

MUI Royale 1.7

Easy GUI Creation For You Kings and Queens of MUI

Andreas Falkenhahn

Inhaltsverzeichnis

1	Allgemeine Information	1
1.1	Einführung	1
1.2	Lizenzvereinbarung	1
1.3	Voraussetzung	2
1.4	Palettenbildschirme Unterstützung	2
2	Über MUI Royale	5
2.1	Geschichte	5
2.2	Zukunft	5
2.3	Häufig gestellte Fragen	5
2.4	Danksagungen	6
3	Grundlegende MUI-Konzepte	7
3.1	Anwendungsbaum	7
3.2	Klassenhierarchie	8
3.3	Automatisches Layout-Modul	8
3.4	Gruppen-Objekte	9
3.5	Objekthandhabung	12
3.6	Benachrichtigungen	12
3.7	Anwendbarkeit	14
3.8	Tastaturabfolgekette	14
3.9	Textformatierungs-codes	14
3.10	Online-Hilfe implementieren	16
3.11	Kontextmenüs	17
3.12	Drag'n'drop	18
3.13	Zeichenkodierung	18
3.14	Hollywood-Brücke	19
3.15	Styleguide	24
4	Tutorial	27
4.1	Tutorial	27
5	Befehlsreferenz	37
5.1	mui.CreateGUI	37
5.2	mui.CreateObject	38
5.3	mui.DoMethod	39
5.4	mui.FreeGUI	40
5.5	mui.FreeImage	40
5.6	mui.FreeObject	41
5.7	mui.Get	41
5.8	mui.HaveObject	42

5.9	mui.IsVersion4	43
5.10	mui.Notify	43
5.11	mui.Request	44
5.12	mui.Set	45
6	Application-Klasse (Anwendung)	47
6.1	Überblick	47
6.2	Application.AboutMUI	47
6.3	Application.AboutMUIRoyale	47
6.4	Application.AddWindow	47
6.5	Application.Base	48
6.6	Application.ConfigChange	48
6.7	Application.DoubleStart	49
6.8	Application.DropObject	49
6.9	Application.HelpFile	49
6.10	Application.Icon	50
6.11	Application.Iconified	50
6.12	Application.Load	51
6.13	Application.Menustrip	52
6.14	Application.OpenConfigWindow	52
6.15	Application.RemoveWindow	53
6.16	Application.Save	53
6.17	Application.SingleTask	54
6.18	Application.Sleep	55
6.19	Application.WindowList	55
7	Area-Klasse (Bereich)	57
7.1	Überblick	57
7.2	Area.Background	57
7.3	Area.BackgroundBrush	58
7.4	Area.BackgroundImage	59
7.5	Area.BackgroundRGB	59
7.6	Area.BottomEdge	60
7.7	Area.ContextMenu	60
7.8	Area.ContextMenuTrigger	61
7.9	Area.ControlChar	61
7.10	Area.CycleChain	62
7.11	Area.Disabled	62
7.12	Area.FixHeight	62
7.13	Area.FixHeightTxt	63
7.14	Area.FixWidth	63
7.15	Area.FixWidthTxt	63
7.16	Area.Font	64
7.17	Area.Frame	64
7.18	Area.FramePhantomHoriz	65
7.19	Area.FrameTitle	66
7.20	Area.Height	66

7.21	Area.Hide	66
7.22	Area.HorizWeight	67
7.23	Area.InnerBottom	67
7.24	Area.InnerLeft	68
7.25	Area.InnerRight	68
7.26	Area.InnerTop	68
7.27	Area.InputMode	69
7.28	Area.LeftEdge	69
7.29	Area.MaxHeight	70
7.30	Area.MaxWidth	70
7.31	Area.Pressed	70
7.32	Area.RightEdge	71
7.33	Area.Selected	71
7.34	Area.ShortHelp	72
7.35	Area.ShowSelState	72
7.36	Area.TopEdge	72
7.37	Area.VertWeight	73
7.38	Area.Weight	73
7.39	Area.Width	74
8	Busy-Klasse (Beschäftigung)	75
8.1	Überblick	75
8.2	Busy.Move	75
8.3	Busy.Speed	75
9	Button-Klasse (Schaltflächen)	77
9.1	Übersicht	77
9.2	Button.HiChar	77
9.3	Button.Label	77
9.4	Button.NoAutoKey	78
9.5	Button.Pressed	78
9.6	Button.Selected	78
9.7	Button.Toggle	79
10	Checkmark-Klasse (Auswahlkästchen)	81
10.1	Übersicht	81
10.2	Checkmark.Selected	81
11	Coloradjust-Klasse (Farbanpassung)	83
11.1	Überblick	83
11.2	Coloradjust.RGB	83
12	Colorfield-Klasse (Farbfeld)	85
12.1	Übersicht	85
12.2	Colorfield.RGB	85

13	Cycle-Klasse (Auswahl-Listefeld)	87
13.1	Übersicht	87
13.2	Cycle.Active	87
14	Dirlist-Klasse (Verzeichnislisten)	89
14.1	Übersicht	89
14.2	Dirlist.AcceptPattern	90
14.3	Dirlist.Directory	90
14.4	Dirlist.DrawersOnly	91
14.5	Dirlist.FilesOnly	91
14.6	Dirlist.FilterDrawers	91
14.7	Dirlist.MultiSelDirs	92
14.8	Dirlist.NumBytes	92
14.9	Dirlist.NumDrawers	92
14.10	Dirlist.NumFiles	93
14.11	Dirlist.Path	93
14.12	Dirlist.RejectIcons	93
14.13	Dirlist.RejectPattern	94
14.14	Dirlist.ReRead	94
14.15	Dirlist.SortDirs	94
14.16	Dirlist.SortHighLow	95
14.17	Dirlist.SortType	95
14.18	Dirlist.Status	95
14.19	Dirlist.Title	96
15	Floattext-Klasse (Fließtext)	97
15.1	Übersicht	97
15.2	Floattext.Justify	97
15.3	Floattext.TabSize	97
15.4	Floattext.Text	98
16	Gauge-Klasse (Fortschrittsbalken)	99
16.1	Übersicht	99
16.2	Gauge.Current	99
16.3	Gauge.Divide	99
16.4	Gauge.Horiz	100
16.5	Gauge.InfoText	100
16.6	Gauge.Max	101

17	Group-Klasse (Gruppen)	103
17.1	Übersicht	103
17.2	Group.ActivePage	104
17.3	Group.AddHead	105
17.4	Group.AddTail	105
17.5	Group.ChildList	106
17.6	Group.Columns	107
17.7	Group.ExitChange	107
17.8	Group.HorizSpacing	108
17.9	Group.InitChange	108
17.10	Group.Insert	109
17.11	Group.PageMode	109
17.12	Group.Remove	110
17.13	Group.SameHeight	111
17.14	Group.SameSize	111
17.15	Group.SameWidth	112
17.16	Group.Spacing	112
17.17	Group.Title	112
17.18	Group.VertSpacing	113
18	HSpace-Klasse (HorizontalAbstand)	115
18.1	Übersicht	115
18.2	HSpace.Width	115
19	Hollywood Klasse	117
19.1	Übersicht	117
19.2	Hollywood.Display	117
19.3	Hollywood.MaxHeight	117
19.4	Hollywood.MaxWidth	118
19.5	Hollywood.MinHeight	118
19.6	Hollywood.MinWidth	119
20	Image-Klasse (Bild)	121
20.1	Übersicht	121
20.2	Image.Brush	121
20.3	Image.FreeHoriz	121
20.4	Image.FreeVert	122
20.5	Image.Source	122
20.6	Image.State	123
21	Imagebutton-Klasse (Bildschaltflächen) ...	125
21.1	Übersicht	125
21.2	Imagebutton.Pressed	125

