAction
TrvCustomAction 206
TrvAction 195
TrvActionTextStyles 102

Description

This action does not introduce any new properties in addition to properties of TrvAction.

This action changes BiDiMode property for styles of the selected text (RVStyle.TextStyles[].BiDiMode).

This action is not used directly. The following actions are inherited from it:

- TrvActionTextLTR 101
- TrvActionTextRTL 102

If you use these actions, it's recommended to set TCustomRichViewEdit.BiDiMode to *rvbdLeftToRight* or *rvbdRightToLeft*.

Text flow direction specified in text styles has higher priority than text flow direction specified in paragraph styles.

See also:

- TrvActionParaBiDi 115
- TrvActionParaLTR 126
- TrvActionParaRTL 128

1.3.3.23 TrvActionTextLTR

TrvActionTextLTR is the action for "Left to Right Text" command.

Unit RichViewActions 50;

Syntax

```
TrvActionTextLTR = class(TrvActionTextBiDi 100)
```

Hierarchy

TObject
TPersistent
TComponent
TBasicAction
TContainedAction
TCustomAction
TAction
TrvCustomAction 206
TrvAction 195
TrvActionTextStyles 102
TrvActionTextBiDi 100

Description

This action changes the default text direction in the selected text to left-to-right. If you use this action, it's recommended to set TCustomRichViewEdit.BiDiMode to *rvbdLeftToRight* or *rvbdRightToLeft*.

This action does not introduce any new properties in addition to properties of TrvAction.

This action changes BiDiMode property for styles of the selected text (RVStyle.TextStyles[].BiDiMode) to *rvbdLeftToRight*.

This is a checkbox-like action, its Checked property is updated depending on the state of selection (or the

current text style).

See also:

- TrvActionTextRTL 102
- TrvActionParaLTR 126
- TrvActionParaRTL 128

1.3.3.24 TrvActionTextRTL

TrvActionTextRTL is the action for "Right to Left Text" command.

Unit RichViewActions 50;

Syntax

```
TrvActionTextRTL = class(TrvActionTextBiDi 100)
```

Hierarchy

TObject
TPersistent
TComponent
TBasicAction
TContainedAction
TCustomAction
TAction
TrvCustomAction 206
TrvAction 195
TrvActionTextStyles 102
TrvActionTextBiDi 100

Description

This action changes the default text direction in the selected text to right-to-left. If you use this action, it's recommended to set TCustomRichViewEdit.BiDiMode to *rvbdLeftToRight* or *rvbdRightToLeft*.

This action does not introduce any new properties in addition to properties of TrvAction.

This action changes BiDiMode property for styles of the selected text (RVStyle.TextStyles[].BiDiMode) to *rvbdRightToLeft*.

This is a checkbox-like action, its Checked property is updated depending on the state of selection (or the current text style).

See also:

- TrvActionTextLTR 101
- TrvActionParaLTR 126
- TrvActionParaRTL 128

1.3.3.25 TrvActionTextStyles

TrvActionTextStyles is a base class for:
- the actions applying changes to text style of the selected fragment of TCustomRichViewEdit component,
- TrvActionInsertHyperlink 138.

Unit RichViewActions 50;

Syntax

```
TrvActionTextStyles = class(TrvAction 195)
```

Hierarchy

TObject
TPersistent

TComponent
TBasicAction
TContainedAction
TCustomAction
TAction
TrvCustomAction 206
TrvAction 195

Description

This action does not introduce any new properties in addition to properties of TrvAction.

This action is not used directly. This action has the following direct descendants:

- TrvActionFonts 93
- TrvActionFontStyle 97
- TrvActionFontStyleEx 98
- TrvActionSSScript 99
- TrvActionTextBiDi 100
- TrvActionFontShrinkGrow 95
- TrvActionFontShrinkOnePoint 96
- TrvActionFontGrowOnePoint 92
- TrvActionInsertHyperlink 138

1.3.4 Paragraph

1.3.4.1 TrvActionAlignCenter

TrvActionAlignCenter is the action for "Align Center" command.

Unit RichViewActions 50;

Syntax

```
TrvActionAlignCenter = class(TrvActionAlignment 106)
```

Hierarchy

TObject
TPersistent
TComponent
TBasicAction
TContainedAction
TCustomAction
TAction
TrvCustomAction 206
TrvAction 195
TrvActionParaStyles 128
TrvActionAlignment 106

Description

This action does not introduce any new properties in addition to properties of TrvAction.

This action aligns the selected paragraphs to the center.

This action changes Alignment property for styles of the selected paragraphs (RVStyle.ParaStyles[].Alignment) to *rvaCenter*.

This is a checkbox-like action, its Checked property is updated depending on the current paragraph style.

1.3.4.2 TrvActionAlignCustomJustify

TrvActionAlignCustomJustify is a base class for the actions changing alignment of selected paragraphs to the both left and right sides.

Unit RichViewActions 50;

Syntax

```
TrvActionAlignCustomJustify = class(TrvActionAlignment 106)
```

Hierarchy

TObject
TPersistent
TComponent
TBasicAction
TContainedAction
TCustomAction
TAction
TrvCustomAction 206
TrvAction 195
TrvActionParaStyles 128
TrvActionAlignment 106

Description

Properties:

LastLineAlignment 104 is applied to the selected paragraphs, if UseLastLineAlignment 104=*True*.

This action is not used directly. The following actions are inherited from it:

- TrvActionAlignLeft 106
- TrvActionAlignRight 107
- TrvActionAlignCenter 103
- TrvActionAlignJustify 105

1.3.4.2.1 Properties

1.3.4.2.1.1 TrvActionAlignCustomJustify.LastLineAlignment, UseLastLineAlignment

Defines the alignment applied to the last line of paragraphs.

```
property LastLineAlignment: TRVLastLineAlignment;
property UseLastLineAlignment: Boolean;
```

If **UseLastLineAlignment**=*True*, **LastLineAlignment** is applied to the selected paragraphs.

Default values:

- **UseLastLineAlignment**:*True*
- **LastLineAlignment**:
  - in TrvActionAlignJustify 105: *rvllaDefault*
  - in TrvActionAlignDistribute: *rvllaJustify*

1.3.4.3 TrvActionAlignDistribute

TrvActionAlignDistribute is the action for "Distribute" command.

Unit RichViewActions 50;

Syntax

```
TrvActionAlignDistribute = class(TrvActionAlignCustomJustify 104)
```

Hierarchy

TObject
TPersistent
TComponent
TBasicAction
TContainedAction
TCustomAction
TAction
TrvCustomAction 206
TrvAction 195
TrvActionParaStyles 128
TrvActionAlignment 106
TrvActionAlignCustomJustify 104

Description

This action does not introduce any new properties in addition to properties of TrvActionAlignCustomJustify.

This action aligns the selected paragraphs both to the right and to the left sides by adding space between all characters.

This action changes Alignment property for styles of the selected paragraphs (RVStyle.ParaStyles[].Alignment) to *rvaDistribute*. Additionally, it may change LastLineAlignment property to LastLineAlignment 104.

This is a checkbox-like action, its Checked property is updated depending on the current paragraph style.

1.3.4.4 TrvActionAlignJustify

TrvActionAlignJustify is the action for "Justify" command.

Unit RichViewActions 50;

Syntax

```
TrvActionAlignJustify = class(TrvActionAlignCustomJustify 104)
```

Hierarchy

TObject
TPersistent
TComponent
TBasicAction
TContainedAction
TCustomAction
TAction
TrvCustomAction 206
TrvAction 195
TrvActionParaStyles 128
TrvActionAlignment 106
TrvActionAlignCustomJustify 104

Description

This action does not introduce any new properties in addition to properties of TrvActionAlignCustomJustify.

This action aligns the selected paragraphs both to the right and to the left sides by adding space between words.

This action changes Alignment property for styles of the selected paragraphs (RVStyle.ParaStyles[].Alignment) to *rvaJustify*. Additionally, it may change LastLineAlignment property to LastLineAlignment 104.

This is a checkbox-like action, its Checked property is updated depending on the current paragraph style.

1.3.4.5 TrvActionAlignLeft

TrvActionAlignLeft is the action for "Align Left" command.

Unit RichViewActions 50;

Syntax

```
TrvActionAlignLeft = class(TrvActionAlignment 106)
```

Hierarchy

TObject
TPersistent
TComponent
TBasicAction
TContainedAction
TCustomAction
TAction
TrvCustomAction 206
TrvAction 195
TrvActionParaStyles 128
TrvActionAlignment 106

Description

This action does not introduce any new properties in addition to properties of TrvAction.

This action aligns the selected paragraphs to the left.

This action changes Alignment property for styles of the selected paragraphs (RVStyle.ParaStyles[].Alignment) to *rvaLeft*.

This is a checkbox-like action, its Checked property is updated depending on the current paragraph style.

1.3.4.6 TrvActionAlignment

TrvActionAlignment is a base class for the actions changing alignment of the selected paragraphs.

Unit RichViewActions 50;

Syntax

```
TrvActionAlignment = class(TrvActionParaStyles 128)
```

Hierarchy

TObject
TPersistent
TComponent
TBasicAction
TContainedAction
TCustomAction
TAction
TrvCustomAction 206
TrvAction 195
TrvActionParaStyles 128

Description

This action does not introduce any new properties in addition to properties of TrvAction.

This action is not used directly. The following actions are inherited from it:

- TrvActionAlignLeft 106
- TrvActionAlignRight 107
- TrvActionAlignCenter 103

- TrvActionAlignJustify 105
- TrvActionAlignDistribute 104

1.3.4.7 TrvActionAlignRight

TrvActionAlignRight is the action for "Align Right" command.

Unit RichViewActions 50;

Syntax

```
TrvActionAlignRight = class(TrvActionAlignment 106)
```

Hierarchy

TObject
TPersistent
TComponent
TBasicAction
TContainedAction
TCustomAction
TAction
TrvCustomAction 206
TrvAction 195
TrvActionParaStyles 128
TrvActionAlignment 106

Description

This action does not introduce any new properties in addition to properties of TrvAction.

This action aligns the selected paragraphs to the right side.

This action changes Alignment property for styles of the selected paragraphs (RVStyle.ParaStyles[].Alignment) to *rvaRight*.

This is a checkbox-like action, its Checked property is updated depending on the current paragraph style.

1.3.4.8 TrvActionClearBoth

TrvActionClearBoth is the action for "Clear Text Flow at Both Sides"

Unit RichViewActions 50;

Syntax

```
TrvActionClearBoth = class(TrvActionClearTextFlow 109)
```

Hierarchy

TObject
TPersistent
TComponent
TBasicAction
TContainedAction
TCustomAction
TAction
TrvCustomAction 206
TrvAction 195
TrvActionClearTextFlow 109

Description

This action does not allow the selected paragraphs flowing around left- and right-aligned pictures. The selected paragraphs will be placed below side-aligned pictures.

This actions calls RichViewEdit.ClearTextFlow(*True*, *True*).

This action does not introduce any new properties in addition to properties of TrvAction.

1.3.4.9 TrvActionClearLeft

TrvActionClearLeft is the action for "Clear Text Flow at Left Side"

Unit RichViewActions[50];

Syntax

```
TrvActionClearLeft = class(TrvActionClearTextFlow[109])
```

Hierarchy

TObject
TPersistent
TComponent
TBasicAction
TContainedAction
TCustomAction
TAction
TrvCustomAction[206]
TrvAction[195]
TrvActionClearTextFlow[109]

Description

This action does not allow the selected paragraphs flowing around left-aligned pictures. The selected paragraphs will be placed below left-aligned pictures. Text flow around right-aligned pictures is allowed.

This actions calls RichViewEdit.ClearTextFlow(*True*, *False*).

This action does not introduce any new properties in addition to properties of TrvAction.

1.3.4.10 TrvActionClearNone

TrvActionClearNone is the action for "Normal Text Flow"

Unit RichViewActions[50];

Syntax

```
TrvActionClearNone = class(TrvActionClearTextFlow[109])
```

Hierarchy

TObject
TPersistent
TComponent
TBasicAction
TContainedAction
TCustomAction
TAction
TrvCustomAction[206]
TrvAction[195]
TrvActionClearTextFlow[109]

Description

This action allows the selected paragraphs flowing around left- and right-aligned pictures.

This actions calls RichViewEdit.ClearTextFlow(*False*, *False*).

This action does not introduce any new properties in addition to properties of TrvAction.

1.3.4.11 TrvActionClearRight

TrvActionClearRight is the action for "Clear Text Flow at Right Side"

Unit RichViewActions 50;

Syntax

```
TrvActionClearRight = class(TrvActionClearTextFlow 109)
```

Hierarchy

TObject
TPersistent
TComponent
TBasicAction
TContainedAction
TCustomAction
TAction
TrvCustomAction 206
TrvAction 195
TrvActionClearTextFlow 109

Description

This action does not allow the selected paragraphs flowing around right-aligned pictures. The selected paragraphs will be placed below right-aligned pictures. Text flow around left-aligned pictures is allowed.

This actions calls RichViewEdit.ClearTextFlow(*False*, *True*).

This action does not introduce any new properties in addition to properties of TrvAction.

1.3.4.12 TrvActionClearTextFlow

TrvActionClearTextFlow is a base class for the actions clearing a text flow around left- and right-aligned pictures.

Unit RichViewActions 50;

Syntax

```
TrvActionClearTextFlow = class(TrvAction 195)
```

Hierarchy

TObject
TPersistent
TComponent
TBasicAction
TContainedAction
TCustomAction
TAction
TrvCustomAction 206
TrvAction 195

Description

This action does not introduce any new properties in addition to properties of TrvAction.

This action is not used directly. The following actions are inherited from it:

- TrvActionClearLeft 108
- TrvActionClearRight 109
- TrvActionClearBoth 107
- TrvActionClearNone 108

1.3.4.13 TrvActionCustomParaListSwitcher

TrvActionCustomParaListSwitcher is a base class for the actions toggling paragraphs' bullets or numbering on/off.

Unit RichViewActions [50];

Syntax

```
TrvActionCustomParaListSwitcher = class(TrvAction [195])
```

Hierarchy

TObject
TPersistent
TComponent
TBasicAction
TContainedAction
TCustomAction
TAction
TrvCustomAction [206]
TrvAction [195]

Description

This action is checked when the current paragraph has a list marker with properties identical to the properties of ListLevels [110]. In this case, applying the action removes bullets and numbering in all selected paragraphs. Otherwise, applying this action applies a list style with levels specified in ListLevels [110] to all selected paragraphs. The 0th list level is applied to paragraphs that did not have list markers; for other paragraphs, level indices are not changed.

This action is not used directly. The following actions are inherited from it:

- TrvActionParaBullets [117]
- TrvActionParaNumbering [127]

1.3.4.13.1 Properties

1.3.4.13.1.1 TrvActionCustomParaListSwitcher.IndentStep

Defines the step for increasing LeftIndent and MarkerIndent of ListLevels [110].

```
property IndentStep: TRVStyleLength;
```

This value is measured in GetControlPanel [208].UnitsProgram [33].

Assigning value to this property resets ListLevels [110] to default value.

Default value:

24

This default value assumes that this property is measured in pixels. For twips, this value may be too small.

See also:

- TrvActionIndent [111].IndentStep [111]
- TrvActionParaList [125].IndentStep [125]

1.3.4.13.1.2 TrvActionCustomParaListSwitcher.ListLevels

Specifies list levels for applying to the selected paragraphs.

```
property ListLevels: TRVListLevelCollection;
```

Subproperties of this property are measured in GetControlPanel [208].UnitsProgram [33].

Default value for TrvActionParaBullets [117]:

9 levels, all with ListType=*rvlstBullet*, FirstIndent=0, MarkerIndent is increased by IndentStep [110]*2 starting from

0, LeftIndent is increased by IndentStep[110]*2 starting from IndentStep[110], Font.Size=12, disk/circle/square cycled 3 times.

Default value for TrvActionParaNumbering[127]:

9 levels, all with ListType=*rvlstDecimal*, FirstIndent=0, MarkerIndent is increased by IndentStep[110]*2 starting from 0, LeftIndent is increased by IndentStep[110]*2 starting from IndentStep[110], Font.Size=10, format sting displays "N."

1.3.4.14 TrvActionIndent

TrvActionIndent is a base class for the actions changing indents in the selected paragraphs.

Unit RichViewActions[50];

Syntax

```
TrvActionIndent = class(TrvActionParaStyles[128])
```

Hierarchy

TObject
TPersistent
TComponent
TBasicAction
TContainedAction
TCustomAction
TAction
TrvCustomAction[206]
TrvAction[195]
TrvActionParaStyles[128]

Description

The actions inherited from this class perform the following operations:

- change LeftIndent property of styles for the selected paragraphs (RVStyle.ParaStyles[].LeftIndent); for paragraphs styles having BiDiMode=*rvbdRightToLeft*, RightIndent is changed instead;
- change list levels for all paragraphs with bullets&numbering (by calling TCustomRichViewEdit.ChangeListLevels).

The actions do not change indents of list styles for bulleted or numbered paragraphs, they promote/demote their levels instead.

To do: to implement changing of LeftIndent property of list style, if bullets or numbering has only one level.

This action is not used directly. The following actions are inherited from it:

- TrvActionIndentDec[112]
- TrvActionIndentInc[112]

1.3.4.14.1 Properties

1.3.4.14.1.1 TrvActionIndent.IndentStep

Defines the step for paragraph indenting or unindenting.

```
property IndentStep: TRVStyleLength;
```

This value is measured in GetControlPanel[208].UnitsProgram[33].

Default value:

24

This default value assumes that this property is measured in pixels. For twips, this value may be too small.

See also:

- TrvActionCustomParaListSwitcher[110].IndentStep[110]

- TrvActionParaList[125].IndentStep[125]

1.3.4.15 TrvActionIndentDec

TrvActionIndentDec is the action for "Decrease Indent" command.

Unit RichViewActions[50];

Syntax

```
TrvActionIndentDec = class(TrvActionIndent[111])
```

Hierarchy

TObject
TPersistent
TComponent
TBasicAction
TContainedAction
TCustomAction
TAction
TrvCustomAction[206]
TrvAction[195]
TrvActionParaStyles[128]
TrvActionIndent[111]

Description

This action does not introduce any new properties in addition to properties of TrvActionIndent.

The action does not allow changing the paragraph indent to a negative value. If TCustomRichViewEdit.LeftMargin +LeftIndent+FirstIndent becomes negative, the action changes FirstIndent to -TCustomRichViewEdit.LeftMargin-LeftIndent (for right-to-left paragraphs, these formulas use RightIndent instead).

1.3.4.16 TrvActionIndentInc

TrvActionIndentInc is the action for "Increase Indent" command.

Unit RichViewActions[50];

Syntax

```
TrvActionIndentInc = class(TrvActionIndent[111])
```

Hierarchy

TObject
TPersistent
TComponent
TBasicAction
TContainedAction
TCustomAction
TAction
TrvCustomAction[206]
TrvAction[195]
TrvActionParaStyles[128]
TrvActionIndent[111]

Description

The action does not allow changing the paragraph indent above the MaxIndent[113].

1.3.4.16.1 Properties

1.3.4.16.1.1 TrvActionIndentInc.IndentMax

Defines the maximal possible value for the indent.

```
property IndentMax: TRVStyleLength;
```

The action does not increase the paragraph's indent above this value.

This value is measured in GetControlPanel[208].UnitsProgram[33].

Default value:

200

This default value assumes that this property is measured in pixels. For twips, this value may be too small.

1.3.4.17 TrvActionLineSpacing

TrvActionLineSpacing is a base class for the actions changing line spacing of the selected paragraphs.

Unit RichViewActions[50];

Syntax

```
TrvActionLineSpacing = class(TrvActionParaStyles[128])
```

Hierarchy

TObject
TPersistent
TComponent
TBasicAction
TContainedAction
TCustomAction
TAction
TrvCustomAction[206]
TrvAction[195]
TrvActionParaStyles[128]

Description

This action does not introduce any new properties in addition to properties of TrvAction.

This action is not used directly. The following actions are inherited from it:

- TrvActionLineSpacing100[113]
- TrvActionLineSpacing150[114]
- TrvActionLineSpacing200[114]

These actions change the following properties of styles of the selected paragraphs (RVStyle.ParaStyles[]):

- LineSpacingType := *rvlsPercent*
- LineSpacing

1.3.4.18 TrvActionLineSpacing100

TrvActionLineSpacing100 is the action for "Single Line Spacing" command.

Unit RichViewActions[50];

Syntax

```
TrvActionLineSpacing100 = class(TrvActionLineSpacing[113])
```

Hierarchy

TObject
TPersistent

TComponent
TBasicAction
TContainedAction
TCustomAction
TAction
TrvCustomAction 206
TrvAction 195
TrvActionParaStyles 128
TrvActionLineSpacing 113

Description

This action does not introduce any new properties in addition to properties of TrvAction.

This actions changes the following properties of styles of the selected paragraphs (RVStyle.ParaStyles[]):

- LineSpacingType := *rvlsPercent*
- LineSpacing := 100

1.3.4.19 TrvActionLineSpacing150

TrvActionLineSpacing150 is the action for "1.5 Line Spacing" command.

Unit RichViewActions 50;

Syntax

```
TrvActionLineSpacing150 = class(TrvActionLineSpacing 113)
```

Hierarchy

TObject
TPersistent
TComponent
TBasicAction
TContainedAction
TCustomAction
TAction
TrvCustomAction 206
TrvAction 195
TrvActionParaStyles 128
TrvActionLineSpacing 113

Description

This action does not introduce any new properties in addition to properties of TrvAction.

This actions changes the following properties of styles of the selected paragraphs (RVStyle.ParaStyles[]):

- LineSpacingType := *rvlsPercent*
- LineSpacing := 150

1.3.4.20 TrvActionLineSpacing200

TrvActionLineSpacing200 is the action for "Double Line Spacing" command.

Unit RichViewActions 50;

Syntax

```
TrvActionLineSpacing200 = class(TrvActionLineSpacing 113)
```

Hierarchy

TObject
TPersistent
TComponent

TBasicAction
TContainedAction
TCustomAction
TAction
TrvCustomAction 206
TrvAction 195
TrvActionParaStyles 128
TrvActionLineSpacing 113

Description

This action does not introduce any new properties in addition to properties of TrvAction.

This actions changes the following properties of styles of the selected paragraphs (RVStyle.ParaStyles[]):

- LineSpacingType := *rvlsPercent*
- LineSpacing := 200

1.3.4.21 TrvActionParaBiDi

TrvActionParaBiDi is a base class for the actions changing the default text flow direction in the selected paragraphs.

Unit RichViewActions 50;

Syntax

```
TrvActionParaBiDi = class(TrvActionParaStyles 128)
```

Hierarchy

TObject
TPersistent
TComponent
TBasicAction
TContainedAction
TCustomAction
TAction
TrvCustomAction 206
TrvAction 195
TrvActionParaStyles 128

Description

This action does not introduce any new properties in addition to properties of TrvAction.

This action changes BiDiMode property for styles of the selected paragraphs (RVStyle.ParaStyles[].BiDiMode).

This action is not used directly. The following actions are inherited from it:

- TrvActionParaLTR 126
- TrvActionParaRTL 128

If you use these actions, it's recommended to set TCustomRichViewEdit.BiDiMode to *rvbdLeftToRight* or *rvbdRightToLeft*.

Text flow direction specified in text styles has higher priority than text flow direction specified in paragraph styles.

See also:

- TrvActionTextBiDi 100
- TrvActionTextLTR 101
- TrvActionTextRTL 102

1.3.4.22 TrvActionParaBorder

TrvActionParaBorder is the action for "Paragraph Border and Background" command.

Unit RichViewActions [50];

Syntax

```
TrvActionParaBorder = class(TrvActionParaStyles [128])
```

Hierarchy

TObject
TPersistent
TComponent
TBasicAction
TContainedAction
TCustomAction
TAction
TrvCustomAction [206]
TrvAction [195]
TrvActionParaStyles [128]

Description

If UserInterface [117]=*False*, the action applies its properties to the selected paragraphs. The list of properties to apply is specified in ValidProperties [117].

If UserInterface [117]=*True*, the action displays a dialog (initialized with properties of the selected paragraphs), assigns the changes to the action's properties, and then applies these properties to the selected paragraph.

This action allows changing attributes of paragraph border and background. For changing other paragraph attributes, see TrvActionParagraph [120].

1.3.4.22.1 Properties

1.3.4.22.1.1 TrvActionParaBorder.Background

Specifies background properties for applying to the selected paragraphs.

```
property Background: TRVBackgroundRect;
```

If UserInterface [117]=*False* and *rvpibBackground_*** in ValidProperties [117], the action applies selected subproperties of this property to the selected paragraphs.

If UserInterface [117]=*True*, the action displays a dialog (initialized with properties of the selected paragraphs); if the user changed values of subproperties of this property in the dialog, corresponding *rvpibBackground_*** values are included in ValidProperties [117], and the action applies them to the selected paragraphs.

This property is applied to the paragraph style's (RVStyle.ParaStyles[]) property of the same name.

Subproperties of this property are measured in the same units (pixels or twips) as properties of the target editor (i.e. in RVStyle.Units).

1.3.4.22.1.2 TrvActionParaBorder.Border

Specifies border properties for applying to the selected paragraphs.

```
property Border: TRVBorder;
```

If UserInterface [117]=*False* and *rvpibBorder_*** in ValidProperties [117], the action applies selected subproperties of this property to the selected paragraphs.

If UserInterface [117]=*True*, the action displays a dialog (initialized with properties of the selected paragraphs); if the user changed values of subproperties of this property in the dialog, corresponding *rvpibBorder_*** values are included in ValidProperties [117], and the action applies them to the selected paragraphs.

This property is applied to the paragraph style's (RVStyle.ParaStyles[]) property of the same name.

Subproperties of this property are measured in the same units (pixels or twips) as properties of the target editor (i.e. in RVStyle.Units).

1.3.4.22.1.3 TrvActionParaBorder.UserInterface

Allows working with or without user interaction.

```
property UserInterface: Boolean;
```

If UserInterface=*False*, the action applies its properties to the selected paragraphs.

If UserInterface=*True*, the action displays a dialog, assigns changes to the action's properties, and then applies these properties to the selected paragraphs.

Default value:

True

1.3.4.22.1.4 TrvActionParaBorder.ValidProperties

Defines a set of properties for applying to the selected paragraphs.

```
type
  TRVParaInfoBorderProperty = (rvpibBackground_Color,
    rvpibBackground_BO_Left, rvpibBackground_BO_Top, rvpibBackground_BO_Right, rvpibBackg
    rvpibBorder_Color, rvpibBorder_Style, rvpibBorder_Width, rvpibBorder_InternalWidth,
    rvpibBorder_BO_Left, rvpibBorder_BO_Top, rvpibBorder_BO_Right, rvpibBorder_BO_Bottom,
    rvpibBorder_Vis_Left, rvpibBorder_Vis_Top, rvpibBorder_Vis_Right, rvpibBorder_Vis_Bot
  TRVParaInfoBorderProperties = set of TRVParaInfoBorderProperty;

property ValidProperties: TRVParaInfoBorderProperty;
```

If UserInterface[117]=*False,* assign a subset of properties (for applying) to ValidProperties .

If UserInterface[117]=*True*, this property is maintained automatically. After displaying a dialog, this property contains a set of properties corresponding to non-blank fields in the dialog.

1.3.4.23 TrvActionParaBullets

TrvActionParaBullets is the action for "Bullets" command.

Unit RichViewActions[50];

Syntax

```
TrvActionParaBullets = class(TrvActionCustomParaListSwitcher[110])
```

Hierarchy

TObject
TPersistent
TComponent
TBasicAction
TContainedAction
TCustomAction
TAction
TrvCustomAction[206]
TrvAction[195]
TrvActionCustomParaListSwitcher[110]

Description

This action applies or removes bullets to the selected paragraphs.

This action does not introduce any new properties in addition to properties of TrvActionCustomParaListSwitcher.

ListLevels[110] must not contains numbered levels.

When applying list, the action tries to reuse existing list styles with identical properties, if they exist.

1.3.4.24 TrvActionParaColor

TrvActionParaColor is the action for changing background color of the selected paragraphs.

Unit RichViewActions[50];

Syntax

```
TrvActionParaColor = class(TrvActionParaCustomColor[119])
```

Hierarchy

TObject
TPersistent
TComponent
TBasicAction
TContainedAction
TCustomAction
TAction
TrvCustomAction[206]
TrvAction[195]
TrvActionCustomColor[197]
TrvActionParaCustomColor[119]

Description

This action does not introduce any new properties and events in addition to properties and events of TrvActionCustomColor[197].

On execution, this action changes Background.Color property of styles of the selected paragraphs (RVStyle.ParaStyles[].Background.Color).

See also the list of color changing actions in the topic about TrvActionCustomColor[197].

1.3.4.25 TrvActionParaColorAndPadding

TrvActionParaColorAndPadding is the action for changing background color and padding of the selected paragraphs.

Unit RichViewActions[50];

Syntax

```
TrvActionParaColorAndPadding = class(TrvActionParaColor[118])
```

Hierarchy

TObject
TPersistent
TComponent
TBasicAction
TContainedAction
TCustomAction
TAction
TrvCustomAction[206]
TrvAction[195]
TrvActionCustomColor[197]
TrvActionParaCustomColor[119]

TrvActionParaColor 118

Description

This action is reserved. It does not have Caption and Hint assigned by the localization procedure. If you use this action, assign its Caption and Hint yourself.

This action does not introduce any new events in addition to events of TrvActionCustomColor 197.

On execution, this action changes Background.Color and Background.BorderOffsets properties of styles of the selected paragraphs (RVStyle.ParaStyles[].Background.Color and RVStyle.ParaStyles[].Background.BorderOffsets). The specific (left/top/right/bottom) padding value specified in Padding 119 property is assigned only if the corresponding property of UsePadding 119 is *True*. If the padding exceeds the paragraph's indents/spacing (LeftIndent/SpaceBefore/RightIndent/SpaceAfter), indents/spacing is increased.

Values of Padding 119 and UsePadding 119 properties must be assigned by the programmer.

1.3.4.25.1 Properties

1.3.4.25.1.1 TrvActionParaColorAndPadding.Padding

Specifies the padding for applying to the selected paragraphs.

```
property Padding: TRVRect;
```

This property is applied to Background.BorderOffsets property of paragraph styles (RVStyle.ParaStyles[].Background.BorderOffsets).

If Padding.Left exceeds LeftIndent property of the paragraph style, LeftIndent is increased.

If Padding.Right exceeds RightIndent property of the paragraph style, RightIndent is increased.

If Padding.Top exceeds SpaceBefore property of the paragraph style, SpaceBefore is increased.

If Padding.Bottom exceeds SpaceAfter property of the paragraph style, SpaceAfter is increased.

UsePadding 119 property specifies which sides of padding are applied.

1.3.4.25.1.2 TrvActionParaColorAndPadding.UsePadding

Specifies sides of paragraphs for applying Padding 119.

```
property UsePadding: TRVBooleanRect;
```

1.3.4.26 TrvActionParaCustomColor

TrvActionParaCustomColor is a base class for the actions changing color properties of the selected paragraphs.

Unit RichViewActions 50;

Syntax

```
TrvActionParaCustomColor = class(TrvActionCustomColor 197)
```

Hierarchy

TObject
TPersistent
TComponent
TBasicAction
TContainedAction
TCustomAction
TAction
TrvCustomAction 206
TrvAction 195

TrvActionCustomColor 197
TrvActionParaCustomColor 119

Description

This action does not introduce any new properties and events in addition to properties and events of TrvActionCustomColor 197.

This action is not used directly. The following actions are inherited from it:

- TrvActionParaColor 118
- TrvActionParaColorAndPadding 118

1.3.4.27 TrvActionParagraph

TrvActionParagraph is the action for "Paragraph" command.

Unit RichViewActions 50;

Syntax

```
TrvActionParagraph = class(TrvActionParaStyles 128)
```

Hierarchy

TObject
TPersistent
TComponent
TBasicAction
TContainedAction
TCustomAction
TAction
TrvCustomAction 206
TrvAction 195
TrvActionParaStyles 128

Description

If UserInterface 124=*False*, the action applies its properties to the selected paragraphs. The list of properties to apply is specified in ValidProperties 124.

If UserInterface 124=*True*, the action displays a paragraph dialog (initialized with properties of the selected paragraphs), assigns the changes to the action's properties, and then applies these properties to the selected paragraph.

This action allows changing almost all paragraph attributes, except for border and background (see TrvActionParaBorder 116).

1.3.4.27.1 Properties

1.3.4.27.1.1 TrvActionParagraph.Alignment

Specifies the alignment for applying to the selected paragraphs.

```
property Alignment: TRVAlignment;
```

If UserInterface 124=*False* and *rvpimAlignment* in ValidProperties 124, the action applies this property to the selected paragraphs.

If UserInterface 124=*True*, the action displays a paragraph dialog (initialized with properties of the selected paragraphs); if the user changed value of this property in the dialog, *rvpimAlignment* is included in ValidProperties 124, and the action applies this property to the selected paragraphs.

This property is applied to the paragraph style's (RVStyle.ParaStyles[]) property of the same name.

1.3.4.27.1.2 TrvActionParagraph.DeleteAllTabs

Instructs to delete all tab-stop positions from styles of the selected paragraphs.

```
property DeleteAllTabs: Boolean;
```

If *True*, all existing tab-stop positions are deleted from styles of the selected paragraphs.

If *False*, only tab-stops specified in TabsToDelete[124] are deleted.

After that, tab-stops from Tabs[124] are added.

All these properties are applied only if ValidProperties[124] contains *rvpimTabs*.

If UserInterface[124]=*True*, you should assign these properties yourself.

If UserInterface[124]=*False*, these properties are defined by the user in a dialog.

1.3.4.27.1.3 TrvActionParagraph.FirstIndent

Specifies the first line indent for applying to the selected paragraphs.

```
property FirstIndent: TRVStyleLength;
```

If UserInterface[124]=*False* and *rvpimFirstIndent* in ValidProperties[124], the action applies this property to the selected paragraphs.

If UserInterface[124]=*True*, the action displays a paragraph dialog (initialized with properties of the selected paragraphs); if the user changed value of this property in the dialog, *rvpimFirstIndent* is included in ValidProperties[124], and the action applies this property to the selected paragraphs.

This property is applied to the paragraph style's (RVStyle.ParaStyles[]) property of the same name.

This value is measured in the same units (pixels or twips) as properties of the target editor (i.e. in RVStyle.Units).

1.3.4.27.1.4 TrvActionParagraph.KeepLinesTogether

Specifies the keep-lines-together option for applying to the selected paragraphs.

```
property KeepLinesTogether: Boolean;
```

If UserInterface[124]=*False* and *rvpimKeepLinesTogether* in ValidProperties[124], the action applies this property to the selected paragraphs.

If UserInterface[124]=*True*, the action displays a paragraph dialog (initialized with properties of the selected paragraphs); if the user changed value of this property in the dialog, *rvpimKeepLinesTogether* is included in ValidProperties[124], and the action applies this property to the selected paragraphs.

This option adds/removes *rvpaoKeepLinesTogether* in the Options property of styles of the selected paragraphs (RVStyle.ParaStyles[].Options).

1.3.4.27.1.5 TrvActionParagraph.KeepWithNext

Specifies the keep-with-next option for applying to the selected paragraphs.

```
property KeepWithNext: Boolean;
```

If UserInterface[124]=*False* and *rvpimKeepWithNext* in ValidProperties[124], the action applies this property to the selected paragraphs.

If UserInterface[124]=*True*, the action displays a paragraph dialog (initialized with properties of the selected paragraphs); if the user changed value of this property in the dialog, *rvpimKeepWithNext* is included in ValidProperties[124], and the action applies this property to the selected paragraphs.

This option adds/removes *rvpaoKeepWithNext* in the Options property of styles of the selected paragraphs (RVStyle.ParaStyles[].Options).

1.3.4.27.1.6 TrvActionParagraph.LastLineAlignment

Specifies the alignment for applying to the last line of selected justified and distributed paragraphs.

```
property LastLineAlignment: TRVLastLineAlignment;
```

If UserInterface[124]=*False* and *rvpimLastLineAlignment* in ValidProperties[124], the action applies this property to the selected paragraphs.

If UserInterface[124]=*True*, the action displays a paragraph dialog (initialized with properties of the selected paragraphs); if the user changed value of this property in the dialog, *rvpimLastLineAlignment* is included in ValidProperties[124], and the action applies this property to the selected paragraphs.

This property is applied to the paragraph style's (RVStyle.ParaStyles[]) property of the same name.

1.3.4.27.1.7 TrvActionParagraph.LeftIndent

Specifies the left indent for applying to the selected paragraphs.

```
property LeftIndent: TRVStyleLength;
```

If UserInterface[124]=*False* and *rvpimLeftIndent* in ValidProperties[124], the action applies this property to the selected paragraphs.

If UserInterface[124]=*True*, the action displays a paragraph dialog (initialized with properties of the selected paragraphs); if the user changed value of this property in the dialog, *rvpimLeftIndent* is included in ValidProperties[124], and the action applies this property to the selected paragraphs.

This property is applied to the paragraph style's (RVStyle.ParaStyles[]) property of the same name.

This value is measured in the same units (pixels or twips) as properties of the target editor (i.e. in RVStyle.Units).

1.3.4.27.1.8 TrvActionParagraph.LineSpacing

Specifies the line spacing value for applying to the selected paragraphs.

```
property LineSpacing: TRVLineSpacingValue;
```

If UserInterface[124]=*False* and *rvpimLineSpacing* in ValidProperties[124], the action applies this property to the selected paragraphs.

If UserInterface[124]=*True*, the action displays a paragraph dialog (initialized with properties of the selected paragraphs); if the user changed value of this property in the dialog, *rvpimLineSpacing* is included in ValidProperties[124], and the action applies this property to the selected paragraphs.

This property is applied to the paragraph style's (RVStyle.ParaStyles[]) property of the same name.

Meaning of this property depends on the value of LineSpacingType[122] property. These two properties are applied together.

If LineSpacingType[122]<>*rvlsPercent*, this value is measured in the same units (pixels or twips) as properties of the target editor (i.e. in RVStyle.Units).

1.3.4.27.1.9 TrvActionParagraph.LineSpacingType

Specifies the line spacing type for applying to the selected paragraphs.

```
property LineSpacingType: TRVLineSpacingType;
```

If UserInterface[124]=*False* and *rvpimLineSpacing* in ValidProperties[124], the action applies this property to the selected paragraphs.

If UserInterface[124]=*True*, the action displays a paragraph dialog (initialized with properties of the selected paragraphs); if the user changed value of this property in the dialog, *rvpimLineSpacing* is included in ValidProperties[124], and the action applies this property to the selected paragraphs.

This property is applied to the paragraph style's (RVStyle.ParaStyles[]) property of the same name.

This property is applied together with LineSpacing 122 property.

1.3.4.27.1.10 TrvActionParagraph.OutlineLevel

Specifies the line spacing type for applying to the selected paragraphs.

```
property OutlineLevel: Integer;
```

If UserInterface 124=*False* and *rvpimOutlineLevel* in ValidProperties 124, the action applies this property to the selected paragraphs.

If UserInterface 124=*True*, the action displays a paragraph dialog (initialized with properties of the selected paragraphs); if the user changed value of this property in the dialog, *rvpimOutlineLevel* is included in ValidProperties 124, and the action applies this property to the selected paragraphs.

This property is applied to the paragraph style's (RVStyle.ParaStyles[]) property of the same name.

1.3.4.27.1.11 TrvActionParagraph.RightIndent

Specifies the right indent for applying to the selected paragraphs.

```
property RightIndent: TRVStyleLength;
```

If UserInterface 124=*False* and *rvpimRightIndent* in ValidProperties 124, the action applies this property to the selected paragraphs.

If UserInterface 124=*True*, the action displays a paragraph dialog (initialized with properties of the selected paragraphs); if the user changed value of this property in the dialog, *rvpimRightIndent* is included in ValidProperties 124, and the action applies this property to the selected paragraphs.

This property is applied to the paragraph style's (RVStyle.ParaStyles[]) property of the same name.

This value is measured in the same units (pixels or twips) as properties of the target editor (i.e. in RVStyle.Units).

1.3.4.27.1.12 TrvActionParagraph.SpaceAfter

Specifies spacing for applying to the bottom side of the selected paragraphs.

```
property SpaceAfter: TRVStyleLength;
```

If UserInterface 124=*False* and *rvpimSpaceAfter* in ValidProperties 124, the action applies this property to the selected paragraphs.

If UserInterface 124=*True*, the action displays a paragraph dialog (initialized with properties of the selected paragraphs); if the user changed value of this property in the dialog, *rvpimSpaceAfter* is included in ValidProperties 124, and the action applies this property to the selected paragraphs.

This property is applied to the paragraph style's (RVStyle.ParaStyles[]) property of the same name.

This value is measured in the same units (pixels or twips) as properties of the target editor (i.e. in RVStyle.Units).

1.3.4.27.1.13 TrvActionParagraph.SpaceBefore

Specifies spacing for applying to the top side of the selected paragraphs.

```
property SpaceBefore: TRVStyleLength;
```

If UserInterface 124=*False* and *rvpimSpaceBefore* in ValidProperties 124, the action applies this property to the selected paragraphs.

If UserInterface 124=*True*, the action displays a paragraph dialog (initialized with properties of the selected paragraphs); if the user changed value of this property in the dialog, *rvpimSpaceBefore* is included in

ValidProperties[124], and the action applies this property to the selected paragraphs.

This property is applied to the paragraph style's (RVStyle.ParaStyles[]) property of the same name.

This value is measured in the same units (pixels or twips) as properties of the target editor (i.e. in RVStyle.Units).

1.3.4.27.1.14 TrvActionParagraph.Tabs

Contains a list of tab-stops for adding to styles of the selected paragraphs.

```
property Tabs: TRVTabInfos;
```

Before applying this property, some (TabsToDelete[124]) or all (DeleteAllTabs[121]) tab-stops are deleted.

All these properties are applied only if ValidProperties[124] contains *rvpimTabs*.

If UserInterface[124]=*True*, you should assign these properties yourself.

If UserInterface[124]=*False*, these properties are defined by the user in a dialog.

Position properties of tabs are measured in the same units (pixels or twips) as properties of the target editor (i.e. in RVStyle.Units).

1.3.4.27.1.15 TrvActionParagraph.TabsToDelete

Contains a list of tab stop positions for deleting from styles of the selected paragraphs.

```
property TabsToDelete: TRVIntegerList;
```

If DeleteAllTabs[121]=*True*, all tab-stops are deleted instead.

After that, tab-stops from Tabs[124] are added.

All these properties are applied only if ValidProperties[124] contains *rvpimTabs*.

If UserInterface[124]=*True*, you should assign these properties yourself.

If UserInterface[124]=*False*, these properties are defined by the user in a dialog.

1.3.4.27.1.16 TrvActionParagraph.UserInterface

Allows working with or without user interaction.

```
property UserInterface: Boolean;
```

If UserInterface=*False*, the action applies its properties to the selected paragraphs.

If UserInterface=*True*, the action displays a paragraph dialog, assigns changes to the action's properties, and then applies these properties to the selected paragraphs.

Default value:

True

1.3.4.27.1.17 TrvActionParagraph.ValidProperties

Defines a set of properties for applying to the selected paragraphs.

```
type
  TRVParaInfoMainProperty = (rvpimFirstIndent, rvpimLeftIndent, rvpimRightIndent,
    rvpimSpaceBefore, rvpimSpaceAfter, rvpimAlignment, rvpimLastLineAlignment,
    rvpimOutlineLevel, rvpimLineSpacing,
    rvpimKeepLinesTogether, rvpimKeepWithNext, rvpimTabs);
  TRVParaInfoMainProperties = set of TRVParaInfoMainProperty;

property ValidProperties: TRVParaInfoMainProperties;
```

If UserInterface[124]=*False,* assign a subset of properties (for applying) to ValidProperties .

If UserInterface[124]=*True*, this property is maintained automatically. After displaying a paragraph dialog, this property contains a set of properties corresponding to non-blank fields in the dialog.

1.3.4.28 TrvActionParaList

TrvActionParaList is the action for "Bullets and Numbering" command.

Unit RichViewActions[50];

Syntax

```
TrvActionParaList = class(TrvActionParaStyles[128])
```

Hierarchy

TObject
TPersistent
TComponent
TBasicAction
TContainedAction
TCustomAction
TAction
TrvCustomAction[206]
TrvAction[195]
TrvActionParaStyles[128]

Description

This action displays a bullet & numbering dialog. Two kinds of dialogs are used, depending on the value of GetControlPanel[208].UserInterface[34]: a general dialog and an HTML dialog.

In the general kind of dialog, the user can choose from predefined templates (may be after customizing them). If the caret is placed in the paragraph having bullets or numbering, the current list style takes the place of one of these templates, so the user can customize it. All templates have 9 list levels (maximal possible count that can be saved in RTF). For numbered styles, the user can choose to continue or to reset numbering, or to create a new list.

In the HTML kind of dialog, templates cannot be customized. All templates have 1 level. The "level" field in this dialog actually does not mean a list level, it affects only indenting (MarkerIndent of list level is IndentStep[125] *level). For numbered styles, the user can choose to continue or to reset numbering.

When applying bullets (without numbered levels), the action tries to reuse existing list styles with the same properties, if they exist. The action may apply properties to TrvActionParaBullets[117] actions.

When applying numbering (if at least one level is numbered), the action uses list styles around the selection (if they exist and have identical properties with the chosen properties). If it's not possible, the action tries to reuse unused list styles (if they exist and have identical properties with the chosen properties). Otherwise, a new list is created. If all levels are numbered, the action may apply properties to TrvActionParaNumbering[127] actions.

1.3.4.28.1 Properties

1.3.4.28.1.1 TrvActionParaList.IndentStep

Defines the step for increasing LeftIndent and MarkerIndent of list levels.

```
property IndentStep: TRVStyleLength;
```

This value is measured in GetControlPanel[208].UnitsProgram[33].

This property is used:

- in an HTML-style list&numbering dialog (if GetControlPanel[208].UserInterface[34]=*rvauiHTML*); this dialog creates a paragraph list with MarkerIndent = level*IndentStep, LeftIndent = (level+1)*IndentStep;

- in a general-style dialog, to adjust indents of a predefines set of styles (indents are adjusted when this action is executed for the first time).

Default value:

24

This default value assumes that this property is measured in pixels. For twips, this value may be too small.

See also:

- TrvActionCustomParaListSwitcher[110].IndentStep[110]
- TrvActionIndent[111].IndentStep[111]

1.3.4.28.1.2 TrvActionParaList.UpdateAllActionsOnForm

Allows updating all TrvActionParaBullets[117] and TrvActionParaNumbering[127] actions on the same form as this action.

```
property UpdateAllActionsOnForm: Boolean;
```

If this property is *True:*

- when the user applies a list without numbered levels, the action assigns these levels to ListLevels[110] of all TrvActionParaBullets[117] actions on the same form/datamodule.
- when the user applies a list with all numbered levels, the action assigns these levels to ListLevels[110] of all TrvActionParaNumbering[127] actions on the same form/datamodule.

Default value:

True

See also properties:

- ActionParaNumbering[126]
- ActionParaBullets[126]

1.3.4.28.1.3 TrvActionParaList.ActionParaNumbering

Defines a link to the "Numbering" action.

```
property ActionParaNumbering: TrvActionParaNumbering[127];
```

When the user applies a list with all numbered levels, the action assigns these levels to ListLevels[110] properties to the action specified in this link. If UpdateAllActionsOnForm[126]=*True*, these levels are assigned to ListLevels[110] of all TrvActionParaNumbering[127] actions on the same form/datamodule.

1.3.4.28.1.4 TrvActionParaList.ActionParaBullets

Defines a link to the "Bullets" action.

```
property ActionParaBullets: TrvActionParaBullets[117];
```

When the user applies a list without numbered levels, the action assigns these levels to ListLevels[110] properties to the action specified in this link. If UpdateAllActionsOnForm[126]=*True*, these levels are assigned to ListLevels[110] of all TrvActionParaBullets[117] actions on the same form/datamodule.

1.3.4.29 TrvActionParaLTR

TrvActionParaLTR is the action for "Left to Right" command.

Unit RichViewActions[50];

Syntax

```
TrvActionParaLTR = class(TrvActionParaBiDi[115])
```

Hierarchy

TObject
TPersistent

TComponent
TBasicAction
TContainedAction
TCustomAction
TAction
TrvCustomAction 206
TrvAction 195
TrvActionParaStyles 102
TrvActionParaBiDi 115

Description

This action changes the default text direction in the selected paragraphs to left-to-right. If you use this action, it's recommended to set TCustomRichViewEdit.BiDiMode to *rvbdLeftToRight* or *rvbdRightToLeft*.

This action does not introduce any new properties in addition to properties of TrvAction.

This action changes BiDiMode property for styles of the selected paragraphs (RVStyle.ParaStyles[].BiDiMode) to *rvbdLeftToRight*.

This is a checkbox-like action, its Checked property is updated depending on the state of selection (or the current paragraph style).

See also:

- TrvActionTextLTR 101
- TrvActionTextRTL 102
- TrvActionParaRTL 128

1.3.4.30 TrvActionParaNumbering

TrvActionParaNumbering is the action for "Numbering" command.

Unit RichViewActions 50;

Syntax

```
TrvActionParaNumbering = class(TrvActionCustomParaListSwitcher 110)
```

Hierarchy

TObject
TPersistent
TComponent
TBasicAction
TContainedAction
TCustomAction
TAction
TrvCustomAction 206
TrvAction 195
TrvActionCustomParaListSwitcher 110

Description

This action applies or removes numbering to the selected paragraphs.

This action does not introduce any new properties in addition to properties of TrvActionCustomParaListSwitcher.

ListLevels 110 should contain numbered levels.

When applying lists, the action does the following:

1. if the paragraph before the selection, the selection or the paragraph after the selection contains a list style with the same properties as ListLevels 110, applies this list style;
2. otherwise, if RVStyle.ListStyles has unused list style with the same properties as ListLevels 110, applies this list style;

3. otherwise, creates a new list style, assigns ListLevels 110 to it, applies this list style.

See also:

- TrvActionParaBullets 117
- TrvActionParaList 125

1.3.4.31 TrvActionParaRTL

TrvActionParaRTL is the action for "Right to Left" command.

Unit RichViewActions 50;

Syntax

```
TrvActionParaRTL = class(TrvActionParaBiDi 115)
```

Hierarchy

TObject
TPersistent
TComponent
TBasicAction
TContainedAction
TCustomAction
TAction
TrvCustomAction 206
TrvAction 195
TrvActionParaStyles 102
TrvActionParaBiDi 115

Description

This action changes the default text direction in the selected paragraphs to right-to-left. If you use this action, it's recommended to set TCustomRichViewEdit.BiDiMode to *rvbdLeftToRight* or *rvbdRightToLeft*.

This action does not introduce any new properties in addition to properties of TrvAction.

This action changes BiDiMode property for styles of the selected paragraphs (RVStyle.ParaStyles[].BiDiMode) to *rvbdRightToLeft*.

This is a checkbox-like action, its Checked property is updated depending on the state of selection (or the current paragraph style).

See also:

- TrvActionTextLTR 101
- TrvActionTextRTL 102
- TrvActionParaLTR 126

1.3.4.32 TrvActionParaStyles

TrvActionParaStyles is a base class for the actions applying changes to the selected paragraphs.

Unit RichViewActions 50;

Syntax

```
TrvActionParaStyles = class(TrvAction 195)
```

Hierarchy

TObject
TPersistent
TComponent
TBasicAction
TContainedAction

TCustomAction
TAction
TrvCustomAction 206
TrvAction 195

Description

This action does not introduce any new properties in addition to properties of TrvAction.

This action is not used directly. This action has the following direct descendants:

- TrvActionParagraph 120
- TrvActionAlignment 106
- TrvActionParaBiDi 115
- TrvActionIndent 111
- TrvActionWordWrap 129
- TrvActionLineSpacing 113
- TrvActionParaBorder 116

1.3.4.33 TrvActionWordWrap

TrvActionWordWrap is the action for "Word Wrap" command.

Unit RichViewActions 50;

Syntax

```
TrvActionWordWrap = class(TrvActionParaStyles 128)
```

Hierarchy

TObject
TPersistent
TComponent
TBasicAction
TContainedAction
TCustomAction
TAction
TrvCustomAction 206
TrvAction 195
TrvActionParaStyles 102

Description

This action turns word wrapping on/off for the selected paragraphs. It excludes/includes *rvpaoNoWrap* from the Options property of styles of the selected paragraphs (RVStyle.ParaStyles[].Options).

This action does not introduce any new properties in addition to properties of TrvAction.

This is a checkbox-like action, its Checked property is updated depending on the current paragraph style (or selection).

1.3.5 Styles

1.3.5.1 TrvActionStyleTemplates

TrvActionStyleTemplates is the action for "Format | Styles" command.

Unit RichViewActions 50;

Syntax

```
TrvActionStyleTemplates = class(TrvAction 195)
```

Hierarchy

TObject
TPersistent
TComponent
TBasicAction
TContainedAction
TCustomAction
TAction
TrvCustomAction 206
TrvAction 195

Description

This action is enabled only if Editor.UseStyleTemplates=*True*.

The action shows a dialog window allowing to manage style templates: to add, to delete, to modify properties, to import (from the special format, RTF and RVF files), to export (to the special format).

When shown initially, the dialog window displays a style template of the current text in the editor.

You can define initial properties of standard style templates using StandardStyleTemplates 130 property.

You can customize the appearance of the dialog window using Images 130, TextStyleImageIndex, ParaStyleImageIndex, ParaTextStyleImageIndex 131 properties.

By default, the style import and export dialogs use localized names of RVF files (*.rvf) and of files containing style templates (*.rvst). If you assign GetControlPanel 208.RVFLocalizable 30, you can change them in the control panel properties: RVFFilter 30, RVStylesFilter, RVStylesExt 32.

This action is used by other actions:

- TrvActionAddStyleTemplate 131 (to open a dialog window)
- TrvActionInsertHyperlink 138 (to get default properties of "Hyperlink" style template)

1.3.5.1.1 Properties

1.3.5.1.1.1 TrvActionStyleTemplates.Images

Images for icons in the styles dialog window.

```
property Images: TCustomImageList;
```

If **Images** is not assigned:

- the tree-view containing a list of styles uses standard 16-color icons;
- a description of styles is displayed without icons.

If **Images** is assigned:

- if at least one of TextStyleImageIndex, ParaStyleImageIndex, ParaTextStyleImageIndex 131 >=0, **Images** and these image indices are used in the tree-view; ImageIndex of this action is used for the root tree node;
- the component searches for the instances of TrvActionFontEx 87, TrvActionParagraph 120, TrvActionParaBorder 116, TrvActionInsertHyperlink 138 actions on the same form; if found, **Images** and the image indices of these actions are used in a style description.

Images are used in the style import dialog window as well.

Generally, you should assign the same TImageList component as used for TActionList/TActionManager containing this action.

1.3.5.1.1.2 TrvActionStyleTemplates.StandardStyleTemplates

Contains a list of style templates used as samples of standard style templates.

```
property StandardStyleTemplates: TRVStyleTemplateCollection;
```

When an user adds a standard style template in the dialog, the action searches in **StandardStyleTemplates** for a style template having this name. If found, the action uses properties of the **StandardStyleTemplates**' item as initial values for the added style template. It not found, the action uses hard-coded values as initial properties.

The following names are used for standard styles: 'Normal', 'Normal Indent', 'No Spacing', 'heading 1'...'heading 9', 'List Paragraph', 'Hyperlink', 'Title', 'Subtitle', 'Emphasis', 'Subtle Emphasis', 'Intense Emphasis', 'Strong', 'Quote', 'Intense Quote', 'Subtle Reference', 'Intense Reference', 'Block Text', 'HTML Variable', 'HTML Code', 'HTML Acronym', 'HTML Definition', 'HTML Keyboard', 'HTML Sample', 'HTML Typewriter', 'HTML Preformatted', 'HTML Cite', 'header', 'footer', 'page number', 'endnote reference', 'footnote reference', 'endnote text', 'footnote text'. Names are case sensitive. These names are not shown to users, their localized versions are shown instead.

Values in this collection are measured in GetControlPanel[208].UnitsProgram[33].

This property may also be used by TrvActionInsertHyperlink, when it adds "Hyperlink" style template.

See also:

TrvActionNew[57].StyleTemplates[58]
TrvActionInsertHyperlink[138].AutoAddHyperlinkStyleTemplate[140].

1.3.5.1.1.3 TrvActionStyleTemplates.*StyleImageIndex

The properties define image indices for icons in the tree-view containing a list of styles.

```
property TextStyleImageIndex: Integer;
property ParaStyleImageIndex: Integer;
property ParaTextStyleImageIndex: Integer;
```

If Images[130] is assigned, and at least one of these properties >= 0, the tree-view in the style management dialog window uses these images:

- **TextStyleImageIndex** for style templates having Kind=*rvstkText* (and a tree node used as a root of for such style templates);
- **ParaStyleImageIndex** for style templates having Kind=*rvstkPara*;
- **ParaTextStyleImageIndex** for style templates having Kind=*rvstkParaText*.

Additionally, ImageIndex of this action is used for the root tree node.

Default value:

-1

1.3.5.2 TrvActionAddStyleTemplate

TrvActionAddStyleTemplate is the action for "Format | Add Style..." command.

Unit RichViewActions[50];

Syntax

```
TrvActionAddStyleTemplate = class(TrvAction[195])
```

Hierarchy

TObject
TPersistent
TComponent
TBasicAction
TContainedAction
TCustomAction
TAction
TrvCustomAction[206]
TrvAction[195]

Description

This action is enabled only if Editor.UseStyleTemplates=*True*.

The action shows a dialog window allowing to manage style templates. When shown initially, the dialog window has a new style template added; its properties are based on the text and paragraph attributes at the position of the caret.

This action can work only if it is linked to TrvActionStyleTemplates[129] action. It can be linked using ActionStyleTemplates[132] property; if this property is not assigned, the action searches for TrvActionStyleTemplates[129] action on the same form.

1.3.5.2.1 Properties

1.3.5.2.1.1 TrvActionAddStyleTemplate.ActionStyleTemplates

Links this action to TrvActionStyleTemplates[129] action.

```
property ActionStyleTemplates: TrvActionStyleTemplates[129];
```

If this property is not assigned, the action searches for TrvActionStyleTemplates[129] action on the same form. If not found, the action does nothing.

TrvActionStyleTemplates[129] defines properties for the style management dialog window.

1.3.5.3 TrvActionStyleInspector

TrvActionStyleInspector is the action for "Format | Style Inspector" command.

Unit RichViewActions[50];

Syntax

```
TrvActionCustomInfoWindow = class(TrvAction[195]);
TrvActionStyleInspector = class(TrvActionCustomInfoWindow);
```

Hierarchy

TObject
TPersistent
TComponent
TBasicAction
TContainedAction
TCustomAction
TAction
TrvCustomAction[206]
TrvAction[195]

Description

This action shows/hides a non-modal window containing information about text and paragraph styles and attributes at the position of the caret.

Assign Control[195] property of this action, and the info window will be updated automatically according to changes in Control[195]. When the action is executed for some editor, the action automatically assign this editor to Control[195].

You can customize the appearance of the info window using Images[132], FontImageIndex and ParaImageIndex[133] properties.

1.3.5.3.1 Properties

1.3.5.3.1.1 TrvActionStyleInspector.Images

Images for icons used in a description of text and paragraph attributes.

```
property Images: TCustomImageList;
```

This image list must contain two images: text and paragraph icons. Their indices should be specified in FontImageIndex and ParaImageIndex[133] properties.

1.3.5.3.1.2 TrvActionStyleInspector.FontImageIndex, ParaImageIndex

Indices in Images[132].

```
property FontImageIndex:Integer;
property ParaImageIndex:Integer;
```

FontImageIndex specifies the index of "font" icon.

ParaImageIndex specifies the index of "paragraph" icon.

Default value:

-1

1.3.5.3.2 Methods

1.3.5.3.2.1 TrvActionStyleInspector.UpdateInfo

Refresh a content of the info window.

```
procedure UpdateInfo(OnlyIfVisible: Boolean = True);
```

Normally, the action updates the window when necessary.

Call this method when GetControlPanel[208].Language[29] is changed.

1.3.5.3.3 Events

1.3.5.3.3.1 TrvActionCustomInfoWindow .OnShow ing

Occurs before showing the info window.

```
property OnShowing: TRVShowFormEvent[235];
```

Example (embedding the info window in the main window TfrmMain, at the right side):

```
procedure TfrmMain.rvActionStyleInspector1Showing(Sender: TrvAction[195];
  Form: TForm);
begin
  Form.Align := alRight;
  Form.Parent := Self;
end;
```

1.3.5.4 TrvActionClearFormat

TrvActionClearFormat is the action for "Format | Clear Format" command.

Unit RichViewActions[50];

Syntax

```
TrvActionClearFormat = class(TrvAction[195])
```

Hierarchy

TObject
TPersistent
TComponent
TBasicAction
TContainedAction
TCustomAction
TAction
TrvCustomAction[206]
TrvAction[195]

Description

This action resets text and/or paragraph attributes in the selected fragment by calling

Editor.ApplyStyleTemplate(-1).

The results depend on the selection. The action clears the paragraph format in the following cases:

- the selection is empty
- the selection includes the whole paragraph completely
- the selection includes several paragraphs.

If the selection is not empty, the action clears its text format.

This action can be used even if Editor.UseStyleTemplates=*False*. In this case, it simply applies the 0th text and the 0th paragraph styles.

This action does not introduce any new properties in addition to properties of TrvAction.

1.3.5.5 TrvActionClearTextFormat

TrvActionClearFormat is the action for "Format | Clear Text Format" command.

Unit RichViewActions 50;

Syntax

```
TrvActionClearTextFormat = class(TrvAction 195)
```

Hierarchy

TObject
TPersistent
TComponent
TBasicAction
TContainedAction
TCustomAction
TAction
TrvCustomAction 206
TrvAction 195

Description

This action resets text attributes in the selected fragment by calling Editor.ApplyTextStyleTemplate(-1, True).

This action can be used even if Editor.UseStyleTemplates=*False*. In this case, it simply applies the 0th text style.

This action does not introduce any new properties in addition to properties of TrvAction.

1.3.6 Insert

1.3.6.1 TrvActionInsertCaption

TrvActionInsertCaption is the action for "Insert Object Caption..." command.

Unit RichViewActions 50;

Syntax

```
TrvActionInsertCaption = class(TrvActionInsertNumSequence 148)
```

Hierarchy

TObject
TPersistent
TComponent
TBasicAction
TContainedAction
TCustomAction
TAction

TrvCustomAction 206
TrvAction 195
TrvActionInsertNumSequence 148

Description

By default, this action allows inserting captions for pictures, "hot pictures" and tables. You can allow captions for additional items using OnCanApply event.

This action displays a dialog window allowing to enter caption for the object at the caret position.

The caption consists of the following parts:

- label
- "numbered sequence" item (name of its sequence=label)
- '. ' string
- used-defined text

Example: *Figure 1. Population dynamics of green zebras*

If UseStyleTemplates=*True* for the target editor, this caption is formatted using 'caption' style template. Otherwise, the current text style is used.

The following caption properties are supported:

- label (equal to a sequence name);
- numbering type (decimal, lower Roman, etc.);
- exclude label or not?
- caption text
- insert above or below?

If UseDefaults 149=*True*, the dialog is initialized with a label 'Figure' (localized string), a decimal numbering type, ExcludeLabel 135 and InsertAbove 135 properties.

When the user clicks the "Insert" button, a caption is inserted above or below the selected object, UseDefaults 149 is reset to *False*, and chosen values are stored in SeqName 148, NumberType 148, ExcludeLabel 135 and InsertAbove 135 properties (they are used to initialize a dialog when it will be displayed for the next time).

1.3.6.1.1 Properties

1.3.6.1.1.1 TrvActionInsertCaption.ExcludeLabel

Specifies whether a label is included in a caption text.

```
property ExcludeLabel: Boolean;
```

This property is used to initialize a dialog. When a dialog is applied, this property is updated automatically.

Example of caption if ExcludeLabel=*False*:

Table 1. Aging Statistics

Example of caption if ExcludeLabel=*True*:

1. Aging Statistics

Initial value:

False

1.3.6.1.1.2 TrvActionInsertCaption.InsertAbove

Specifies where a caption is inserted: above or below the object.

```
property InsertAbove: Boolean;
```

This property is used to initialize a dialog. When a dialog is applied, this property is updated automatically.

Initial value:

False

1.3.6.2 TrvActionInsertCustomPageNumber

TrvActionInsertCustomPageNumber is a base class for actions inserting page numbers and page counts.

Unit RichViewActions[50];

Syntax

```
TrvActionInsertCustomPageNumber = class(TrvAction[195])
```

Hierarchy

TObject
TPersistent
TComponent
TBasicAction
TContainedAction
TCustomAction
TAction
TrvCustomAction[206]
TrvAction[195]

Description

This action is not used directly. It is a parent for the following actions:

- TrvActionInsertPageNumber[150]
- TrvActionInsertPageCount[149]

1.3.6.2.1 Properties

1.3.6.2.1.1 TrvActionInsertCustomPageNumber.NumberType

A type of numbering for the inserted item (decimal, lower Roman, etc.)

```
property NumberType: TRVPageNumberType;
```

Default value

rvpntDecimal

1.3.6.3 TrvActionInsertFile

TrvActionInsertFile is the action for "Insert | File" command.

Unit RichViewActions[50];

Syntax

```
TrvActionInsertFile = class(TrvActionCustomFileIO[54])
```

Hierarchy

TObject
TPersistent
TComponent
TBasicAction
TContainedAction
TCustomAction
TAction
TrvCustomAction[206]
TrvAction[195]
TrvActionCustomIO[54]
TrvActionCustomFileIO[54]

Description

This action displays a file opening dialog and inserts the chosen file in the caret position.

In addition to formats supported by TrvActionOpen[59] action, this action supports:

- HTML (not only if rvHtmlViewImporter[222], but also if rvHtmlImporter[221] is used)
- formats supported by Microsoft Office file import converters.

1.3.6.3.1 Properties

1.3.6.3.1.1 TrvActionInsertFile.Filter

Defines a set of file formats appearing in the file opening dialog.

```
property Filter: TrvFileImportFilterSet[234];
```

If *ffiCustom* is included, formats listed in the CustomFilter[54] property are used.

Name and extension of RichView Format (RVF) may be customized, see GetControlPanel[208].RVFFilter[30].

Default value:

all formats

1.3.6.3.2 Methods

1.3.6.3.2.1 TrvActionInsertFile.InsertFile

Inserts the specified file.

```
    function InsertFile(rve: TCustomRichViewEdit; const FileName: String;
      FileFormat: TrvFileImportFilter[234];
      ConverterOrCustomIndex: Integer=0;
      rvc: TRVOfficeConverter=nil;
      Silent: Boolean = False): Boolean;
```

Parameters:

rve – editor where to insert file.

FileName – name of the file to insert.

FileFormat – format of this file.

ConverterCustomFilterIndex used only if **FileFormat** is *ffiCustom* or *ffiOfficeConverters*.

If **FileFormat** = *ffiCustom*, **ConverterCustomFilterIndex** identifies a custom file format. Custom formats are numbered from 1 in the order they are listed in the CustomFilter[54] property.

If **FileFormat** = *ffiOfficeConverters*, **ConverterCustomFilterIndex** identifies an import format, as an index in **rvc**.ImportConverters.

rvc is used if **FileFormat** = *ffiOfficeConverters*.

Silent – if **False**, the action displays an error message if the insertion was not successful.

Return value:

True if the insertion was successful.

If the insertion failed, this method displays an error message.

This method does not change values of the action's properties.

1.3.6.4 TrvActionInsertHLine

TrvActionInsertHLine is the action for "Insert | Horizontal Line" command.

Unit RichViewActions[50];

Syntax

```
TrvActionInsertHLine = class(TrvAction 195)
```

Hierarchy

TObject
TPersistent
TComponent
TBasicAction
TContainedAction
TCustomAction
TAction
TrvCustomAction 206
TrvAction 195

Description

This action inserts a horizontal line ("break" item type) in the caret position in TCustomRichViewEdit component.

The inserted line has the specified Color 138 and Width 138, and Style 138.

1.3.6.4.1 Properties

1.3.6.4.1.1 TrvActionInsertHLine.Color

Specifies the color for horizontal line.

```
property Color: TColor;
```

Default value:

clWindowText

1.3.6.4.1.2 TrvActionInsertHLine.Style

Specifies the style for horizontal line.

```
property Style: TRVBreakStyle;
```

Default value:

rvbsLine

1.3.6.4.1.3 TrvActionInsertHLine.Width

Specifies the width for horizontal line.

```
property Width: TRVStyleLenth;
```

This value is measured in GetControlPanel 208.UnitsProgram 33.

Default value:

1

This default value assumes that this property is measured in pixels. For twips, this value may be too small.

1.3.6.5 TrvActionInsertHyperlink

TrvActionInsertHyperlink is the action for "Insert | Hyperlink" command.

Unit RichViewActions 50;

Syntax

```
TrvActionInsertHyperlink = class(TrvActionTextStyles 102)
```

Hierarchy

TObject
TPersistent

TComponent
TBasicAction
TContainedAction
TCustomAction
TAction
TrvCustomAction 206
TrvAction 195
TrvActionTextStyles 102

Description

This action can perform the following tasks:

- inserting a new text hyperlink, with the specified target and text;
- converting the selected text and (optionally) pictures to hyperlinks (with the specified target);
- removing hyperlink from the selected fragment.

When this action is executed, it displays a dialog for defining hyperlink attributes. But before the dialog is shown, the action adjusts the selection in TCustomRichViewEdit: if the selection includes a hyperlink partially (or if there is no selection but the caret is inside a link), this hyperlink is selected completely. If the selection is still empty, the action will insert a new hyperlink. Otherwise it will convert the selected fragment to hyperlinks (or will change targets for existing hyperlinks in the selection). If the user will enter an empty target, all links will be removed instead.

By default, the action stores targets of hyperlinks in items' tags. You can override this behavior using OnApplyHyperlinkToItem 146 and OnGetHyperlinkTargetFromItem 146 events.

Using style templates

Visual appearance of a hyperlink depends on whether style templates are used 17.

If style templates are not used: the action allows defining text and background colors for hyperlinks (including "hover" colors). The chosen colors will be applied to all new links. When converting a text to a hypertext, properties listed in ValidProperties 143 are applied to its style (and the original style is assigned to its NextStyleNo property).

When converting a hypertext to a text, the same set of properties are changed. But instead of properties of this action, properties of RVStyle.TextStyles[0] are applied (and -1 is assigned to NextStyleNo property).

The dialog window has a button invoking a new dialog for changing the link color.

If style templates are used: the action applies StyleTemplateName 142 to hyperlinks. When converting a hypertext to a text, the action clears a link to style template.

The dialog window has a combo-box allowing to choose a style template. Only style templates having Kind=rvstkText are listed.

Methods

In addition to the operations performed on the execution, the action has several useful methods:

- GetHyperlinkStyleNo 144 returns the index of hypertext style for the given text style;
- GetNormalStyleNo 145 returns the index of non-hypertext style for the given text style;
- DetectURL 143 analyses text to the left of the caret; if it is URL, converts it to hyperlink;
- TerminateHyperlink 146 converts the current text style to non-hypertext, if the caret is at the end of hyperlink;
- GoToLink 145 opens the hyperlink target in the external browser; for local links, it moves the caret to the target *checkpoint*.

1.3.6.5.1 Properties

1.3.6.5.1.1 TrvActionInsertHyperlink.ActionStyleTemplates

Links this action to TrvActionStyleTemplates[129] action.

```
property ActionStyleTemplates: TrvActionStyleTemplates[129];
```

This property is used only if style templates are used[17] (i.e. the target editor's UseStyleTemplates=*True*).

If this link is not defined, the first found TrvActionStyleTemplates[129] action on the same form/datamodule is used.

See StyleTemplateName[142] for details.

1.3.6.5.1.2 TrvActionInsertHyperlink.AutoAddHyperlinkStyleTemplate

Defines whether the action may add "Hyperlink" style template.

```
property AutoAddHyperlinkStyleTemplate: Boolean;
```

This property is used only if style templates are used[17] (i.e. the target editor's UseStyleTemplates=*True*).

See StyleTemplateName[142] for details.

Default value:

True

1.3.6.5.1.3 TrvActionInsertHyperlink.BackColor

Defines the background color for hyperlinks.

```
property BackColor: TColor;
```

This property is used only if style templates are not used[17] (i.e. the target editor's UseStyleTemplates=*False*). If style templates are used, the action applies StyleTemplateName[142] instead.

This property is used when converting text to hypertext, if *rvhlBackColor* is included in ValidProperties[143].

A value of this property can be changed by the user in the hyperlink properties dialog.

Default value:

clNone

1.3.6.5.1.4 TrvActionInsertHyperlink.Color

Defines the text color for hyperlinks.

```
property Color: TColor;
```

This property is used only if style templates are not used[17] (i.e. the target editor's UseStyleTemplates=*False*). If style templates are used, the action applies StyleTemplateName[142] instead.

This property is used when converting text to hypertext, if *rvhlColor* is included in ValidProperties[143].

A value of this property can be changed by the user in the hyperlink properties dialog.

Default value:

clBlue

1.3.6.5.1.5 TrvActionInsertHyperlink.Cursor

Defines the hypertext cursor for the links.

```
property Cursor: TCursor;
```

This property is used only if style templates are not used[17] (i.e. the target editor's UseStyleTemplates=*False*). If style templates are used, the action applies StyleTemplateName[142] instead.

This property is used when converting text to hypertext, if *rvhlCursor* is included in ValidProperties 143.

Default value:

crJump

1.3.6.5.1.6 TrvActionInsertHyperlink.HoverBackColor

Defines the background color for hyperlinks under the mouse pointer.

```
property HoverBackColor: TColor;
```

This property is used only if style templates are not used 17 (i.e. the target editor's UseStyleTemplates=*False*). If style templates are used, the action applies StyleTemplateName 142 instead.

This property is used when converting text to hypertext, if *rvhlHoverBackColor* is included in ValidProperties 143.

A value of this property can be changed by the user in the hyperlink properties dialog.

Default value:

clNone (transparent)

1.3.6.5.1.7 TrvActionInsertHyperlink.HoverColor

Defines the text color for hyperlinks under the mouse pointer.

```
property HoverColor: TColor;
```

This property is used only if style templates are not used 17 (i.e. the target editor's UseStyleTemplates=*False*). If style templates are used, the action applies StyleTemplateName 142 instead.

This property is used when converting text to hypertext, if *rvhlHoverColor* is included in ValidProperties 143.

A value of this property can be changed by the user in the hyperlink properties dialog.