22	Label-Klasse (Beschriftung)	127
22.1	Übersicht	127
22.2	Label.Centered	127
22.3	Label.DoubleFrame	127
22.4	Label.FreeVert	128
22.5	Label.Key	128
22.6	Label.LeftAligned	128
22.7	Label.SingleFrame.....	129
23	Listtree-Klasse (Baumlisten)	131
23.1	Übersicht	131
23.2	Listtree.Active	132
23.3	Listtree.AutoLineHeight.....	132
23.4	Listtree.Close	133
23.5	Listtree.DoubleClick.....	134
23.6	Listtree.DragDropSort.....	134
23.7	Listtree.EmptyNodes	134
23.8	Listtree.Exchange	135
23.9	Listtree.FindName	136
23.10	Listtree.GetEntry	137
23.11	Listtree.Insert	138
23.12	Listtree.MinLineHeight.....	140
23.13	Listtree.Move	140
23.14	Listtree.Open	141
23.15	Listtree.Quiet	142
23.16	Listtree.Remove	142
23.17	Listtree.Rename	143
23.18	Listtree.Sort	144
23.19	Listtree.SortHook	144
23.20	Listtree.Title.....	145
24	Listtreenode-Klasse (Baumlistenknoten) ..	147
24.1	Übersicht	147
24.2	Listtreenode.Frozen	147
24.3	Listtreenode.Name	147
24.4	Listtreenode.NoSign	147
24.5	Listtreenode.Open.....	148
25	Listview-Klasse (Listenansicht)	149
25.1	Übersicht	149
25.2	Listview.Active.....	150
25.3	Listview.AdjustHeight	150
25.4	Listview.AdjustWidth	151
25.5	Listview.AutoLineHeight	151
25.6	Listview.AutoVisible	151
25.7	Listview.Clear.....	152

25.8	Listview.ClickColumn	152
25.9	Listview.DefClickColumn	152
25.10	Listview.DoubleClick	153
25.11	Listview.Entries	153
25.12	Listview.Exchange	153
25.13	Listview.First	154
25.14	Listview.GetEntry	154
25.15	Listview.GetSelection	154
25.16	Listview.Input	155
25.17	Listview.Insert	155
25.18	Listview.InsertPosition	156
25.19	Listview.Jump	156
25.20	Listview.MinLineHeight	157
25.21	Listview.Move	157
25.22	Listview.MultiSelect	158
25.23	Listview.Quiet	158
25.24	Listview.Remove	159
25.25	Listview.Rename	159
25.26	Listview.Select	160
25.27	Listview.ScrollerPos	160
25.28	Listview.SelectChange	161
25.29	Listview.Sort	161
25.30	Listview.SortFunc	162
25.31	Listview.TitleClick	163
25.32	Listview.Visible	163
26	Listviewcolumn-Klasse	
	(Listenansichtspalten)	165
26.1	Übersicht	165
26.2	Listviewcolumn.Bar	165
26.3	Listviewcolumn.Col	165
26.4	Listviewcolumn.Delta	166
26.5	Listviewcolumn.Hidden	166
26.6	Listviewcolumn.MinWidth	166
26.7	Listviewcolumn.MaxWidth	167
26.8	Listviewcolumn.PreParse	167
26.9	Listviewcolumn.Title	168
26.10	Listviewcolumn.Weight	168
27	Menu-Klasse (Menü)	169
27.1	Übersicht	169
27.2	Menu.AddHead	169
27.3	Menu.AddTail	169
27.4	Menu.Disabled	170
27.5	Menu.Insert	170
27.6	Menu.Remove	171
27.7	Menu.Title	171

28	MenuItem-Klasse (Menüelement)	173
28.1	Übersicht	173
28.2	MenuItem.CommandString	173
28.3	MenuItem.Disabled	173
28.4	MenuItem.Exclude	174
28.5	MenuItem.Selected	174
28.6	MenuItem.Shortcut	174
28.7	MenuItem.Title	175
28.8	MenuItem.Type	175
29	Menustrip-Klasse (Menüleisten)	177
29.1	Übersicht	177
29.2	Menustrip.AddHead	178
29.3	Menustrip.AddTail	178
29.4	Menustrip.Insert	179
29.5	Menustrip.Remove	179
30	Notify-Klasse (Benachrichtigung)	181
30.1	Übersicht	181
30.2	Notify.AppMessage	181
30.3	Notify.Class	182
30.4	Notify.ExportID	182
30.5	Notify.HelpLine	182
30.6	Notify.HelpNode	183
30.7	Notify.ID	183
30.8	Notify.MUIClass	183
30.9	Notify.NoNotify	184
30.10	Notify.NotifyData	184
30.11	Notify.Revision	185
30.12	Notify.UserData	185
30.13	Notify.Version	185
31	Numericbutton-Klasse (Nummerschaltflächen)	187
31.1	Übersicht	187
32	Popdrawer-Klasse (Verzeichnisdialog)	189
32.1	Übersicht	189
32.2	Popdrawer.Acknowledge	189
32.3	Popdrawer.Active	189
32.4	Popdrawer.AdvanceOnCR	190
32.5	Popdrawer.Contents	190
32.6	Popdrawer.SaveMode	190
32.7	Popdrawer.Title	191

33	Popfile-Klasse (Dateialog)	193
33.1	Übersicht	193
33.2	Popfile.AcceptPattern	193
33.3	Popfile.Acknowledge	193
33.4	Popfile.Active	194
33.5	Popfile.AdvanceOnCR	194
33.6	Popfile.Contents	194
33.7	Popfile.Pattern	195
33.8	Popfile.RejectIcons	195
33.9	Popfile.RejectPattern	195
33.10	Popfile.SaveMode	196
33.11	Popfile.ShowPattern	196
33.12	Popfile.Title	196
34	Popfont-Klasse (Schriftdialog)	197
34.1	Übersicht	197
34.2	Popfont.Acknowledge	197
34.3	Popfont.Active	197
34.4	Popfont.AdvanceOnCR	198
34.5	Popfont.Contents	198
34.6	Popfont.FixedWidthOnly	198
34.7	Popfont.MaxHeight	199
34.8	Popfont.MinHeight	199
34.9	Popfont.Title	199
35	Poplist-Klasse (Listendialog)	201
35.1	Übersicht	201
35.2	Poplist.Acknowledge	201
35.3	Poplist.AdvanceOnCR	202
35.4	Poplist.Contents	202
36	Poppen-Klasse (Stiftdialog)	203
36.1	Übersicht	203
36.2	Poppen.RGB	203
36.3	Poppen.Title	203
37	Prop-Klasse (Proportional)	205
37.1	Übersicht	205
37.2	Prop.Decrease	205
37.3	Prop.Entries	205
37.4	Prop.First	206
37.5	Prop.Horiz	206
37.6	Prop.Increase	206
37.7	Prop.Slider	207
37.8	Prop.UseWinBorder	207
37.9	Prop.Visible	208

38	Radio-Klasse	209
38.1	Übersicht	209
38.2	Radio.Active	209
39	Rectangle-Klasse (Rechteck)	211
39.1	Übersicht	211
39.2	Rectangle.BarTitle	211
39.3	Rectangle.HBar	211
39.4	Rectangle.VBar	212
40	Register-Klasse	213
40.1	Übersicht	213
40.2	Register.ActivePage	213
40.3	Register.AddPage	214
40.4	Register.Closable	214
40.5	Register.ClosePage	215
40.6	Register.CloseRequest	215
40.7	Register.GetPageID	215
40.8	Register.InsertPage	216
40.9	Register.Pages	217
40.10	Register.Position	217
41	Scale-Klasse (Skala)	219
41.1	Übersicht	219
41.2	Scale.Horiz	219
42	Scrollbar-Klasse (Bildlaufleiste)	221
42.1	Übersicht	221
42.2	Scrollbar.IncDecSize	221
42.3	Scrollbar.Type	221
43	Scrollgroup-Klasse (Bildlaufgruppen)	223
43.1	Übersicht	223
43.2	Scrollgroup.FreeHoriz	223
43.3	Scrollgroup.FreeVert	224
43.4	Scrollgroup.UseWinBorder	224

44	Slider-Klasse (Schieberegler)	225
44.1	Übersicht	225
44.2	Slider.Format	225
44.3	Slider.Horiz	226
44.4	Slider.Level	226
44.5	Slider.Max	226
44.6	Slider.Min	226
44.7	Slider.Pressed	227
44.8	Slider.Quiet	227
44.9	Slider.Reverse	227
44.10	Slider.Stringify	228
45	String-Klasse (Zeichenketten)	229
45.1	Übersicht	229
45.2	String.Accept	229
45.3	String.Acknowledge	229
45.4	String.AdvanceOnCR	230
45.5	String.Contents	230
45.6	String.Copy	230
45.7	String.CursorPos	231
45.8	String.Cut	231
45.9	String.Insert	231
45.10	String.MarkEnd	232
45.11	String.MarkStart	232
45.12	String.MaxLen	232
45.13	String.Paste	233
45.14	String.Redo	233
45.15	String.Reject	233
45.16	String.Secret	234
45.17	String.Undo	234
46	Text-Klasse	235
46.1	Übersicht	235
46.2	Text.Contents	235
46.3	Text.HiChar	235
46.4	Text.PreParse	236
46.5	Text.SetMax	236
46.6	Text.SetMin	237
46.7	Text.SetVMax	237
47	Texteditor-Klasse	239
47.1	Übersicht	239
47.2	Texteditor.ActiveObjectOnClick	239
47.3	Texteditor.Align	240
47.4	Texteditor.AreaMarked	240
47.5	Texteditor.AutoClip	240