Default value:

clBlue

1.3.6.5.1.8 TrvActionInsertHyperlink.HoverEffects

Defines the effects (such as underlining) for hyperlinks under the mouse pointer.

```
property HoverEffects: TRVHoverEffects;
```

This property is used only if style templates are not used 17 (i.e. the target editor's UseStyleTemplates=*False*). If style templates are used, the action applies StyleTemplateName 142 instead.

This property is used when converting text to hypertext, if *rvhlHoverEffects* is included in ValidProperties 143.

A value of this property can be changed by the user in the hyperlink properties dialog.

Default value:

[]

1.3.6.5.1.9 TrvActionInsertHyperlink.HoverUnderlineColor

Defines the underline color for hyperlinks under the mouse pointer.

```
property HoverUnderlineColor: TColor;
```

This property is used only if style templates are not used 17 (i.e. the target editor's UseStyleTemplates=*False*). If style templates are used, the action applies StyleTemplateName 142 instead.

This property is used when converting text to hypertext, if *rvhlHoverUnderlineColor* is included in ValidProperties 143.

A value of this property can be changed by the user in the hyperlink properties dialog.

Default value:

clNone (instructs using UnderlineColor[143])

1.3.6.5.1.10 TrvActionInsertHyperlink.SetToPictures

Specifies whether this action should convert pictures to hot-pictures and vice versa.

```
property SetToPictures: Boolean;
```

If this property is *False*, the action does nothing with the selected pictures.

Default value:

True

1.3.6.5.1.11 TrvActionInsertHyperlink.SpaceFiller

Specifies the string used for replacing space characters in hyperlink targets (by EncodeTarget[144]).

```
property SpaceFiller: String;
```

Default value:

' ' (space character, so nothing is replaced)

1.3.6.5.1.12 TrvActionInsertHyperlink.Style

Defines the font style for hyperlinks.

```
property Style: TFontStyles;
```

This property is used only if style templates are not used[17] (i.e. the target editor's UseStyleTemplates=*False*). If style templates are used, the action applies StyleTemplateName[142] instead.

This property is used when converting text to hypertext, if *rvhlBold*, *rvhlItalic*, *rvhlUnderline*, or *rvhlStrikeout* are included in ValidProperties[143].

For example:

- if *rvhlBold* is included in ValidProperties[143] and *fsBold* is included in Style, hyperlinks will be bold;
- if *rvhlBold* is included in ValidProperties[143] and *fsBold* is not included in Style, hyperlinks will not be bold;
- if *rvhlBold* is not included in ValidProperties[143] (default), boldness is not changed.

Default value:

[*fsUnderline*]

1.3.6.5.1.13 TrvActionInsertHyperlink.StyleTemplateName

Defines the name of a style template for applying to hyperlinks.

```
property StyleTemplateName: String;
```

This property is used only if style templates are used[17] (i.e. the target editor's UseStyleTemplates=*True*).

This property may be changed by an user in the dialog window.

This style template must be a text style template (Kind=*rvskText*), otherwise it will not be listed in a dialog window.

To get a style template for a hyperlink, the action performs the following steps:

1. The action searches in the target editor's Style.StyleTemplates for the style template named **StyleTemplateName**. If found, it is used. Otherwise,
2. The action searches for "Hyperlink" style template. If found, it is used. Otherwise,
3. If AutoAddHyperlinkStyleTemplate[140]=*True*, the action adds "Hyperlink" style template to the target editor's Style.StyleTemplates. Properties of this new style templates may be taken from the linked[140] TrvActionStyleTemplates[129] action, if it has "Hyperlink" in its StandardStyleTemplates[130]; otherwise, the action uses hard-coded default values.

Default value:

'Hyperlink'

1.3.6.5.1.14 TrvActionInsertHyperlink.UnderlineColor

Defines the underline color for hyperlinks.

```
property UnderlineColor: TColor;
```

This property is used only if style templates are not used[17] (i.e. the target editor's UseStyleTemplates=*False*). If style templates are used, the action applies StyleTemplateName[142] instead.

This property is used when converting text to hypertext, if *rvhlHoverUnderlineColor* is included in ValidProperties[143].

A value of this property can be changed by the user in the hyperlink properties dialog.

Default value:

clNone (instructs using Color[140])

1.3.6.5.1.15 TrvActionInsertHyperlink.ValidProperties

Defines properties distinguishing hyperlinks from normal text, if style templates are not used.

```
type
  TRVHyperlinkProperty = (rvhlBold, rvhlItalic, rvhlUnderline, rvhlUnderlineColor,
    rvhlStrikeout, rvhlColor, rvhlBackColor, rvhlHoverColor, rvhlHoverBackColor,
    rvhlHoverUnderlineColor, rvhlHoverUnderline, rvhlCursor, rvhlJump);
  TRVHyperlinkProperties = set of TRVHyperlinkProperty;

property ValidProperties: TRVHyperlinkProperties;
```

This property is used only if style templates are not used[17] (i.e. the target editor's UseStyleTemplates=*False*). If style templates are used, the action applies StyleTemplateName[142] instead.

When this action converts text to hypertext, properties of this action listed in this property are applied:

- BackColor[140]
- Color[140]
- HoverBackColor[141]
- HoverColor[141]
- HoverEffects[141]
- HoverUnderlineColor[141]
- UnderlineColor[143]
- Style[142]

When this action converts text to non-hypertext, properties of RVStyle.TextStyles[0] listed in this property are applied.

Default value:

[*rvhlUnderline, rvhlColor, rvhlBackColor, rvhlHoverColor, rvhlHoverBackColor, rvhlCursor, rvhlJump*] (hyperlink with the standard hyperlink cursor, blue and underlined)

1.3.6.5.2 Methods

1.3.6.5.2.1 TrvActionInsertHyperlink.DetectURL

Helps to detect URLs on typing in **rve**.

```
procedure DetectURL(rve: TCustomRichViewEdit);
```

This action checks the text to the left of the caret position, and if its a URL, converts it to a hyperlink.

The function does nothing of the text is already a hyperlink, or if the selection in **rve** is not empty.

The method uses GetHyperlinkStyleNo[144] for this conversion.

Example:

```
procedure TForm1.RichViewEdit1KeyPress(Sender: TObject; var Key: Char);
begin
  if Key in [#13, #9, ' ', ',', ';'] then begin
    rvActionInsertHyperlink1.DetectURL(RichViewEdit1);
    rvActionInsertHyperlink1.TerminateHyperlink[146](RichViewEdit1);
  end;
end;
```

1.3.6.5.2.2 TrvActionInsertHyperlink.EncodeTarget

Encodes hypertext target before assigning it to the item's tag.

```
function EncodeTarget(const Target: String): String;
```

The function returns the processed **Target** parameter.

It removes the line break characters (#13 and #10) and replaces the space characters with SpaceFiller[142].

Normally, the returned value is the same as **Target** (because the action does not allow entering multiline targets, and SpaceFiller[142]=' ').

1.3.6.5.2.3 TrvActionInsertHyperlink.GetHyperlinkStyleNo

Returns the index of hypertext style for the given style.

```
function GetHyperlinkStyleNo(rve: TCustomRichViewEdit; StyleNo: Integer=-1): Integer;
```

If **StyleNo**=-1, the action returns the index of hypertext style for **rve**.CurTextStyleNo, otherwise for **StyleNo**.

The appearance of the returned style depends on **rve**.UseStyleTemplates.

If style templates are not used[17], the method returns the index of style (in **rve**.Style.TextStyles) having all properties of the specified style, except for the properties defined in ValidProperties[143]. For these properties, values specified in properties of this action are used.

If style templates are used[17], the method returns the index of style (in **rve**.Style.TextStyles) having all properties of the specified style, by with the style template StyleTemplateName[142] applied.

If a style with the required properties already exists in **rve**.Style.TextStyles, its index is returned. Otherwise, a new style is created and its index is returned.

This method is used in DetectURL[143] method, or when inserting/applying hyperlink. You can use it in other cases when you need a hypertext style index.

For example, in TCustomRichViewEdit.OnReadHyperlink event:

```
procedure TForm1.RichViewEdit1ReadHyperlink(Sender: TCustomRichView;
  const Target, Extras: String; DocFormat: TRVLoadFormat;
  var StyleNo: Integer; var ItemTag: TRVTag; var ItemName: TRVRawByteString);
begin
  if DocFormat=rvlfURL then
    StyleNo := rvActionInsertHyperlink1.GetHyperlinkStyleNo(RichViewEdit1);
  ItemTag := rvActionInsertHyperlink1.EncodeTarget[144](Target);
end;
```

See also methods:

- GetNormalStyleNo[145]

1.3.6.5.2.4 TrvActionInsertHyperlink.GetNormalStyleNo

Returns the index of non-hypertext style for the given style.

```
function GetNormalStyleNo(rve: TCustomRichViewEdit; StyleNo: Integer=-1): Integer;
```

If **StyleNo**=-1, the action returns the index of non-hypertext style for **rve**.CurTextStyleNo, otherwise for **StyleNo**.

The appearance of the returned style depends on **rve**.UseStyleTemplates.

If style templates are not used [17], the method returns the index of style (in **rve**.Style.TextStyles) having all properties of the specified style, except for the properties defined in ValidProperties [143]. For these properties, values of properties of **rve**.Style.TextStyles[0] are used.

If style templates are used [17], the method returns the index of style (in **rve**.Style.TextStyles) having all properties of the specified style, by with link to a style template cleared.

If a style with the required properties already exists in **rve**.Style.TextStyles, its index is returned. Otherwise, a new style is created and its index is returned.

This method is used in TerminateHyperlink [146] method, or when user entered an empty target in the dialog.

See also methods:

- GetHyperlinkStyleNo [144]

1.3.6.5.2.5 TrvActionInsertHyperlink.GoToLink

These methods execute the given hyperlink.

```
procedure GoToLink(rve: TCustomRichViewEdit; id: Integer);
procedure GoToLink(rve: TCustomRichViewEdit; const Target: String);
```

The first version of the method should be called in rve.OnJump event:

```
procedure TForm3.RichViewEdit1Jump(Sender: TObject; id: Integer);
begin
  rvActionInsertHyperlink1.GoToLink(RichViewEdit1, id);
end;
```

The second version of this method can be called from rve.OnJump or from any place of code.

If the link's target (stored in its tag) is started from '#', the action moves the caret to the *checkpoint* of the same name (but without '#').

Otherwise, the action opens the link's target in external application (using ShellExecute function).

On error (no such checkpoint, or ShellExecute returned an error code), an error message is displayed.

See also:

- GoToCheckpoint [228] function

1.3.6.5.2.6 TrvActionInsertHyperlink.SetTags

Applies the action to the selection.

```
procedure SetTags(rve: TCustomRichViewEdit; const Target: String);
```

Normally, calling this method is not necessary. It is called automatically when the action applies **Target** to the selection.

If **Target** is not empty, the action converts the selection to hyperlink(s).

If **Target** is empty, the action removes hyperlinks from the selection.

1.3.6.5.2.7 TrvActionInsertHyperlink.TerminateHyperlink

Helps to terminate hyperlinks when typing in **rve**.

```
procedure TerminateHyperlink(rve: TCustomRichViewEdit);
```

If the selection is empty, the current text item is a hyperlink, and the caret is at the end of this item, this function assigns the index non-hypertext style to the current text style (**rve**.CurTextStyleNo). GetNormalStyleNo [145] is used.

Example:

```
procedure TForm1.RichViewEdit1KeyPress(Sender: TObject; var Key: Char);
begin
  if Key in [#13, #9, ' ', ',', ';'] then begin
    rvActionInsertHyperlink1.DetectURL [143](RichViewEdit1);
    rvActionInsertHyperlink1.TerminateHyperlink(RichViewEdit1);
  end;
end;
```

When the user types something at the end of a hyperlink, new characters are added to the hyperlink's text. But when he/she types a space character (or tab, or comma, or Enter, or semicolon), the current text style becomes a non-hypertext, so this and subsequent characters will not be added to the hyperlink's text.

1.3.6.5.3 Events

1.3.6.5.3.1 TrvActionInsertHyperlink.OnApplyHyperlinkToItem

Occurs when converting item of unknown type to/from hyperlink

```
type
  TRVApplyHyperlinkToItemEvent = procedure (Sender: TObject;
    Editor: TCustomRichViewEdit; RVData: TCustomRVData; ItemNo: Integer;
    const Target: String) of object;

property OnApplyHyperlinkToItem: TRVApplyHyperlinkToItemEvent;
```

For text and pictures, the action applies hypertext (or removes hypertext) automatically.

For other types of items, this event occurs.

The item is defined by **Editor** and **ItemNo** (it is the **ItemNo**-th item in **Editor**, where **Editor** can be a main editor or cell inplace editor). **RVData** is **Editor**.RVData.

When converting to hypertext, **Target** is not empty.

When converting from hypertext, **Target** is empty.

In this event, you can perform an editing operation on this item.

1.3.6.5.3.2 TrvActionInsertHyperlink.OnGetHyperlinkTargetFromItem

Occurs when the action needs to get a hyperlink target from item of unknown type.

```
type
  TRVGetHyperlinkTargetFromItem = procedure (Sender: TObject;
    Editor: TCustomRichViewEdit; RVData: TCustomRVData; ItemNo: Integer;
    var Target: String) of object;

property OnGetHyperlinkTargetFromItem: TRVGetHyperlinkTargetFromItem;
```

For text and pictures, the action gets targets automatically. For items of other types, this event occurs.

If you return empty **Target**, the action will treat this item as non-hypertext.

The item is defined by **Editor** and **ItemNo** (it is the **ItemNo**-th item in **Editor**, where **Editor** can be a main editor

or cell inplace editor). **RVData** is **Editor**.RVData.

1.3.6.5.3.3 TrvActionInsertHyperlink.OnHyperlinkForm

Allows displaying your own hypertext dialog.

```
type
  TRVHyperlinkFormEvent = procedure (Sender: TObject; InsertNew: Boolean;
    var Text, Target: String; var Proceed: Boolean) of object;

property OnHyperlinkForm: TRVHyperlinkFormEvent;
```

Use this event if you want to replace the default dialog used in this action.

Input parameters:

If **InsertNew**=*True*, this event is called for inserting a new hyperlink.

Text – the selected text, valid only if **InsertNew**=*False*.

Target – hypertext target of selected text, use only if **InsertNew**=*False*. It is not empty only in case when the selection contains only hyperlinks with equal targets.

Proceed = *True*.

Output parameters:

Text – text for new hyperlink, used only if **InsertNew**=*True*.

Target – hyperlink target. You can return empty string only if **InsertNew**=*False* (to remove hyperlinks from the selection)

Proceed must be *True* if the user pressed "OK", and *False* if the user pressed "Cancel".

1.3.6.6 TrvActionInsertNumber

TrvActionInsertNumber is the action for "Insert Number..." command.

Unit RichViewActions [50];

Syntax

```
TrvActionInsertNumber = class(TrvActionInsertNumSequence [148])
```

Hierarchy

TObject
TPersistent
TComponent
TBasicAction
TContainedAction
TCustomAction
TAction
TrvCustomAction [206]
TrvAction [195]
TrvActionInsertNumSequence [148]

Description

This action displays a dialog window allowing to enter properties of an item to insert.

The following properties are supported:

- sequence name (identifier);
- numbering type (decimal, lower Roman, etc.);
- continue numbering / start from the specified value.

If UseDefaults[149]=*True*, the dialog is initialized with a sequence name 'Numbering' (localized string) and a decimal numbering type.

When the user clicks the "Insert" button, a "numbered sequence" item is inserted in the caret position, UseDefaults[149] is reset to *False*, and chosen values are stored in SeqName[148] and NumberType[148] properties (they are used to initialize a dialog when it will be displayed for the next time).

1.3.6.7 TrvActionInsertNumSequence

TrvActionInsertNumSequence is a base class for the actions inserting a "numbered sequence" item.

Unit RichViewActions[50];

Syntax

```
TrvActionInsertNumSequence = class(TrvAction[195])
```

Hierarchy

TObject
TPersistent
TComponent
TBasicAction
TContainedAction
TCustomAction
TAction
TrvCustomAction[206]
TrvAction[195]

Description

This action is not used directly.

The following actions are inherited from it:

- TrvActionInsertNumber
- TrvActionInsertCaption

1.3.6.7.1 Properties

1.3.6.7.1.1 TrvActionInsertNumSequence.NumberType

A type of numbering for the inserted item (decimal, lower Roman, etc.)

```
property NumberType: TRVSeqType;
```

If UseDefaults[149]=*False*, this property is used to initialize a numbering type field in a dialog.

This property is updated automatically to the value entered in this field, when this dialog is applied.

1.3.6.7.1.2 TrvActionInsertNumSequence.SeqName

A name (identifier) of the numbered sequence. The inserted item will continue numbering started by items having the same sequence name.

```
property SeqName: String;
```

If UseDefaults[149]=*False*, this property is used to initialize a sequence name field in a dialog.

This property is updated automatically to the value entered in this field, when this dialog is applied.

The following sequence names are not allowed (the "Insert" button is disabled if they are entered):

- empty strings
- strings started from '@'

1.3.6.7.1.3 TrvActionInsertNumSequence.UseDefaults

Specifies whether values of SeqName[148] and NumberType[148] properties are used.

```
property UseDefaults: Boolean
```

When this property is *True*, the following values are used to initialize a dialog:

- sequence name = 'Numbering' (localized) for TrvActionInsertNumber or 'Figure' (localized) for TrvActionInsertCaption;
- numbering type = decimal.

When this property is *False*, values of SeqName[148] and NumberType[148] properties are used instead.

After an insertion, this property is reset to *False* automatically.

Initial value:

True

1.3.6.8 TrvActionInsertPageBreak

TrvActionInsertPageBreak is the action for "Insert Page Break" command.

Unit RichViewActions[50];

Syntax

```
TrvActionInsertPageBreak = class(TrvAction[195])
```

Hierarchy

TObject
TPersistent
TComponent
TBasicAction
TContainedAction
TCustomAction
TAction
TrvCustomAction[206]
TrvAction[195]

Description

This action inserts a page break in the caret position (or, when called from a table cell, before the current row).

This action does not introduce any new properties in addition to properties of TrvAction.

See also:

- TrvActionRemovePageBreak[204]

1.3.6.9 TrvActionInsertPageCount

TrvActionInsertPageCount is the action for "Insert | Page Count" command.

Unit RichViewActions[50];

Syntax

```
TrvActionInsertPageCount = class(TrvActionInsertCustomPageNumber[136])
```

Hierarchy

TObject
TPersistent
TComponent
TBasicAction
TContainedAction
TCustomAction

TAction
TrvCustomAction 206
TrvAction 195
TrvActionInsertCustomPageNumber 136

Description

This action inserts "page count" field in the position of the caret. The number is formatted according to NumberType 136 property.

See also

- TrvActionInsertPageNumber 150

1.3.6.10 TrvActionInsertPageNumber

TrvActionInsertPageNumber is the action for "Insert | Page Number" command.

Unit RichViewActions 50;

Syntax

```
TrvActionInsertPageNumber = class(TrvActionInsertCustomPageNumber 136)
```

Hierarchy

TObject
TPersistent
TComponent
TBasicAction
TContainedAction
TCustomAction
TAction
TrvCustomAction 206
TrvAction 195
TrvActionInsertCustomPageNumber 136

Description

This action inserts "page number" field in the position of the caret. The number is formatted according to NumberType 136 property.

See also

- TrvActionInsertPageCount 149

1.3.6.11 TrvActionInsertPicture

TrvActionInsertPicture is the action for "Insert | Picture" command.

Unit RichViewActions 50;

Syntax

```
TrvActionInsertPicture = class(TrvActionCustomIO 54)
```

Hierarchy

TObject
TPersistent
TComponent
TBasicAction
TContainedAction
TCustomAction
TAction
TrvCustomAction 206
TrvAction 195

TrvActionCustomIO 54

Description

This action inserts a page picture from file in the caret position. Unlike all other file opening actions, this action detects file format by the file extension, not by the file filter index.

The following additional properties can be applied to inserted pictures:

- VAlign 153 – the image alignment relative to text;
- Spacing 152 – padding (spacing between the image itself and its border);
- OuterHSpacing, OuterVSpacing 152 – horizontal and vertical spacing around the image border;
- BorderWidth, BorderColor 151 – border width and color;
- BackgroundColor 151 – background color;
- MaxImageSize 152 – maximal allows size (the image will be stretched if it exceeds this size);
- StoreFileName 152, StoreFileNameInItemName 153 allow storing file name with the image.

1.3.6.11.1 Properties

1.3.6.11.1.1 TrvActionInsertPicture.BackgroundColor

The property specify a background color for inserted images.

```
property BackgroundColor: TColor;
```

Default value

clNone (no background color)

1.3.6.11.1.2 TrvActionInsertPicture.BorderWidth, BorderColor

The property specify a border width and color for inserted images.

```
property BorderWidth: TRVStyleLength;
property BorderColor: TColor;
```

Assign a positive value to BorderWidth to add border to inserted pictures.

BorderWidth is measured in GetControlPanel 208.UnitsProgram 33.

Default value

- BorderWidth: 0
- BorderColor: clBlack

1.3.6.11.1.3 TrvActionInsertPicture.DefaultExt

Specifies the default file extension.

```
property DefaultExt: String;
```

DefaultExt specifies a file extension that is appended automatically to the selected file name, unless the selected file name already includes a registered extension. If the user selects a file name with an extension that is unregistered, DefaultExt is appended to the unregistered extension.

This property is assigned to DefaultExt property of the file opening dialog (TOpenPictureDialog).

Default value:

'bmp'

1.3.6.11.1.4 TrvActionInsertPicture.Filter

Determines the file masks (filters) available in the file opening dialog.

```
property Filter: String;
```

If this property is not empty, it is assigned to the Filter property of the file opening dialog (TOpenPictureDialog). Otherwise, a filter containing all registered graphic formats is created.

Default value:

" (empty string)

See also:

- TrvActionItemProperties.GraphicFilter 202
- TrvActionItemProperties.BackgroundGraphicFilter 202
- TrvActionTableProperties.BackgroundGraphicFilter 188

1.3.6.11.1.5 TrvActionInsertPicture.MaxImageSize

Defines the maximal image size for inserting without scaling.

```
property MaxImageSize: TRVStyleLength;
```

A positive value limits sizes of inserted images. If width or height of the inserted picture exceeds this value, the image is scaled down proportionally. The actual image is not changed, it's just inserted scaled (its *rvepImageHeight* and *rvepImageWidth* properties are assigned, see the help file on TRVExtraItemProperty).

This value is measured in GetControlPanel 208.UnitsProgram 33.

Default value:

0 (no scaling)

1.3.6.11.1.6 TrvActionInsertPicture.OuterHSpacing, OuterVSpacing

Specifies outer spacing for inserted images (spacing around the image border)

```
property OuterHSpacing: TRVStyleLength;
property OuterVSpacing: TRVStyleLength;
```

OuterHSpacing defines a horizontal spacing, OuterVSpacing defines a vertical spacing,

Assign a positive values to these property to add spacing around inserted pictures.

These values are measured in GetControlPanel 208.UnitsProgram 33.

Default value

0

1.3.6.11.1.7 TrvActionInsertPicture.Spacing

Specifies padding for inserted images (spacing between the image and its border)

```
property Spacing: TRVStyleLength;
```

Assign a positive value to this property to add padding for inserted pictures.

This value is measured in GetControlPanel 208.UnitsProgram 33.

Default value

0

1.3.6.11.1.8 TrvActionInsertPicture.StoreFileName

Specifies whether file names (full paths) are stored in item properties of inserted pictures.

```
property StoreFileName: Boolean;
```

If this property is *True*, path to the graphic file is stored in the document in the *rvespImageFileName* item property (see the RichView help file, TRVExtraItemStrProperty type).This is a recommended way to store path to graphic files. You can use these file names when saving HTML files (Include *rvsoUseItemImageFileNames* in TrvActionSave.SaveOptions 71 and TrvActionExport.SaveOptions 64).

Default value

False

See also:

- TrvActionPasteSpecial.StoreFileName 79
- TrvActionItemProperties.StoreImageFileName 203
- TrvActionTableProperties.StoreImageFileName 188
- TCustomRichView.RTFReadProperties.StoreImagesFileNames
- HTML Import (for images inserted with HTML, paths to graphic files are always stored in *rvespImageFileName*)

1.3.6.11.1.9 TrvActionInsertPicture.StoreFileNameInItemName

Specifies whether file names (full paths) are stored in item names of inserted pictures.

```
property StoreFileNameInItemName: Boolean;
```

If this property is *True*, path to the graphic file is stored in the document in the item name (item text).

It's not recommended to use this option. Use StoreFileName 152 instead.

Default value

False

See also:

- TrvActionPasteSpecial.StoreFileNameInItemName 80
- TrvActionItemProperties 201 (does not support storing paths in item names)
- RTF reading (does not support storing paths in item names)
- TRVHtmlViewImporter 222.StoreFileNameInItemName
- TRVHtmlImporter 221.Options (*rvhtmloImgSrcToImageName* option)

1.3.6.11.1.10 TrvActionInsertPicture.VAlign

Specifies the vertical alignment for inserted pictures.

```
property VAlign: TRVVAlign;
```

Default value:

rvvaBaseline

1.3.6.12 TrvActionInsertSymbol

TrvActionInsertSymbol is the action for "Insert | Symbol" command.

Unit RichViewActions 50;

Syntax

```
TrvActionInsertSymbol = class(TrvAction 195)
```

Hierarchy

TObject
TPersistent
TComponent
TBasicAction
TContainedAction
TCustomAction
TAction
TrvCustomAction 206
TrvAction 195

Description

This action inserts a character in the caret position in TCustomRichViewEdit component. Text style for characters inserted by this action have *rvprDoNotAutoSwitch* in Protection.

1.3.6.12.1 Properties

1.3.6.12.1.1 TrvActionInsertSymbol.AlwaysInsertUnicode

Specifies whether a character is always inserted as Unicode.

```
property AlwaysInsertUnicode: Boolean;
```

If *False*, value of Unicode property of text style used to insert a character depends on the settings made in the dialog.

If *True*, value of Unicode property of text style used to insert a character is always *True*.

Default value:

False for Delphi 4-2007, *True* for Delphi 2009 and newer

1.3.6.12.1.2 TrvActionInsertSymbol.DisplayUnicodeBlocks

Shows or hides combo box allowing to choose Unicode blocks.

```
property DisplayUnicodeBlocks: Boolean;
```

This combo box can be displayed when the user chooses "Unicode" in the character set combo box (or if SymbolType 154 =*rvisUnicode*).

It allows restricting a range of Unicode characters displayed in the grid to the specified block.

Names of Unicode blocks are not localized (always displayed in English).

Default value:

True

1.3.6.12.1.3 TrvActionInsertSymbol.SymbolType

Specifies how the user can choose a character.

```
type
  TRVInsertSymbolType = (rvisBoth, rvisUnicode, rvisSingleByte);

property SymbolType: TRVInsertSymbolType;
```

Value	File Type
rvisBoth	The user selects a character set, then a character. There is a special character set: "Unicode". If it is selected, the user can choose any Unicode character.
rvisUnicode	The combo box for selecting character sets is hidden. The user chooses a Unicode character.
rvisSingleByte	The user selects a character set, then a character.

When choosing a Unicode character (in *rvisBoth* or *rvisUnicode* modes), a combo box with Unicode blocks can be displayed, see DisplayUnicodeBlocks 154.

Even if ANSI character is chosen (in *rvisBoth* or *rvisSingleByte* modes), it can be inserted using a Unicode text style, see AlwaysInsertUnicode 154.

Default value:

rvisBoth

1.3.6.12.1.4 TrvActionInsertSymbol.UseCurrentFont

Defines how the initial value for a character's font is chosen.

```
property UseCurrentFont: Boolean;
```

If *False*: "Symbol" font is used when the dialog is displayed for the first time. The font of the last chosen character is used for subsequent calls.

If *True*: the font of the current text style is used.

1.3.6.13 TrvActionInsertText

TrvActionInsertText is the action for text insertion.

Unit RichViewActions 50;

Syntax

```
TrvActionInsertText = class(TrvAction 195)
```

Hierarchy

TObject
TPersistent
TComponent
TBasicAction
TContainedAction
TCustomAction
TAction
TrvCustomAction 206
TrvAction 195

Description

This action calls OnInsertText 155 event and inserts the returned text in the caret position. This action does not have predefined Caption and Hint. Assign these properties yourself.

For example, this action may be used to insert the current date or time.

This action does not introduce any new properties in addition to properties of TrvAction.

See also:

- TrvActionEvent 199

1.3.6.13.1 Events

1.3.6.13.1.1 TrvActionInsertText.OnInsertText

Requests text for insertion.

```
type
  TRVAInsertTextEvent = procedure (Sender: TObject; Editor: TCustomRichViewEdit;
    var Text: String) of object;

property OnInsertText: TRVAInsertTextEvent;
```

The string assigned to **Text** will be inserted in **Editor**.

1.3.7 Notes

1.3.7.1 TrvActionCustomInsertSidenote

TrvActionCustomInsertSidenote is a base class for the actions inserting sidenotes and text box items.

Unit RichViewActions [50];

Syntax

```
TrvActionCustomInsertSidenote = class(TrvActionInsertNote [160])
```

Hierarchy

TObject
TPersistent
TComponent
TBasicAction
TContainedAction
TCustomAction
TAction
TrvCustomAction [206]
TrvAction [195]
TrvActionInsertNote [160]

Description

This action introduces properties that are assigned to properties of an inserted sidenote (or a text box item):

- BoxProperties [157]
- BoxPosition [156]

Text in the note's Document is formatted either according to StyleTemplateName [157] (if style templates are used) or according to Font [161] (otherwise).

This class is not used directly. The following actions are inherited from TrvActionCustomInsertSidenote:

- TrvActionInsertSidenote [161]
- TrvActionInsertTextBox [162]

1.3.7.1.1 Properties

1.3.7.1.1.1 TrvActionCustomInsertSidenote.BoxPosition

Specifies position properties for the floating box.

```
type
  TRVABoxPosition = class (TRVBoxPosition);
property BoxPosition: TRVABoxPosition;
```

This property is assigned to BoxPosition property of the inserted sidenote / text box item.

Default values are different in inherited actions.

TrvActionInsertSidenote [161]**:**

- HorizontalAnchor=*rvhanMainTextArea*;
- HorizontalPositionKind=*rvfpkAlignment*;
- HorizontalAlignment=*rvfphalRight*;
- VerticalAnchor=*rvvanParagraph*;
- VerticalPositionKind=*rvfpkAbsPosition*;
- VerticalOffset=0.

TrvActionInsertTextBox [162]**:**

- HorizontalAnchor=*rvhanMainTextArea*;
- HorizontalPositionKind=*rvfpkAlignment*;
- HorizontalAlignment=*rvfphalCenter*;

- VerticalAnchor=*rvvanLine*;
- VerticalPositionKind=*rvfpkAbsPosition*;
- VerticalOffset=20.

1.3.7.1.1.2 TrvActionCustomInsertSidenote.BoxProperties

Specifies size, background, border, vertical alignment properties for the floating box.

```
type
  TRVABoxProperties = class (TRVBoxProperties);
property BoxProperties: TRVABoxProperties;
```

This property is assigned to BoxProperties property of the inserted sidenote / text box item.

Default values

- WidthType=*rvbwtRelPage*;
- Width=20000;

1.3.7.1.1.3 TrvActionCustomInsertSidenote.StyleTemplateName

Specifies the name of a style template that will be applied to a paragraph of a sidenote's Document.

```
property StyleTemplateName: String;
```

This property is used only if the target editor's UseStyleTemplates=*True*. Otherwise, the 0-th paragraph style and Font 161 are used.

Default values are different in inherited actions:

TrvActionInsertSidenote 161**:**

'Sidenote Text' (note: this style template is standard for RichViewActions, but it is not standard for Microsoft Word)

TrvActionInsertTextBox 162**:**

'Normal'

1.3.7.2 TrvActionEditNote

TrvActionEditNote allows editing a footnote, an endnote, a sidenote, or a text box in a non-modal dialog window.

Unit RichViewActions 50;

Syntax

```
TrvActionInsertNote = class(TrvCustomEditorAction 208)
```

Hierarchy

TObject
TPersistent
TComponent
TBasicAction
TContainedAction
TCustomAction
TAction
TrvCustomAction 206
TrvAction 195
TrvCustomEditorAction 208

Description

ScaleRichView note: it's not recommended to use this action for TSRichViewEdit. Use TsrvActionEditNote instead.

This action can be executed directly, or it is called by TrvActionInsertFootnote 159, TrvActionInsertEndnote 159,

TrvActionInsertSidenote[161], TrvActionInsertTextBox[162].

It's recommended to assign Control[195] property for this action; otherwise, this property is assigned automatically when the action is executed for the first time.

When executed, the action shows a non-modal dialog window containing TRichViewEdit for editing subdocuments of items of the following types:

- footnote;
- endnote;
- sidenote;
- text box item.

If there is no item of these types in the caret position of the target editor, the action can move the caret to the next such item, if JumpToNextNote[158]=*True*. If there are no such items, a note editor is shown disabled.

If a note editor window is already visible when this action is executed, the action activates its window.

After the execution, the window is still open and the action reacts on the caret movement in the target editor (more exactly, in Control[195]): the window always shows a document for the current note. The user can edit the note document. It is saved to the note (as an undoable operation) automatically in the following cases:

- a caret movement in Control[195];
- saving Control[195] to a file or a stream;
- printing or displaying a print preview of Control[195];
- showing or hiding the note editor window;
- assigning a different value to Control[195].

If you do not want to save recent changes, undo them in the note editor (there is no "Cancel Changes" command).

Editing in the note editor

By default, a note editor is shown in a floating window, without any additional controls in this window. If you attempt to use toolbar buttons located on another form, you will see the problem: clicking the button activates another form, and the command will be executed for an editor in this form.

So you have two main options to implement UI for a note editor:

- use OnFormCreate[209] event to insert additional control in the note editor window (such as menu or toolbar)
- insert the note window in the main form, and switch RVAControlPanel.DefaultControl[26] when the note editor acquires and releases the input focus (see the example in the topic about OnShowing and OnHide[209])

The latter is implemented in the ActionTest demos.

1.3.7.2.1 Properties

1.3.7.2.1.1 TrvActionEditNote.JumpToNextNote

Allows moving the caret to the next note in the target editor, when this action is executed.

```
property JumpToNextNote: Boolean;
```

The action always shows a note editor window on execution.

If **JumpToNextNote**=*True*, and the current item in the target editor is not a note when the action is executed, the action finds the next note in the target editor, moves the caret to it, and then shows a note editor window.

Default value:

True

1.3.7.3 TrvActionInsertEndnote

TrvActionInsertEndnote inserts a new endnote in the document.

Unit RichViewActions 50;

Syntax

```
TrvActionInsertEndnote = class(TrvActionInsertNote 160)
```

Hierarchy

TObject
TPersistent
TComponent
TBasicAction
TContainedAction
TCustomAction
TAction
TrvCustomAction 206
TrvAction 195
TrvActionInsertNote 160

Description

ScaleRichView note: it's not recommended to use this action for TSRichViewEdit. Use TsrvActionInsertEndnote instead.

TrvActionInsertEndnote inserts a new endnote in the document, then executes ActionEditNote 161 to open an editor for this note.

If TrvActionEditNote 157's editor is active when this action is executed, this action inserts a reference to the parent endnote (TRVNoteReferenceItemInfo item).

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 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 161 property.

Note: "endnote reference" and "endnote text" style templates are added by default to TrvActionNew.StyleTemplates, so they are included in new documents.

1.3.7.4 TrvActionInsertFootnote

TrvActionInsertFootnote inserts a new footnote in the document.

Unit RichViewActions 50;

Syntax

```
TrvActionInsertFootnote = class(TrvActionInsertNote 160)
```

Hierarchy

TObject
TPersistent
TComponent
TBasicAction
TContainedAction

TCustomAction
TAction
TrvCustomAction 206
TrvAction 195
TrvActionInsertNote 160

Description

ScaleRichView note: it's not recommended to use this action for TSRichViewEdit. Use TsrvActionInsertFootnote instead.

TrvActionInsertFootnote inserts a new footnote in the document, then executes ActionEditNote 161 to open an editor for this note.

If TrvActionEditNote 157's editor is active when this action is executed, this action inserts a reference to the parent footnote (TRVNoteReferenceItemInfo item).

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 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 161 property.

Note: "footnote reference" and "footnote text" style templates are added by default to TrvActionNew.StyleTemplates, so they are included in new documents.

1.3.7.5 TrvActionInsertNote

TrvActionInsertNote is a base class for the actions inserting footnotes, endnotes, sidenotes and text boxes.

Unit RichViewActions 50;

Syntax

```
TrvActionInsertNote = class(TrvAction 195)
```

Hierarchy

TObject
TPersistent
TComponent
TBasicAction
TContainedAction
TCustomAction
TAction
TrvCustomAction 206
TrvAction 195

Description

This class is not used directly.

The following actions are inherited from TrvActionInsertNote:

- TrvActionInsertFootnote 159
- TrvActionInsertEndnote 159
- TrvActionInsertSidenote 161
- TrvActionInsertTextBox 162

1.3.7.5.1 Properties

1.3.7.5.1.1 TrvActionInsertNote.ActionEditNote

Links this action to TrvActionEditNote[157] action.

```
property ActionEditNote: TrvActionEditNote[157];
```

The action executes ActionEditNote after inserting a note.

If this link is not defined, the first found TrvActionEditNote[157] action on the same form/datamodule is used. If not found, a window for editing the inserted note is not shown.

1.3.7.5.1.2 TrvActionInsertNote.Font

Default font for text in a note's document.

```
type
  TRVAFont = TFont;
property Font: TRVAFont;
```

This value is used if UseStyleTemplates=*False* for the target editor, or the required style template does not exist in Style.StyleTemplates of the target editor. Otherwise, note text is formatted according to a style template.

Style template names:

- "footnote text" for footnotes,
- "endnote text" for endnotes,
- StyleTemplateName[157] for sidenotes and text boxes (default values: "Sidenote Text" for sidenotes, "Normal" for text boxes)

Default value:

'Arial', 10

1.3.7.6 TrvActionInsertSidenote

TrvActionInsertSidenote inserts a new sidenote (a note in a floating box) in the document.

Unit RichViewActions[50];

Syntax

```
TrvActionCustomInsertSidenote = class(TrvActionCustomInsertSidenote[156])
```

Hierarchy

TObject
TPersistent
TComponent
TBasicAction
TContainedAction
TCustomAction
TAction
TrvCustomAction[206]
TrvAction[195]
TrvActionInsertNote[160]
TrvActionCustomInsertSidenote[156]

Description

ScaleRichView note: it's not recommended to use this action for TSRichViewEdit. Use TsrvActionInsertSidenote instead.

TrvActionInsertSidenote inserts a new sidenote in the document, then executes ActionEditNote[161] to open an editor for this note.

If TrvActionEditNote[157]'s editor is active when this action is executed, this action inserts a reference to the parent sidenote (TRVNoteReferenceItemInfo item).

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 property.

If style templates are used, the sidenote and the sidenote reference characters are formatted using RefStyleTemplateName[162] 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[157] style template. Otherwise, or if this style template does not exist, sidenote text is formatted using Font[161] property.

The following properties are assigned to the inserted sidenote:

- BoxProperties[157] (default: width: 20% of page width, height: auto)
- BoxPosition[156] (default: horizontal - right aligned to the main text area, vertical - aligned to the top of the paragraph)

1.3.7.6.1 Properties

1.3.7.6.1.1 TrvActionInsertSidenote.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: String;
```

This property is used only if the target editor's UseStyleTemplates=*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)

1.3.7.7 TrvActionInsertTextBox

TrvActionInsertTextBox inserts a new text box item in the document.

Unit RichViewActions[50];

Syntax

```
TrvActionInsertTextBox = class(TrvActionCustomInsertSidenote[156])
```

Hierarchy

TObject
TPersistent
TComponent
TBasicAction
TContainedAction
TCustomAction
TAction
TrvCustomAction[206]
TrvAction[195]
TrvActionInsertNote[160]
TrvActionCustomInsertSidenote[156]

Description

ScaleRichView note: it's not recommended to use this action for TSRichViewEdit. Use

TsrvActionInsertTextBox instead.

TrvActionInsertTextBox inserts a new text box item in the document, then executes ActionEditNote[161] to open an editor for its document.

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 property.

If style templates are used, this text item's paragraph and text are formatted using StyleTemplateName[157] style template. Otherwise, or if this style template does not exist, this text is formatted using Font[161] property.

The following properties are assigned to the inserted text box:

- BoxProperties[157] (default: width: 20% of page width, height: auto)
- BoxPosition[156] (default: horizontal - centered in the main text area, vertical - 20 pixels below the top of the line)

1.3.8 Tables

1.3.8.1 TrvActionInsertTable

TrvActionInsertTable is the action for "Table | Insert Table" command.

Unit RichViewActions[50];

Syntax

```
TrvActionInsertTable = class(TrvAction[195])
```

Hierarchy

TObject
TPersistent
TComponent
TBasicAction
TContainedAction
TCustomAction
TAction
TrvCustomAction[206]
TrvAction[195]

Description

The action can insert a table in two ways:

- using a table dialog (when executed)
- using a special table size window (when ShowTableSizeDialog[171] is called).

In the table dialog, the user can specify a count of rows and columns and table width modes:

- autosize (table.BestWidth=0; widths of table cells are specified in pixels (or twips) so that the table has width not exceeding the current document width)
- fit window (table.BestWidth=-100 (i.e. 100%); widths of table cells are not specified)
- custom size (table.BestWidth is defined in the dialog, in pixels (or twips) or percent; widths of table cells are not specified)

When inserting using a table size window, table.BestWidth is specified in BestWidth[164]. If it is zero, cells widths are defined like in "autosize" mode.

If the user checked "remember dimensions for new tables" in the dialog, RowCount[169], ColCount[167] and BestWidth[164] properties are set to the values entered in the dialog.

Before the insertion:

- the properties of this action are assigned to the table properties of the same names;
- RowsKeepTogether[169] and RowsVAlign[169] are assigned to properties of all rows of the table,
- OnInserting[171] event occurs.

The user can assign properties of this action by checking "Default" check box in the table properties dialog displayed by TrvActionTableProperties[187] or TrvActionItemProperties[201].

1.3.8.1.1 Properties

1.3.8.1.1.1 TrvActionInsertTable.BackgroundPicture

Defines the table background image.

```
property BackgroundPicture: TPicture;
```

BackgroundPicture.Graphic is assigned to the table's BackgroundImage

1.3.8.1.1.2 TrvActionInsertTable.BackgroundStyle

Defines the table background style.

```
property BackgroundStyle: TRVItemBackgroundStyle;
```

This property is assigned to the table property of the same name.

Default value:

rvbsColor

1.3.8.1.1.3 TrvActionInsertTable.BestWidth

Defines the table width.

```
property BestWidth: TRVHTMLLength;
```

This value is used:

- as an initial value for initializing a table dialog (on the action execution),
- directly (when ShowTableSizeDialog[171] is called).

If the user chose to remember dimensions for new tables, value from the dialog is assigned to this property.

This property (or the value from the table dialog) is assigned to table.BestWidth property. 0 means a default width, negative values mean percent, positive values mean GetControlPanel[208].UnitsProgram[33].

If it is zero (i.e. a default table width), the action sets BestWidth properties of all table cells to positive values so that the table width does not exceed the current document width. Otherwise, BestWidth properties of all cells are 0.

Default value:

0

1.3.8.1.1.4 TrvActionInsertTable.BorderColor

Defines the table border color (or color of dark sides of 3D table border)

```
property BorderColor: TColor;
```

This property is assigned to the table property of the same name.

Default value:

clWindowText

1.3.8.1.1.5 TrvActionInsertTable.BorderHSpacing

Defines the spacing between the left table border and the cells, and between the right table border and the cells.

```
property BorderHSpacing: TRVStyleLength;
```

This value is measured in GetControlPanel[208].UnitsProgram[33].

This property is assigned to the table property of the same name (converted to Units of the target editor).

Default value:

2

This default value assumes that this property is measured in pixels. For twips, this value may be too small.

1.3.8.1.1.6 TrvActionInsertTable.BorderLightColor

Defines the color of light sides of 3D table border

```
property BorderLightColor: TColor;
```

This property is assigned to the table property of the same name.

Default value:

clBtnHighlight

1.3.8.1.1.7 TrvActionInsertTable.BorderStyle

Defines the style of table border.

```
property BorderStyle: TRVTableBorderStyle;
```

This property is assigned to the table property of the same name.

Default value:

rvtbColor

1.3.8.1.1.8 TrvActionInsertTable.BorderVSpacing

Defines the spacing between the top table border and the cells, and between the bottom table border and the cells.

```
property BorderVSpacing: TRVStyleLength;
```

This value is measured in GetControlPanel 208.UnitsProgram 33.

This property is assigned to the table property of the same name (converted to Units of the target editor).

Default value:

2

This default value assumes that this property is measured in pixels. For twips, this value may be too small.

1.3.8.1.1.9 TrvActionInsertTable.BorderWidth

Defines the width of table border.

```
property BorderWidth: TRVStyleLength;
```

This value is measured in GetControlPanel 208.UnitsProgram 33.

This property is assigned to the table property of the same name (converted to Units of the target editor).

0 hides the border.

Default value:

1

This default value assumes that this property is measured in pixels. For twips, this value may be too small.

1.3.8.1.1.10 TrvActionInsertTable.CellBorderColor

Defines the default color of cell borders (or of dark sides of 3D cell borders).

```
property CellBorderColor: TColor;
```

This property is assigned to the table property of the same name.

Default value:
clWindowText

1.3.8.1.1.11 TrvActionInsertTable.CellBorderLightColor

Defines the default color of light sides of 3D cell borders.

```
property CellBorderLightColor: TColor;
```

This property is assigned to the table property of the same name.

Default value:
clBtnHighlight

1.3.8.1.1.12 TrvActionInsertTable.CellBorderStyle

Defines the style of cells borders.

```
property CellBorderStyle: TRVTableBorderStyle;
```

This property is assigned to the table property of the same name.

Default value:
rvtbColor

1.3.8.1.1.13 TrvActionInsertTable.CellBorderWidth

Defines the width of cells borders.

```
property CellBorderWidth: TRVStyleLength;
```

This value is measured in GetControlPanel[208].UnitsProgram[33].

This property is assigned to the table property of the same name (converted to Units of the target editor).

0 hides the border.

Default value:
1
This default value assumes that this property is measured in pixels. For twips, this value may be too small.

1.3.8.1.1.14 TrvActionInsertTable.CellHPadding,CellVPadding

The properties define spacing between border and content in cells.

```
property CellHPadding: TRVStyleLength;
property CellVPadding: TRVStyleLength;
property CellPadding: TRVStyleLength;
```

These values are measured in GetControlPanel[208].UnitsProgram[33].

CellHPadding defines a horizontal spacing, CellVPadding defines a vertical spacing.

These properties are assigned to the table properties of the same names (converted to Units of the target editor).

CellPadding is used to simplify access to these properties. Assigning value to CellPadding assigns it to CellHPadding and CellVPadding. CellPadding returns (CellHPadding+CellVPadding)/2.

Default value:
1
This default value assumes that these properties are measured in pixels. For twips, this value may be too small.

1.3.8.1.1.15 TrvActionInsertTable.CellHSpacing

Defines the horizontal spacing between cells (i.e. spacing between table columns).

```
property CellHSpacing: TRVStyleLength;
```

This value is measured in GetControlPanel[208].UnitsProgram[33].

This property is assigned to the table property of the same name (converted to Units of the target editor).

Default value:

2

This default value assumes that this property is measured in pixels. For twips, this value may be too small.

1.3.8.1.1.16 TrvActionInsertTable.CellVSpacing

Defines the vertical spacing between cells (i.e. spacing between table rows).

```
property CellVSpacing: TRVStyleLength;
```

This value is measured in GetControlPanel[208].UnitsProgram[33].

This property is assigned to the table property of the same name (converted to Units of the target editor).

Default value:

2

This default value assumes that this property is measured in pixels. For twips, this value may be too small.

1.3.8.1.1.17 TrvActionInsertTable.ColBandSize, RowBandSize

The properties defining count of columns/rows in alternating column/row bands.

```
property ColBandSize: Integer;
property RowBandSize: Integer;
```

RowBandSize specify the count of rows in row bands.

ColBandSize specify the count of columns in column bands.

These properties are assigned to the table properties of the same names.

Default value:

1

See also

- HeadingRowColor and other colors[171]

1.3.8.1.1.18 TrvActionInsertTable.ColCount

Defines the count of table columns.

```
property ColCount: Integer;
```

This property is used as an initial value for initializing a table dialog. If the user chose to remember dimensions for new tables, value from the dialog is assigned to this property.

This property is ignored in ShowTableSizeDialog[171] method.

Default value:

2

1.3.8.1.1.19 TrvActionInsertTable.Color

Defines the background table color.

```
property Color: TColor;
```

This property is assigned to the table property of the same name.

Default value:

clNone

1.3.8.1.1.20 TrvActionInsertTable.HeadingRow Count

Defines the count of heading rows (rows repeated on each page containing this table).

```
property HeadingRowCount: Integer;
```

This property is assigned to the table property of the same name.

Default value:

0

1.3.8.1.1.21 TrvActionInsertTable.HOutermostRule

Allows showing horizontal lines between the top and the bottom table borders and cells.

```
property HOutermostRule: Boolean;
```

This property is assigned to the table property of the same name.

Horizontal rules are drawn only if HRuleWidth 168 >0.

Default value:

False

1.3.8.1.1.22 TrvActionInsertTable.HRuleColor

Defines the color of horizontal rules.

```
property HRuleColor: TColor;
```

This property is assigned to the table property of the same name.

Horizontal rules are drawn only if HRuleWidth 168 >0.

Default value:

clWindowText

1.3.8.1.1.23 TrvActionInsertTable.HRuleWidth

Defines the width of horizontal rules.

```
property HRuleWidth: TRVStyleLength;
```

This value is measured in GetControlPanel 208 .UnitsProgram 33 .

This property is assigned to the table property of the same name (converted to Units of the target editor).

0 hides rules.

Default value:

0

1.3.8.1.1.24 TrvActionInsertTable.ItemText

Defines the item text (item name) for the table.

```
property ItemText: String;
```

This property is used as a parameter of InsertItem method.

Default value:

" (empty string)

1.3.8.1.1.25 TrvActionInsertTable.RowCount

Defines the count of table rows.

```
property RowCount: Integer;
```

This property is used as an initial value for initializing a table dialog. If the user chose to remember dimensions for new tables, value from the dialog is assigned to this property.

This property is ignored in ShowTableSizeDialog 171 method.

Default value:

2

1.3.8.1.1.26 TrvActionInsertTable.RowsKeepTogether

Defines the "keep together" property for all rows of inserted tables.

```
property RowsKeepTogether: Boolean;
```

This property is assigned to KeepTogether properties of all rows of inserted tables.

If *True*, the component avoids breaking pages inside rows (pagination adds page breaks between rows, when possible)

ScaleRichView note: TSRichViewEdit does not support page breaks inside table cells yet.

Default value:

False

1.3.8.1.1.27 TrvActionInsertTable.RowsVAlign

Defines the default vertical alignment of content of cells of all rows of inserted tables.

```
property RowsVAlign: TRVCellVAlign;
```

This property is assigned to VAlign properties of all rows of inserted tables.

Default value:

rvcTop

1.3.8.1.1.28 TrvActionInsertTable.TableOptions

Defines the table options.

```
property TableOptions: TRVTableOptions;
```

This property is assigned to the table property of the same name.

Default value:

[*rvtoEditing, rvtoRowSizing, rvtoColSizing, rvtoRowSelect, rvtoColSelect*]

1.3.8.1.1.29 TrvActionInsertTable.TablePrintOptions

Defines the table printing options.

```
property TablePrintOptions: TRVTablePrintOptions;
```

This property is assigned to the table's PrintOptions property.

Default value:

[*rvtoHalftoneBorders*,*rvtoRowsSplit*]

1.3.8.1.1.30 TrvActionInsertTable.VisibleBorders

Defines visible sides of a table border.

```
property VisibleBorders: TRVBooleanRect;
```

This property is assigned to the table property of the same name.

Default value:

(*True, True, True, True*)

1.3.8.1.1.31 TrvActionInsertTable.VOutermostRule

Allows showing vertical lines between the left and the right table borders and cells.

```
property VOutermostRule: Boolean;
```

This property is assigned to the table property of the same name.

Vertical rules are drawn only if VRuleWidth 170 >0.

Default value:

False

1.3.8.1.1.32 TrvActionInsertTable.VRuleColor

Defines the color of vertical rules.

```
property VRuleColor: TColor;
```

This property is assigned to the table property of the same name.

Vertical rules are drawn only if VRuleWidth 170 >0.

Default value:

clWindowText

1.3.8.1.1.33 TrvActionInsertTable.VRuleWidth

Defines the width of vertical rules.

```
property VRuleWidth: TRVStyleLength;
```

This value is measured in GetControlPanel 208 .UnitsProgram 33 .

This property is assigned to the table property of the same name (converted to Units of the target editor).

0 hides rules.

Default value:

0

1.3.8.1.1.34 TrvActionInsertTable's row and column colors

A set of properties defining default colors of cells in specific rows and columns.

```
property HeadingRowColor: TColor;
property LastRowColor: TColor;
property OddRowsColor: TColor;
property EvenRowsColor: TColor;
property FirstColumnColor: TColor;
property LastColumnColor: TColor;
property OddColumnsColor: TColor;
property EvenColumnsColor: TColor;
```

These properties are assigned to the table properties of the same names.

Default values

clNone

See also

- ColBandSize, RowBandSize

1.3.8.1.2 Methods

1.3.8.1.2.1 TrvActionInsertTable.Show TableSizeDialog

These methods display a table sizing window. When this window is closed, a table with the specified count of rows and columns is inserted in **Target**.

```
procedure ShowTableSizeDialog(Target: TCustomRichViewEdit; Button: TControl); overload;
procedure ShowTableSizeDialog(Target: TCustomRichViewEdit; const ButtonRect: TRect); over
```

The first version of this method displays this window relative to the position of **Button**.

In the second version of this method, this position is specified in **ButtonRect** (some third-party toolbar component do not use controls for buttons, so the first version of this method cannot be used).

If the user presses Escape while this window is open, the window is closed and insertion is canceled.

1.3.8.1.3 Events

1.3.8.1.3.1 TrvActionInsertTable.OnCloseTableSizeDialog

Occurs when a table sizing window (shown by ShowTableSizeDialog 171 method) is closed.

```
property OnCloseTableSizeDialog: TNotifyEvent;
```

This event can be used to uncheck a button pressed before calling ShowTableSizeDialog 171 method (if you have some button that must be pressed while this window is displayed).

1.3.8.1.3.2 TrvActionInsertTable.OnInserting

Occurs before **table** is inserted.

```
type
    TRVAInsertTableEvent = procedure (Sender: TrvActionInsertTable; table: TRVTableItemIn

property OnInserting: TRVAInsertTableEvent;
```

This event occurs after all the action's properties are applied to **table**. You can make additional changes in this event.

1.3.8.2 TrvActionTableCell

TrvActionTableCell is a base class for actions performing operations on table and table cells.

Unit RichViewActions 50;

Syntax

```
TrvActionTableCell = class(TrvAction 195)
```

Hierarchy

TObject
TPersistent
TComponent
TBasicAction
TContainedAction
TCustomAction
TAction
TrvCustomAction 206
TrvAction 195

Description

This action does not introduce any new properties in addition to properties of TrvAction.

This action is not used directly. This action has the following direct descendants:

- TrvActionTableRCBase 189
- TrvActionTableDeleteTable 183
- TrvActionTableMergeCells 186
- TrvActionTableSplitCells 193
- TrvActionTableSelectTable 191
- TrvActionTableSelectCell 189
- TrvActionTableMultiCellAttributes 187
- TrvActionTableProperties 187

1.3.8.3 TrvActionTableCellAllBorders

TrvActionTableCellAllBorders is the action for "Table | Cell Borders | All Borders" command.

Unit RichViewActions 50;

Syntax

```
TrvActionTableCellAllBorders = class(TrvActionTableCellBorder 173)
```

Hierarchy

TObject
TPersistent
TComponent
TBasicAction
TContainedAction
TCustomAction
TAction
TrvCustomAction 206
TrvAction 195
TrvActionTableCell 172
TrvActionTableMultiCellAttributes 187
TrvActionTableCellBorder 173

Description

The action makes all sides of borders of the selected table cells (or the edited cell) visible/invisible.

This action does not introduce any new properties in addition to properties of TrvAction.

See also:

- TrvActionTableCellLeftBorder 174
- TrvActionTableCellRightBorder 175
- TrvActionTableCellTopBorder 179
- TrvActionTableCellBottomBorder 173
- TrvActionTableCellNoBorders 175

1.3.8.4 TrvActionTableCellBorder

TrvActionTableCellBorder is a base class for the actions changing cell borders.

Unit RichViewActions 50;

Syntax

```
TrvActionTableCellBorder = class(TrvActionTableMultiCellAttributes 187)
```

Hierarchy

TObject
TPersistent
TComponent
TBasicAction
TContainedAction
TCustomAction
TAction
TrvCustomAction 206
TrvAction 195
TrvActionTableCell 172
TrvActionTableMultiCellAttributes 187

Description

This action does not introduce any new properties in addition to properties of TrvAction.

This action is not used directly. The following actions are inherited from it:

- TrvActionTableCellLeftBorder 174
- TrvActionTableCellRightBorder 175
- TrvActionTableCellTopBorder 179
- TrvActionTableCellBottomBorder 173
- TrvActionTableCellAllBorders 172
- TrvActionTableCellNoBorders 175

These actions change table.Cells[].VisibleBorders properties for the selected cells (or the edited cell). All these actions (except for TrvActionTableCellNoBorders 175) are checkbox-like buttons: they are checked if the corresponding sides of cells borders are visible.

1.3.8.5 TrvActionTableCellBottomBorder

TrvActionTableCellBottomBorder is the action for "Table | Cell Borders | Bottom Border" command.

Unit RichViewActions 50;

Syntax

```
TrvActionTableCellBottomBorder = class(TrvActionTableCellBorder 173)
```

Hierarchy

TObject
TPersistent
TComponent

TBasicAction
TContainedAction
TCustomAction
TAction
TrvCustomAction 206
TrvAction 195
TrvActionTableCell 172
TrvActionTableMultiCellAttributes 187
TrvActionTableCellBorder 173

Description

The action makes the bottom borders of the selected table cells (or the edited cell) visible/invisible.

This action does not introduce any new properties in addition to properties of TrvAction.

See also:

- TrvActionTableCellLeftBorder 174
- TrvActionTableCellRightBorder 175
- TrvActionTableCellTopBorder 179
- TrvActionTableCellAllBorders 172
- TrvActionTableCellNoBorders 175

1.3.8.6 TrvActionTableCellLeftBorder

TrvActionTableCellLeftBorder is the action for "Table | Cell Borders | Left Border" command.

Unit RichViewActions 50;

Syntax

```
TrvActionTableCellLeftBorder = class(TrvActionTableCellBorder 173)
```

Hierarchy

TObject
TPersistent
TComponent
TBasicAction
TContainedAction
TCustomAction
TAction
TrvCustomAction 206
TrvAction 195
TrvActionTableCell 172
TrvActionTableMultiCellAttributes 187
TrvActionTableCellBorder 173

Description

The action makes the left borders of the selected table cells (or the edited cell) visible/invisible.

This action does not introduce any new properties in addition to properties of TrvAction.

See also:

- TrvActionTableCellRightBorder 175
- TrvActionTableCellTopBorder 179
- TrvActionTableCellBottomBorder 173
- TrvActionTableCellAllBorders 172
- TrvActionTableCellNoBorders 175

1.3.8.7 TrvActionTableCellNoBorders

TrvActionTableCellNoBorders is the action for "Table | Cell Borders | No Borders" command.

Unit RichViewActions 50;

Syntax

```
TrvActionTableCellNoBorders = class(TrvActionTableCellBorder 173)
```

Hierarchy

TObject
TPersistent
TComponent
TBasicAction
TContainedAction
TCustomAction
TAction
TrvCustomAction 206
TrvAction 195
TrvActionTableCell 172
TrvActionTableMultiCellAttributes 187
TrvActionTableCellBorder 173

Description

The action makes all sides of border of the selected table cells (or the edited cell) invisible.

This action does not introduce any new properties in addition to properties of TrvAction.

This is a redundant action, because it only provides a subset of functionality of TrvActionTableCellAllBorders 172.

See also:

- TrvActionTableCellLeftBorder 174
- TrvActionTableCellRightBorder 175
- TrvActionTableCellTopBorder 179
- TrvActionTableCellBottomBorder 173
- TrvActionTableCellAllBorders 172

1.3.8.8 TrvActionTableCellRightBorder

TrvActionTableCellRightBorder is the action for "Table | Cell Borders | Right Border" command.

Unit RichViewActions 50;

Syntax

```
TrvActionTableCellRightBorder = class(TrvActionTableCellBorder 173)
```

Hierarchy

TObject
TPersistent
TComponent
TBasicAction
TContainedAction
TCustomAction
TAction
TrvCustomAction 206
TrvAction 195
TrvActionTableCell 172
TrvActionTableMultiCellAttributes 187
TrvActionTableCellBorder 173

Description

The action makes the right borders of the selected table cells (or the edited cell) visible/invisible.

This action does not introduce any new properties in addition to properties of TrvAction.

See also:

- TrvActionTableCellLeftBorder 174
- TrvActionTableCellTopBorder 179
- TrvActionTableCellBottomBorder 173
- TrvActionTableCellAllBorders 172
- TrvActionTableCellNoBorders 175

1.3.8.9 TrvActionTableCellRotation

TrvActionTableCellRotation is a base class for the actions changing a rotation in table cells.

Unit RichViewActions 50;

Syntax

```
TrvActionTableCellRotation = class(TrvActionTableMultiCellAttributes 187)
```

Hierarchy

TObject
TPersistent
TComponent
TBasicAction
TContainedAction
TCustomAction
TAction
TrvCustomAction 206
TrvAction 195
TrvActionTableCell 172
TrvActionTableMultiCellAttributes 187

Description

This action does not introduce any new properties in addition to properties of TrvAction.

This action is not used directly. The following actions are inherited from it:

- TrvActionTableCellRotationNone 176
- TrvActionTableCellRotation90 177
- TrvActionTableCellRotation180 177
- TrvActionTableCellRotation270 178

1.3.8.10 TrvActionTableCellRotationNone

TrvActionTableCellRotationNone is the action for "Table | No Cell Rotation" command.

Unit RichViewActions 50;

Syntax

```
TrvActionTableCellRotationNone = class(TrvActionTableCellRotation 176)
```

Hierarchy

TObject
TPersistent
TComponent
TBasicAction
TContainedAction
TCustomAction
TAction

TrvCustomAction 206
TrvAction 195
TrvActionTableCell 172
TrvActionTableMultiCellAttributes 187
TrvActionTableCellRotation 176

Description

The action resets the rotation of the selected table cells (or the edited cell) to 0°. It changes table.Cells[].Rotation to *rvrotNone*. If the cell was rotated vertically (90° or 270°), it also exchanges table.Cells[].BestWidth and BestHeight.

This action does not introduce any new properties in addition to properties of TrvAction.

See also:

- TrvActionTableCellRotation90 177
- TrvActionTableCellRotation180 177
- TrvActionTableCellRotation270 178

1.3.8.11 TrvActionTableCellRotation90

TrvActionTableCellRotation90 is the action for "Table | Rotate Cell by 90°" command.

Unit RichViewActions 50;

Syntax

```
TrvActionTableCellRotation90 = class(TrvActionTableCellRotation 176)
```

Hierarchy

TObject
TPersistent
TComponent
TBasicAction
TContainedAction
TCustomAction
TAction
TrvCustomAction 206
TrvAction 195
TrvActionTableCell 172
TrvActionTableMultiCellAttributes 187
TrvActionTableCellRotation 176

Description

The action sets the rotation of the selected table cells (or the edited cell) to 90°. It changes table.Cells[].Rotation to *rvrot90*. If the cell was rotated horizontally (0° or 180°), it also exchanges table.Cells[].BestWidth and BestHeight.

This action does not introduce any new properties in addition to properties of TrvAction.

See also:

- TrvActionTableCellRotationNone 176
- TrvActionTableCellRotation180 177
- TrvActionTableCellRotation270 178

1.3.8.12 TrvActionTableCellRotation180

TrvActionTableCellRotation180 is the action for "Table | Rotate Cell by 180°" command.

Unit RichViewActions 50;

Syntax

```
TrvActionTableCellRotation180 = class(TrvActionTableCellRotation 176)
```

Hierarchy

TObject
TPersistent
TComponent
TBasicAction
TContainedAction
TCustomAction
TAction
TrvCustomAction 206
TrvAction 195
TrvActionTableCell 172
TrvActionTableMultiCellAttributes 187
TrvActionTableCellRotation 176

Description

The action sets the rotation of the selected table cells (or the edited cell) to 180°. It changes table.Cells[].Rotation to *rvrot180*. If the cell was rotated vertically (90° or 270°), it also exchanges table.Cells[].BestWidth and BestHeight.

This action does not introduce any new properties in addition to properties of TrvAction.

See also:

- TrvActionTableCellRotationNone 176
- TrvActionTableCellRotation90 177
- TrvActionTableCellRotation270 178

1.3.8.13 TrvActionTableCellRotation270

TrvActionTableCellRotation270 is the action for "Table | Rotate Cell by 270°" command.

Unit RichViewActions 50;

Syntax

```
TrvActionTableCellRotation270 = class(TrvActionTableCellRotation 176)
```

Hierarchy

TObject
TPersistent
TComponent
TBasicAction
TContainedAction
TCustomAction
TAction
TrvCustomAction 206
TrvAction 195
TrvActionTableCell 172
TrvActionTableMultiCellAttributes 187
TrvActionTableCellRotation 176

Description

The action sets the rotation of the selected table cells (or the edited cell) to 270°. It changes table.Cells[].Rotation to *rvrot270*. If the cell was rotated horizontally (0° or 180°), it also exchanges table.Cells[].BestWidth and BestHeight.

This action does not introduce any new properties in addition to properties of TrvAction.

See also:

- TrvActionTableCellRotationNone 176
- TrvActionTableCellRotation90 177
- TrvActionTableCellRotation180 177

1.3.8.14 TrvActionTableCellTopBorder

TrvActionTableCellTopBorder is the action for "Table | Cell Borders | Top Border" command.

Unit RichViewActions 50;

Syntax

```
TrvActionTableCellTopBorder = class(TrvActionTableCellBorder 173)
```

Hierarchy

TObject
TPersistent
TComponent
TBasicAction
TContainedAction
TCustomAction
TAction
TrvCustomAction 206
TrvAction 195
TrvActionTableCell 172
TrvActionTableMultiCellAttributes 187
TrvActionTableCellBorder 173

Description

The action makes the top borders of the selected table cells (or the edited cell) visible/invisible.

This action does not introduce any new properties in addition to properties of TrvAction.

See also:

- TrvActionTableCellLeftBorder 174
- TrvActionTableCellRightBorder 175
- TrvActionTableCellBottomBorder 173
- TrvActionTableCellAllBorders 172
- TrvActionTableCellNoBorders 175

1.3.8.15 TrvActionTableCellVAlign

TrvActionTableCellVAlign is a base class for the actions changing a vertical alignment in table cells.

Unit RichViewActions 50;

Syntax

```
TrvActionTableCellVAlign = class(TrvActionTableMultiCellAttributes 187)
```

Hierarchy

TObject
TPersistent
TComponent
TBasicAction
TContainedAction
TCustomAction
TAction
TrvCustomAction 206
TrvAction 195
TrvActionTableCell 172

TrvActionTableMultiCellAttributes 187

Description

This action does not introduce any new properties in addition to properties of TrvAction.

This action is not used directly. The following actions are inherited from it:

- TrvActionTableCellVAlignTop 182
- TrvActionTableCellVAlignMiddle 181
- TrvActionTableCellVAlignBottom 180
- TrvActionTableCellVAlignDefault 180

1.3.8.16 TrvActionTableCellVAlignBottom

TrvActionTableCellVAlignBottom is the action for "Table | Align Cell to the Bottom" command.

Unit RichViewActions 50;

Syntax

```
TrvActionTableCellVAlignBottom = class(TrvActionTableCellVAlign 179)
```

Hierarchy

TObject
TPersistent
TComponent
TBasicAction
TContainedAction
TCustomAction
TAction
TrvCustomAction 206
TrvAction 195
TrvActionTableCell 172
TrvActionTableMultiCellAttributes 187
TrvActionTableCellVAlign 179

Description

The action aligns contents of the selected table cells (or the edited cell) to bottom. It changes table.Cells[].VAlign to *rvcBottom*.

This action does not introduce any new properties in addition to properties of TrvAction.

See also:

- TrvActionTableCellVAlignTop 182
- TrvActionTableCellVAlignMiddle 181
- TrvActionTableCellVAlignDefault 180

1.3.8.17 TrvActionTableCellVAlignDefault

TrvActionTableCellVAlignDefault is the action for "Table | Default Cell Vertical Alignment" command.

Unit RichViewActions 50;

Syntax

```
TrvActionTableCellVAlignDefault = class(TrvActionTableCellVAlign 179)
```

Hierarchy

TObject
TPersistent
TComponent
TBasicAction

TContainedAction
TCustomAction
TAction
TrvCustomAction 206
TrvAction 195
TrvActionTableCell 172
TrvActionTableMultiCellAttributes 187
TrvActionTableCellVAlign 179

Description

The action sets the default alignment for the selected table cells (or the edited cell). It changes table.Cells[].VAlign to *rvcVDefault*. In this mode, cells use table.Rows[].VAlign.

This action does not introduce any new properties in addition to properties of TrvAction.

See also:

- TrvActionTableCellVAlignTop 182
- TrvActionTableCellVAlignMiddle 181
- TrvActionTableCellVAlignBottom 180

1.3.8.18 TrvActionTableCellVAlignMiddle

TrvActionTableCellVAlignMiddle is the action for "Table | Align Cell to the Middle" command.

Unit RichViewActions 50;

Syntax

```
TrvActionTableCellVAlignMiddle = class(TrvActionTableCellVAlign 179)
```

Hierarchy

TObject
TPersistent
TComponent
TBasicAction
TContainedAction
TCustomAction
TAction
TrvCustomAction 206
TrvAction 195
TrvActionTableCell 172
TrvActionTableMultiCellAttributes 187
TrvActionTableCellVAlign 179

Description

The action aligns contents of the selected table cells (or the edited cell) to the middle. It changes table.Cells[].VAlign to *rvcMiddle*.

This action does not introduce any new properties in addition to properties of TrvAction.

See also:

- TrvActionTableCellVAlignTop 182
- TrvActionTableCellVAlignBottom 180
- TrvActionTableCellVAlignDefault 180

1.3.8.19 TrvActionTableCellVAlignTop

TrvActionTableCellVAlignTop is the action for "Table | Align Cell to the Top" command.

Unit RichViewActions 50;

Syntax

```
TrvActionTableCellVAlignTop = class(TrvActionTableCellVAlign 179)
```

Hierarchy

TObject
TPersistent
TComponent
TBasicAction
TContainedAction
TCustomAction
TAction
TrvCustomAction 206
TrvAction 195
TrvActionTableCell 172
TrvActionTableMultiCellAttributes 187
TrvActionTableCellVAlign 179

Description

The action aligns contents of the selected table cells (or the edited cell) to top. It changes table.Cells[].VAlign to *rvcTop*.

This action does not introduce any new properties in addition to properties of TrvAction.

See also:

- TrvActionTableCellVAlignMiddle 181
- TrvActionTableCellVAlignBottom 180
- TrvActionTableCellVAlignDefault 180

1.3.8.20 TrvActionTableDeleteCols

TrvActionTableDeleteCols is the action for "Table | Delete Columns" command.

Unit RichViewActions 50;

Syntax

```
TrvActionTableDeleteColumns = class(TrvActionTableRCBase 189)
```

Hierarchy

TObject
TPersistent
TComponent
TBasicAction
TContainedAction
TCustomAction
TAction
TrvCustomAction 206
TrvAction 195
TrvActionTableCell 172
TrvActionTableRCBase 189

Description

This action deletes all table columns containing selected cells (or the edited cell).

This action does not introduce any new properties in addition to properties of TrvAction.

See also:

- TrvActionTableDeleteRows 183

1.3.8.21 TrvActionTableDeleteRows

TrvActionTableDeleteRows is the action for "Table | Delete Rows" command.

Unit RichViewActions 50;

Syntax

```
TrvActionTableDeleteRows = class(TrvActionTableRCBase 189)
```

Hierarchy

TObject
TPersistent
TComponent
TBasicAction
TContainedAction
TCustomAction
TAction
TrvCustomAction 206
TrvAction 195
TrvActionTableCell 172
TrvActionTableRCBase 189

Description

This action deletes all table rows containing selected cells (or the edited cell).

This action does not introduce any new properties in addition to properties of TrvAction.

See also:

- TrvActionTableDeleteCols 182

1.3.8.22 TrvActionTableDeleteTable

TrvActionTableDeleteTable is the action for "Table | Delete Table" command.

Unit RichViewActions 50;

Syntax

```
TrvActionTableDeleteTable = class(TrvActionTableCell 172)
```

Hierarchy

TObject
TPersistent
TComponent
TBasicAction
TContainedAction
TCustomAction
TAction
TrvCustomAction 206
TrvAction 195
TrvActionTableCell 172

Description

The action deletes the table at the caret position.

This action does not introduce any new properties in addition to properties of TrvAction.

1.3.8.23 TrvActionTableGrid

TrvActionTableGrid is the action for "Table | Show Grid Lines" command.

Unit RichViewActions 50;

Syntax

```
TrvActionTableGrid = class(TrvAction 195)
```

Hierarchy

TObject
TPersistent
TComponent
TBasicAction
TContainedAction
TCustomAction
TAction
TrvCustomAction 206
TrvAction 195

Description

The action shows/hides grid lines in tables. Grid lines are shown in places of invisible borders (zero border width, or sides hidden by VisibleBorders properties).

The action includes/excludes rvoShowGridLines in Options of the target editor.

This action does not introduce any new properties in addition to properties of TrvAction.

1.3.8.24 TrvActionTableInsertColLeft

TrvActionTableInsertColLeft is the action for "Table | Insert Column Left" command.

Unit RichViewActions 50;

Syntax

```
TrvActionTableInsertColLeft = class(TrvActionTableRCBase 189)
```

Hierarchy

TObject
TPersistent
TComponent
TBasicAction
TContainedAction
TCustomAction
TAction
TrvCustomAction 206
TrvAction 195
TrvActionTableCell 172
TrvActionTableRCBase 189

Description

This action inserts one new column to the left of the selected cells (or of the edited cell).

This action does not introduce any new properties in addition to properties of TrvAction.

See also:

- TrvActionTableInsertRowsAbove 185
- TrvActionTableInsertRowsBelow 186
- TrvActionTableInsertColRight 185

1.3.8.25 TrvActionTableInsertColRight

TrvActionTableInsertColRight is the action for "Table | Insert Column Right" command.

Unit RichViewActions 50;

Syntax

```
TrvActionTableInsertColRight = class(TrvActionTableRCBase 189)
```

Hierarchy

TObject
TPersistent
TComponent
TBasicAction
TContainedAction
TCustomAction
TAction
TrvCustomAction 206
TrvAction 195
TrvActionTableCell 172
TrvActionTableRCBase 189

Description

This action inserts one new column to the right of the selected cells (or of the edited cell).

This action does not introduce any new properties in addition to properties of TrvAction.

See also:

- TrvActionTableInsertRowsAbove 185
- TrvActionTableInsertRowsBelow 186
- TrvActionTableInsertColLeft 184

1.3.8.26 TrvActionTableInsertRowsAbove

TrvActionTableInsertRowsAbove is the action for "Table | Insert Row Above" command.

Unit RichViewActions 50;

Syntax

```
TrvActionTableInsertRowsAbove = class(TrvActionTableRCBase 189)
```

Hierarchy

TObject
TPersistent
TComponent
TBasicAction
TContainedAction
TCustomAction
TAction
TrvCustomAction 206
TrvAction 195
TrvActionTableCell 172
TrvActionTableRCBase 189

Description

This action inserts one new row above the selected cells (or above the edited cell).

This action does not introduce any new properties in addition to properties of TrvAction.

See also:

- TrvActionTableInsertRowsBelow 186
- TrvActionTableInsertColLeft 184
- TrvActionTableInsertColRight 185

1.3.8.27 TrvActionTableInsertRowsBelow

TrvActionTableInsertRowsBelow is the action for "Table | Insert Row Below" command.

Unit RichViewActions 50;

Syntax

```
TrvActionTableInsertRowsBelow = class(TrvActionTableRCBase 189)
```

Hierarchy

TObject
TPersistent
TComponent
TBasicAction
TContainedAction
TCustomAction
TAction
TrvCustomAction 206
TrvAction 195
TrvActionTableCell 172
TrvActionTableRCBase 189

Description

This action inserts new rows below the selected cells (or below the edited cell).

When inserting rows to the end of the table, if AllowMultiple 186 = *True*, the action requests a number of rows to insert and inserts this count of rows.

See also:

- TrvActionTableInsertRowsAbove 185
- TrvActionTableInsertColLeft 184
- TrvActionTableInsertColRight 185

1.3.8.27.1 Properties

1.3.8.27.1.1 TrvActionTableInsertRow sBelow .Allow Multiple

Allowing inserting several rows at the end of table at once.