47.6	Texteditor.Clear	241
47.7	Texteditor.Color	241
47.8	Texteditor.ColorMap	242
47.9	Texteditor.Columns	242
47.10	Texteditor.Contents	242
47.11	Texteditor.ConvertTabs	243
47.12	Texteditor.Copy	243
47.13	Texteditor.CursorX	243
47.14	Texteditor.CursorY	244
47.15	Texteditor.Cut	244
47.16	Texteditor.Erase	244
47.17	Texteditor.ExportHook	245
47.18	Texteditor.ExportWrap	245
47.19	Texteditor.FixedFont	246
47.20	Texteditor.GetSelection	246
47.21	Texteditor.GetText	246
47.22	Texteditor.HasChanged	247
47.23	Texteditor.ImportHook	248
47.24	Texteditor.ImportWrap	249
47.25	Texteditor.Insert	250
47.26	Texteditor.Mark	250
47.27	Texteditor.MarkAll	251
47.28	Texteditor.MarkNone	251
47.29	Texteditor.Paste	251
47.30	Texteditor.PasteColors	252
47.31	Texteditor.PasteStyles	252
47.32	Texteditor.ReadOnly	252
47.33	Texteditor.Redo	253
47.34	Texteditor.RedoAvailable	253
47.35	Texteditor.Replace	253
47.36	Texteditor.Rows	254
47.37	Texteditor.Scrollbar	254
47.38	Texteditor.Search	254
47.39	Texteditor.SetBold	255
47.40	Texteditor.SetColor	256
47.41	Texteditor.SetItalic	256
47.42	Texteditor.SetUnderline	257
47.43	Texteditor.StyleBold	258
47.44	Texteditor.StyleItalic	258
47.45	Texteditor.StyleUnderline	258
47.46	Texteditor.TabSize	259
47.47	Texteditor.Undo	259
47.48	Texteditor.UndoAvailable	259
47.49	Texteditor.UndoLevels	260
47.50	Texteditor.WrapBorder	260
47.51	Texteditor.WrapMode	260
47.52	Texteditor.WrapWords	261

48 Toolbar-Klasse (Symbolleiste) 263

48.1	Übersicht	263
48.2	Toolbar.Active	263
48.3	Toolbar.BarPos	264
48.4	Toolbar.BarSpacer	264
48.5	Toolbar.BarSpacerSpacing	264
48.6	Toolbar.Borderless	265
48.7	Toolbar.BottomBarFrameSpacing	265
48.8	Toolbar.BottomInnerSpacing	265
48.9	Toolbar.Columns	266
48.10	Toolbar.DisMode	266
48.11	Toolbar.DontMove	267
48.12	Toolbar.EnableKeys	267
48.13	Toolbar.Frame	267
48.14	Toolbar.Free	268
48.15	Toolbar.FreeHoriz	268
48.16	Toolbar.FreeVert	269
48.17	Toolbar.Horiz	269
48.18	Toolbar.HorizInnerSpacing	269
48.19	Toolbar.HorizSpacing	270
48.20	Toolbar.HorizTextGfxSpacing	270
48.21	Toolbar.IgnoreAppearance	270
48.22	Toolbar.LabelPos	271
48.23	Toolbar.LeftBarFrameSpacing	271
48.24	Toolbar.MouseOver	272
48.25	Toolbar.NtRaiseActive	272
48.26	Toolbar.Raised	272
48.27	Toolbar.RightBarFrameSpacing	273
48.28	Toolbar.Rows	273
48.29	Toolbar.Scale	273
48.30	Toolbar.Scaled	274
48.31	Toolbar.SpacersSize	274
48.32	Toolbar.SpecialSelect	274
48.33	Toolbar.Sunny	275
48.34	Toolbar.TextOverUseShine	275
48.35	Toolbar.TopBarFrameSpacing	275
48.36	Toolbar.TopInnerSpacing	276
48.37	Toolbar.VertSpacing	276
48.38	Toolbar.VertTextGfxSpacing	277
48.39	Toolbar.ViewMode	277

49 Toolbarbutton-Klasse (Symbolleistenschaltfläche) 279

49.1	Übersicht	279
49.2	Toolbarbutton.Disabled	279
49.3	Toolbarbutton.DisImage	279
49.4	Toolbarbutton.Exclude	279

49.5	Toolbarbutton.Hide	280
49.6	Toolbarbutton.Image	280
49.7	Toolbarbutton.Immediate	281
49.8	Toolbarbutton.NoClick	281
49.9	Toolbarbutton.Pressed	281
49.10	Toolbarbutton.Selected	281
49.11	Toolbarbutton.SelImage	282
49.12	Toolbarbutton.ShortHelp	282
49.13	Toolbarbutton.Sleep	282
49.14	Toolbarbutton.Toggle	283
50	VSpace-Klasse (VertikalAbstand)	285
50.1	Übersicht	285
50.2	VSpace.Height	285
51	Virtgroup-Klasse (Virtuelle-Gruppen)	287
51.1	Übersicht	287
51.2	Virtgroup.Height	287
51.3	Virtgroup.Horiz	287
51.4	Virtgroup.Input	288
51.5	Virtgroup.Left	288
51.6	Virtgroup.Top	288
51.7	Virtgroup.Width	289
52	Volumelist-Klasse (Datenträgerlisten)	291
52.1	Übersicht	291
52.2	Volumelist.Title	292
53	Window-Klasse (Fenster)	293
53.1	Übersicht	293
53.2	Window.Activate	293
53.3	Window.ActiveObject	293
53.4	Window.AppWindow	294
53.5	Window.Borderless	294
53.6	Window.CloseGadget	294
53.7	Window.CloseRequest	295
53.8	Window.DefaultObject	295
53.9	Window.DepthGadget	295
53.10	Window.DragBar	296
53.11	Window.Height	296
53.12	Window.LeftEdge	297
53.13	Window.Menustrip	297
53.14	Window.MouseObject	298
53.15	Window.MuiID	298
53.16	Window.NeedsMouseObject	299
53.17	Window.NoMenus	299

53.18	Window.Open	299
53.19	Window.PubScreen	300
53.20	Window.ScreenTitle	300
53.21	Window.ScreenToBack	300
53.22	Window.ScreenToFront	301
53.23	Window.Sleep	301
53.24	Window.SizeGadget	301
53.25	Window.Snapshot	302
53.26	Window.ToBack	302
53.27	Window.ToFront	302
53.28	Window.TopEdge	303
53.29	Window.UseBottomBorderScroller	303
53.30	Window.UseLeftBorderScroller	304
53.31	Window.UseRightBorderScroller	304
53.32	Window.Title	305
53.33	Window.Width	305
Anhang A Lizenzen		307
A.1	MUI Lizenzen	307
A.2	Expat Lizenzen	307
A.3	GPL Lizenzen	307
Index		315

1 Allgemeine Information

1.1 Einführung

MUI Royale ist ein Plugin für Hollywood, mit dem Sie einfach MUI-GUIs mit Hollywood erstellen können. Das GUI-Layout kann bequem mithilfe einer XML-Datei definiert werden, die von MUI Royale im laufenden Betrieb in eine MUI-GUI konvertiert wird. Einfacher geht es nicht!

MUI Royale unterstützt über 40 MUI-Klassen einschließlich beliebter Drittanbieter-Klassen wie `TextEditor.mcc` und `TheBar.mcc`. Das Erstellen und Verwalten von Menuskripten wird ebenfalls vollständig unterstützt. Das Highlight von MUI Royale ist jedoch sicherlich seine eingebaute Hollywood MUI-Klasse. Diese Klasse ermöglicht die dynamische Einbettung kompletter Hollywood-Displays in MUI-GUIs, mit denen das Beste aus Hollywood und MUI zu einer leistungsstarken Anwendung kombiniert werden kann.