```
property AllowMultiple: Boolean;
```

If *True*, when inserting rows to the end of the table, the number of new rows is requested from the user.

Default value:

True

1.3.8.28 TrvActionTableMergeCells

TrvActionTableMergeCells is the action for "Table | Merge Cells" command.

Unit RichViewActions 50;

Syntax

```
TrvActionTableMergeCells = class(TrvActionTableCell 172)
```

Hierarchy

TObject

TPersistent
TComponent
TBasicAction
TContainedAction
TCustomAction
TAction
TrvCustomAction 206
TrvAction 195
TrvActionTableCell 172

Description

The action merges the selected table cells in one.

This action does not introduce any new properties in addition to properties of TrvAction.

1.3.8.29 TrvActionTableMultiCellAttributes

TrvActionTableMultiCellAttributes is the base class for actions changing properties of table cells.

Unit RichViewActions 50;

Syntax

```
TrvActionTableMultiCellAttributes = class(TrvActionTableCell 172)
```

Hierarchy

TObject
TPersistent
TComponent
TBasicAction
TContainedAction
TCustomAction
TAction
TrvCustomAction 206
TrvAction 195
TrvActionTableCell 172

Description

This action does not introduce any new properties in addition to properties of TrvAction.

This action is not used directly. This action has the following direct descendants:

- TrvActionTableCellVAlign 179
- TrvActionTableCellBorder 173

1.3.8.30 TrvActionTableProperties

TrvActionTableProperties is the action for "Table | Table Properties" command.

Unit RichViewActions 50;

Syntax

```
TrvActionTableProperties = class(TrvActionTableCell 172)
```

Hierarchy

TObject
TPersistent
TComponent
TBasicAction
TContainedAction
TCustomAction

TAction
TrvCustomAction 206
TrvAction 195
TrvActionTableCell 172

Description

The action displays a dialog for editing table properties. If the table has no selection (i.e. the caret is placed to the right or to the left of the table), this dialog has one page ("Table"). If there is a multicell selection or an edited cell, the dialog has two additional pages ("Rows" and "Cells").

The same dialog is shown by TrvActionItemProperties 201 action. But TrvActionItemProperties 201 edits properties of an item at the position of caret. For example, if a picture inside a table is selected, TrvActionItemProperties 201 edits this picture, while TrvActionTableProperties edits this table.

The user can check "Default" check box in this dialog to assign properties of new tables.

1.3.8.30.1 Properties

1.3.8.30.1.1 TrvActionTableProperties.ActionInsertTable

Specifies the "insert table" action for applying default properties to.

```
property ActionInsertTable: TrvActionInsertTable 163;
```

If this link is not defined, the first found TrvActionInsertTable 163 action on the same form/datamodule is used.

If "Default" checkbox is checked, when properties are applied, they are assigned to properties of this action as well, so they become default properties for new tables

See also

- TRVActionItemProperties 201.ActionInsertTable 202

1.3.8.30.1.2 TrvActionTableProperties.BackgroundGraphicFilter

Determines the file masks (filters) available in the file opening dialog displayed to open a picture to assign to a background picture of table or selected cells.

```
property BackgroundGraphicFilter: String;
```

If this property is not empty, it is assigned to the Filter property of the file opening dialog (TOpenPictureDialog). Otherwise, a filter containing all registered graphic formats is created.

Default value:

" (empty string)

See also:

- TrvActionInsertPicture.Filter 151
- TrvActionItemProperties.GraphicFilter 202
- TrvActionItemProperties.BackgroundGraphicFilter 202

1.3.8.30.1.3 TrvActionTableProperties.StoreImageFileName

Specifies whether file names (full paths) are stored for background image of table and cells assigned by this action.

```
property StoreImageFileName: Boolean;
```

If this property is *True*, path to the graphic file is stored in the document:

- in the *rvespImageFileName* item property (see the RichView help file, TRVExtraItemStrProperty type) for tables;
- in BackgroundImageFileName cell property for table cells.

This is a recommended way to store path to graphic files. You can use these file names when saving HTML files (Include *rvsoUseItemImageFileNames* in TrvActionSave.SaveOptions 71 and TrvActionExport.SaveOptions 64).

Default value

False

See also:

- TrvActionPasteSpecial.StoreFileName 79
- TrvActionInsertPicture.StoreFileName 152
- TrvActionItemProperties.StoreImageFileName 203
- TCustomRichView.RTFReadProperties.StoreImagesFileNames
- HTML Import (for images inserted with HTML, paths to graphic files are always stored in *rvespImageFileName*)

1.3.8.31 TrvActionTableRCBase

TrvActionTableRCBase is a base class for the actions for insertion and deletion of table rows and columns.

Unit RichViewActions 50;

Syntax

```
TrvActionTableRCBase = class(TrvActionTableCell 172)
```

Hierarchy

TObject
TPersistent
TComponent
TBasicAction
TContainedAction
TCustomAction
TAction
TrvCustomAction 206
TrvAction 195
TrvActionTableCell 172

Description

This action does not introduce any new properties in addition to properties of TrvAction.

This action is not used directly. The following actions are inherited from it:

- TrvActionTableInsertRowsAbove 185
- TrvActionTableInsertRowsBelow 186
- TrvActionTableInsertColLeft 184
- TrvActionTableInsertColRight 185
- TrvActionTableDeleteRows 183
- TrvActionTableDeleteCols 182
- TrvActionTableSelectCols 190
- TrvActionTableSelectRows 190

1.3.8.32 TrvActionTableSelectCell

TrvActionTableSelectCell is the action for "Table | Select Cell" command.

Unit RichViewActions 50;

Syntax

```
TrvActionTableSelectCell = class(TrvActionTableCell 172)
```

Hierarchy

TObject
TPersistent
TComponent
TBasicAction

TContainedAction
TCustomAction
TAction
TrvCustomAction 206
TrvAction 195
TrvActionTableCell 172

Description

The action selects the edited table cell.

This action does not introduce any new properties in addition to properties of TrvAction.

See also:

- TrvActionTableSelectCols 190
- TrvActionTableSelectRows 190
- TrvActionTableSelectTable 191

1.3.8.33 TrvActionTableSelectCols

TrvActionTableSelectCols is the action for "Table | Select Columns" command.

Unit RichViewActions 50;

Syntax

```
TrvActionTableSelectCols = class(TrvActionTableRCBase 189)
```

Hierarchy

TObject
TPersistent
TComponent
TBasicAction
TContainedAction
TCustomAction
TAction
TrvCustomAction 206
TrvAction 195
TrvActionTableCell 172
TrvActionTableRCBase 189

Description

The action selects table columns containing selected cells (or the edited cell).

This action does not introduce any new properties in addition to properties of TrvAction.

See also:

- TrvActionTableSelectRows 190
- TrvActionTableSelectCell 189
- TrvActionTableSelectTable 191

1.3.8.34 TrvActionTableSelectRows

TrvActionTableSelectRows is the action for "Table | Select Rows" command.

Unit RichViewActions 50;

Syntax

```
TrvActionTableSelectRows = class(TrvActionTableRCBase 189)
```

Hierarchy

TObject
TPersistent
TComponent
TBasicAction
TContainedAction
TCustomAction
TAction
TrvCustomAction 206
TrvAction 195
TrvActionTableCell 172
TrvActionTableRCBase 189

Description

The action selects table rows containing selected cells (or the edited cell).

This action does not introduce any new properties in addition to properties of TrvAction.

See also:

- TrvActionTableSelectCols 190
- TrvActionTableSelectCell 189
- TrvActionTableSelectTable 191

1.3.8.35 TrvActionTableSelectTable

TrvActionTableSelectTable is the action for "Table | Select Table" command.

Unit RichViewActions 50;

Syntax

```
TrvActionTableSelectTable = class(TrvActionTableCell 172)
```

Hierarchy

TObject
TPersistent
TComponent
TBasicAction
TContainedAction
TCustomAction
TAction
TrvCustomAction 206
TrvAction 195
TrvActionTableCell 172

Description

The action selects all cells in the table at the caret position.

Note: a selection containing a table, and a selection containing all selected table cells are different selections, although they look identically.

When all cells are selected (for example, after execution of this action), the user can perform table operations (for example, inserting rows or merging cells). When the user presses Delete key, all cells are cleared. TCustomRichViewEdit can copy such selection to the Clipboard in RVF and text format, but cannot copy in RTF format yet.

When the table is selected, the user cannot perform table operations When the user presses Delete key, the table is deleted. TCustomRichViewEdit can copy such selection to the Clipboard in all supported formats.

See also:

- TrvActionTableSelectCols 190
- TrvActionTableSelectRows 190

- TrvActionTableSelectCell 189

1.3.8.36 TrvActionTableSort

TrvActionTableSort is the action for "Table | Sort" command.

Unit RichViewActions 50;

Syntax

```
TrvActionTableSort = class(TrvActionTableCell 172)
```

Hierarchy

TObject
TPersistent
TComponent
TBasicAction
TContainedAction
TCustomAction
TAction
TrvCustomAction 206
TrvAction 195
TrvActionTableCell 172

Description

The action sorts the table rows. If the action is called when more than one row are selected, it sorts the selected rows; otherwise, it sorts all rows.

Table heading rows are never sorted. Additionally, you can exclude the top [selected] row from sorting.

The action displays a dialog window allowing to specify parameters for sorting:

- column to sort by;
- ascending/descending order;
- sort as numbers or as text; if as text, case sensitivity

This action does not introduce any new properties in addition to properties of TrvAction.

1.3.8.37 TrvActionTableSplit

TrvActionTableSplit is the action for "Table | Split Table" command.

Unit RichViewActions 50;

Syntax

```
TrvActionTableSplit = class(TrvActionTableCell 172)
```

Hierarchy

TObject
TPersistent
TComponent
TBasicAction
TContainedAction
TCustomAction
TAction
TrvCustomAction 206
TrvAction 195
TrvActionTableCell 172

Description

This action splits the current table into two tables. The second table starts from the first selected row.

This action does not introduce any new properties in addition to properties of TrvAction.

1.3.8.38 TrvActionTableSplitCells

TrvActionTableSplitCells is the action for "Table | Split Cells" command.

Unit RichViewActions [50];

Syntax

```
TrvActionTableSplitCells = class(TrvActionTableCell [172])
```

Hierarchy

TObject
TPersistent
TComponent
TBasicAction
TContainedAction
TCustomAction
TAction
TrvCustomAction [206]
TrvAction [195]
TrvActionTableCell [172]

Description

The action displays a dialog where the user can choose:

- to unmerge the selected cells (if the selection contains cells having values of ColSpan or RowSpan properties greater than 1);
- to split the selected cells into the specified number or rows and columns (optionally with preliminary merging).

This action does not introduce any new properties in addition to properties of TrvAction.

1.3.8.39 TrvActionTableToText

TrvActionTableToText is the action for "Table | Convert to Text" command.

Unit RichViewActions [50];

Syntax

```
TrvActionTableToText = class(TrvActionTableCell [172])
```

Hierarchy

TObject
TPersistent
TComponent
TBasicAction
TContainedAction
TCustomAction
TAction
TrvCustomAction [206]
TrvAction [195]
TrvActionTableCell [172]

Description

This action converts the current table to text (in other words, it removes the table without removing its content).

The action allows choosing a delimiter to insert between text from cells: line break, tab, semicolon, or comma. Delimiters are inserted using the current text style.

This action does not introduce any new properties in addition to properties of TrvAction.

1.3.9 Addict 3&4 Support

1.3.9.1 TrvActionAddictSpell3

TrvActionAddictSpell3 is the action for "Spell Check" command, using the Addictive Software's spellchecker[219].

Unit RichViewActions[50];

Syntax

```
TrvActionAddictSpell3 = class(TrvAction[195])
```

Hierarchy

TObject
TPersistent
TComponent
TBasicAction
TContainedAction
TCustomAction
TAction
TrvCustomAction[206]
TrvAction[195]

Description

This action is available only if Addictive Software's spellchecker[219] is used.

This action is enabled only if GetControlPanel[208].RVAddictSpell3[30] is assigned.

The action starts a spelling checking process; when a misspelled word is found, a dialog is displayed to suggests a replacement.

This action does not introduce any new properties in addition to properties of TrvAction.

1.3.9.2 TrvActionAddictThesaurus3

TrvActionAddictThesaurus3 is the action for "Thesaurus" command, using the Addictive Software's spellchecker[219].

Unit RichViewActions[50];

Syntax

```
TrvActionAddictThesaurus3 = class(TrvAction[195])
```

Hierarchy

TObject
TPersistent
TComponent
TBasicAction
TContainedAction
TCustomAction
TAction
TrvCustomAction[206]
TrvAction[195]

Description

This action is available only if Addictive Software's spellchecker[219] is used.

This action is enabled only if GetControlPanel[208].RVAddictThesaurus3[30] is assigned.

The action offers synonyms for the current word.

This action does not introduce any new properties in addition to properties of TrvAction.

1.3.10 Others

1.3.10.1 TrvAction

TrvAction is a base class for all RichViewActions working with TCustomRichViewEdit.

Unit RichViewActions[50];

Syntax

```
TrvAction = class(TrvCustomAction[206])
```

Hierarchy

TObject
TPersistent
TComponent
TBasicAction
TContainedAction
TCustomAction
TAction
TrvCustomAction[206]

Description

This action is not used directly, but all RichViewActions working with TCustomRichViewEdit are inherited from it.

1.3.10.1.1 Properties

1.3.10.1.1.1 TrvAction.Control

Defines the editor for this action.

```
property Control: TCustomRVControl;
```

If an editor is assigned to this property, the action works with this component.

Controls of the following types can be assigned to this property:

- TRichViewEdit
- TDBRichViewEdit
- TSRichViewEdit (ScaleRichView)
- TDBSRichViewEdit (ScaleRichView)

Otherwise, if GetControlPanel[208].DefaultControl[26] is assigned, the action works with it.

Otherwise, if an editor component is focused, the action works with it.

Otherwise, it works with the first found editor component.

Special actions

The following actions automatically assign this property, if it was not assigned before the first execution:

- TrvActionStyleInspector[132];
- TrvActionEditNote[157].

They need this property assigned, because they display information for the current item in the target editor, even when the caret is moved after the execution of the action.

1.3.10.2 TrvActionBackground

TrvActionBackground is the action for "Background" command.

Unit RichViewActions[50];

Syntax

```
TrvActionBackground = class(TrvAction[195])
```

Hierarchy

TObject
TPersistent
TComponent
TBasicAction
TContainedAction
TCustomAction
TAction
TrvCustomAction[206]
TrvAction[195]

Description

This action changes background (Color, BackgroundBitmap, BackgroundStyle) and "padding" (LeftMargin, TopMargin, RightMargin, BottomMargin) properties of the target editor. "Padding" can be changed only if CanChangeMargins[196]=*True*.

if *rvfoSaveBack* and/or *rvfoSaveLayout* are included in the target editor's RVFOptions property, the action changes background and/or margin properties as an editing operation (and these changes can be undone by the user).

The action calls the following events:

- TCustomRichViewEdit.OnChange (called by TCustomRichViewEdit.Change, see above)
- GetControlPanel[208].OnMarginsChanged[39] (only if the action changed values of LeftMargin, TopMargin, RightMargin, BottomMargin properties of TCustomRichViewEdit)
- GetControlPanel[208].OnBackgroundChange[36]

1.3.10.2.1 Properties

1.3.10.2.1.1 TrvActionBackground.CanChangeMargins

Specifies whether the action can change margin properties of the target TRichViewEdit control

```
property CanChangeMargins: Boolean;
```

If *True*, the "Padding" button is added to the action's dialog, allowing to define values for LeftMargin, RightMargin, TopMargin, BottomMargin properties of the target TRichViewEdit control.

It's recommended to hide this button when working with TSRichViewEdit controls from ScaleRichView.

Default value

True

1.3.10.2.1.2 TrvActionBackground.ImageFileName

Returns the file name (full path) of image file assigned to background bitmap.

```
property ImageFileName: String;
```

You can use this property to store this path somewhere.

The following events can be used for this:

- OnChange[197]
- GetControlPanel[208].OnBackgroundChange[36]

1.3.10.2.2 Events

1.3.10.2.2.1 TrvActionBackground.OnChange

Occurs after the background is changed by this action.

```
property OnChange: TRVAEditEvent[232];
```

You can use either this event or GetControlPanel[208].OnBackgroundChange[36].

1.3.10.3 TrvActionColor

TrvActionColor is the action for "Background Color" command.

Unit RichViewActions[50];

Syntax

```
TrvActionCustomColor = class(TrvActionCustomColor[197])
```

Hierarchy

TObject
TPersistent
TComponent
TBasicAction
TContainedAction
TCustomAction
TAction
TrvCustomAction[206]
TrvAction[195]
TrvActionCustomColor[197]

Description

The action changes the target editor's Color property.

if *rvfoSaveBack* is included in the target editor's RVFOptions property, this change is made as an editing operation (and can be undone by the user)

The action calls GetControlPanel[208].OnBackgroundChange[36] event.

This action does not introduce any new properties and events in addition to properties and events of TrvActionCustomColor[197].

1.3.10.4 TrvActionCustomColor

TrvActionCustomColor is a base class of the actions for color changing.

Unit RichViewActions[50];

Syntax

```
TrvActionCustomColor = class(TrvAction[195])
```

Hierarchy

TObject
TPersistent
TComponent
TBasicAction
TContainedAction
TCustomAction
TAction
TrvCustomAction[206]
TrvAction[195]

Description

This action is not used directly. The following actions are inherited from it:

- TrvActionFontCustomColor 87
- TrvActionFontColor 86
- TrvActionFontBackColor 85
- TrvActionParaCustomColor 119
- TrvActionParaColor 118
- TrvActionColor 197

1.3.10.4.1 Properties

1.3.10.4.1.1 TrvActionCustomColor.CallerControl

Specifies the control (for example a toolbar button) which executed this action.

```
property CallerControl: TControl;
```

Assign this property if you use Delphi 4 or 5. For the newer versions of Delphi, the caller control is determined automatically (using ActionComponent property).

A color-picker window is positioned relative to this control. If this control is undefined, a color-picker window is displayed at the mouse pointer position.

1.3.10.4.1.2 TrvActionCustomColor.Color

Specifies the color to apply.

```
property Color: TColor;
```

See the topic about UserInterface 198 property for details.

Default value for TrvActionCustomColor:

clNone

Default value for TrvActionFontColor 86:

clWindowText

1.3.10.4.1.3 TrvActionCustomColor.UserInterface

Defines the way how the user chooses a color.

```
type
  TrvaColorInterface = (rvacNone, rvacColorDialog, rvacAdvanced);

property UserInterface: TrvaColorInterface;
```

Value	Meaning
rvacNone	Color is not chosen by the user. When the action is executed, value specified in Color 198 property is applied.
rvacColorDialog	Color is chosen using TColorDialog. Value of Color 198 property before the execution is ignored. When the color is chosen, it is assigned to Color 198 property.
rvacAdvanced	Color is chosen using a special popup color-picker window. Value of Color 198 property before the execution is ignored. When the color is chosen, it is assigned to Color 198 property.

This action uses GetControlPanel 208.ColorDialog 25, if assigned. Otherwise it creates a temporal color dialog.

In *rvacAdvanced* mode, GetControlPanel 208.OnGetActionControlCoords 38 may be used to position a color-

picker window.

Default value:

rvacAdvanced

1.3.10.4.2 Events

1.3.10.4.2.1 TrvActionCustomColor.OnShow ColorPicker

Occurs before the color-picker window is displayed.

```
property OnShowColorPicker: TNotifyEvent;
```

This event occurs if UserInterface[198] = *rvacAdvanced*.

1.3.10.4.2.2 TrvActionCustomColor.OnHideColorPicker

Occurs when the color-picker window is destroyed.

```
property OnHideColorPicker: TNotifyEvent;
```

This event occurs if UserInterface[198] = *rvacAdvanced*.

1.3.10.5 TrvActionEvent

TrvActionEvent is the action to perform custom operations.

Unit RichViewActions[50];

Syntax

```
TrvActionEvent = class(TrvAction[195])
```

Hierarchy

TObject
TPersistent
TComponent
TBasicAction
TContainedAction
TCustomAction
TAction
TrvCustomAction[206]
TrvAction[195]

Description

This action calls OnExecute[199] and OnUpdate[200] events allowing to implement custom commands.

This action does not have predefined Caption and Hint. Assign these properties yourself.

This action does not introduce any new properties in addition to properties of TrvAction.

See also:

- TrvActionInsertText[155]

1.3.10.5.1 Events

1.3.10.5.1.1 TrvActionEvent.OnExecute

Occurs when the action is executed.

```
type
  TRVAEvent = procedure(Sender: TObject;
    Editor: TCustomRichViewEdit) of object;
```

```
property OnExecute: TRVAEvent;
```

1.3.10.5.1.2 TrvActionEvent.OnUpdate

Occurs when the application is idle or when the action list updates.

```
type
  TRVAEvent = procedure(Sender: TObject;
    Editor: TCustomRichViewEdit) of object;

property OnUpdate: TRVAEvent;
```

Update values of Enabled and Checked properties (and may be values of other action's properties) in this event.

This event is not called if Disabled[207]=*True*, or GetControlPanel[208].ActionsEnabled[25]=*False*.

1.3.10.6 TrvActionFillColor

TrvActionFillColor is the action for "Fill Color" command.

Unit RichViewActions[50];

Syntax

```
TrvActionFillColor = class(TrvAction[195])
```

Hierarchy

TObject
TPersistent
TComponent
TBasicAction
TContainedAction
TCustomAction
TAction
TrvCustomAction[206]
TrvAction[195]

Description

This action can change a background color of:

- text (RVStyle.TextStyles[].BackColor),
- paragraph (RVStyle.ParaStyles[].Background.Color),
- table cell (table.Cells[].Color)
- table (table.Color).

For paragraphs, it can also change padding (RVStyle.ParaStyles[].Background.BorderOffsets).

The action displays a dialog where the user can define a color and an object for application.

This is a redundant action, because these properties can be changed by other actions. But it may be convenient to have an action allowing to change all these properties.

1.3.10.6.1 Properties

1.3.10.6.1.1 TrvActionFillColor.AllowApplyingTo

Contains objects that can be changed by this action.

```
type
  TRVAFillColorApplyToElement = (rvafcaText, rvafcaParagraph, rvafcaCell, rvafcaTable);
  TRVAFillColorApplyToSet = set of TRVAFillColorApplyToElement;

property AllowApplyingTo: TRVAFillColorApplyToSet;
```

The action can change a background color only for objects listed in this property.

Default value:

[*rvafcaText..rvafcaTable*]

1.3.10.7 TrvActionHide

TrvActionHide switches visibility of the selected fragment.

Unit RichViewActions 50;

Syntax

```
TrvActionHide = class(TrvAction 195)
```

Hierarchy

TObject
TPersistent
TComponent
TBasicAction
TContainedAction
TCustomAction
TAction
TrvCustomAction 206
TrvAction 195

Description

This action makes the selected text and objects hidden (or shows them if they are already hidden).

This action does not introduce any new properties in addition to properties of TrvAction.

See also

- TrvActionShowSpecialCharacters 205

1.3.10.8 TrvActionItemProperties

TrvActionItemProperties is the action for "Object Properties" command.

Unit RichViewActions 50;

Syntax

```
TrvActionItemProperties = class(TrvAction 195)
```

Hierarchy

TObject
TPersistent
TComponent
TBasicAction
TContainedAction
TCustomAction
TAction
TrvCustomAction 206
TrvAction 195

Description

This action displays a dialog for changing properties of items of the following types:

- pictures and "hot-pictures"
- "breaks" (horizontal lines)
- tables
- sidenotes and text box items
- numbered sequences
- page numbers

- other (using OnCustomItemPropertiesDialog 203 event)

The user can check "Default" check box in this dialog to assign properties of new tables and horizontal lines.

See also:

- TrvActionTableProperties 187

1.3.10.8.1 Properties

1.3.10.8.1.1 TrvActionItemProperties.ActionInsertHLine

Specifies the "insert horizontal line" action for applying default properties to.

```
property ActionInsertHLine: TrvActionInsertHLine 137;
```

If this link is not defined, the first found TrvActionInsertHLine 137 action on the same form/datamodule is used.

If "Default" checkbox is checked, when properties are applied, they are assigned to properties of this action as well, so they become default properties for new horizontal lines.

See also:

- ActionInsertTable 202

1.3.10.8.1.2 TrvActionItemProperties.ActionInsertTable

Specifies the "insert table" action for applying default properties to.

```
property ActionInsertTable: TrvActionInsertTable 163;
```

If this link is not defined, the first found TrvActionInsertTable 163 action on the same form/datamodule is used.

If "Default" checkbox is checked, when properties are applied, they are assigned to properties of this action as well, so they become default properties for new tables

See also:

- ActionInsertHLine 202
- TrvActionTableProperties 187.ActionInsertTable 188

1.3.10.8.1.3 TrvActionItemProperties.BackgroundGraphicFilter

Determines the file masks (filters) available in the file opening dialog displayed to open a picture to assign to a background picture of table or selected cells.

```
property BackgroundGraphicFilter: String;
```

If this property is not empty, it is assigned to the Filter property of the file opening dialog (TOpenPictureDialog). Otherwise, a filter containing all registered graphic formats is created.

Default value:

" (empty string)

See also:

- TrvActionInsertPicture.Filter 151
- GraphicFilter 202
- TrvActionTableProperties.BackgroundGraphicFilter 188

1.3.10.8.1.4 TrvActionItemProperties.GraphicFilter

Determines the file masks (filters) available in the file opening dialog for pictures and "hot-pictures".

```
property GraphicFilter: String;
```

If this property is not empty, it is assigned to the Filter property of the file opening dialog (TOpenPictureDialog). Otherwise, a filter containing all registered graphic formats is created.

Default value:

" (empty string)

See also:

- TrvActionInsertPicture.Filter 151
- TrvActionItemProperties.BackgroundGraphicFilter 202
- TrvActionTableProperties.BackgroundGraphicFilter 188

1.3.10.8.1.5 TrvActionItemProperties.StoreImageFileName

Specifies whether file names (full paths) are stored for images assigned by this action.

```
property StoreImageFileName: Boolean;
```

If this property is *True*, path to the graphic file is stored in the document:

- in the *rvespImageFileName* item property (see the RichView help file, TRVExtraItemStrProperty type) for tables, pictures and "hot-pictures";
- in BackgroundImageFileName cell property for table cells.

This is a recommended way to store path to graphic files. You can use these file names when saving HTML files (Include *rvsoUseItemImageFileNames* in TrvActionSave.SaveOptions 71 and TrvActionExport.SaveOptions 64).

Default value

False

See also:

- TrvActionPasteSpecial.StoreFileName 79
- TrvActionInsertPicture.StoreFileName 152
- TrvActionTableProperties.StoreImageFileName 188
- TCustomRichView.RTFReadProperties.StoreImagesFileNames
- HTML Import (for images inserted with HTML, paths to graphic files are always stored in *rvespImageFileName*)

1.3.10.8.2 Events

1.3.10.8.2.1 TrvActionItemProperties.OnCanApply

Requests whether a custom dialog can be displayed for the given item.

```
type
  TRVCanApplyEvent = procedure (Sender: TObject; Editor: TCustomRichViewEdit;
    Item: TCustomRVItemInfo; var CanApply: Boolean) of object;

property OnCanApply: TRVCanApplyEvent;
```

This event occurs only if OnCustomItemPropertiesDialog 203 event is assigned.

A dialog will be displayed for **Item** in the **Editor** (for the selected item, if only one item is selected), if **CanApply** is set to *True*.

1.3.10.8.2.2 TrvActionItemProperties.OnCustomItemPropertiesDialog

Allows displaying a dialog for editing properties of different item types, or replacing dialogs for items of the standard types, or forbidding displaying a standard dialog for items of standard types.

```
type
  TRVCustomItemPropertiesDialog = procedure (Sender: TObject;
    Editor: TCustomRichViewEdit; var DoDefault: Boolean) of object;

property OnCustomItemPropertiesDialog: TRVCustomItemPropertiesDialog;
```

A dialog must be displayed for the current item in the **Editor** (or for the selected item, if only one item is selected).

Assign *False* to **DoDefault** if you do not want to display a standard dialog after calling this event.

See also:

- OnCanApply 203

1.3.10.9 TrvActionRemoveHyperlinks

TrvActionRemoveHyperlinks is the action for "Remove Hyperlinks" command.

Unit RichViewActions 50;

Syntax

```
TrvActionRemoveHyperlinks = class(TrvAction 195)
```

Hierarchy

TObject
TPersistent
TComponent
TBasicAction
TContainedAction
TCustomAction
TAction
TrvCustomAction 206
TrvAction 195

Description

This action converts all hyperlinks in the selected fragment to a plain text. This action does the same work as TrvActionInsertHyperlink 138, if the user entered an empty link target.

This action is a wrapper for TrvActionInsertHyperlink 138 action referenced in ActionInsertHyperlink property.

1.3.10.9.1 Properties

1.3.10.9.1.1 TrvActionRemoveHyperlinks.ActionInsertHyperlink

Links this action to TrvActionInsertHyperlink 138 action.

```
property ActionInsertHyperlink: TrvActionInsertHyperlink 138;
```

If this link is not defined, the first found TrvActionInsertHyperlink 138 action on the same form/datamodule is used. If no TrvActionInsertHyperlink 138 action found, this action is disabled.

1.3.10.10 TrvActionRemovePageBreak

TrvActionRemovePageBreak is the action for "Remove Page Break" command.

Unit RichViewActions 50;

Syntax

```
TrvActionRemovePageBreak = class(TrvAction 195)
```

Hierarchy

TObject
TPersistent
TComponent
TBasicAction
TContainedAction
TCustomAction
TAction
TrvCustomAction 206
TrvAction 195

Description

This action removes a page break from the paragraph at the caret position (or, when called from a table cell, from the current table row).

This action does not introduce any new properties in addition to properties of TrvAction.

See also:

- TrvActionInsertPageBreak 149

1.3.10.11 TrvActionShowSpecialCharacters

TrvActionShowSpecialCharacters is the action for "Non-printing Characters" command.

Unit RichViewActions 50;

Syntax

```
TrvActionShowSpecialCharacters = class(TrvAction 195)
```

Hierarchy

TObject
TPersistent
TComponent
TBasicAction
TContainedAction
TCustomAction
TAction
TrvCustomAction 206
TrvAction 195

Description

The action shows/hides:

- marks for non-printing characters (spaces, non breaking spaces, tabs, soft hyphens, paragraphs separators, line separators) and placeholders of floating items;
- hidden items.

The set of non-printing characters to display is defined in RVVisibleSpecialCharacters global variable (declared in RVStyle.pas).

The action includes/excludes *rvoShowSpecialCharacters* and *rvoShowHiddenText* from TCustomRichViewEdit.Options property and reformats TCustomRichViewEdit.

This action assumes that these options are included and excluded from Options together.

This action does not introduce any new properties in addition to properties of TrvAction.

See also

- TrvActionHide 201

1.3.10.12 TrvActionVAlign

TrvActionVAlign is the action for "Object Position" command.

Unit RichViewActions 50;

Syntax

```
TrvActionVAlign = class(TrvAction 195)
```

Hierarchy

TObject
TPersistent

TComponent
TBasicAction
TContainedAction
TCustomAction
TAction
TrvCustomAction 206
TrvAction 195

Description

This action displays a dialog for changing VAlign property for the items of the following types:

- pictures
- *hot-pictures*
- *bullets* (image lists)
- *hotspots* (image lists)
- controls
- label items
- numbered sequences
- custom item types inherited from TRVRectItemInfo

This action does not introduce any new properties in addition to properties of TrvAction.

1.3.10.13 TrvCustomAction

TrvCustomAction is a base class for all RichViewActions.

Unit RichViewActions 50;

Syntax (normal)

```
TrvCustomAction = class(TAction)
```

Syntax (if TNT Controls 224 are used)

```
TrvCustomAction = class(TAction, ITntAction)
```

Hierarchy

TObject
TPersistent
TComponent
TBasicAction
TContainedAction
TCustomAction
TAction

Description

This action is not used directly, but all RichViewActions are inherited from it.

Main properties:

- ControlPanel 207 links this action to a control panel defining additional properties for the action.
- Disabled 207 allows to disable this action.

Direct descendants:

- TrvAction 195
- TrvActionPageSetup 64

1.3.10.13.1 Properties

1.3.10.13.1.1 TrvCustomAction.Caption

Represents the caption of client controls and menu items.

If TNT Controls 224 are used, this property overrides the standard action's Caption property.

```
property Caption: WideString;
```

This property is defined only if TNT Controls 224 are used. It overrides the standard Caption property.

Caption holds the string that is the caption for the action when it is set. The value of Caption is propagated to client controls and menu items.

See also properties:

- Hint 207

1.3.10.13.1.2 TrvCustomAction.ControlPanel

Links this action with a control panel.

```
property ControlPanel: TRVAControlPanel 23;
```

A control panel defines properties used for many actions.

If this property is not assigned (*nil*), the action uses MainRVAControlPanel 235.

See also:

- GetControlPanel 208.

1.3.10.13.1.3 TrvCustomAction.Disabled

Allows to disable the action.

```
property Disabled: Boolean;
```

If *False*, the action is disabled and enabled automatically.

If *True*, the action is disabled.

Default value:

False

See also properties:

- Enabled

See also:

- GetControlPanel 208.ActionsEnabled 25

1.3.10.13.1.4 TrvCustomAction.Hint

Indicates the Help hint for client controls and menu items.

If TNT Controls 224 are used, this property overrides the standard action's Hint property.

```
property Hint: WideString;
```

This property is defined only if TNT Controls 224 are used. It overrides the standard Hint property.

Hint holds the string that is the hint for that action when it is set. This value is propagated to client controls and menu items.

See also properties:

- Caption 207

1.3.10.13.2 Methods

1.3.10.13.2.1 TrvCustomAction.GetControlPanel

Returns a control panel used by this action.

```
function GetControlPanel: TRVAControlPanel;
```

If ControlPanel is assigned, the method returns it. Otherwise it returns MainRVAControlPanel.

1.3.10.14 TrvCustomEditorAction

TrvCustomEditorAction is a base class for action displaying a non-modal window containing TRichViewEdit component.

Unit RichViewActions;

Syntax

```
TrvCustomEditorAction = class(TrvAction)
```

Hierarchy

TObject
TPersistent
TComponent
TBasicAction
TContainedAction
TCustomAction
TAction
TrvCustomAction
TrvAction

Description

When executed, this action shows Form containing SubDocEditor.

This class is not used directly. The following actions are inherited from TrvCustomEditorAction:

- TrvActionEditNote

1.3.10.14.1 Properties

1.3.10.14.1.1 TrvCustomEditorAction.Form

Returns a form containing TRichViewEdit control.

```
property Form: TForm;
```

This form is created when the action is executed for the first time. This form is destroyed together with the action: on closing, it only hides itself.

See also

- OnFormCreate
- OnShowing and OnHide
- SubDocEditor

1.3.10.14.1.2 TrvCustomEditorAction.SubDocEditor

Returns a TRichViewEdit control inserted in Form.

```
property SubDocEditor: TRichViewEdit;
```

This control is created and destroyed together with Form,

1.3.10.14.2 Events

1.3.10.14.2.1 TrvCustomEditorAction.OnFormCreate

Occurs after the Form[208] is created.

```
property OnFormCreate: TRVShowFormEvent[235];
```

See also:

- OnShowing, OnHide[209] (contains example)

1.3.10.14.2.2 TrvCustomEditorAction.OnShowing, OnHide

Occurs before Form[208] is shown, and when it is hidden.

```
property OnShowing: TRVShowFormEvent[235];
property OnHide: TNotifyEvent;
```

Example

This example shows how to work with a form displayed by TrvActionEditNote[157].

Let the main form TfrmMain have:

- RichViewEdit: TRichViewEdit – the main editor;
- RVAControlPanel1: TRVAControlPanel[23];
- RVAPopupMenu1: TRVAPopupMenu[40];
- rvActionEditNote1: TrvActionEditNote;
- rvActionStyleInspector1: TrvActionStyleInspector[132];
- cmbFont: TRVFontComboBox[46];
- cmbFontSize: TRVFontSizeComboBox[46];
- cmbStyles: TRVStyleTemplateComboBox[47].

We want to insert Form[208] at the bottom of TfrmMain.

Initial property settings:

- RVAControlPanel1.DefaultControl[26] := RichViewEdit1;
- rvActionStyleInspector1.Control[195] := RichViewEdit1;
- cmbFont.Editor[44] := RichViewEdit1;
- cmbFontSize.Editor[44] := RichViewEdit1;
- cmbStyles.Editor[48] := RichViewEdit1

```
// OnFormCreate[209] event
procedure TfrmMain.rvActionEditNoteFormCreate(Sender: TrvAction[195]; Form: TForm);
begin
  // assigning properties of the note editor
  rvActionEditNote1.SubDocEditor.OnEnter := NoteEditorEnter;
  rvActionEditNote1.SubDocEditor.OnExit := NoteEditorExit;
  rvActionEditNote1.SubDocEditor.PopupMenu := RVAPopupMenu1;
  // moving the form to the bottom of frmMain
  Form.Align := alBottom;
  Form.Height := 100;
  Form.Parent := Self;
end;
```

```
// OnShowing event
procedure TfrmMain.rvActionEditNoteShowing(Sender: TrvAction; Form: TForm);
begin
  // leaving the current note in RichViewEdit1 highlighted after moving the input focus
  RichViewEdit1.ForceFieldHighlight := True;
end;
```

```
// OnHide event
procedure TfrmMain.rvActionEditNoteHide(Sender: TObject);
begin
  // restoring old values of properties
  RichViewEdit1.ForceFieldHighlight := False;
  // focusing the main editor
  if RichViewEdit1.CanFocus then
    RichViewEdit1.SetFocus;
end;
```

```
// OnEnter: occurs when the input focus is moved to the note editor
procedure TfrmMain.NoteEditorEnter(Sender: TObject);
begin
  // making the note editor default
  RVAControlPanel1.DefaultControl := rvActionsResource.rvActionEditNote1.SubDocEditor;
  UpdateLinkedControls;
end;
```

```
// OnExit: making the main editor default
procedure TfrmMain.NoteEditorExit(Sender: TObject);
begin
  RVAControlPanel1.DefaultControl := RichViewEdit1;
  UpdateLinkedControls;
end;
```

```
procedure TfrmMain.UpdateLinkedControls;
begin
  rvActionStyleInspector1.Control := RVAControlPanel1.DefaultControl;
  cmbFont.Editor := RVAControlPanel1.DefaultControl;
  cmbFontSize.Editor := RVAControlPanel1.DefaultControl;
  cmbStyles.Editor := RVAControlPanel1.DefaultControl;
end;
```

Note: a note editor is special, it redirects many actions (for example, TrvActionNew 57, TrvActionSave 69, TrvActionPrint 65) to the main editor.

1.4 Localization of RichViewActions

Changing UI language

RichViewActions require localization, even if you use only one UI language.

Initially, the language specified in TRVAControlPanel.Language 29 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 218 to localize them.

For changing UI language, use RVA_ChooseLanguage 217 function. The topic about this function contains an example.

Other localization functions are declared in RVALocalize 212 unit.

Hiding untranslated controls

It is possible that some new features added to RichViewActions were not translated to all languages yet.

Such controls can be hidden by ShowUntranslatedControls 236 global variable.

Excluding languages

You can reduce the size of your application by excluding some languages.

To exclude languages, define one more of the following compiler defines in the options of your project:

- RVA_NO_ARMENIAN
- RVA_NO_BULGARIAN
- RVA_NO_BYELORUSSIAN
- RVA_NO_CATALAN
- RVA_NO_CHINESE_SIMPL
- RVA_NO_CHINESE_TRAD
- RVA_NO_CZECH
- RVA_NO_DANISH
- RVA_NO_DUTCH
- RVA_NO_ENGLISH_US
- RVA_NO_FARSI
- RVA_NO_FINNISH
- RVA_NO_FRENCH
- RVA_NO_GERMAN
- RVA_NO_HINDI
- RVA_NO_HUNGARIAN
- RVA_NO_ITALIAN
- RVA_NO_LITHUANIAN
- RVA_NO_MALAY
- RVA_NO_NORWEGIAN
- RVA_NO_POLISH
- RVA_NO_PORTUGUESE_BR
- RVA_NO_PORTUGUESE_PT
- RVA_NO_ROMANIAN
- RVA_NO_RUSSIAN
- RVA_NO_SLOVAK
- RVA_NO_SPANISH
- RVA_NO_SWEDISH
- RVA_NO_THAI
- RVA_NO_TURKISH
- RVA_NO_UKRAINIAN

Resource language

There is one special language: "Resource language". It contains the same string as in "English (US)", but they are stored in resources. You can use this language together with an external localization tool.

By default, this language is excluded. You can enable it by removing the following line from RichViewActions.inc:

{$DEFINE RVA_NO_RESOURCELANGUAGE}

Adding a new language

If you want to add a translation to you language (for example, Greek):

1. Open RVAL_EngUS.pas, rename it to RVAL_Greek.pas. Add RVAL_Greek in "uses" of RVALocalize.pas. (If you use Delphi/C++Builder 2009, save this unit as UTF-8).
2. Translate all text in this unit (do not translate comments!)
3. Change the call of RVA_RegisterLanguage 214 to RVA_RegisterLanguage('Greek', Ελληνικά', GREEK_CHARSET, @Messages).
4. Remove the call of RVA_SwitchLanguage 214.
5. Send the translated file to ***svt@trichview.com*** for including in the main installation of RichViewActions. Your file will be maintained (it will be corrected when new messages will be added in the future versions of

RichViewActions)

1.4.1 RVALocalize unit

This unit contains procedures and functions working with a language of user interface in RichViewActions 14.

Procedures and functions declared in this unit

Procedures returning names of registered UI languages:

- RVA_EnumLanguages 212
- RVA_FillLanguageList 213

Functions returning properties of the current language:

- RVA_GetLanguageName 214
- RVA_GetCharset 213
- RVA_GetHelpFile 213

Functions returning a translated string:

- RVA_GetS, RVA_GetSH, RVA_GetPC 214
- RVA_TranslateUnits 215
- RVA_GetProgressMessage, RVA_GetPrintingMessage 215

Procedure for changing the current language:

- RVA_SwitchLanguage 214

Other procedures

The following procedures and functions are declared in other units, but related to localization:

- RVA_ChooseLanguage 217 (displays a dialog to choose a language name, changes the current language)
- RVA_LocalizeForm 218 (localizes all actions on the specified form/datamodule)
- RVALocalizeRuler 219 (localizes TRVRuler component)
- GetAddictSpellLanguage, GetAddictThesLanguage 218 (localizes Addict 3 219 components)
- RVA_GetColorName 218 (returns the color name)

Additional information

See also:

- Localization of RichViewActions 210

1.4.1.1 Procedures and Functions

1.4.1.1.1 RVA_EnumLanguages procedure

This procedure allows to enumerate all UI languages available for RichViewActions.

Unit RVALocalize 212.

```
//type
//  TGetStrProc = procedure (const S: string) of object;
procedure RVA_EnumLanguages(Proc: TGetStrProc);
```

This procedure calls **Proc** one time for each language name. Only English language names are passed to this procedure.

Names of languages can be assigned to TRVAControlPanel.Language 29 or used as a parameter for RVA_SwitchLanguage 214 procedure.

See also:

- RVA_FillLanguageList 213

1.4.1.1.2 RVA_FillLanguageList procedure

This procedure adds all UI language names available for RichViewActions in **sl**.

Unit RVALocalize[212].

```
type TRVALocStrings = TStrings;
procedure RVA_FillLanguageList(sl: TRVALocStrings; English: Boolean = True;
  Native: Boolean = False);
```

English	Native	Result
True	False	English language names
False	True	Native language names
True	True	Strings in format: "English language name - native language name" (or English language name, if they are equal)
False	False	English language names

Names of languages can be assigned to TRVAControlPanel.Language[29] or used as a parameter for RVA_SwitchLanguage[214] procedure.

1.4.1.1.3 RVA_GetCharset function

This method returns the charset for the current UI language.

Unit RVALocalize[212].

```
function RVA_GetCharset(ControlPanel: TComponent=nil): TFontCharset;
```

If **ControlPanel** is a TRVAControlPanel[23] component, the function uses it. Otherwise, it uses MainRVAControlPanel[235].

The name of the current language is returned as ControlPanel.Language[29], or by RVA_GetLanguageName[214] function.

The current language can be changed by assigning a new value to ControlPanel.Language[29], or by calling RVA_SwitchLanguage[214] procedure.

This charset is ignored in Delphi/C++Builder 2009 or newer, because Unicode strings do not need charsets. Use this function for Delphi/C++Builder 4-2007.

RichViewActions apply this charset to all fonts in all dialogs displayed by RichViewActions.

You should use this property to apply to your own forms (if they support a localization) and to Screen.HintFont and Screen.MenuFont:

```
Screen.HintFont.Charset := RVA_GetCharset; // Delphi 6-2007 is required
Screen.MenuFont.Charset := RVA_GetCharset; // Delphi 6-2007 is required
```

1.4.1.1.4 RVA_GetHelpFile function

This method returns the help file name for the current UI language.

Unit RVALocalize[212].

```
function RVA_GetHelpFile(ControlPanel: TComponent): String;
```

If **ControlPanel** is a TRVAControlPanel[23] component.

The name of the current language is returned as **ControlPanel**.Language[29], or by RVA_GetLanguageName[214] function.

The current language can be changed by assigning a new value to **ControlPanel**.Language [29], or by calling RVA_SwitchLanguage [214] procedure.

See also:

- TRVAControlPanel.UseHelpFiles [34]

1.4.1.1.5 RVA_GetLanguageName function

This method returns the name of the current UI language.

Unit RVALocalize [212].

```
function RVA_GetLanguageName(ControlPanel: TComponent=nil): TRVALanguageName [216];
```

If **ControlPanel** is a TRVAControlPanel [23] component, the function uses it. Otherwise, it uses MainRVAControlPanel [235].

The same value is returned in ControlPanel.Language [29] property.

The current language can be changed by assigning a new value to ControlPanel.Language [29], or by calling RVA_SwitchLanguage [214] procedure.

1.4.1.1.6 RVA_GetS function

This method returns a localized string.

Unit RVALocalize [212].

```
type
  TRVAMessageID = (rvam_Empty, rvam_menu_File, rvam_menu_Edit, ...);
function RVA_GetS(MsgID: TRVAMessageID; ControlPanel: TComponent=nil): String;
function RVA_GetSH(MsgID: TRVAMessageID; ControlPanel: TComponent=nil): String;
function RVA_GetPC(MsgID: TRVAMessageID; ControlPanel: TComponent=nil): PChar;
```

If **ControlPanel** is a TRVAControlPanel [23] component, the functions use it. Otherwise, they use MainRVAControlPanel [235].

These functions return strings corresponding to MsgID, translated to the current language. The name of the current language is returned as ControlPanel.Language [29], or by RVA_GetLanguageName [214] function.

The current language can be changed by assigning a new value to ControlPanel.Language [29], or by calling RVA_SwitchLanguage [214] procedure.

RVA_GetS just returns a string.

RVA_GetSH(MsgId) returns ' '+RVA_GetS(MsgId)+' ' (used for assigning to captions of radio groups and group boxes).

RVA_GetPC returns the same value as RVA_GetS, but not as a String but as PChar.

Note: if RichViewActions are compiled with TNT Controls [224], these functions return WideString.

1.4.1.1.7 RVA_SwitchLanguage procedure

Changes the current language.

```
procedure RVA_SwitchLanguage(const LanguageName: TRVALanguageName;
  ControlPanel: TComponent=nil);
```

If **ControlPanel** is a TRVAControlPanel [23] component, the function uses it. Otherwise, it uses MainRVAControlPanel [235].

Assigning a new value to ControlPanel.Language [29] has the same effect.

To get a list of valid languages, use RVA_EnumLanguages [212] or RVA_FillLanguageList [213].

You can use either English or native language name as a parameter.

To complete a language changing, the following procedures must be called after RVA_SwitchLanguage:

- RVA_LocalizeForm 218 for localization of Captions and Hints of all actions on the given form/datamodule;
- RVALocalizeRuler 219 for localizations of hints in TRVRuler component;
- GetAddictSpellLanguage(RVA_GetLanguageName), GetAddictThesLanguage(RVA_GetLanguageName) 218 for assigning UILanguage property of TRVAddictSpell3 and TRVThesaurus3 (only if Addict 3 219 is used);
- RVA_GetCharset 213 for changing Charsets of menus, hints and your forms;
- RVA_GetS 214 for localizing standard menu items in your menu (such as "File", "Edit"...).

See also:

- RVA_ChooseLanguage 217

1.4.1.1.8 RVA_TranslateUnits procedure

Translates units of measurement to the current language.

```
procedure RVA_TranslateUnits(SL: TStrings; ControlPanel: TComponent=nil);
```

If **ControlPanel** is a TRVAControlPanel 23 component, the function uses it. Otherwise, it uses MainRVAControlPanel 235.

SL must have 6 items corresponding to the following units, in the specified order:

- Inches,
- Centimeters,
- Millimeters,
- Picas,
- Pixels,
- Points.

The name of the current language is returned as ControlPanel.Language 29, or by RVA_GetLanguageName 214 function.

The current language can be changed by assigning a new value to ControlPanel.Language 29, or by calling RVA_SwitchLanguage 214 procedure.

1.4.1.1.9 Functions for progress messages

```
function RVA_GetProgressMessage(Operation: TRVLongOperation;
  ControlPanel: TComponent=nil): String;
function RVA_GetPrintingMessage(PageCompleted: Integer; Step:TRVPrintingStep;
  ControlPanel: TComponent=nil): String;
```

If **ControlPanel** is a TRVAControlPanel 23 component, the functions use it. Otherwise, they use MainRVAControlPanel 235.

RVA_GetProgressMessage returns a localized text message describing the specified long operation. It can be used in TRichViewEdit.OnProgress event.

RVA_GetPrintingMessage returns a localized text message describing the current printing progress. It can be used in TRVPrint.OnSendingToPrinter event.

The name of the current language is returned as ControlPanel.Language 29, or by RVA_GetLanguageName 214 function.

The current language can be changed by assigning a new value to ControlPanel.Language 29, or by calling RVA_SwitchLanguage 214 procedure.

Example:

This example assumes that Application.Hint is displayed in a status bar.

```
// TRVAControlPanel.OnDownload 38
procedure TForm3.RVAControlPanel1Download(Sender: TrvAction;
  const Source: String);
begin
```

```
  if Source='' then
    Application.Hint := ''
  else
    Application.Hint := RVAFormat(RVA_GetS(rvam_msg_Downloading), [Source]);
end;
{-------------------------------------------------------------------------------}
// TRVPrint.OnSendingToPrinter
procedure TForm3.RVPrint1SendingToPrinter(Sender: TCustomRichView;
  PageCompleted: Integer; Step: TRVPrintingStep);
begin
  Application.Hint := RVA_GetPrintingMessage(PageCompleted, Step);
end;
{-------------------------------------------------------------------------------}
// Reading or writing
procedure TForm3.RichViewEdit1Progress(Sender: TCustomRichView;
  Operation: TRVLongOperation; Stage: TRVProgressStage; PercentDone: Byte);
begin
  case Stage of
    rvpstgStarting:
      begin
        ProgressBar1.Position := 0;
        ProgressBar1.Visible := True;
        Application.Hint := RVA_GetProgressMessage(Operation);
      end;
    rvpstgRunning:
      begin
        ProgressBar1.Position := PercentDone;
        ProgressBar1.Update;
      end;
    rvpstgEnding:
      begin
        Application.Hint := '';
        ProgressBar1.Visible := False;
      end;
  end;
end;
```

1.4.1.2 Types

1.4.1.2.1 TRVALanguageName

A type for UI language name.

Unit RVALocalize[212].

```
type
  TRVALanguageName = type String;
```

See also:

- TRVAControlPanel.Language[29]
- RVA_GetLanguageName[214]
- RVA_SwitchLanguage[214]

1.4.2 RichViewActions unit

1.4.2.1 Procedures and functions

1.4.2.1.1 RVA_ChooseLanguage function

Displays a dialog for choosing UI language and applies the chosen language.

Unit RichViewActions.

```
function RVA_ChooseLanguage: Boolean;
```

This function displays a dialog containing a list with all available UI languages.

If the user pressed "OK", this function calls RVA_SwitchLanguage[214] and returns *True*.

You may need to perform additional operations on the language changing, see RVA_SwitchLanguage[214] for details.

In Delphi 4-2007, this function displays language names in English.

In new versions of Delphi (or if TNT Controls[224] are used), this function displays both English and native language names.

Example (from the ActionTest demo):

```
// button at the right side of status bar
procedure TForm3.Button1Click(Sender: TObject);
begin
  if RVA_ChooseLanguage then
    Localize;
end;

procedure TForm3.FormCreate(Sender: TObject);
begin
  ...
  Localize;
  ...
end;

procedure TForm3.Localize;
var Index: Integer;
begin
  // Fonts (for Delphi 4-2007)
  Font.Charset            := RVA_GetCharset[213];
  StatusBar1.Font.Charset := RVA_GetCharset[213];
  // Fonts (for Delphi 6-2007)
  Screen.HintFont.Charset := RVA_GetCharset[213];
  Screen.MenuFont.Charset := RVA_GetCharset[213];
  // Localizing all actions on rvActionsResource
  RVA_LocalizeForm[218](rvActionsResource);
  // Localizing all actions on this form
  RVA_LocalizeForm[218](Self);
  // Localizing ruler
  RVALocalizeRuler[219](RVRuler1);
  // Localizing menus
  mitFile.Caption := RVA_GetS[214](rvam_menu_File);
  mitEdit.Caption := RVA_GetS[214](rvam_menu_Edit);
  // Styles
  rvActionsResource.rvActionStyleInspector1.UpdateInfo[133];
  RVStyleTemplateComboBox1.Localize[49];
  // Localizing combobox with units of measurement
```

```
  Index := cmbUnits.ItemIndex;
  RVA_TranslateUnits[215](cmbUnits.Items);
  cmbUnits.ItemIndex := Index;
  // Only if Addict 3 or 4[219] is used:
  RVAddictSpell31.UILanguage := GetAddictSpellLanguage[218](RVA_GetLanguageName);
  RVThesaurus31.UILanguage := GetAddictThesLanguage[218](RVA_GetLanguageName);
end;
```

1.4.2.1.2 RVA_GetColorName function

Returns a localized color name.

Unit RichViewActions.

```
function RVA_GetColorName(Color: TColor): String;
```

For *clNone*, it returns 'Transparent' (translated to the current language).

For *clWindow, clBtnFace, clWindowText*, it returns 'Auto' (translated to the current language).

For 40 standard colors, it returns their localized names.

For other colors, it returns 'RGB(NN,NN,NN)' string.

See also:

- Localization of RichViewActions[210].

1.4.2.1.3 RVA_LocalizeForm procedure

Applies the current language to all RichViewActions on **Form**.

Unit RichViewActions.

```
procedure RVA_LocalizeForm(Form: TComponent);
```

Form is a form or datamodule. This procedure changes Captions and Hints of all actions (inherited from TrvCustomAction[206]) owned by this form/datamodule.

The name of the current language is returned as Action.GetControlPanel[208].Language[29].

The current language can be changed by assigning a new value to TRVAControlPanel.Language[29], or by calling RVA_SwitchLanguage[214] procedure.

See also:

- RVALocalize unit[212].

1.4.3 Other units

1.4.3.1 GetAddictSpellLanguage, GetAddictThesLanguage functions

The functions convert a language of RichViewActions to languages used by Addict 3[219] components.

Unit RVAAddictLanguages.

```
function GetAddictSpellLanguage(const s: String): TSpellLanguageType;
function GetAddictThesLanguage(const s: String): TThesaurusLanguageType;
```

Example:

```
  RVAddictSpell31.UILanguage := GetAddictSpellLanguage(RVA_GetLanguageName[214]);
  RVThesaurus31.UILanguage := GetAddictThesLanguage(RVA_GetLanguageName[214]);
```

1.4.3.2 RVALocalizeRuler procedure

Localizes a TRVRuler component.

Unit RVALocRuler.

```
procedure RVALocalizeRuler(Ruler: TRVRuler);
```

The method assigns localized strings to hints displayed by **Ruler**.

See also:

- Localization of RichViewActions 210
- RVALocalize unit 212

1.5 Third-party tools

Components

The following additional components can be optionally used in RichViewActions:

- Addict 3 or 4 219 spellchecker and thesaurus (shareware)
- other spellcheckers (freeware and shareware)
- Indy 220 (freeware) or CleverComponents 221 (shareware) for downloading external images
- RvHtmlImporter 221 for inserting and pasting HTML files (freeware)
- RvHtmlViewImporter 222 for opening, inserting and pasting HTML files (freeware)
- RichViewXML 223 for opening and saving documents in XML format (freeware)
- Toolbar 2000 224 (free for non-commercial use), TBX 224, SpTBXLib 223 for providing alternative versions of TRVAPopupMenu 40
- ThemeEngine 223 for providing a skinned user interface (discontinued)
- TNT Controls 224 for Unicode support in Delphi 4-2007.

Toolbar Images

By default, RichViewActions use simple 16-color 16x16 images.

In Delphi 2009+ (or if you use a thirdparty image list component supporting semitransparent images), you can use TRichView icons 225, or the following alternative toolbar images:

- GlyFX 225
- Glyfz 226
- Fugue Icons 227
- FamFamFam Silk Icons 227

1.5.1 Components

1.5.1.1 Addict 3 and 4

Addict 3 and 4 is a spelling check and thesaurus component suite. Visit ***www.addictive-software.com*** for additional information.

All registered users of RichView Package get a 10% discount when registering Addict Professional + PlusPack (spell checker and thesaurus).

Installing

1. Install the Addict 3 or 4 packages.
2. Install ***RichView parser for Addict 3 and 4***.
3. Open **RichViewActions.inc**, remove the dot from the line {.$DEFINE USERVADDICT3}.
4. Open the RichViewActions package. Add a link to the parser's package in "Requires" section. Install the package 14 (or rebuild it, if already installed).

5. Create TRVAddictSpell3 and TRVThesaurus3 components, assign them to RVAddictSpell3 [30] and RVAddicthesaurus3 [30] properties of TRVAControlPanel [23] component (if you do not see these properties, steps 1-4 were not completed)
6. Create and use TrvActionAddictSpell3 [194] and TrvActionAddictThesaurus3 [194] actions.

Using

Assign TrvActionAddictSpell3 [194] and TrvActionAddictThesaurus3 [194] to menu items or toolbar buttons.

If you need an autocorrection feature, see RVA_Addict3AutoCorrect [228] procedure.

When changing the UI language, you should change languages of TRVAddictSpell3 and TRVThesaurus3 components using GetAddictSpellLanguage and GetAddictThesLanguage [218] functions (see the example in the topic about RVA_ChooseLanguage [217] function).

Live spelling check

The examples in this section assume that RVAddictSpell31: TRVAddictSpell3 component is placed on the Form1 form.

To enable a live spelling check, assign OnSpellingCheck event of TCustomRichViewEdit component:

```
procedure TForm1.RichViewEdit1SpellingCheck(Sender: TCustomRichView;
  const AWord: String; StyleNo: Integer; var Misspelled: Boolean);
begin
  Misspelled := not RVAddictSpell31.WordAcceptable(AWord);
end;
```

If you want to start a live spelling check when the user starts editing, assign *rvlspOnChange* to LiveSpellingMode property of TCustomRichViewEdit component.

If you want to start a live spelling check when the user loaded a document, use OnOpenFile [62] event of TrvActionOpen [59] action:

```
procedure TForm1.RvActionOpen1OpenFile(Sender: TObject; Editor: TCustomRichViewEdit;
  const FileName: String; FileFormat: TrvFileOpenFilter; CustomFilterIndex: Integer);
begin
  Editor.StartLiveSpelling;
end;
```

When the user pressed "Add" or "Ignore All" in the spell dialog, red underlines must be removed from all occurrences of the added/ignored word. Use RVAddictSpell31.OnParserIgnoreWord event:

```
procedure TForm3.RVAddictSpell31ParserIgnoreWord(Sender: TObject;
  Before: Boolean; State: Integer);
begin
  if not Before and (State in [IgnoreState_IgnoreAll, IgnoreState_Add]) then
    RichViewEdit1.LiveSpellingValidateWord(RVAddictSpell31.CurrentWord);
end;
```

1.5.1.2 Indy

HTML files (imported with RvHtmlImporter [221] or RvHtmlViewImporter [222]) and RTF files can contain links to images located on web server. If you want to use them, it's required to download them inside OnImportPicture event of TCustomRichViewEdit component.

RichViewActions can do it for you using Indy (TIdHTTP) component.

Indy is included in new versions of Delphi, or it can be ***downloaded***.

If the support for both Indy and CleverComponents [221] is configured, Indy is used.

Installing

1. Make sure that Indy packages are installed.
2. Open **RichViewActions.inc**, remove the dot from the line {.$DEFINE USEINDY}.
3. Open the RichViewActions package. Add a link to the Indy package in "Requires" section. Install the package 14 (or rebuild it, if already installed).
4. Create TIdHTTP component, assign it to IdHTTP 29 property of TRVAControlPanel 23 component. (if you do not see this property, steps 1-3 were not completed).

See also:

- TRVAControlPanel.OnDownload 38 event

1.5.1.3 CleverComponents

HTML files (imported with RvHtmlImporter 221 or RvHtmlViewImporter 222) and RTF files can contain links to images located on web server. If you want to use them, it's required to download them inside OnImportPicture event of TCustomRichViewEdit component.

RichViewActions can do it for you using CleverComponents (TClHTTP) component.

Visit ***http://www.clevercomponents.com*** for additional information.

If the support for both Indy 220 and CleverComponents is configured, Indy is used.

Installing

1. Make sure that Indy packages are installed.
2. Open **RichViewActions.inc**, remove the dot from the line {.$DEFINE USECLEVERCOMPONENTS}.
3. Open the RichViewActions package. Add a link to the CleverComponent package in "Requires" section. Install the package 14 (or rebuild it, if already installed).
4. Create TclHTTP component, assign it to IdHTTP 29 property of TRVAControlPanel 23 component. (if you do not see this property, steps 1-3 were not completed).

See also:

- TRVAControlPanel.OnDownload 38 event

1.5.1.4 RvHtmlImporter

RVHTMLImporter allows importing HTML files in TCustomRichView components.

RichViewAction support RVHTMLImporter in the following actions:

- TrvActionInsertFile 136 (RVHTMLImporter replaces the MS Office HTML import converter in the file insertion dialog)
- TrvActionPasteSpecial 78
- TrvActionPaste 77 (pasted if not RVF or non-empty RTF available in the Clipboard)

RVHTMLImporter can be downloaded from ***www.trichview.com/resources/***.

Installing

1. Install the RVHTMLImporter package.
2. Open **RichViewActions.inc**, remove the dot from the line {.$DEFINE USERVHTML}.
3. Open the RichViewActions package. Add a link to the RVHTMLImporter package in "Requires" section. Install the package 14 (or rebuild it, if already installed).
4. Create TRVHTMLImporter component, assign it to RVHTMLImporter 31 property of TRVAControlPanel 23

component (if you do not see this property, steps 1-3 were not completed)

Make sure that *ffiHTML* is in Filter[137] property of TrvActionInsertFile[136] action.

Include *rvddHTML* in AcceptPasteFormats property of the editor.

See also:

- PasteHTML[228] procedure
- RvHtmlViewImporter[222]

1.5.1.5 RvHtmlViewImporter

RVHTMLViewImporter allows loading HTML files in RichView using HTML parser from THTMLViewer (freeware with source).

If activated, RichViewActions support HTML files in the following actions:

- TrvActionOpen[59]
- TrvActionSaveAs[73] (RVHTMLViewImporter is not used for saving, but HTML is added in the "Save As" action because it is added in the "Open" action)
- TrvActionInsertFile[136] (if both RvHtmlViewImporter and RvHtmlImporter[221] are used, this action uses RvHtmlViewImporter)
- TrvActionPasteSpecial[78] (if both RvHtmlViewImporter and RvHtmlImporter[221] are used, this action uses RvHtmlViewImporter)
- TrvActionPaste[77] (pasted if not RVF or non-empty RTF available in the Clipboard; if both RvHtmlViewImporter and RvHtmlImporter[221] are used, this action uses RvHtmlViewImporter)

THTMLViewer can be downloaded from ***code.google.com/p/thtmlviewer/***.

RvHtmlViewImporter can be downloaded from ***www.trichview.com/resources/***.

Installing

1. Read the installation instructions in RvHtmlViewImporter and make the proper changes in THTMLViewer packages and source code.
2. Install HTMLViewer.
3. Install RVHTMLViewImporter.
4. Open **RichViewActions.inc**, remove the dot from the line {.$DEFINE USERVHTMLVIEWER}.
5. Open the RichViewActions package. Add a link to the RVHTMLViewImporter package in "Requires" section. Install the package[14] (or rebuild it, if already installed).
6. Create THTMLViewer and TRVHTMLViewImporter components, assign THTMLViewer component to HTMLViewer property of TRVHTMLViewImporter component, assign TRVHTMLViewImporter component to RVHTMLViewImporter[31] property of TRVAControlPanel[23] component (if you do not see this property, steps 1-5 were not completed)

Make sure that *ffiHTML* is included in Filter property of TrvActionInsertFile[136], TrvActionOpen[59], *ffeHTML* is included in Filter property of TrvActionSaveAs[73], TrvActionExport[62].

Include *rvddHTML* in AcceptPasteFormats property of the editor.

See also:

- PasteHTML[228] procedure
- RvHtmlImporter[221]

1.5.1.6 RichViewXML

RichViewXML provides an alternative to RVF way for lossless storing of RichView documents.

If activated, RichViewActions support RichView XML files in the following actions:

- TrvActionOpen 59
- TrvActionSaveAs 73
- TrvActionInsertFile 136

RichViewXML can be downloaded from ***www.trichview.com/resources/xml/***.

Installing

1. Install the RichViewXML package.
2. Open **RichViewActions.inc**, remove the dot from the line {.$DEFINE USERVXML}.
3. Open the RichViewActions package. Add a link to the RichViewXML package in "Requires" section. Install the package 14 (or rebuild it, if already installed).
4. Create TRichViewXML component, assign it to RVXML 32 property of TRVAControlPanel 23 component (if you do not see this property, steps 1-3 were not completed)

Make sure that *ffiXML* is in Filter of TrvActionOpen 59, TrvActionInsertFile 136, *ffeXML* in Filter of TrvActionSaveAs 73, TrvActionExport 62.

You can change a file filter for XML files: see TRVAControlPanel.XMLFilter 35 property.

1.5.1.7 SpTBXLib

SpTBXLib is an expansion package for TB2K 224 components that adds Unicode support and skinning. You can download it from ***www.silverpointdevelopment.com/sptbxlib/***.

If activated, TRVASPTBXPopupMenu 40 component is available.

Installing

1. Install the SpTBXLib package.
2. Open **RichViewActions.inc**, remove the dot from the line {.$DEFINE USESPTBX}.
3. Open the RichViewActions package. Add a link to the SpTBXLib package in "Requires" section. Install the package 14 (or rebuild it, if already installed).

1.5.1.8 ThemeEngine

ThemeEngine is discontinued. This product is no longer developed.

ThemeEngine is a library of components allowing to change visual appearance and behavior of your application with use of the powerful set of tools including visual components, convenient designers, easy-to-use non-visual components, the optimized graphic algorithms and a set of ready themes. ThemeEngine can add themes to forms, to the menu and to the elements of the management. Handreds stable easy to use components and thousands free skins. You can download it from ***www.ksdev.com***.

If activated, the most of standard controls on RichViewActions dialogs are replaced with ThemeEngine controls. Colors of other controls are changed according to the selected theme.

ThemeEngine's controls support Unicode.

Installing

1. Open ThemeEngine package and set "Explicit rebuild" option. For Delphi 4-7, open the package, click "Options" in the package window, on the tab "Description" set "Build control"="Explicit rebuild". For newer

version of Delphi, open the package, right click in the Project Manager and choose "Options", on the page "Description" set "Build control"="Explicit rebuild".

2. Install the ThemeEngine package.
3. Open **RichViewActions.inc**, remove the dot from the line {.$DEFINE USERVKSDEVTE}.
4. Open the RichViewActions package. Add a link to the ThemeEngine package in "Requires" section. Install the package [14] (or rebuild it, if already installed).

See ActionTest_TE.dpr demo.

1.5.1.9 TNT Controls

TMS Unicode Component Pack controls allow you to develop applications that take advantage of the Unicode capabilities of Windows NT/2000/XP/2003/Vista without abandoning Delphi, C++Builder or Windows 95/98/ME.You can download them from ***www.tmssoftware.com***.

Since Delphi/C++Builder 2009 features a built-in support of Unicode, TMS Unicode Component Pack should be used only in older versions of Delphi.

If activated, the most of standard controls on RichViewActions dialogs are replaced with TNT controls. Caption [207] and Hints [207] properties of RichViewActions are changed to Unicode. TRVAPopup [40] menu, find and replace dialogs support Unicode.

Installing

1. Open TMS Unicode Component Pack packages and set "Explicit rebuild" option. For Delphi 4-7, open the package, click "Options" in the package window, on the tab "Description" set "Build control"="Explicit rebuild". For newer version of Delphi, open the package, right click in the Project Manager and choose "Options", on the page "Description" set "Build control"="Explicit rebuild".
2. Install TMS Unicode Component Pack.
3. Open **RichViewActions.inc**, remove the dot from the line {.$DEFINE USERVTNT}.
4. Open the RichViewActions package. Add a link to the TMS Unicode Component Pack package in "Requires" section. Install the package [14] (or rebuild it, if already installed).

See ActionTest_TNT.dpr demo.

1.5.1.10 TBX

SpTBXLib is an expansion package for TB2K [224] components. You can download it from ***code.google.com/p/tbxlib/***

If activated, TRVATBXPopupMenu [40] component is available.

Installing

1. Install the TBX package.
2. Open **RichViewActions.inc**, remove the dot from the line {.$DEFINE USETBX}.
3. Open the RichViewActions package. Add a link to the TBX package in "Requires" section. Install the package [14] (or rebuild it, if already installed).

1.5.1.11 Toolbar2000

Toolbar2000 is a set of components designed to mimic the look and behavior of Office 2000's menus and toolbars. You can download it from ***www.jrsoftware.org***.

It has the following extensions:

- TBX 224
- SpTBXLib 223

If activated, TRVATBPopupMenu 40 component is available.

Installing

1. Install the Toolbar2000 package.
2. Open **RichViewActions.inc**, remove the dot from the line {.$DEFINE USETB2K}.
3. Open the RichViewActions package. Add a link to the Toolbar2000 package in "Requires" section. Install the package 14 (or rebuild it, if already installed).

1.5.2 Toolbar Images

1.5.2.1 TRichView Icons

TRichView Icons

This set of icons was created specially for RichViewActions.

It is used in this manual, and in ActionTest demos for Delphi 2009 and newer (including Ribbon demos).

The set includes 16x16 icons for almost all actions, and 32x32 icons for the main actions. Additionally, it includes several icons for using in the scrollbar areas of TSRichViewEdit (ScaleRichView).

For Delphi 2009+ only (because of semitransparent images).

The images are free for registered TRichView users. You can use these images in your applications. Any other use requires a special permission. You cannot sell these images or any graphic work created basing on them.

Additional information

Additional information is available on the support forum.

1.5.2.2 GlyFX toolbar images

GlyFX Images

If you want to use modern looking images, consider ***GlyFX*** toolbar images.

There are 4 datamodules available.

Two datamodules based on GlyFX Windows XP icon sets:

- XP Common (free)
- XP Core II
- XP Word processing.

Two datamodules based on GlyFX Windows Vista icon sets:

- Vista Common (free)
- Vista Core II
- Vista Word processing.

The corresponding sets must be purchased in order to use these datamodules. After purchasing, request datamodules from GlyFX.

All registered users of TRichView get a 25% discount when registering all GlyFX products.

Datamodules:

- *dmActionsGlyFX.pas* uses XP-style images without alpha channel (based on bmp). Can be used in Delphi 4+
- *dmActionsGlyFXAlpha.pas* uses XP-style images with alpha channel (based on png). Can be used in Delphi 2009+

- *dmActionsGlyFXAero.pas* uses Vista-style images without alpha channel (based on bmp). Can be used in Delphi 4+
- *dmActionsGlyFXAeroAlpha.pas* uses Vista-style images with alpha channel (based on png). Can be used in Delphi 2009+

All datamodules include 6 image lists: normal, highlighted disabled; all in 16x16 and 24x24.

Additionally, projects with GlyFX and TRibbon are available for TRichViewEdit and TSRichViewEdit.

Installing

It's not necessary to install files to use these datamodules.

However, if you want to use one of these modules at design time instead of dmActions.pas:

1. Open **RichViewActions.inc** and remove the dot from one of the following lines:
   {.$DEFINE USEGLYFX} // to use dmActionsGlyFX (XP-style, 1 bit transparency)
   {.$DEFINE USEGLYFXALPHA} // to use dmActionsGlyFXAlpha (XP-style, 1 byte transparency, D2009+)
   {.$DEFINE USEGLYFXAERO} // to use dmActionsGlyFXAero (Vista-style, 1 bit transparency)
   {.$DEFINE USEGLYFXAEROALPHA} // to use dmActionsGlyFXAeroAlpha (Vista-style, 1 byte transparency, D2009+)
2. Remove dmActions.pas from the RichViewActions package
3. Install (or recompile) the package.

Additional information

Additional information is available on the support forum.

1.5.2.3 Glyfz toolbar images

Glyfz Images

If you want to use toolbar images looking very close to MS Word 2007/2010 images, consider ***Glyfz*** toolbar images.

We have the following demos:

- analog of Ribbon\ActionTestRibbon demo for TRichViewEdit
- analog of ActionTestTabsRibbon demo for ScaleRichView.

These projects can be used only in Delphi 2009+, because they require TRibbon component, and because images are semitransparent.

They use the following sets of Glyfz images:

- Office 2007 Ribbon Bar Icons, Core Set 1
- Office 2007 Ribbon Bar Icons, Core Set 2
- Office 2007 Ribbon Bar Icons, Word Processing Set 1
- Office 2007 Ribbon Bar Icons, Word Processing Set 2

The corresponding sets must be purchased in order to use these projects. After purchasing, request this project from Glyfz.

The projects use 16x16 and 32x32 images (32x32 images are used only for several commands)

Additional information

Additional information is available on the support forum.

1.5.2.4 Fugue Icons toolbar images

Fugue Icons

If you want to use modern looking images, consider ***Fugue Icons*** toolbar images.

Fugue Icons – Copyright © 2009 Yusuke Kamiyamane.

The icons can be used free for any personal or commercial projects under a ***Creative Commons Attribution 3.0 license***.

If you can't or don't want to place link to the icons author website , please visit ***p.yusukekamiyamane.com*** and purchase a royalty-free license.

Download a datamodule with these icons from ***http://www.trichview.com/resources/actions/dmactionsfugue.zip***

For Delphi 2009+ only (because of semitransparent images)

The datamodule contains 2 identical sets of icons: with and without shadows.

Installing

It's not necessary to install files to use this datamodule.

However, If you want to use the datamodule at design time instead of dmActions.pas:

1. Open **RichViewActions.inc** and remove the dot from one of the following line:
   {.$DEFINE USEFUGUEICONS} // to use dmActionsFugue (1 byte transparency, D2009+)
2. Remove dmActions.pas from the RichViewActions package
3. Install (or recompile) the package.

Additional information

Additional information is available on the support forum.

1.5.2.5 FamFamFam Silk Icons toolbar images

FamFamFam Silk Icons

A FamFamFam Icons datamodule for RichViewActions is available.

You can download it from ***http://www.trichview.com/resources/actions/dmactionssilk.zip***

For Delphi 2009+ only (because of semitransparent images)

Silk Icons - Copyright © Mark James, ***http://www.famfamfam.com***.

The icons are licensed under a Creative Commons Attribution 2.5 or 3.0 License, details can be found on the author's website.

Unfortunately, this set contains not too many icons for table operations.

Additional information

Additional information is available on the support forum.

1.6 Procedures and Functions

1.6.1 RichViewActions unit

1.6.1.1 ConvertToUnicode procedure

Converts all text in **rv** to Unicode.

Unit RichViewActions

```
procedure ConvertToUnicode(rv: TCustomRichView);
```

This procedure changes Unicode properties of all **rv**.Style.TextStyles to *True*, and converts all non-Unicode text in **rv** to Unicode.

This method is called by PasteHTML 228.

1.6.1.2 GoToCheckpoint function

Moves the caret in **rv** to the checkpoint having the specified name.

Unit RichViewActions

```
function GoToCheckpoint(rv: TCustomRichViewEdit; const CPName: String): Boolean;
```

Returns *True* if the caret was successfully moved. Returns *False* if the checkpoint is not found.

This function is used in TrvActionInsertHyperlink.GoToLink 145 method.

1.6.1.3 PasteHTML procedure

Pastes HTML from the Clipboard to **rve** (if available)

Unit RichViewActions.