MUI Royale enthält umfangreiche Dokumentationen in verschiedenen Formaten wie PDF, HTML, AmigaGuide und CHM, die die Grundlagen der MUI-Programmierung im Detail beschreiben und eine praktische Funktions- und Klassenreferenz bieten. Ein Schritt-für-Schritt-Tutorial, das Sie zu Ihrem ersten MUI-Programm führt, ist ebenfalls enthalten. Darüber hinaus sind fast 20 Beispielskripte im Distributionsarchiv enthalten, von denen viele direkte Ports aus dem MUI 3.8 SDK von Stefan Stuntz sind. Aber es gibt auch originelle Entwicklungen, die zeigen, wie man Video- und Song-Player mit der kombinierten Leistung von Hollywood und MUI erstellt.

All dies macht MUI Royale zum wahrhaft königlichen MUI-Erlebnis, das sorgfältig für Sie, Könige und Königinnen von MUI, gefertigt wurde!

1.2 Lizenzvereinbarung

MUI Royale ist © Copyright 2012-2018 bei Andreas Falkenhahn (im folgenden "der Autor" genannt). Alle Rechte vorbehalten.

Das Programm wird zur Verfügung gestellt "wie es ist" und der Autor kann für keinerlei Schäden, welcher Natur sie auch immer sein mögen, verantwortlich gemacht werden. Sie benutzen dieses Programm völlig auf eigene Gefahr und eigenes Risiko. Der Autor gibt keinerlei Garantien in Verbindung mit der Benutzung dieses Programmes, nicht einmal die Garantie der Funktionstüchtigkeit.

Dieses Programm kann frei weitergegeben werden solange die folgenden drei Bedingungen erfüllt sind:

1. Es dürfen keine Änderungen am Programm vorgenommen werden.
2. Das Programm darf nicht verkauft werden.
3. Wenn Sie das Programm auf einer Coverdisk veröffentlichen möchten, müssen Sie erst um Erlaubnis fragen.

Dieses Programm benutzt das Magic User Interface (MUI), welches (C) Copyright 1992-97 by Stefan Stuntz ist. Siehe [Abschnitt A.1 \[MUI-Lizenz\]](#), [Seite 307](#), für Details.

Dieses Programm benutzt Expat (C) Copyright 1998, 1999, 2000 Thai Open Source Software Center Ltd und Clark Cooper. (C) Copyright 2001, 2002, 2003, 2004, 2005, 2006 Expat-Autoren. Siehe [Abschnitt A.2 \[Expat-Lizenz\]](#), [Seite 307](#), für Details.

Dieses Programm benutzt TextEditor.mcc von Allan Odgaard und dem TextEditor.mcc Open Source Team. Siehe [Abschnitt A.3 \[LGPL-Lizenz\]](#), [Seite 307](#), für Details.

Dieses Programm benutzt TheBar.mcc von Alfonso Ranieri und dem TheBar.mcc Open Source Team. Siehe [Abschnitt A.3 \[LGPL-Lizenz\]](#), [Seite 307](#), für Details.

Dieses Programm benutzt codesets.library von Alfonso Ranieri und dem codesets.library Open Source Team. Siehe [Abschnitt A.3 \[LGPL-Lizenz\]](#), [Seite 307](#), für Details.

Das Handbuch von MUI Royale basiert auf dem MUI 3.8 Software Development Kit © Copyright 1992-97 by Stefan Stuntz. Einige weitergehende Informationen in diesem Handbuch stammen von den TextEditor.mcc und TheBar.mcc Entwicklerunterlagen.

Alle Warenzeichen sind Eigentum ihrer jeweiligen Firmen.

FÜR DIESES PROGRAMM GIBT ES KEINE GARANTIE, SOWEIT ES DIE ANZUWENDENDEN GESETZE ZULASSEN. SOFERN ANDERSWO NICHTS GEGENTEILIGES GESCHRIEBEN STEHT STELLEN DER AUTOR UND/ODER DRITTE DAS PROGRAMM "SO WIE ES IST" ZUR VERFÜGUNG, OHNE IRGEND-EINE GARANTIE, WEDER DIREKT NOCH INDIREKT. DIES BEINHÄLTET, IST ABER NICHT DARAUF BESCHRÄNKT, VERKÄUFLICHKEIT UND EIGNUNG FÜR EINEN BESTIMMTEN VERWENDUNGSZWECK. DAS VOLLSTÄNDIGE RISIKO DER QUALITÄT UND AUSFÜHRBARKEIT DES PROGRAMMS LIEGT BEIM ANWENDER. SOLLTE SICH DAS PROGRAMM ALS DEFEKT HERAUSSTELLEN, LIEGEN ALLE KOSTEN FÜR SERVICE, INSTANDSETZUNG ODER NACHBESSERUNG BEIM ANWENDER.

KEIN COPYRIGHT-INHABER ODER DRITTER, DER DAS PROGRAMM WIE OBEN ERLAUBT WEITERVERKAUFT, KANN FÜR SCHÄDEN IRGENDWELCHER ART HAFTBAR GEMACHT WERDEN (DIES BEINHÄLTET, IST ABER NICHT BESCHRÄNKT AUF, DATENVERLUST INFOLGE UNFÄHIGKEIT DES PROGRAMMS, MIT ANDEREN PROGRAMMEN ZUSAMMENZUARBEITEN), SELBST WENN EIN SOLCHER INHABER ODER DRITTER AUF DIE MÖGLICHKEIT EINES SOLCHEN SCHADENS HINGEWIESEN WURDE, AUSSER ES BESTEHT EINE SCHRIFTLICHE EINWILLIGUNG ODER WIRD VOM GESETZ VERLANGT.

1.3 Voraussetzung

- Hollywood 5.2 oder besser
- MUI 3.8 oder besser
- 68020+ oder PowerPC
- codesets.library für UTF-8 Unterstützung
- Die Toolbar-Klasse benötigt TheBar.mcc 26.11 oder höher
- Die Texteditor-Klasse benötigt TextEditor.mcc 15.47 oder besser
- optional: CyberGraphX oder Picasso96 für RTG-Bildschirme
- optional: guigfx.library und render.library für Paletten-Bildschirme

1.4 Palettenbildschirme Unterstützung

Ab MUI Royale 1.4 kann das Plugin auch auf Paletten-basierten Bildschirmen verwendet werden, d.h. auf Bildschirmen mit Tiefen von 8 Bit und darunter. Diese Fähigkeit erfordert

jedoch die Installation von `guigfx.library` und `render.library`. Andernfalls funktioniert MUI Royale nicht auf Paletten-basierten Bildschirmen.

Wenn Ihr Programm nicht nur MUI-Fenster verwendet, sondern zusätzlich auch normale Hollywood-Displays öffnen muss, müssen Sie auch das Plananarama-Plugin installieren, das Hollywood auf palettenbasierten Bildschirmen ausführen lässt. Sie müssen zuerst `@REQUIRE Plananarama` aufrufen, bevor Sie `@REQUIRE MUI Royale` aufrufen, d.h. Ihr Skript muss so beginnen:

```
@REQUIRE "plananarama"  
@REQUIRE "muiroyale"
```

Wenn Ihr Skript nur MUI-Fenster verwendet, benötigen Sie jedoch kein Plananarama. `guigfx.library` und `render.library` sind dann ausreichend.

2 Über MUI Royale

2.1 Geschichte

In der Datei `history.txt` finden Sie ein vollständiges auf englisch verfasstes Änderungsprotokoll von MUI Royale.

2.2 Zukunft

Hier sind einige Dinge, die auf meiner Liste stehen:

- Unterstützung für Drag'n'Drop
- Unterstützung für mehr MUI-Klassen

2.3 Häufig gestellte Fragen

In diesem Abschnitt werden einige häufig gestellte Fragen behandelt. Bitte lesen Sie sie zuerst, bevor Sie in der Mailingliste oder im Forum nachfragen, da Ihr Problem hier möglicherweise behandelt wurde.

F: Die GUI ist nicht skalierbar. Ist es nicht einer der großen Vorteile von MUI, dass seine GUIs immer in der Größe veränderbar sind und MUI sich automatisch darum kümmert?

A: Ja, aber es gibt einige Dinge, die Sie beachten müssen, damit die Größenanpassung funktioniert. Wenn sich in Ihrer GUI ein Gadget befindet, das eine feste Größe hat, müssen Sie es mit größenverstellbaren `<rectangle>` Objekten auf seinen Seiten auffüllen. Dann wird Ihre GUI wieder skalierbar sein. Stellen Sie sich zum Beispiel vor, Sie haben ein 64x64 `<image>` Objekt in einer horizontalen Gruppe in Ihrem Fenster. MUI kann die Größe dieses Fensters nicht ändern, es sei denn, Sie fügen den Seiten Ihres Bild-Objekts `Rectangle`-Objekte (Rechteck) hinzu, da MUI ein Objekt finden muss, dessen Größe angepasst werden kann. Daher müssen Sie darauf achten, dass nicht veränderbare Objekte in Ihrer GUI immer mit veränderbaren aufgefüllt werden. Übrigens, seien Sie vorsichtig mit dem Tag `<label>`: Objekte der Label-Klasse (Beschriftung) sind nicht skalierbar! Wenn Sie in der Größe veränderbare Beschriftungen verwenden möchten, verwenden Sie stattdessen `<text>`.

F: Warum sind Beschriftungen nicht skalierbar?

A: Ehrlich gesagt, ich weiß es nicht. Es gibt ein `Label.FreeVert` Attribut, aber das entsprechende `Label.FreeHoriz` fehlt. Ich weiß nicht wirklich, warum Stefan Stuntz es so gemacht hat, aber ich bin mir sicher, dass dahinter ein paar Gründe stecken.

F: Warum gibt es keine Attribute wie `Application.Author`, `Application.Copyright`, `Application.Description`, `Application.Title` und `Application.Version`?

A: Diese Attribute wurden bewusst weggelassen, um Kollisionen mit Hollywood zu vermeiden. Hollywood bietet bereits die Präprozessor-Anweisungen `@APPAUTHOR`, `@APPCOPYRIGHT`, `@APPDESCRIPTION`, `@APTITLE`, und `@APPVERSION`. Der Inhalt dieser Präprozessor-Anweisung wird beim Erstellen automatisch an das MUI-Anwendungs-Objekt übergeben.

2.4 Danksagungen

MUI Royale wurde von Andreas Falkenhahn geschrieben. Danke an Stefan Stuntz für sein wunderbares MUI-Toolkit, Alfonso Ranieri für TheBar.mcc und codesets.library, Allan Odgaard für TextEditor.mcc und die Open-Source-Teams für TheBar.mcc und TextEditor.mcc, um diese Klassen zu pflegen und alte Fehler zu beheben.

Ein besonderer Dank geht an Helmut Haake und Dominic Widmer für die Übersetzung des Handbuchs ins Deutsche. Fehler oder Verbesserungsvorschläge bzgl. des deutschen Handbuchs bitte an das Übersetzungsteam richten, welches unter handbuch@gmx.ch oder <https://amiga-resistance.info> erreicht werden kann.

Wenn Sie mich kontaktieren möchten, senden Sie bitte eine E-Mail an andreas@airsoftsoftwair.de oder nutzen Sie das Kontaktformular unter <http://www.hollywood-mal.com>.

3 Grundlegende MUI-Konzepte

3.1 Anwendungsbaum

Eine MUI-Anwendung besteht aus einem (manchmal sehr) großen Objektbaum. Der Stamm dieses Baums ist immer eine Instanz der Application-Klasse (Anwendung), Anwendungs-Objekt genannt. Dieses Anwendungs-Objekt behandelt die verschiedenen Kommunikationskanäle wie Benutzereingaben über Fenster, ARexx Befehle oder Commodities-Nachrichten.

Ein Anwendungs-Objekt selbst würde ausreichen, um nicht-GUI-Programme mit nur ARexx und Commodities-Funktionen zu erstellen. Wenn Sie Fenster mit vielen netten Gadgets und anderen Benutzeroberflächen-Dingen haben wollen, müssen Sie Ihrer Anwendung Fenster-Objekte hinzufügen. Das Anwendungs-Objekt ist in der Lage beliebig, viele untergeordnete Objekte zu verwalten. Die Anzahl der Fenster ist nicht begrenzt.

Fenster-Objekte sind Instanzen der Window-Klasse und behandeln alle Aktionen, die mit dem Öffnen, Schließen, Verschieben, Ändern der Größe und Auffrischen von Intuitionsfenstern verbunden sind. Ein Fenster für sich selbst ist jedoch nicht von großem Nutzen, wenn kein Inhalt angezeigt werden soll. Fenster-Objekte benötigen deshalb immer ein sogenanntes Stamm-Objekt.

Mit diesem Stamm-Objekt erreichen wir schließlich die Gadget bezogenen Klassen des MUI-Systems. Diese Gadget bezogenen Klassen sind alle Unterklassen der Area-Klasse (Bereich), Sie beschreiben eine Rechteckregion mit einigen klassenabhängigen Inhalten. Viele verschiedene Klassen wie Zeichenketten, Schaltflächen, Auswahlkästchen oder Listenansichten sind verfügbar, aber die wichtigste Unterklasse der Area-Klasse ist wahrscheinlich die Group-Klasse (Gruppen). Instanzen dieser Klasse können mit einer beliebigen Anzahl untergeordneter Objekte umgehen und die Größe und Position dieser untergeordneten Objekte mit verschiedenen Attributen steuern. Natürlich können diese untergeordneten Elemente wieder Gruppen-Objekte mit anderen Gruppen von untergeordneten Elementen sein. Da Ihr Fenster normalerweise mehr als nur ein Objekt enthalten soll, muss das Stamm-Objekt eines Fensters immer ein Group-Klassen-Objekt sein.

MUI Royale GUIs werden vollständig mit der XML-Markup-Sprache definiert. Hier ist ein Beispiel, wie ein Anwendungsbaum in einer XML-Definition aussehen könnte:

```
<?xml version="1.0" encoding="iso-8859-1"?>
<application base="HELLOWORLD">
  <window title="Example GUI" muuid="MAIN">
    <vgroup>
      <listview>
        <column/>
      </listview>
      <string/>
      <hgroup>
        <button>Add</button>
        <button>Remove</button>
      </hgroup>
    </vgroup>
  </window>
```

</application>

Da diese ersten Abschnitte sehr wichtig sind, um zu verstehen, wie MUI funktioniert, hier eine kurze Zusammenfassung:

Eine Anwendung besteht aus genau einem Anwendungs-Objekt. Dieses Anwendungs-Objekt kann eine beliebige Anzahl von untergeordneten Objekten haben, von denen jedes ein Fenster-Objekt ist. Jedes Fenster-Objekt enthält ein Stamm-Objekt, das vom Typ Group-Klasse (Gruppen) sein muss. Dieses Gruppen-Objekt behandelt wieder eine beliebige Zahl von untergeordneten Objekten, entweder anderen Gruppen-Objekten oder einigen Benutzeroberflächenelementen wie Zeichenketten, Schieberegler oder Schaltflächen.

3.2 Klassenhierarchie

Es gibt zwei wichtige Oberklassen, die Sie kennen müssen, da fast alle anderen MUI-Klassen untergeordnete Elemente dieser beiden Oberklassen sind: Die erste ist die Notify-Klasse die das MUI-Ereignisbenachrichtigungssystem verarbeitet. Mit der Notify-Klasse können Ihre MUI-Objekte bestimmte Ereignisse überwachen und bei jedem Auftreten dieser Ereignisse Benachrichtigungen auslösen. Siehe [Abschnitt 30.1 \[Notify-Klasse\]](#), [Seite 181](#), für Details.

Die zweite wichtige Oberklasse ist die Area-Klasse. Die ist die Oberklasse für alle MUI-Objekte, die eine visuelle Darstellung in Form eines Gadgets in einem MUI-Fenster haben. Daher können Sie alle Attribute der Area-Klasse für alle Ihre MUI-Gadgets verwenden. Mit den Attributen `Area.Width` und `Area.Height` können Sie beispielsweise die Dimensionen einer Listenansicht, eines Texteditors oder eines Zeichenketten-Gadgets ermitteln, da dies alles Unterklassen der Area-Klasse sind. Siehe [Abschnitt 7.1 \[Area-Klasse\]](#), [Seite 57](#), für Details.