```
procedure PasteHTML(rve: TCustomRichViewEdit);
```

This procedure is available only if RvHtmlViewImporter 222 or RvHtmlImporter 221 is used.

The procedure does the following operations:

- loading HTML from the Clipboard to a temporal TRichView component;
- if **rve**.RTFReadProperties.UnicodeMode=*rvruOnlyUnicode*, converting a document in this temporal TRichView component to Unicode (see ConvertToUnicode 228);
- inserting the content of this temporal TRichView component in **rve**, in the caret position (via TMemoryStream in RVF format).

This procedure is used in TrvActionPaste 77 and TrvActionPasteSpecial 78 actions.

1.6.1.4 RVA_Addict3AutoCorrect function

Helps to implement autocorrection of spelling on typing.

Unit RichViewActions.

```
function RVA_Addict3AutoCorrect(rve: TCustomRichViewEdit;
  ControlPanel: TRVAControlPanel 23=nil): Boolean;
```

This function is available only if Addict 3 or 4 219 is used.

If **ControlPanel**=*nil*, the function uses MainRVAControlPanel 235.

The function works only if TRVAddictSpell3 component is assigned to **ControlPanel**.RVAddictSpell3 30 property, and the dictionary for autocorrection is active (for English language, it is **autocorrect.adu**).

The function checks if a correction exists for the word to the left of the caret. If found, the word is replaced.

This function should be called from OnKeyPress event of TCustomRichViewEdit component:

```
procedure TForm3.RichViewEdit1KeyPress(Sender: TObject; var Key: Char);
```

```
begin
  if (Key in ['''', #13, #9]) or (Pos(Key, RichViewEdit1.Delimiters)<>0) then
    RVA_Addict3AutoCorrect(RichViewEdit1);
end;
```

Return value:

True if a correction is pefromed

1.6.1.5 RVA_ChooseStyle function

Displays a dialog for choosing a visual style (Delphi XE2+) and applies the chosen style.

Unit RichViewActions.

```
function RVA_ChooseStyle: Boolean;
```

This function requires Delphi XE2 or newer. When called from applications compiled in older version of Delphi, it displays an error message.

1.6.1.6 RVA_EditorControlFunctionDef function

The function allows for actions inherited from TrvCustomEditAction 84 working with additional editor types.

Unit RichViewActions;

```
function RVA_EditorControlFunctionDef(Control: TControl;
    Command: TRVAEditorControlCommand): Boolean;
```

This function is a default value of RVA_EditorControlFunction 236 variable.

It allows working with:

- TCustomEdit (TEdit, TMemo, TRichEdit and others);
- TComboBox (having csDropDown or csSimple style).

1.6.1.7 RVA_ConvertToPixels, RVA_ConvertToTwips procedures

The procedures convert all properties measured in TRVAControlPanel 23.UnitsProgram 33 for all actions on the **Form**.

Unit RichViewActions.

```
procedure RVA_ConvertToPixels(Form: TComponent);
procedure RVA_ConvertToTwips(Form: TComponent);
```

Form is a form or datamodule. This procedure changes values of properties of all actions (inherited from TrvCustomAction 206) owned by this form/datamodule.

RVA_ConvertToPixels must be called before assigning *rvstuPixels* to TRVAControlPanel 23.UnitsProgram 33.

RVA_ConvertToTwips must be called before assigning *rvstuTwips* to TRVAControlPanel 23.UnitsProgram 33.

These procedures do not change value of TRVAControlPanel 23.UnitsProgram 33.

If you have actions linked to different control panels on this form/datamodule, you must perform this conversion for all these control panels at the same time (first, call **RVA_ConvertToPixels/RVA_ConvertToTwips**, next, change UnitsProgram 33 for all these control panels).

1.6.1.8 RvHtmlHelp function

Opens an HTML help file (*.CHM)

Unit RichViewActions.

```
function RvHtmlHelp(HelpFileName: TRVUnicodeString): Boolean;
```

HelpFileName is a name of CHM file, for example 'RichViewActions.chm'. It may include a topic name, for

example 'RichViewActions.chm::/RvHtmlHelp.htm' displays this topic.

1.6.2 RVARibbonUtils unit

This unit contains procedures and functions for working with TRibbon control.

- FillActionClientContents 230
- RemoveEllipsis 230, RemoveEllipsisFromRibbon 230
- LocalizeRibbonPage 230
- SetRulerColors 230

1.6.2.1 FillActionClientContents procedure

For all action items in Manager, copies Action.Hint to CommandProperties.Content;

Unit RVARibbonUtils 230.

```
procedure FillActionClientContents(Manager: TActionManager);
```

1.6.2.2 RemoveEllipsis, RemoveEllipsisFromRibbon procedures

Unit RVARibbonUtils 230.

```
procedure RemoveEllipsis(Form: TComponent);
procedure RemoveEllipsisFromRibbon(Manager: TActionManager);
```

RemoveEllipsis removes ending '...' from captions of all actions on the form. Not recommended for using.

RemoveEllipsisFromRibbon removes ending '...' from captions of all Manager.ActionBars[].Items[].

1.6.2.3 LocalizeRibbonPage procedure

Assigns Caption and KeyTip to the ribbon **Page**.

Unit RVARibbonUtils 230.

```
procedure LocalizeRibbonPage(Page: TRibbonPage; CaptionID: TRVAMessageID);
```

This procedure assigns the message identified by **CaptionID** to **Page**.Caption, after removing '&' from it. If it finds '&', it assigns a hot key from this message to **Page**.KeyTip.

1.6.2.4 SetRulerColors procedure

Changes **Ruler**'s colors according to **ColorMap**.

Unit RVARibbonUtils 230.

```
procedure SetRulerColors(Ruler: TRVRuler; ColorMap: TCustomRibbonColorMap);
```

1.6.3 Other units

1.6.3.1 LoadCSV function

The function loads CSV ("comma-separated values") files as a table. Only ANSI files can be loaded.

Unit rvcsv.

```
function LoadCSV(const FileName: String; rv: TCustomRichView;
  Separator: TRVAnsiChar; UseQuotes: Boolean;
  TextStyleNo, ParaNo: Integer): TRVTableItemInfo;
```

Parameters:

FileName – file to load.

rv – TCustomRichView component where you plan to insert the loaded table (it is not inserted by this function).

Separator – character used to separate values. The most common possible values:

- ',' (comma)
- ';' (semicolon)
- ' ' (space)
- #9 (tab)

UseQuotes – if *True*, double-quotes are used to allow separators inside text values (in the input file); double double-quotes are treated as double-quote character.

TextStyleNo, **ParaNo** – styles of text and paragraph used to add text

Return value:

the loaded table; *nil* in case of failure.

Example:

```
var table: TRVTableItemInfo;
...
table := LoadCSV(FileName, RichViewEdit1, ',', True, 0, 0);
if table<>nil then
  RichViewEdit1.InsertItem('', table);
```

You can implement inserting CSV files using TRVAControlPanel.OnCustomFileOperation 37 event.

1.6.3.2 MarkSubString functions

Marks all occurrences of **s** in **rv**.

Unit MarkSearch.

```
function MarkSubStringA(const s: TRVAnsiString;
  Color, BackColor: TColor; IgnoreCase, WholeWords: Boolean; rv: TCustomRichView;
  RVData: TCustomRVData=nil): Integer;
function MarkSubStringW(const s: TRVUnicodeString;
  Color, BackColor: TColor; IgnoreCase, WholeWords: Boolean; rv: TCustomRichView;
  RVData: TCustomRVData=nil): Integer;
```

All occurrences of **s** are marked with the specified **Color** and **BackColor**. **IgnoreCase** and **WholeWords** specify search options.

This operation is not an editing operation, it cannot be undone.

These functions are not used in RichViewActions. They are included in RichViewActions for a convenience.

1.6.3.3 NormalizeRichView procedure

Fixes possible problems in TCustomRichView component.

Unit RVNormalize.

```
procedure NormalizeRichViewNormalizeRichView(RVData: TCustomRVData;
  FirstItemNo: Integer=0; Recursive: Boolean=True);
```

This procedure does not update undo/redo buffer in editor, so it must be called before the first editing operations after clearing and creating/loading document.

This procedure:

- fixes incorrect "new line" flags
- fixes paragraphs consisting only of list markers without items
- fixes incorrect ParaNo assignment
- removes empty text items (from places where they should not be)
- combines adjacent text items of the same text style and tag

This procedure fixes issues 1-3 and 5-7 listed in the TRichView help topic "Valid Documents"

Example:

```
    NormalizeRichView(rv.RVData);
    rv.Format;
```

This procedure is called, for example, after loading HTML files with rvHtmlImporter 221.

Optional parameters:

FirstItemNo defines the first item to check. The procedure checks items from FirstItemNo to RVData.ItemCount-1. By default, all items are checked.

If **Recursive**=*True*, the procedure checks subdocuments of RVData (table cells).

1.6.3.4 RVChangeCharCase, RVGetCharCase procedures

These procedures allow changing character case in the selected fragment of TCustomRichViewEdit component.

Unit RVCharCase;

```
type
  TRVCharCase = (rvccLowerCase, rvccUpperCase, rvccTitleWord);

procedure RVChangeCharCase(rve: TCustomRichViewEdit; CharCase: TRVCharCase);
procedure RVGetCharCase(rve: TCustomRichViewEdit;
  var AllUpperCase, AllLowerCase: Boolean);
```

RVChangeCharCase changes character case in the selection of **rve** according to **CharCase** parameter.

Option	Meaning
rvccLowerCase	change all characters to lower case
rvccUpperCase	change all characters to upper case
rvccTitleWord	change the first character of each word to upper case, other characters to lower case.

For Unicode text items, this action requires NT-based systems (WinNT4/2000/XP/Vista). In Win9x/ME, Unicode text items are skipped.

RVGetCharCase returns *True* in **AllUpperCase** parameter if all characters in the selection are in upper case.

RVGetCharCase returns *True* in **AllLowerCase** parameter if all characters in the selection are in lower case.

See also:

- TrvActionCharCase 74

1.7 Types

1.7.1 TRVAEditEvent

This type is used for several events.

```
type
  TRVAEditEvent = procedure (Sender: TrvAction;
    Edit: TCustomRichViewEdit) of object;
```

1.7.2 TRVAHFInfo

TRVAHFInfo is the class used for storing properties of a header and a footer in RichViewActions 14.

Unit RichViewActions

Syntax

```
TRVAHFInfo = class(TPersistent);
```

Hierarchy

TObject
TPersistent

Description

This is the type of TRVAControlPanel.Header and Footer 28 properties.

This class has the following properties.

Text: String – text displayed in header/footer. This text may contain the following codes:

- &p – index of the current page (counted from TRVAControlPanel.FirstPageNumber 28);
- &P – count of pages;
- &d – current date (DateToStr)
- &t – current time (DateTimeToString(..., ShortTimeFormat, Time))
- && – '&' character.

Default value: '' (empty string) for footer, '- &p -' for header.

Alignment: TAlignment – horizontal alignment of text. Default value: *taCenter*.

PrintOnFirstPage: Boolean – specifies whether the header/footer is printed on the first page. Default value: *True*.

Font: TRVAFont – font for printing text. Default value: Arial, 10 (where TRVAFont = class(TFont)).

RichViewActions print header and footer by assigning a temporal handler of TRVPrint.OnPagePrepaint event. If TRVPrint component (specified in TRVAControlPanel.RVPrint 31) already has this event assigned, the temporal event handler calls the old handler before drawing text.

1.7.3 TRVASpellString, TRVASpellStrings, TRVASpellStringList

Unit RichViewActions.

If TNT Controls are not used, these strings are Unicode in Delphi 2009 and newer, and ANSI in older versions of Delphi:

```
type
  TRVASpellString = String;
  TRVASpellStrings = TStrings;
  TRVASpellStringList = TStringList;
```

If TNT Controls are used 224, these strings are always Unicode:

```
type
  TRVASpellString = TRVUnicodeString; // WideString
  TRVASpellStrings = TTntStrings;
  TRVASpellStringList = TTntStringList;
```

1.7.4 TrvFileExportFilter, TrvFileSaveFilter

Types for Filter property of "Save As" and "Export" actions.

Unit RichViewActions;

```
type
```

```
  TrvFileExportFilter = (ffeRVF, ffeRTF, ffeXML, ffeTextANSI, ffeTextUnicode,
    ffeCustom, ffeHTMLCSS, ffeHTML, ffeDocX, ffeOfficeConverters);
  TrvFileSaveFilter = ffeRVF..ffeHTMLCSS;
  TrvFileExportFilterSet = set of TrvFileExportFilter;
  TrvFileSaveFilterSet   = set of TrvFileSaveFilter;
```

The declaration above includes the full set of values, but the actual declaration depends on the third-party tools used in the project.

ffeXML is declared only if RichViewXML 223 is used.

ffeHTMLCSS is included in TrvFileSaveFilter only if RvHtmlViewImporter 222 is used.

Value	File Type
ffeRVF	RichView Format (RVF)
ffeRTF	Rich Text Format (RTF)
ffeXML	Format of RichViewXML 223 (XML)
ffeTextANSI	ANSI text files (TXT)
ffeTextUnicode	Unicode (UTF-16) text files (TXT)
ffeCustom	Custom file formats defined in CustomFilter property of the corresponding action
ffeHTMLCSS	HTML files (HTM, HTML) saved by TCustomRichViewEdit.SaveHTMLEx method
ffeHTML	Simplified HTML files (HTM, HTML) saved by TCustomRichViewEdit.SaveHTML method
ffeDocX	Microsoft Word 2007+ Format (DocX)
ffeOfficeConverters	Formats supported by Microsoft Office text export converters

See also:

- TrvActionSaveAs.Filter 74
- TrvActionExport.Filter 63

1.7.5 TrvFileImportFilter, TrvFileOpenFilter

Types for Filter property of "Open" and "Insert File" actions.

Unit RichViewActions;

```
type
  TrvFileImportFilter = (ffiRVF, ffiRTF, ffiXML, ffiTextANSI, ffiTextUnicode,
    ffiTextAuto, ffiCustom, ffiHTML, ffiOfficeConverters);
  TrvFileOpenFilter = ffiRVF..ffiHTML;
  TrvFileImportFilterSet = set of TrvFileImportFilter;
  TrvFileOpenFilterSet   = set of TrvFileOpenFilter;
```

The declaration above includes the full set of values, but the actual declaration depends on the third-party tools used in the project.

ffiXML is declared only if RichViewXML 223 is used.

ffiHTML is declared only if RvHtmlImporter 221 or RvHtmlViewImporter 222 are used.

ffiHTML is included in TrvFileOpenFilter only if RvHtmlViewImporter 222 is used.

Value	File Type
ffiRVF	RichView Format (RVF)
ffiRTF	Rich Text Format (RTF)
ffiXML	Format of RichViewXML 223 (XML)
ffiTextANSI	ANSI text files (TXT)
ffiTextUnicode	Unicode (UTF-16) text files (TXT)
ffiTextAuto	ANSI or Unicode (UTF-16) text files (TXT), autodetected
ffiCustom	Custom file formats defined in CustomFilter property of the corresponding action
ffiHTML	HTML files (HTM, HTML)
ffiOfficeConverters	Formats supported by Microsoft Office text export converters

See also:

- TrvActionOpen.Filter 61
- TrvActionInsertFile.Filter 137

1.7.6 TRVShowFormEvent

This type is used in the following events:

- TrvActionCustomInfoWindow.OnShowing 133
- TrvCustomEditorAction.OnShowing 209
- TrvCustomEditorAction.OnFormCreate 209

```
type
  TRVShowFormEvent = procedure (Sender: TrvAction 195; Form: TForm) of object;
```

1.8 Global Variables

1.8.1 MainRVAControlPanel

The main control panel for RichViewActions.

Unit RichViewActions;

```
var MainRVAControlPanel: TRVAControlPanel 23;
```

If no other control panels are created, this variable returns a default control panel object (not belonging to any form or datamodule).

If an application contains a single TRVControlPanel on a form, it becomes the main control panel automatically, and MainRVAControlPanel returns this control panel.

Another control panel can be made a main one if you call its Activate 36 method.

1.8.2 RVA_EditForceDefControl

Specifies a priority of non-TCustomRichViewEdit editors for editing actions.

```
var
  RVA_EditForceDefControl: Boolean = False;
```

By default, focused non-TCustomRichViewEdit controls have higher priority in editing actions 84 (such as Cut, Copy, Paste) than Control 195 or GetControlPanel 208.DefaultControl 26 properties.

For example, in the ActionTest demo, if you select text in the font name combo box and press Ctrl+X, the selected text from this combo box will be cut to the Clipboard, the action ignores RVAControlPanel1.DefaultControl[26]=RichViewEdit1.

If you want to change this priority, assign RVA_EditForceDefControl := True.

1.8.3 RVA_EditorControlFunction

This variable refers to the function allowing for the actions inherited from TrvCustomEditAction[84] to work with additional editor types.

Unit RichViewActions;

```
type
  TRVAEditorControlCommand = (rvaeccIsEditorControl, rvaeccHasSelection,
    rvaeccCanPaste, rvaeccCanPasteText,
    rvaeccCanUndo, rvaeccCopy, rvaeccCut,
    rvaeccPaste, rvaeccPasteText, rvaeccUndo);
  TRVAEditorControlCommandFunction =
    function (Control: TControl; Command: TRVAEditorControlCommand): Boolean;
var
    RVA_EditorControlFunction: TRVAEditorControlCommandFunction;
```

By default, this variable is initialized with RVA_EditorControlFunctionDef[229] function.

Command	The function must...
rvaeccIsEditorControl	... return True if Control is a supported editor control.
rvaeccHasSelection	... return True if the selection in Control is not empty
rvaeccCanPaste	... return True if Control can paste data from the Clipboard.
rvaeccCanPasteText	... return True if Control can paste text from the Clipboard.
rvaeccCanUndo	... return True if Control can undo the last editing operation
rvaeccCopy	... copy the selection from Control to the Clipboard
rvaeccCut	... cut the selection from Control to the Clipboard
rvaeccPaste	... paste data from the Clipboard into Control
rvaeccPasteText	... paste text from the Clipboard into Control
rvaeccUndo	... undo the last editing operation in Control

1.8.4 ShowUntranslatedControls

This variable allows hiding controls in RichViewActions dialog that were not translated yet.

Unit RichViewActions;

```
var
  ShowUntranslatedControls: Boolean = True;
```

1.9 How to

1.9.1 How to change RVF format name

By default, RVF (RichView Format) is the main file format of RichViewActions.

If you want to change file extension and the format name, you can do it in the following way.

Place RVAControlPanel1: TRVAControlPanel 23 component on the form. Change the following properties (in the code or in the Object Inspector):

```
RVAControlPanel1.RVFLocalizable 30 := False;
RVAControlPanel1.RVFFilter 30 := 'SuperApp Files (*.saf)|*.saf';
RVAControlPanel1.DefaultExt 27 := 'saf';
RVAControlPanel1.RVFormatTitle 31 := 'SuperApp Format';
```

Additionally, you can change names of files used by TrvActionStyleTemplates 129 to save and load style templates:

```
RVAControlPanel1.RVStylesFilter 32 := 'SuperApp Styles (*.sas)|*.sas';
RVAControlPanel1.RVStylesExt 32 := 'sas';
```

Now, if you want to localize these properties 210, you need to assign them yourself every time when the application language is changed.

Index

- A -

ActionEditNote 161
TrvActionInsertNote 161
ActionFind 77, 82
TrvActionFindNext 77
TrvActionReplace 82
ActionFont 44
TCustomRVFontComboBox 44
ActionInsertHLine 202
TrvActionItemProperties 202
ActionInsertHyperlink 204
TrvActionRemoveHyperlinks 204
ActionInsertTable 188, 202
TrvActionItemProperties 202
TrvActionTableProperties 188
ActionList 41
TRVAPopupMenu 41
TRVASPTBXPopupMenu 41
TRVATBPopupMenu 41
TRVATBXPopupMenu 41
ActionNew 60
TrvActionOpen 60
ActionPageSetup 67
TrvActionPrintPreview 67
ActionParaBullets 126
TrvActionParaList 126
ActionParaNumbering 126
TrvActionParaList 126
ActionPrint 67
TrvActionPrintPreview 67
ActionReplace 76
TrvActionFind 76
ActionSave 57, 73
TrvActionCustomNew 57
TrvActionSaveAs 73
ActionSaveAs 69
TrvActionSave 69
ActionsEnabled 25
TRVAControlPanel 25
ActionStyleTemplates 132, 140
TrvActionAddStyleTemplate 132
TrvActionInsertHyperlink 140
Activate 36
TRVAControlPanel 36
AddColorNameToHints 25
TRVAControlPanel 25
AddDefaultCharset 45
TRVFontCharsetComboBox 45
AddFormat 80
TrvActionPasteSpecial 80
Addict 3 219
Alignment 120, 122
TrvActionParagraph 120, 122
AllowApplyingTo 200
TrvActionFillColor 200
AllowMultiple 186
TrvActionTableInsertRowsBelow 186
AllStylesImageIndex 48
TRVStyleTemplateComboBox 48
TRVStyleTemplateListBox 48
AlwaysInsertUnicode 154
TrvActionInsertSymbol 154
AutoAddHyperlinkStyleTemplate 140
TrvActionInsertHyperlink 140
AutoApplySymbolCharset 88
TrvActionFontEx 88
AutoDeleteUnusedStyles 25
TRVAControlPanel 25
AutoUpdateFileName 55
TrvActionCustomIO 55

- B -

BackColor 88, 140
TrvActionFontEx 88
TrvActionInsertHyperlink 140
Background 116
TrvActionParaBorder 116
BackgroundColor 151
TrvActionInsertPicture 151
BackgroundGraphicFilter 188, 202
TrvActionItemProperties 202
TrvActionTableProperties 188
BackgroundPicture 164
TrvActionInsertTable 164
BackgroundStyle 164
TrvActionInsertTable 164
BestWidth 164
TrvActionInsertTable 164
Border 116
TrvActionParaBorder 116
BorderColor 151, 164
TrvActionInsertPicture 151
TrvActionInsertTable 164

BorderHSpacing 164
TrvActionInsertTable 164
BorderLightColor 165
TrvActionInsertTable 165
BorderStyle 165
TrvActionInsertTable 165
BorderVSpacing 165
TrvActionInsertTable 165
BorderWidth 151, 165
TrvActionInsertPicture 151
TrvActionInsertTable 165
BoxPosition 156
TrvActionCustomInsertSidenote 156
BoxProperties 157
TrvActionCustomInsertSidenote 157

- C -

CallerControl 198
TrvActionCustomColor 198
CanChangeMargins 196
TrvActionBackground 196
CanCloseDoc 71
TrvActionSave 71
Caption 207
TrvCustomAction 207
CellBorderColor 165
TrvActionInsertTable 165
CellBorderLightColor 166
TrvActionInsertTable 166
CellBorderStyle 166
TrvActionInsertTable 166
CellBorderWidth 166
TrvActionInsertTable 166
CellHPadding 166
TrvActionInsertTable 166
CellHSpacing 167
TrvActionInsertTable 167
CellPadding 166
TrvActionInsertTable 166
CellVPadding 166
TrvActionInsertTable 166
CellVSpacing 167
TrvActionInsertTable 167
CharScale 89
TrvActionFontEx 89
CharSpacing 89
TrvActionFontEx 89
ClearFormatImageIndex 48
TRVStyleTemplateComboBox 48
TRVStyleTemplateListBox 48
CleverComponents 221
ClHTTP 25
TRVAControlPanel 25
CloseDialog 77, 82
TrvActionFind 77
TrvActionReplace 82
ColBandSize 167
TrvActionInsertTable 167
ColCount 167
TrvActionInsertTable 167
Color 138, 140, 168, 198
TrvActionCustomColor 198
TrvActionInsertHLine 138
TrvActionInsertHyperlink 140
TrvActionInsertTable 168
ColorDialog 25
TRVAControlPanel 25
Control 195
TrvAction 195
ControlPanel 48, 207
TrvCustomAction 207
TRVStyleTemplateComboBox 48
TRVStyleTemplateListBox 48
ConvertToUnicode 228
CreateDirectoryForHTMLImages 63, 69
TrvActionExport 63
TrvActionSave 69
Cursor 140
TrvActionInsertHyperlink 140
CustomFilter 54, 60
TrvActionCustomFileIO 54
TrvActionOpen 60

- D -

DefaultCharsetCaption 45
TRVFontCharsetComboBox 45
DefaultColor 26
TRVAControlPanel 26
DefaultControl 26
TRVAControlPanel 26
DefaultCustomFilterIndex 26
TRVAControlPanel 26
DefaultDocParameters 26
TRVAControlPanel 26
DefaultExt 27, 151
TRVAControlPanel 27
TrvActionInsertPicture 151
DefaultFileFormat 27

DefaultFileFormat 27
TRVAControlPanel 27
DefaultFileName 27
TRVAControlPanel 27
DefaultMargin 27
TRVAControlPanel 27
DeleteAllTabs 121
TrvActionParagraph 121
DetectURL 143
TrvActionInsertHyperlink 143
DialogFontName 28
TRVAControlPanel 28
DialogFontSize 28
TRVAControlPanel 28
DialogTitle 55, 60
TrvActionCustomIO 55
TrvActionOpen 60
Disabled 207
TrvCustomAction 207
DisableWhenUnmodified 70
TrvActionSave 70
DisplayUnicodeBlocks 154
TrvActionInsertSymbol 154
Documents 70
TrvActionSave 70

- E -

Editor 44, 48
TCustomRVFontComboBox 44
TRVStyleTemplateComboBox 48
TRVStyleTemplateListBox 48
EncodeTarget 144
TrvActionInsertHyperlink 144
EvenColumnsColor 171
TrvActionInsertTable 171
EvenRowsColor 171
TrvActionInsertTable 171
ExcludeLabel 135
TrvActionInsertCaption 135
ExportToFile 64
TrvActionExport 64

- F -

FamFamFam Silk Icons toolbar images 227
FileName 55
TrvActionCustomIO 55
FileTitle 63, 70
TrvActionExport 63
TrvActionSave 70
FillActionClientContents 230
Filter 61, 63, 74, 137, 151
TrvActionExport 63
TrvActionInsertFile 137
TrvActionInsertPicture 151
TrvActionOpen 61
TrvActionSaveAs 74
FindDoc 72
TrvActionSave 72
FirstColumnColor 171
TrvActionInsertTable 171
FirstIndent 121
TrvActionParagraph 121
FirstPageNumber 28
TRVAControlPanel 28
Font 94, 161
TrvActionFonts 94
TrvActionInsertNote 161
FontImageIndex 133
TrvActionStyleInspector 133
FontName 45, 47
TRVFontCharsetComboBox 45
TRVFontSizeComboBox 47
FontSizeDouble 91
TrvActionFontEx 91
FontStyleEx 89
TrvActionFontEx 89
Footer 28
TRVAControlPanel 28
Form 208
TrvCustomEditorAction 208
Fugue Icons toolbar images 227

- G -

GetAddictSpellLanguage 218
GetAddictThesLanguage 218
GetControlPanel 208
TrvCustomAction 208
GetHyperlinkStyleNo 144
TrvActionInsertHyperlink 144
GetNormalStyleNo 145
TrvActionInsertHyperlink 145
GlyFX toolbar images 225
Glyfz toolbar images 226
GoToCheckpoint 228
GoToLink 145
TrvActionInsertHyperlink 145
GraphicFilter 202

GraphicFilter 202
TrvActionItemProperties 202

- H -

Header 28
TRVAControlPanel 28
HeadingRowColor 171
TrvActionInsertTable 171
HeadingRowCount 168
TrvActionInsertTable 168
Hint 207
TrvCustomAction 207
HOutermostRule 168
TrvActionInsertTable 168
HoverBackColor 141
TrvActionInsertHyperlink 141
HoverColor 141
TrvActionInsertHyperlink 141
HoverEffects 141
TrvActionInsertHyperlink 141
HoverUnderlineColor 141
TrvActionInsertHyperlink 141
How to
change RVF format name 237
How to change RVF format name 237
HRuleColor 168
TrvActionInsertTable 168
HRuleWidth 168
TrvActionInsertTable 168

- I -

IdHTTP 29
TRVAControlPanel 29
If 104
ImageFileName 196
TrvActionBackground 196
ImagePrefix 63, 70
TrvActionExport 63
TrvActionSave 70
Images 48, 130, 132
TrvActionStyleInspector 132
TrvActionStyleTemplates 130
TRVStyleTemplateComboBox 48
TRVStyleTemplateListBox 48
IndentMax 113
TrvActionIndentInc 113
IndentStep 110, 111, 125
TrvActionCustomParaListSwitcher 110
TrvActionIndent 111
TrvActionParaList 125
Indy 220
InitialDir 56, 61
TrvActionCustomIO 56
TrvActionOpen 61
InsertAbove 135
TrvActionInsertCaption 135
InsertFile 137
TrvActionInsertFile 137
Installing RichViewActions 14
ItemText 169
TrvActionInsertTable 169

- J -

JumpToNextNote 158
TrvActionEditNote 158

- K -

KeepLinesTogether 121
TrvActionParagraph 121
KeepWithNext 121
TrvActionParagraph 121

- L -

Language 29
TRVAControlPanel 29
LastColumnColor 171
TrvActionInsertTable 171
LastFilterIndex 61
TrvActionOpen 61
LastLineAlignment 104, 122
TrvActionAlignCustomJustify 104
TrvActionParagraph 122
LastRowColor 171
TrvActionInsertTable 171
LeftIndent 122
TrvActionParagraph 122
LineSpacing 122
TrvActionParagraph 122
LineSpacingType 122
TrvActionParagraph 122
ListLevels 110
TrvActionCustomParaListSwitcher 110
LoadCSV 230
LoadFile 61
TrvActionOpen 61

Localization of RichViewActions 210
Localize 49
TRVStyleTemplateComboBox 49
TRVStyleTemplateListBox 49
LocalizeRibbonPage 230
LostFormatWarning 71
TrvActionSave 71

- M -

MainRVAControlPanel 235
MarginsUnits 65
TrvActionPageSetup 65
MarkSubStringA 231
MarkSubStringW 231
MaxImageSize 152
TrvActionInsertPicture 152
Maximized 67
TrvActionPrintPreview 67
MaxSize 91, 92
TrvActionFontGrow 91
TrvActionFontGrowOnePoint 92
MaxSuggestionsCount 41
TRVAPopupMenu 41
TRVASPTBXPopupMenu 41
TRVATBPopupMenu 41
TRVATBXPopupMenu 41
MetafileCompatibility 29
TRVAControlPanel 29
MinSize 95, 96
TrvActionFontShrink 95
TrvActionFontShrinkOnePoint 96

- N -

NormalizeRichView 231
NumberType 136, 148
TrvActionInsertCustomPageNumber 136
TrvActionInsertNumSequence 148

- O -

OddColumnsColor 171
TrvActionInsertTable 171
OddRowsColor 171
TrvActionInsertTable 171
OnAddStyle 36
TRVAControlPanel 36
OnApplyHyperlinkToItem 146
TrvActionInsertHyperlink 146
OnBackgroundChange 36
TRVAControlPanel 36
OnCanApply 203
TrvActionItemProperties 203
OnCanPaste 80
TrvActionPasteSpecial 80
OnChange 65, 197
TrvActionBackground 197
TrvActionPageSetup 65
OnClickAllStyles 50
TRVStyleTemplateComboBox 50
TRVStyleTemplateListBox 50
OnCloseTableSizeDialog 171
TrvActionInsertTable 171
OnCreateTeForm 36
TRVAControlPanel 36
OnCustomFileOperation 37
TRVAControlPanel 37
OnCustomItemPropertiesDialog 203
TrvActionItemProperties 203
OnCustomPaste 81
TrvActionPasteSpecial 81
OnDocumentFileChange 72
TrvActionSave 72
OnDownload 38
TRVAControlPanel 38
OnExecute 199
TrvActionEvent 199
OnFormCreate 209
TrvCustomEditorAction 209
OnGetActionControlCoords 38
TRVAControlPanel 38
OnGetHyperlinkTargetFromItem 146
TrvActionInsertHyperlink 146
OnGetPreviewFormClass 68
TrvActionPrintPreview 68
OnHide 209
TrvCustomEditorAction 209
OnHideColorPicker 199
TrvActionCustomColor 199
OnHyperlinkForm 147
TrvActionInsertHyperlink 147
OnInserting 171
TrvActionInsertTable 171
OnInsertText 155
TrvActionInsertText 155
OnLiveSpellAdd 42
TRVAPopupMenu 42
TRVASPTBXPopupMenu 42
TRVATBPopupMenu 42

OnLiveSpellAdd 42
TRVATBXPopupMenu 42
OnLiveSpellGetSuggestions 42
TRVAPopupMenu 42
TRVASPTBXPopupMenu 42
TRVATBPopupMenu 42
TRVATBXPopupMenu 42
OnLiveSpellIgnoreAll 43
TRVAPopupMenu 43
TRVASPTBXPopupMenu 43
TRVATBPopupMenu 43
TRVATBXPopupMenu 43
OnLiveSpellWordReplace 43
TRVAPopupMenu 43
TRVASPTBXPopupMenu 43
TRVATBPopupMenu 43
TRVATBXPopupMenu 43
OnMarginsChanged 39
TRVAControlPanel 39
OnNew 57
TrvActionCustomNew 57
OnOpenFile 62
TrvActionOpen 62
OnReplaceAllEnd 82
TrvActionReplace 82
OnReplaceAllStart 83
TrvActionReplace 83
OnReplacing 83
TrvActionReplace 83
OnSaving 72
TrvActionSave 72
OnShowColorPicker 199
TrvActionCustomColor 199
OnShowing 81, 133, 209
TrvActionCustomInfoWindow 133
TrvActionPasteSpecial 81
TrvCustomEditorAction 209
OnStyleNeeded 39
TRVAControlPanel 39
OnUpdate 200
TrvActionEvent 200
OuterHSpacing 152
TrvActionInsertPicture 152
OuterVSpacing 152
TrvActionInsertPicture 152
OutlineLevel 123
TrvActionParagraph 123

- P -

Padding 119
TrvActionParaColorAndPadding 119
ParaImageIndex 133
TrvActionStyleInspector 133
ParaStyleImageIndex 48, 131
TrvActionStyleTemplates 131
TRVStyleTemplateComboBox 48
TRVStyleTemplateListBox 48
ParaTextStyleImageIndex 48, 131
TrvActionStyleTemplates 131
TRVStyleTemplateComboBox 48
TRVStyleTemplateListBox 48
PasteHTML 228
Percent 96
TrvActionFontShrinkGrow 96
PixelBorders 29
TRVAControlPanel 29

- R -

RefStyleTemplateName 162
TrvActionInsertSidenote 162
RemoveEllipsis 230
RemoveEllipsisFromRibbon 230
Reset 59
TrvActionNew 59
RichViewActions
downloading images 220, 221
FamFamFam Silk Icons toolbar images 227
Fugue Icons toolbar images 227
GlyFX toolbar images 225
Glyfz toolbar images 226
history 18
inserting and pasting HTML 221
installing 14
localizing 210
opening-inserting-saving HTML 222
opening-inserting-saving XML 223
Style Templates 17
ThemeEngine 223
Third-party tools 219
TNT Controls 224
TRichView toolbar images 225
using 14
using Addict 3 components 219
using SpTBXLib 223
using TBX 224

RichViewActions
using Toolbar 2000 224
RichViewActions overview 14
RichViewActions.Overview 14
RichViewXML 223
RightIndent 123
TrvActionParagraph 123
RowBandSize 167
TrvActionInsertTable 167
RowCount 169
TrvActionInsertTable 169
RowsKeepTogether 169
TrvActionInsertTable 169
RowsVAlign 169
TrvActionInsertTable 169
RVA_Addict3AutoCorrect 228
RVA_ChooseLanguage 217
RVA_ChooseStyle 229
RVA_ConvertToPixels 229
RVA_ConvertToTwips 229
RVA_EditForceDefControl 235
RVA_EditorControlFunction 236
RVA_EditorControlFunctionDef 229
RVA_EnumLanguages 212
RVA_FillLanguageList 213
RVA_GetCharset 213
RVA_GetColorName 218
RVA_GetHelpFile 213
RVA_GetLanguageName 214
RVA_GetPC 214
RVA_GetPrintingMessage 215
RVA_GetProgressMessage 215
RVA_GetS 214
RVA_GetSH 214
RVA_LocalizeForm 218
RVA_SwitchLanguage 214
RVA_TranslateUnits 215
RVAAddictLanguages unit 218
RVAddictSpell3 30
TRVAControlPanel 30
RVAddictThesaurus3 30
TRVAControlPanel 30
RVALocalize 212
RVALocalizeRuler 219
RVALocRuler unit 219
RVARibbonUtils unit 230
RVChangeCharCase 232
RVCharCase 232
rvcsv unit 230
RVFFilter 30
TRVAControlPanel 30
RVFLocalizable 30
TRVAControlPanel 30
RVFormatTitle 31
TRVAControlPanel 31
RVGetCharCase 232
RvHtmlHelp 229
RVHTMLImporter 31, 221
TRVAControlPanel 31
RVHTMLViewImporter 31, 222
TRVAControlPanel 31
RVNormalize 231
RVPrint 31
TRVAControlPanel 31
RVStylesExt 32
TRVAControlPanel 32
RVStylesFilter 32
TRVAControlPanel 32
RVXML 32
TRVAControlPanel 32

- S -

SaveOptions 64, 71
TrvActionExport 64
TrvActionSave 71
SearchScope 32
TRVAControlPanel 32
SeqName 148
TrvActionInsertNumSequence 148
SetRulerColors 230
SetTags 145
TrvActionInsertHyperlink 145
SetToPictures 142
TrvActionInsertHyperlink 142
ShowAllStyles 49
TRVStyleTemplateComboBox 49
TRVStyleTemplateListBox 49
ShowClearFormat 49
TRVStyleTemplateComboBox 49
TRVStyleTemplateListBox 49
ShowReplaceAllSummary 82
TrvActionReplace 82
ShowSoftPageBreaks 32
TRVAControlPanel 32
ShowTableSizeDialog 171
TrvActionInsertTable 171
ShowUntranslatedControls 236
SmartHeadings 49

SmartHeadings 49
TRVStyleTemplateComboBox 49
TRVStyleTemplateListBox 49
SpaceAfter 123
TrvActionParagraph 123
SpaceBefore 123
TrvActionParagraph 123
SpaceFiller 142
TrvActionInsertHyperlink 142
Spacing 152
TrvActionInsertPicture 152
SpTBXLib 223
StandardStyleTemplates 130
TrvActionStyleTemplates 130
StoreFileName 79, 152
TrvActionInsertPicture 152
TrvActionPasteSpecial 79
StoreFileNameInItemName 80, 153
TrvActionInsertPicture 153
TrvActionPasteSpecial 80
StoreImageFileName 188, 203
TrvActionItemProperties 203
TrvActionTableProperties 188
Style 138, 142
TrvActionInsertHLine 138
TrvActionInsertHyperlink 142
Style Templates in RichViewActions 17
StyleTemplateName 142, 157
TrvActionCustomInsertSidenote 157
TrvActionInsertHyperlink 142
StyleTemplates 58
TrvActionNew 58
SubDocEditor 208
TrvCustomEditorAction 208
SubSuperScriptType 89
TrvActionFontEx 89
SuppressNextErrorMessage 71
TrvActionSave 71
SymbolType 154
TrvActionInsertSymbol 154

- T -

TableGridStyle 33
TRVAControlPanel 33
TableOptions 169
TrvActionInsertTable 169
TablePrintOptions 170
TrvActionInsertTable 170
Tabs 124
TrvActionParagraph 124
TabsToDelete 124
TrvActionParagraph 124
TBX 224
TCustomRVFontComboBox 43
TCustomRVFontComboBox.ActionFont 44
TCustomRVFontComboBox.Editor 44
TerminateHyperlink 146
TrvActionInsertHyperlink 146
TextStyleImageIndex 48, 131
TrvActionStyleTemplates 131
TRVStyleTemplateComboBox 48
TRVStyleTemplateListBox 48
ThemeEngine 223
Title 66, 68
TrvActionPrint 66
TrvActionQuickPrint 68
TNT Controls 224
Toolbar 2000 224
TRichView toolbar images 225
TRVA2LiveSpellAddEvent type 42
TRVA2LiveSpellGetSuggestionsEvent type 42
TRVA2LiveSpellIgnoreAllEvent 43
TRVA2LiveSpellReplaceEvent type 43
TRVABoxPosition 156
TRVABoxProperties 157
TRVACanPasteEvent 80
TrvaColorInterface 198
TRVAControlPanel 23
TRVAControlPanel.ActionsEnabled 25
TRVAControlPanel.Activate 36
TRVAControlPanel.AddColorNameToHints 25
TRVAControlPanel.AutoDeleteUnusedStyles 25
TRVAControlPanel.CIHTTP 25
TRVAControlPanel.ColorDialog 25
TRVAControlPanel.DefaultColor 26
TRVAControlPanel.DefaultControl 26
TRVAControlPanel.DefaultCustomFilterIndex 26
TRVAControlPanel.DefaultDocParameters 26
TRVAControlPanel.DefaultExt 27
TRVAControlPanel.DefaultFileFormat 27
TRVAControlPanel.DefaultFileName 27
TRVAControlPanel.DefaultMargin 27
TRVAControlPanel.DialogFontName 28
TRVAControlPanel.DialogFontSize 28
TRVAControlPanel.FirstPageNumber 28
TRVAControlPanel.Footer 28
TRVAControlPanel.Header 28
TRVAControlPanel.IdHTTP 29

TRVAControlPanel.Language 29
TRVAControlPanel.MetafileCompatibility 29
TRVAControlPanel.OnAddStyle 36
TRVAControlPanel.OnBackgroundChange 36
TRVAControlPanel.OnCreateTeForm 36
TRVAControlPanel.OnCustomFileOperation 37
TRVAControlPanel.OnDownload 38
TRVAControlPanel.OnGetActionControlCoords 38
TRVAControlPanel.OnMarginsChanged 39
TRVAControlPanel.OnStyleNeeded 39
TRVAControlPanel.PixelBorders 29
TRVAControlPanel.RVAddictSpell3 30
TRVAControlPanel.RVAddictThesaurus3 30
TRVAControlPanel.RVFFilter 30
TRVAControlPanel.RVFLocalizable 30
TRVAControlPanel.RVFormatTitle 31
TRVAControlPanel.RVHTMLImporter 31
TRVAControlPanel.RVHTMLViewImporter 31
TRVAControlPanel.RVPrint 31
TRVAControlPanel.RVStylesExt 32
TRVAControlPanel.RVStylesFilter 32
TRVAControlPanel.RVXML 32
TRVAControlPanel.SearchScope 32
TRVAControlPanel.ShowSoftPageBreaks 32
TRVAControlPanel.TableGridStyle 33
TRVAControlPanel.UnitsDisplay 33
TRVAControlPanel.UnitsProgram 33
TRVAControlPanel.UseHelpFiles 34
TRVAControlPanel.UserInterface 34
TRVAControlPanel.UseTextCodePageDialog 34
TRVAControlPanel.UseXPThemes 35
TRVAControlPanel.XMLFilter 35
TRVAControlPanel.XMLLocalizable 35
TrvAction 195
TrvAction.Control 195
TrvActionAddictSpell3 194
TrvActionAddictThesaurus3 194
TrvActionAddStyleTemplate 131
TrvActionAddStyleTemplate.ActionStyleTemplates 132
TrvActionAlignCenter 103
TrvActionAlignCustomJustify 104
TrvActionAlignCustomJustify.LastLineAlignment 104
TrvActionAlignCustomJustify.UseLastLineAlignment 104
TrvActionAlignDistribute 104
TrvActionAlignJustify 105
TrvActionAlignLeft 106
TrvActionAlignment 106
TrvActionAlignRight 107
TrvActionBackground 195
TrvActionBackground.CanChangeMargins 196
TrvActionBackground.ImageFileName 196
TrvActionBackground.OnChange 197
TrvActionCharCase 74
TrvActionClearBoth 107
TrvActionClearFormat 133
TrvActionClearLeft 108
TrvActionClearNone 108
TrvActionClearRight 109
TrvActionClearTextFlow 109
TrvActionClearTextFormat 134
TrvActionColor 197
TrvActionCopy 75
TrvActionCustomColor 197
TrvActionCustomColor.CallerControl 198
TrvActionCustomColor.Color 198
TrvActionCustomColor.OnHideColorPicker 199
TrvActionCustomColor.OnShowColorPicker 199
TrvActionCustomColor.UserInterface 198
TrvActionCustomFileIO 54
TrvActionCustomFileIO.CustomFilter 54
TrvActionCustomInfoWindow 132
TrvActionCustomInfoWindow.OnShowing 133
TrvActionCustomInsertSidenote 156
TrvActionCustomInsertSidenote.BoxPosition 156
TrvActionCustomInsertSidenote.BoxProperties 157
TrvActionCustomInsertSidenote.StyleTemplateName 157
TrvActionCustomIO 54
TrvActionCustomIO.AutoUpdateFileName 55
TrvActionCustomIO.DialogTitle 55
TrvActionCustomIO.FileName 55
TrvActionCustomIO.InitialDir 56
TrvActionCustomNew 56
TrvActionCustomNew.ActionSave 57
TrvActionCustomNew.OnNew 57
TrvActionCustomParaListSwitcher 110
TrvActionCustomParaListSwitcher.IndentStep 110
TrvActionCustomParaListSwitcher.ListLevels 110
TrvActionCut 75
TrvActionEditNote 157
TrvActionEditNote.JumpToNextNote 158
TrvActionEvent 199
TrvActionEvent.OnExecute 199
TrvActionEvent.OnUpdate 200
TrvActionExport 62
TrvActionExport.CreateDirectoryForHTMLImages 63

TrvActionExport.ExportToFile 64
TrvActionExport.FileTitle 63
TrvActionExport.Filter 63
TrvActionExport.ImagePrefix 63
TrvActionExport.SaveOptions 64
TrvActionFillColor 200
TrvActionFillColor.AllowApplyingTo 200
TrvActionFind 76
TrvActionFind.ActionReplace 76
TrvActionFind.CloseDialog 77
TrvActionFindNext 77
TrvActionFindNext.ActionFind 77
TrvActionFontAllCaps 85
TrvActionFontBackColor 85
TrvActionFontBold 86
TrvActionFontColor 86
TrvActionFontCustomColor 87
TrvActionFontEx 87
TrvActionFontEx.AutoApplySymbolCharset 88
TrvActionFontEx.BackColor 88
TrvActionFontEx.CharScale 89
TrvActionFontEx.CharSpacing 89
TrvActionFontEx.FontSizeDouble 91
TrvActionFontEx.FontStyleEx 89
TrvActionFontEx.SubSuperScriptType 89
TrvActionFontEx.UnderlineColor 90
TrvActionFontEx.UnderlineType 90
TrvActionFontEx.ValidProperties 90
TrvActionFontEx.VShift 90
TrvActionFontGrow 91
TrvActionFontGrow.MaxSize 91
TrvActionFontGrowOnePoint 92
TrvActionFontGrowOnePoint.MaxSize 92
TrvActionFontItalic 92
TrvActionFontOverline 93
TrvActionFonts 93
TrvActionFonts.Font 94
TrvActionFonts.UserInterface 94
TrvActionFontShrink 94
TrvActionFontShrink.MinSize 95
TrvActionFontShrinkGrow 95
TrvActionFontShrinkGrow.Percent 96
TrvActionFontShrinkOnePoint 96
TrvActionFontShrinkOnePoint.MinSize 96
TrvActionFontStrikeout 96
TrvActionFontStyle 97
TrvActionFontStyleEx 98
TrvActionFontUnderline 98
TrvActionHide 201
TrvActionIndent 111
TrvActionIndent.IndentStep 111
TrvActionIndentDec 112
TrvActionIndentInc 112
TrvActionIndentInc.IndentMax 113
TrvActionInsertCaption 134
TrvActionInsertCaption.ExcludeLabel 135
TrvActionInsertCaption.InsertAbove 135
TrvActionInsertCustomPageNumber 136
TrvActionInsertCustomPageNumber.NumberType 136
TrvActionInsertEndnote 159
TrvActionInsertFile 136
TrvActionInsertFile.Filter 137
TrvActionInsertFile.InsertFile 137
TrvActionInsertFootnote 159
TrvActionInsertHLine 137
TrvActionInsertHLine.Color 138
TrvActionInsertHLine.Style 138
TrvActionInsertHLine.Width 138
TrvActionInsertHyperlink 138
TrvActionInsertHyperlink.ActionStyleTemplates 140
TrvActionInsertHyperlink.AutoAddHyperlinkStyleTemplate 140
TrvActionInsertHyperlink.BackColor 140
TrvActionInsertHyperlink.Color 140
TrvActionInsertHyperlink.Cursor 140
TrvActionInsertHyperlink.DetectURL 143
TrvActionInsertHyperlink.EncodeTarget 144
TrvActionInsertHyperlink.GetHyperlinkStyleNo 144
TrvActionInsertHyperlink.GetNormalStyleNo 145
TrvActionInsertHyperlink.GoToLink 145
TrvActionInsertHyperlink.HoverBackColor 141
TrvActionInsertHyperlink.HoverColor 141
TrvActionInsertHyperlink.HoverEffects 141
TrvActionInsertHyperlink.HoverUnderlineColor 141
TrvActionInsertHyperlink.OnApplyHyperlinkToItem 146
TrvActionInsertHyperlink.OnGetHyperlinkTargetFromItem 146
TrvActionInsertHyperlink.OnHyperlinkForm 147
TrvActionInsertHyperlink.SetTags 145
TrvActionInsertHyperlink.SetToPictures 142
TrvActionInsertHyperlink.SpaceFiller 142
TrvActionInsertHyperlink.Style 142
TrvActionInsertHyperlink.StyleTemplateName 142
TrvActionInsertHyperlink.TerminateHyperlink 146
TrvActionInsertHyperlink.UnderlineColor 143
TrvActionInsertHyperlink.ValidProperties 143

TrvActionInsertNote 160
TrvActionInsertNote.ActionEditNote 161
TrvActionInsertNote.Font 161
TrvActionInsertNumber 147
TrvActionInsertNumSequence 148
TrvActionInsertNumSequence.NumberType 148
TrvActionInsertNumSequence.SeqName 148
TrvActionInsertNumSequence.UseDefaults 149
TrvActionInsertPageBreak 149
TrvActionInsertPageCount 149
TrvActionInsertPageNumber 150
TrvActionInsertPicture 150
TrvActionInsertPicture.BackgroundColor 151
TrvActionInsertPicture.BorderColor 151
TrvActionInsertPicture.BorderWidth 151
TrvActionInsertPicture.DefaultExt 151
TrvActionInsertPicture.Filter 151
TrvActionInsertPicture.MaxImageSize 152
TrvActionInsertPicture.OuterHSpacing 152
TrvActionInsertPicture.OuterVSpacing 152
TrvActionInsertPicture.Spacing 152
TrvActionInsertPicture.StoreFileName 152
TrvActionInsertPicture.StoreFileNameInItemName 153
TrvActionInsertPicture.VAlign 153
TrvActionInsertSidenote 161
TrvActionInsertSidenote.RefStyleTemplateName 162
TrvActionInsertSymbol 153
TrvActionInsertSymbol.AlwaysInsertUnicode 154
TrvActionInsertSymbol.DisplayUnicodeBlocks 154
TrvActionInsertSymbol.SymbolType 154
TrvActionInsertSymbol.UseCurrentFont 155
TrvActionInsertTable 163
TrvActionInsertTable.BackgroundPicture 164
TrvActionInsertTable.BackgroundStyle 164
TrvActionInsertTable.BestWidth 164
TrvActionInsertTable.BorderColor 164
TrvActionInsertTable.BorderHSpacing 164
TrvActionInsertTable.BorderLightColor 165
TrvActionInsertTable.BorderStyle 165
TrvActionInsertTable.BorderVSpacing 165
TrvActionInsertTable.BorderWidth 165
TrvActionInsertTable.CellBorderColor 165
TrvActionInsertTable.CellBorderLightColor 166
TrvActionInsertTable.CellBorderStyle 166
TrvActionInsertTable.CellBorderWidth 166
TrvActionInsertTable.CellHPadding 166
TrvActionInsertTable.CellHSpacing 167
TrvActionInsertTable.CellPadding 166
TrvActionInsertTable.CellVPadding 166
TrvActionInsertTable.CellVSpacing 167
TrvActionInsertTable.ColBandSize 167
TrvActionInsertTable.ColCount 167
TrvActionInsertTable.Color 168
TrvActionInsertTable.EvenColumnsColor 171
TrvActionInsertTable.EvenRowsColor 171
TrvActionInsertTable.FirstColumnColor 171
TrvActionInsertTable.HeadingRowColor 171
TrvActionInsertTable.HeadingRowCount 168
TrvActionInsertTable.HOutermostRule 168
TrvActionInsertTable.HRuleColor 168
TrvActionInsertTable.HRuleWidth 168
TrvActionInsertTable.ItemText 169
TrvActionInsertTable.LastColumnColor 171
TrvActionInsertTable.LastRowColor 171
TrvActionInsertTable.OddColumnsColor 171
TrvActionInsertTable.OddRowsColor 171
TrvActionInsertTable.OnCloseTableSizeDialog 171
TrvActionInsertTable.OnInserting 171
TrvActionInsertTable.RowBandSize 167
TrvActionInsertTable.RowCount 169
TrvActionInsertTable.RowsKeepTogether 169
TrvActionInsertTable.RowsVAlign 169
TrvActionInsertTable.ShowTableSizeDialog 171
TrvActionInsertTable.TableOptions 169
TrvActionInsertTable.TablePrintOptions 170
TrvActionInsertTable.VisibleBorders 170
TrvActionInsertTable.VOutermostRule 170
TrvActionInsertTable.VRuleColor 170
TrvActionInsertTable.VRuleWidth 170
TrvActionInsertText 155
TrvActionInsertText.OnInsertText 155
TrvActionInsertTextBox 162
TrvActionItemProperties 201
TrvActionItemProperties.ActionInsertHLine 202
TrvActionItemProperties.ActionInsertTable 202
TrvActionItemProperties.BackgroundGraphicFilter 202
TrvActionItemProperties.GraphicFilter 202
TrvActionItemProperties.OnCanApply 203
TrvActionItemProperties.OnCustomItemPropertiesDialog 203
TrvActionItemProperties.StoreImageFileName 203
TrvActionLineSpacing 113
TrvActionLineSpacing100 113
TrvActionLineSpacing150 114
TrvActionLineSpacing200 114

TrvActionNew 57
TrvActionNew.Reset 59
TrvActionNew.StyleTemplates 58
TrvActionOpen 59
TrvActionOpen.ActionNew 60
TrvActionOpen.CustomFilter 60
TrvActionOpen.DialogTitle 60
TrvActionOpen.Filter 61
TrvActionOpen.InitialDir 61
TrvActionOpen.LastFilterIndex 61
TrvActionOpen.LoadFile 61
TrvActionOpen.OnOpenFile 62
TrvActionPageSetup 64
TrvActionPageSetup.MarginsUnits 65
TrvActionPageSetup.OnChange 65
TrvActionParaBiDi 115
TrvActionParaBorder 116
TrvActionParaBorder.Background 116
TrvActionParaBorder.Border 116
TrvActionParaBorder.UserInterface 117
TrvActionParaBorder.ValidProperties 117
TrvActionParaBullets 117
TrvActionParaColor 118
TrvActionParaColorAndPadding 118
TrvActionParaColorAndPadding.Padding 119
TrvActionParaColorAndPadding.UsePadding 119
TrvActionParaCustomColor 119
TrvActionParagraph 120
TrvActionParagraph.Alignment 120, 122
TrvActionParagraph.DeleteAllTabs 121
TrvActionParagraph.FirstIndent 121
TrvActionParagraph.KeepLinesTogether 121
TrvActionParagraph.KeepWithNext 121
TrvActionParagraph.LastLineAlignment 122
TrvActionParagraph.LeftIndent 122
TrvActionParagraph.LineSpacing 122
TrvActionParagraph.LineSpacingType 122
TrvActionParagraph.OutlineLevel 123
TrvActionParagraph.RightIndent 123
TrvActionParagraph.SpaceAfter 123
TrvActionParagraph.SpaceBefore 123
TrvActionParagraph.Tabs 124
TrvActionParagraph.TabsToDelete 124
TrvActionParagraph.UserInterface 124
TrvActionParagraph.ValidProperties 124
TrvActionParaList 125
TrvActionParaList.ActionParaBullets 126
TrvActionParaList.ActionParaNumbering 126
TrvActionParaList.IndentStep 125
TrvActionParaList.UpdateAllActionsOnForm 126
TrvActionParaLTR 126
TrvActionParaNumbering 127
TrvActionParaRTL 128
TrvActionParaStyles 128
TrvActionPaste 77
TrvActionPasteAsText 78
TrvActionPasteSpecial 78
TrvActionPasteSpecial.AddFormat 80
TrvActionPasteSpecial.OnCanPaste 80
TrvActionPasteSpecial.OnCustomPaste 81
TrvActionPasteSpecial.OnShowing 81
TrvActionPasteSpecial.StoreFileName 79
TrvActionPasteSpecial.StoreFileNameInItemName 80
TrvActionPrint 65
TrvActionPrint.Title 66
TrvActionPrintPreview 66
TrvActionPrintPreview.ActionPageSetup 67
TrvActionPrintPreview.ActionPrint 67
TrvActionPrintPreview.Maximized 67
TrvActionPrintPreview.OnGetPreviewFormClass 68
TrvActionQuickPrint 68
TrvActionQuickPrint.Title 68
TrvActionRedo 81
TrvActionRemoveHyperlinks 204
TrvActionRemoveHyperlinks.ActionInsertHyperlink 204
TrvActionRemovePageBreak 204
TrvActionReplace 81
TrvActionReplace.ActionFind 82
TrvActionReplace.CloseDialog 82
TrvActionReplace.OnReplaceAllEnd 82
TrvActionReplace.OnReplaceAllStart 83
TrvActionReplace.OnReplacing 83
TrvActionReplace.ShowReplaceAllSummary 82
TrvActionSave 69
TrvActionSave.ActionSaveAs 69
TrvActionSave.CanCloseDoc 71
TrvActionSave.CreateDirectoryForHTMLImages 69
TrvActionSave.DisableWhenUnmodified 70
TrvActionSave.Documents 70
TrvActionSave.FileTitle 70
TrvActionSave.FindDoc 72
TrvActionSave.ImagePrefix 70
TrvActionSave.LostFormatWarning 71
TrvActionSave.OnDocumentFileChange 72
TrvActionSave.OnSaving 72
TrvActionSave.SaveOptions 71

TrvActionSave.SuppressNextErrorMessage 71
TrvActionSaveAs 73
TrvActionSaveAs.ActionSave 73
TrvActionSaveAs.Filter 74
TrvActionSelectAll 83
TrvActionShowSpecialCharacters 205
TrvActionSSScript 99
TrvActionStyleInspector 132
TrvActionStyleInspector.FontImageIndex 133
TrvActionStyleInspector.Images 132
TrvActionStyleInspector.ParaImageIndex 133
TrvActionStyleInspector.UpdateInfo 133
TrvActionStyleTemplates 129
TrvActionStyleTemplates.Images 130
TrvActionStyleTemplates.ParaStyleImageIndex 131
TrvActionStyleTemplates.ParaTextStyleImageIndex 131
TrvActionStyleTemplates.StandardStyleTemplates 130
TrvActionStyleTemplates.TextStyleImageIndex 131
TrvActionSubscript 99
TrvActionSuperscript 100
TrvActionTableCell 172
TrvActionTableCellAllBorders 172
TrvActionTableCellBorder 173
TrvActionTableCellBottomBorder 173
TrvActionTableCellLeftBorder 174
TrvActionTableCellNoBorders 175
TrvActionTableCellRightBorder 175
TrvActionTableCellRotation 176
TrvActionTableCellRotation180 177
TrvActionTableCellRotation270 178
TrvActionTableCellRotation90 177
TrvActionTableCellRotationNone 176
TrvActionTableCellTopBorder 179
TrvActionTableCellVAlign 179
TrvActionTableCellVAlignBottom 180
TrvActionTableCellVAlignDefault 180
TrvActionTableCellVAlignMiddle 181
TrvActionTableCellVAlignTop 182
TrvActionTableDeleteCols 182
TrvActionTableDeleteRows 183
TrvActionTableDeleteTable 183
TrvActionTableGrid 184
TrvActionTableInsertColLeft 184
TrvActionTableInsertColRight 185
TrvActionTableInsertRowsAbove 185
TrvActionTableInsertRowsBelow 186
TrvActionTableInsertRowsBelow.AllowMultiple 186
TrvActionTableMergeCells 186
TrvActionTableMultiCellAttributes 187
TrvActionTableProperties 187
TrvActionTableProperties.ActionInsertTable 188
TrvActionTableProperties.BackgroundGraphicFilter 188
TrvActionTableProperties.StoreImageFileName 188
TrvActionTableRCBase 189
TrvActionTableSelectCell 189
TrvActionTableSelectCols 190
TrvActionTableSelectRows 190
TrvActionTableSelectTable 191
TrvActionTableSort 192
TrvActionTableSplit 192
TrvActionTableSplitCells 193
TrvActionTableToText 193
TrvActionTextBiDi 100
TrvActionTextLTR 101
TrvActionTextRTL 102
TrvActionTextStyles 102
TrvActionUndo 83
TrvActionVAlign 205
TrvActionWordWrap 129
TRVACustomPasteEvent 81
TRVAddStyleEvent 36
TrvaDocumentInfo 70
TRVADownloadEvent 38
TRVAEditEvent 232
TRVAEditorControlCommand 236
TRVAEditorControlCommandFunction 236
TRVAEvent 199, 200
TRVAFileOperation 37
TRVAFillColorApplyToElement 200
TRVAFillColorApplyToSet 200
TRVAHFInfo 233
TRVAInsertTableEvent 171
TRVAInsertTextEvent 155
TRVALanguageName 216
TRVALiveSpellAddEvent type 42
TRVALiveSpellGetSuggestionsEvent type 42
TRVALiveSpellIgnoreAllEvent type 43
TRVALiveSpellReplaceEvent type 43
TRVAMessageID 214
TRVAPopupActionBar 40
TRVAPopupMenu 40
TRVAPopupMenu.ActionList 41
TRVAPopupMenu.MaxSuggestionsCount 41
TRVAPopupMenu.OnLiveSpellAdd 42
TRVAPopupMenu.OnLiveSpellGetSuggestions 42

TRVAPopupMenu.OnLiveSpellIgnoreAll 43
TRVAPopupMenu.OnLiveSpellWordReplace 43
TRVApplyHyperlinkToItemEvent 146
TRVAReplacingEvent 83
TRVASearchScope 32
TRVASpellString 233
TRVASpellStringList 233
TRVASpellStrings 233
TRVASPTBXPopupMenu 40
TRVASPTBXPopupMenu.ActionList 41
TRVASPTBXPopupMenu.MaxSuggestionsCount 41
TRVASPTBXPopupMenu.OnLiveSpellAdd 42
TRVASPTBXPopupMenu.OnLiveSpellGetSuggestions 42
TRVASPTBXPopupMenu.OnLiveSpellIgnoreAll 43
TRVASPTBXPopupMenu.OnLiveSpellWordReplace 43
TRVATBPopupMenu 40
TRVATBPopupMenu.ActionList 41
TRVATBPopupMenu.MaxSuggestionsCount 41
TRVATBPopupMenu.OnLiveSpellAdd 42
TRVATBPopupMenu.OnLiveSpellGetSuggestions 42
TRVATBPopupMenu.OnLiveSpellIgnoreAll 43
TRVATBPopupMenu.OnLiveSpellWordReplace 43
TRVATBXPopupMenu 40
TRVATBXPopupMenu.ActionList 41
TRVATBXPopupMenu.MaxSuggestionsCount 41
TRVATBXPopupMenu.OnLiveSpellAdd 42
TRVATBXPopupMenu.OnLiveSpellGetSuggestions 42
TRVATBXPopupMenu.OnLiveSpellIgnoreAll 43
TRVATBXPopupMenu.OnLiveSpellWordReplace 43
TRVAUserInterface 34
TRVCanApplyEvent 203
TRVCharCase 232
TRVCreateTeFormEvent 36
TrvCustomAction 206
TrvCustomAction.Caption 207
TrvCustomAction.ControlPanel 207
TrvCustomAction.Disabled 207
TrvCustomAction.GetControlPanel 208
TrvCustomAction.Hint 207
TrvCustomEditAction 84
TrvCustomEditorAction 208
TrvCustomEditorAction.Form 208
TrvCustomEditorAction.OnFormCreate 209
TrvCustomEditorAction.OnHide 209
TrvCustomEditorAction.OnShowing 209
TrvCustomEditorAction.SubDocEditor 208
TRVCustomFileOperationEvent 37
TRVCustomItemPropertiesDialog 203
TrvCustomPrintAction 74
TrvFileExportFilter 233
TrvFileExportFilterSet 233
TrvFileImportFilter 234
TrvFileImportFilterSet 234
TrvFileOpenFilter 234
TrvFileOpenFilterSet 234
TrvFileSaveFilter 233
TrvFileSaveFilterSet 233
TRVFontCharsetComboBox 44
TRVFontCharsetComboBox.AddDefaultCharset 45
TRVFontCharsetComboBox.DefaultCharsetCaption 45
TRVFontCharsetComboBox.FontName 45
TRVFontComboBox 46
TRVFontInfoMainProperties 90
TRVFontInfoMainProperty 90
TRVFontSizeComboBox 46
TRVFontSizeComboBox.FontName 47
TRVGetFormClassEvent 68
TRVGetHyperlinkTargetFromItem 146
TRVHyperlinkFormEvent 147
TRVHyperlinkProperties 143
TRVHyperlinkProperty 143
TrvPaperMarginsUnits 65
TRVParaInfoBorderProperties 117
TRVParaInfoBorderProperty 117
TRVParaInfoMainProperties 124
TRVParaInfoMainProperty 124
TRVSaveFileEvent 72
TRVShowFormEvent 133, 235
TRVStyleNeededEvent 39
TRVStyleTemplateComboBox 47
TRVStyleTemplateComboBox.AllStylesImageIndex 48
TRVStyleTemplateComboBox.ClearFormatImageIndex 48
TRVStyleTemplateComboBox.ControlPanel 48
TRVStyleTemplateComboBox.Editor 48
TRVStyleTemplateComboBox.Images 48
TRVStyleTemplateComboBox.Localize 49
TRVStyleTemplateComboBox.OnClickAllStyles 50
TRVStyleTemplateComboBox.ParaStyleImageIndex 48
TRVStyleTemplateComboBox.ParaTextStyleImageIndex 48

TRVStyleTemplateComboBox.ShowAllStyles 49
TRVStyleTemplateComboBox.ShowClearFormat 49
TRVStyleTemplateComboBox.SmartHeadings 49
TRVStyleTemplateComboBox.TextStyleImageIndex 48
TRVStyleTemplateListBox.AllStylesImageIndex 48
TRVStyleTemplateListBox.ClearFormatImageIndex 48
TRVStyleTemplateListBox.ControlPanel 48
TRVStyleTemplateListBox.Editor 48
TRVStyleTemplateListBox.Images 48
TRVStyleTemplateListBox.Localize 49
TRVStyleTemplateListBox.OnClickAllStyles 50
TRVStyleTemplateListBox.ParaStyleImageIndex 48
TRVStyleTemplateListBox.ParaTextStyleImageIndex 48
TRVStyleTemplateListBox.ShowAllStyles 49
TRVStyleTemplateListBox.ShowClearFormat 49
TRVStyleTemplateListBox.SmartHeadings 49
TRVStyleTemplateListBox.TextStyleImageIndex 48

- U -

UnderlineColor 90, 143
TrvActionFontEx 90
TrvActionInsertHyperlink 143
UnderlineType 90
TrvActionFontEx 90
UnitsDisplay 33
TRVAControlPanel 33
UnitsProgram 33
TRVAControlPanel 33
UpdateAllActionsOnForm 126
TrvActionParaList 126
UpdateInfo 133
TrvActionStyleInspector 133
UseCurrentFont 155
TrvActionInsertSymbol 155
UseDefaults 149
TrvActionInsertNumSequence 149
UseHelpFiles 34
TRVAControlPanel 34
UseLastLineAlignment 104
TrvActionAlignCustomJustify 104
UsePadding 119
TrvActionParaColorAndPadding 119
UserInterface 34, 94, 117, 124, 198
TRVAControlPanel 34
TrvActionCustomColor 198
TrvActionFonts 94
TrvActionParaBorder 117
TrvActionParagraph 124
UseTextCodePageDialog 34
TRVAControlPanel 34
UseXPThemes 35
TRVAControlPanel 35
Using RichViewActions 14

- V -

ValidProperties 90, 117, 124, 143
TrvActionFontEx 90
TrvActionInsertHyperlink 143
TrvActionParaBorder 117
TrvActionParagraph 124
VAlign 153
TrvActionInsertPicture 153
VisibleBorders 170
TrvActionInsertTable 170
VOutermostRule 170
TrvActionInsertTable 170
VRuleColor 170
TrvActionInsertTable 170
VRuleWidth 170
TrvActionInsertTable 170
VShift 90
TrvActionFontEx 90

- W -

Width 138
TrvActionInsertHLine 138

- X -

XMLFilter 35
TRVAControlPanel 35
XMLLocalizable 35
TRVAControlPanel 35