3.3 Automatisches Layout-Modul

Eines der wichtigsten und mächtigsten Eigenschaften von MUI ist sein dynamisches Layout-Modul. Im Gegensatz zu anderen verfügbaren Benutzeroberflächen-Tools braucht sich der Programmierer einer MUI-Anwendung nicht um Gadget-Größen und Positionen zu kümmern. MUI verarbeitet alle notwendigen Berechnungen automatisch und macht jedes Programm vollständig für Bildschirm, Fenstergröße und Schriftart sensibel, ohne die geringste Interaktion des Programmierers zu benötigen.

Aus der Sicht eines Programmierers müssen Sie lediglich einige rechteckige Bereiche definieren, die die Objekte enthalten sollen, die Sie in Ihrem Fenster sehen möchten. Objekte der Group-Klasse (Gruppen) werden für diesen Zweck verwendet. Diese Objekte sind nicht selbst sichtbar, sondern teilen ihren untergeordneten Elementen mit, ob sie horizontal oder vertikal erscheinen sollen.

Für ein automatisches und dynamisches Layout ist es wichtig, dass jedes einzelne Objekt seine minimalen und maximalen Dimensionen kennt. Bevor ein Fenster geöffnet wird, fragt MUI alle seine Gadgets nach diesen Werten und berechnet daraus die äußersten Größen des Fensters.

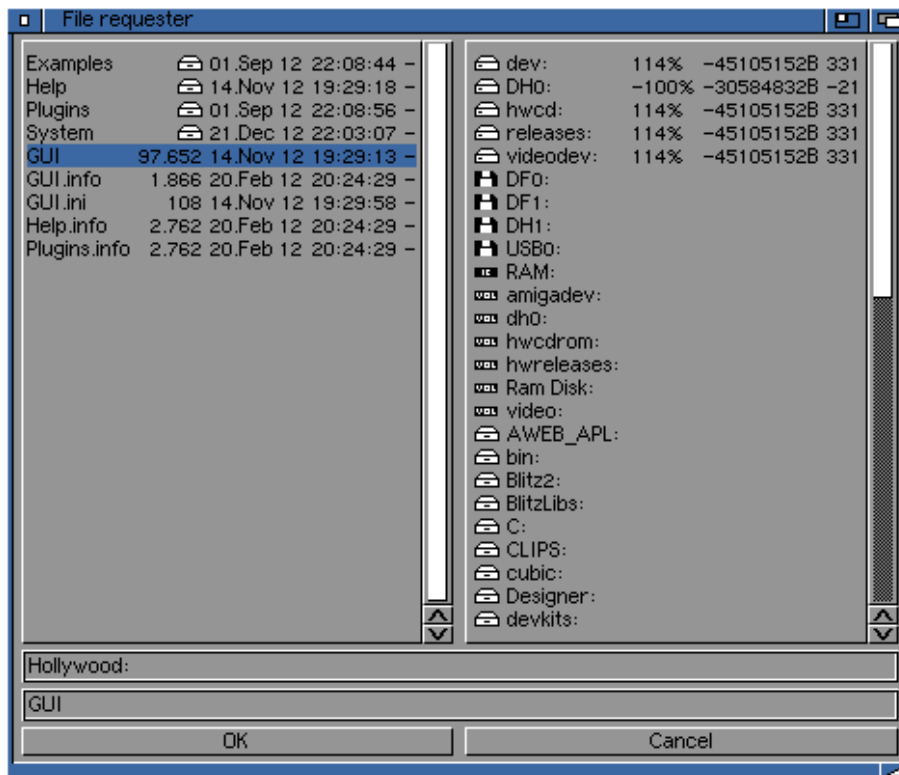
Sobald das Fenster geöffnet ist, findet das Layout statt. Beginnend mit der aktuellen Fenstergröße wird das Stamm-Objekt und alle seine untergeordneten Elemente abhängig vom Typ der Gruppe ihres übergeordneten Elementes und einigen zusätzlichen Attributen platziert. Der Algorithmus stellt sicher, dass Objekte niemals kleiner oder größer als ihre maximale Größe werden.

Die wichtige Sache bei diesem Mechanismus ist, dass die Objektplatzierung von der Fenstergröße abhängt. Dies ermöglicht eine sehr einfache Implementierung eines Größen-Gadgets: Wenn der Benutzer ein Fenster in der Größe ändert, startet MUI einfach einen neuen Layout-Prozess und berechnet automatisch die Positionen und Größen der Objekte neu. Es ist keine Programmiererinteraktion erforderlich.

Daher sollten Sie niemals explizit feste Größen für Ihre Fenster- sowie deren Gadgets definieren. Stattdessen berechnet MUI diese automatisch, abhängig von den aktuellen Benutzereinstellungen. Obwohl es möglich ist, mit `Window.Width` und `Window.Height` feste Größen für Ihr Fenster zu verwenden, ist es gegen die MUI-Philosophie, diese tatsächlich zu verwenden. Außerdem sollten die Gadget-Gegenstücke `Area.FixWidth` und `Area.FixHeight` nur verwendet werden, wenn Sie einen sehr guten Grund dafür haben. Normalerweise sollten Sie die Layoutberechnung sowie die Gadget- und Fenstergrößenbestimmung dem automatischen Layout-Modul von MUI überlassen, damit der Benutzer die ideale visuelle Darstellung Ihres GUI-Layouts erhält.

3.4 Gruppen-Objekte

Wie oben erwähnt, definiert ein Programmierer das Design eines Fensters, indem Objekte horizontal oder vertikal gruppiert werden. Betrachten wir als kleines Beispiel ein einfaches Dateidialogfenster:



Dieses Fenster besteht aus zwei Listenansicht-Objekten, zwei Zeichenketten-Gadgets und zwei Schaltflächen. Um MUI mitzuteilen, wie diese Objekte platziert werden sollen, müssen Sie Gruppen um sie herum definieren. Hier besteht das Fenster aus einer vertikalen Gruppe,

die eine horizontale Gruppe mit beiden Listen als erstes untergeordnetes Element, das Pfad-Gadget als zweites untergeordnetes Element, das Datei-Gadget als drittes untergeordnetes Element und wieder eine horizontale Gruppe mit beiden Knöpfen als viertes untergeordnetes Element enthält.

In einer XML-Datei könnte die GUI-Definition dieses Fensters dann so aussehen:

```
<?xml version="1.0" encoding="iso-8859-1"?>
<application base="FILEREQ">
  <window title="File requester">
    <vgroup>
      <hgroup>
        <listview><column/></listview>
        <listview><column/></listview>
      </hgroup>
      <string/>
      <string/>
      <hgroup>
        <button>OK</button>
        <button>Cancel</button>
      </hgroup>
    </vgroup>
  </window>
</application>
```

Diese XML-Definition reicht völlig aus, um den Inhalt unseres Fensters zu definieren, alle notwendigen Größen und Positionen werden vom MUI-System automatisch berechnet.

Um zu verstehen, wie diese Berechnungen funktionieren, ist es wichtig zu wissen, dass alle grundlegenden Objekte (z.B. Zeichenketten, Schaltflächen, Listen) eine feste minimale und eine maximale Größe haben. Gruppen-Objekte berechnen ihre minimalen und maximalen Größen von ihren untergeordneten Elementen, je nachdem, ob sie horizontal oder vertikal sind:

Horizontale Gruppen:

- Die minimale Breite einer horizontalen Gruppe ist die Summe aller minimalen Breiten ihrer untergeordneten Elemente.
- Die maximale Breite einer horizontalen Gruppe ist die Summe aller maximalen Breiten ihrer untergeordneten Elemente.
- Die minimale Höhe einer horizontalen Gruppe ist die größte minimale Höhe ihrer untergeordneten Elemente.
- Die maximale Höhe einer horizontalen Gruppe ist die kleinste maximale Höhe ihrer untergeordneten Elemente.

Vertikale Gruppen:

- Die minimale Höhe einer vertikalen Gruppe ist die Summe aller Mindesthöhen ihrer untergeordneten Elemente.
- Die maximale Höhe einer vertikalen Gruppe ist die Summe aller maximalen Höhen ihrer untergeordneten Elemente.

- Die minimale Breite einer vertikalen Gruppe ist die größte minimale Breite ihrer untergeordneten Elemente.
- Die maximale Breite einer vertikalen Gruppe ist die kleinste maximale Breite ihrer untergeordneten Elemente.

Vielleicht klingt dieser Algorithmus ein wenig kompliziert, aber in der Tat ist es wirklich geradlinig und stellt sicher, dass Objekte nicht kleiner werden als ihr Minimum oder größer als ihre maximale Größe.

Bevor ein Fenster geöffnet wird, fragt es sein Stamm-Objekt (normalerweise ein Gruppen-Objekt), um die minimale und maximale Größen zu berechnen. Diese Größen werden als Begrenzungsmaße des Fensters verwendet, die kleinste mögliche Fenstergröße führt dazu, dass alle Objekte in ihrer Mindestgröße angezeigt werden.

Sobald die minimalen und maximalen Größen berechnet sind, beginnt der Layoutprozess. Das Stamm-Objekt wird angewiesen, sich selbst in dem Rechteck zu platzieren, das durch die aktuelle Fenstergröße definiert wird. Diese Fenstergröße wird entweder vom Programmierer festgelegt oder resultiert aus einer Fenstergrößenänderung durch den Benutzer. Wenn ein Objekt aufgefordert wird, sich selbst zu gestalten, legt es einfach seine Position und seine Abmessungen auf das gegebene Rechteck fest. Im Fall eines Gruppen-Objekts verteilt ein mehr oder weniger komplizierter Algorithmus den gesamten verfügbaren Raum zwischen seinen untergeordneten Elementen und teilt sie dem Layout mit.

Dieser "mehr oder weniger komplizierte Algorithmus" ist für die Objektanordnung verantwortlich. Abhängig von einigen Attributen des Gruppen-Objekts (horizontal oder vertikal, ...) und einigen Attributen der untergeordneten Objekte (minimale und maximale Dimensionen, ...) wird der Raum verteilt und die untergeordneten Elemente platziert.

Ein kleines Beispiel macht die Dinge klarer. Schauen wir uns an, was in einem Fenster passiert, das nur drei horizontal gruppierte Farbfeld-Objekte enthält:



Farbfeld-Objekte haben eine minimale Breite und Höhe von einem Pixel und keine maximale Breite und Höhe. Da wir eine horizontale Gruppe haben, ergibt die oben erläuterte Minmax-Berechnung eine minimale Breite von drei Pixeln und eine minimale Höhe von einem Pixel für das Stamm-Objekt des Fensters (die horizontale Gruppe, die die Farbfelder enthält). Maximale Dimensionen der Gruppe sind unbegrenzt. Mithilfe dieser Ergebnisse kann MUI die Fensterbegrenzungsdimensionen berechnen, indem einige Abstandswerte und die Dicke der Fensterumrandung hinzugefügt werden.

Sobald die minimalen und maximalen Abmessungen berechnet sind, kann das Fenster durch den Programmierer oder einer benutzerdefinierten Größe geöffnet werden. Diese Größe ist der Ausgangspunkt für die folgenden Layoutberechnungen. Stellen wir uns für unser kleines Beispiel vor, dass die aktuelle Fenstergröße 100 Pixel breit und 50 Pixel hoch ist.

MUI subtrahiert die Fensterrahmen und einige Fensterinnenabstände und teilt dem Stamm-Objekt das Layout selbst mit, in das Rechteck links = 5, oben = 20, Breite = 90, Höhe = 74. Da unser Stamm-Objekt in diesem Fall eine horizontale Gruppe ist, weiß es, dass jedes Farbfeld die volle Höhe von 74 Pixeln erhalten kann und dass die verfügbare Breite von 90 Pixeln von allen drei Feldern geteilt werden muss. Die resultierenden Felder erhalten somit eine Breite von $90/3 = 30$ Pixel.

Das ist die grundlegende Art, wie MUIs Layoutsystem funktioniert. Es gibt viel mehr Möglichkeiten, das Layout zu beeinflussen. Sie können z.B. Objekten unterschiedliche Wichtigkeiten zuweisen, Abstände zwischen den Objekten definieren oder sogar zweidimensionale Gruppen erstellen. Diese komplexen Layout-Probleme werden in der Dokumentation der Gruppen behandelt.

3.5 Objekthandhabung

MUI Royale verwendet XML-Dateien zum Erstellen von MUI-Objekten. Beim Erstellen von Objekten können Sie alle Attribute verwenden, die im Bereich Anwendbarkeit der zugehörigen Attributdokumentation mit dem Symbol "I" gekennzeichnet sind. Um beispielsweise ein Zeichenketten-Objekt mit einem Zeichenkettenartigen Rahmen, einer maximalen Länge von 80 und dem Anfangsinhalt "foobar" zu erstellen, müssten Sie den folgenden XML-Code verwenden:

```
<string id="mystring" frame="string" contents="foobar" maxlen="80"/>
```

Sobald Ihr Objekt fertig ist, können Sie mit dem kommunizieren beginnen, indem Sie eines seiner Attribute setzen respektive abrufen oder indem Sie ihm Methoden senden. Zu diesem Zweck ist es wichtig, dass Sie Ihrem Objekt einen Identifikator geben, indem Sie den Tag "id" verwenden, damit die Funktionen `mui.Set()`, `mui.Get()` und `mui.DoMethod()` Ihr Objekt finden können. Hier ein Beispiel, wie Sie mit Ihrem Zeichenketten-Gadget kommunizieren:

```
mui.Set("mystring", "contents", "look")
s$ = mui.Get("mystring", "contents")
DebugPrint("Always " .. s$ .. " on the bright side of life.")
```

Wie bereits oben erwähnt, sind alle Attribute und Methoden in der Dokumentation, die mit dieser Distribution geliefert wird, vollständig dokumentiert.

3.6 Benachrichtigungen

Das zentrale Element zur Steuerung einer MUI-Anwendung ist der Benachrichtigungsmechanismus. Um zu verstehen, wie er funktioniert, ist es wichtig zu wissen, dass die meisten Objekte viele Attribute enthalten, die ihren aktuellen Zustand definieren. Die Benachrichtigung ermöglicht es, auf Änderungen dieser Attribute zu reagieren.

Attribute werden entweder direkt vom Programmierer (mit einem Aufruf von `mui.Set()`) oder durch die Benutzermanipulation einiger Gadgets geändert. Wenn er z.B. den aktiven Eintrag eines Auswahl-Listefeld-Gadgets ändert, wird das `Cycle.Active` Attribut fortlaufend aktualisiert und spiegelt den aktuell aktiven Eintrag wider.

Wenn Sie informiert werden möchten, wenn sich der Wert eines bestimmten Objektattributs ändert, müssen Sie zunächst eine Benachrichtigung für dieses Attribut in Ihrer Objektdeklaration einrichten. Dies geschieht direkt in der XML-Datei mit dem Tag "notify":

```
<cycle id="mycycle" notify="active">
```

```

    <item>One</item>
    <item>Two</item>
    <item>Three</item>
</cycle>

```

Wenn Sie mehrere Benachrichtigungen für dasselbe MUI-Objekt überwachen möchten, müssen Sie sie mit einem Semikolon trennen, z.B:

```

<cycle id="mycycle" notify="active; appmessage">
...
</cycle>

```

Beachten Sie, dass Sie Benachrichtigungen zur Laufzeit auch mit dem Befehl `mui.Notify()` in Ihrem Code einrichten oder entfernen können.

Als nächstes müssen Sie einen Ereignis-Handler für Ereignisse von MUI Royale mit dem Hollywood-Befehl `InstallEventHandler()` einrichten. Sie müssen eine Hollywood-Funktion übergeben, die als Ereignishandler für den Befehl `InstallEventHandler()` fungiert. Hier ist ein Beispiel:

```
InstallEventHandler({MUIRoyale = p_MUIHandler})
```

Immer wenn ein MUI-Ereignis auftritt, ruft MUI Royale die Callback-Funktion für den Ereignishandler auf, die Sie an `InstallEventHandler()` übergeben haben. Die Funktion erhält eine Tabelle als Parameter mit den folgenden initialisierten Feldern:

Action: Initialisiert "MUIRoyale".

Class: Enthält den Namen der MUI-Klasse, von der dieses Ereignis stammt, z.B. "Cycle".

Attribute: Enthält den Namen des Klassenattributs, das das Ereignis ausgelöst hat, z.B. "Active".

ID: Enthält die ID des MUI-Objekts, das dieses Ereignis ausgelöst hat, z.B. "mycycle".

TriggerValue: Enthält den aktuellen Wert des Attributs, das dieses Ereignis ausgelöst hat. Sie können dies auch herausfinden, indem Sie ein `mui.Get()` für das Objekt ausführen, aber es ist bequemer, den aktuellen Wert direkt zu Ihrer Ereignis-Callback-Funktion zu bekommen.

MUIUserData: Wenn dem Objekt bestimmte Benutzerdaten zugewiesen wurden, wird es an die Ereignis-Handler-Callback-Funktion in diesem Tag übergeben. Siehe [Abschnitt 30.12 \[Notify.UserData\], Seite 185](#), für Details.

NotifyData: Wenn dem Objekt bestimmte Benachrichtigungsdaten zugewiesen wurden, wird es an die Callback-Funktion des Ereignishandlers in diesem Tag übergeben. Siehe [Abschnitt 30.10 \[Notify.NotifyData\], Seite 184](#), für Details.

3.7 Anwendbarkeit

In der Dokumentation jedes Objektattributs finden Sie Informationen über die Anwendbarkeit dieses Attributs. Die Attribut-Anwendbarkeit wird in Form einer Kombination der vier Buchstaben I, S, G und N beschrieben. Dies teilt Ihnen die verschiedenen Kontexte mit, in denen das Attribut verwendet werden kann.

Hier ist eine Erklärung der verschiedenen Anwendbarkeitskontexte:

- I Attribute können beim Erstellen des Objekts in der XML-Datei verwendet werden. (Initialisierungszeit)
- S Attribut kann zur Laufzeit mit `mui.Set()` verwendet werden.
- G Attribut kann zur Laufzeit mit `mui.Get()` verwendet werden.
- N Benachrichtigungen für dieses Attribut sind möglich, indem Sie den Tag "Notify" in der XML-Definition verwenden oder die Funktion `mui.Notify()` zur Laufzeit aufrufen.

Wenn zum Beispiel ein Attribut eine Anwendbarkeit von nur "I" hat, dann kann dieses Attribut nur während der Objektinitialisierungszeit verwendet werden. Es kann später nicht mit `mui.Set()` geändert werden. Wenn ein Attribut eine Anwendbarkeit von nur "S" hat, ist es andererseits nicht möglich, das Attribut bereits zur Initialisierungszeit in der XML-Datei anzugeben. Attribute, die eine Anwendbarkeit von "ISGN" haben, können in allen Kontexten verwendet werden.

3.8 Tastaturabfolgekette

Einer der großen Vorteile von MUI ist, dass es auch leistungsstarke Funktionen zur Steuerung Ihrer GUI über die Tastatur bietet. Zu diesem Zweck sollten Sie eine Tastaturabfolgekette für alle Ihre Fenster einrichten. Immer wenn der Benutzer die TAB-Taste drückt, wird das nächste MUI-Objekt in der Tastaturabfolgekette aktiviert.

Alles, was Sie tun müssen, um der Tastaturabfolgekette ein MUI-Objekt hinzuzufügen, ist das Attribut `Area.CycleChain` auf `True` zu setzen. Um beispielsweise der Tastaturabfolgekette eine Listenansicht hinzuzufügen, führen Sie einfach Folgendes in Ihrer XML-Definition aus:

```
<listview cyclechain="true">
  <column/>
</listview>
```

Objekte vom Typ `Button`-Klasse müssen nicht explizit zur Tastaturabfolgekette hinzugefügt werden. Als Komfortfunktion fügt MUI Royale automatisch alle Tasten zur Tastaturabfolgekette hinzu. Wenn Sie dies nicht möchten, müssen Sie `Area.CycleChain` für alle Ihre Schaltflächen auf `False` setzen, obwohl es dafür eigentlich keinen vernünftigen Grund gibt. Bitte stellen Sie sicher, dass Sie alle Ihre MUI-Objekte immer der Tastaturabfolgekette hinzufügen, da dies Ihre GUI für Nur-Tastatur-Benutzer viel attraktiver macht.

3.9 Textformatierungscodes

Das Text-Modul von MUI ist sehr leistungsfähig und ermöglicht es Ihnen, spezielle Zeichen-codes für die Textformatierung on-the-fly einzufügen. Zum Beispiel ist es möglich, Textausrichtung und Schriftstil (fett, kursiv, usw.) mit Formatierungscodes festzulegen. Mit MUI

3.9 oder besser ist es sogar möglich, Bilder (aus einer Hollywood-Pinselquelle) in Ihre Text-Objekte einzubetten.

Formatierungs-codes beginnen immer mit einem Escape-Zeichen, gefolgt von einer Zeichenfolge, die den Formatierungscode beschreibt. In der Dezimalschreibweise entspricht das Escape-Zeichen dem ASCII-Code 27, der in Oktalzahl 33 und in Hexadezimalschreibweise \$1B ist. Bei der Verwendung von Formatierungs-codes ist Vorsicht geboten, da sie in XML-Dateien und Hollywood-Quelldateien unterschiedlich verwendet werden. In XML-Dateien wird die Oktalnotation verwendet, d.h. Sie starten eine Escape-Sequenz mit einem umgekehrten Schrägstrich und der Oktalzahl 33 ('\33'). In Hollywood-Quellcodes hingegen werden Oktalzahlen nach einem umgekehrten Schrägstrich jedoch nicht unterstützt. Hollywood erwartet immer nach einem umgekehrten Schrägstrich den ASCII-Code in Dezimalschreibweise. Deshalb müssen Sie '\27' verwenden, um eine Escape-Sequenz aus Hollywood-Codes zu initiieren.

Um diesen Unterschied ein wenig besser zu veranschaulichen, lassen Sie uns zwei Beispiele betrachten. Hier ist ein Beispiel für das Erstellen eines fett formatierten Text-Objekts in einer XML-Datei. Fett gedruckter Text wird aktiviert, indem das Zeichen "b" nach dem Escape-Zeichen verwendet wird:

```
<text id="mytext">\33bBold text</text>
```

Sie können sehen, dass die Oktalnotation hier verwendet wird, da XML-Dateien nach einem umgekehrten Schrägstrich eine Oktalzahl erwarten. In Hollywood ist dies jedoch anders, da Hollywood nach einem umgekehrten Schrägstrich ein Dezimalzeichen erwartet. So müssen Sie hier die Escape-Codes angeben, wenn Sie sie aus einer Hollywood-Quelldatei verwenden:

```
mui.Set("mytext", "contents", "\27bBold text")
```

Sie können sehen, dass der Code derselbe ist, außer dass wir \27b anstelle von \33b verwenden, da Hollywood nach einem umgekehrten Schrägstrich immer Dezimalzahlen anstelle von Oktalzahlen verwendet.

Wenn Sie XML-Code aus dem Hollywood-Skript an MUI Royale übergeben und Textformatierungs-codes verwenden möchten, müssen Sie sicherstellen, dass der umgekehrte Schrägstrich entsprechend entkoppelt wird, z.B. so:

```
mui.CreateObject("<text id=\"mytext\">\\33bBold text</text>")
```

Der umgekehrte Schrägstrich muss hier weggelassen werden, da er im XML-Code wörtlich vorhanden sein muss. Wenn wir nur einen umgekehrten Schrägstrich verwenden würden, würde das Escape-Zeichen in das XML eingefügt werden, was nicht erlaubt ist.

Die folgenden Formatierungs-codes werden derzeit unterstützt:

- \33u Stellt den Stil auf Unterstreichung ein.
- \33b Stellt den Stil auf fett ein.
- \33i Stellt den Stil kursiv ein.
- \33n Stellt den Stil wieder auf normal ein.
- \33<n> Verwendet die Stiftnummer n (2..9) als vorderen Stift. n muss ein gültiger DrawInfo-Stift sein, wie in `intuition/screens.h` angegeben.

- `\33P[RRGGBB]` Ändert die Frontfarbe in die angegebene RGB-Farbe. Die RGB-Farbe muss in Form von sechs Hexadezimalziffern RRGGBB angegeben werden. Dies ist nur auf True-Color-Bildschirmen möglich. Es erfordert auch mindestens MUI 3.9.
- `\33P[AARRGGBB]` Ändert die Frontfarbe in die angegebene RGB-Farbe und wendet die Alpha-Mischung mit der angegebenen Intensität an. Die RGB-Farbe muss in Form von sechs Hexadezimalziffern RRGGBB angegeben werden, denen zwei Alpha-Kanalziffern AA vorangestellt werden müssen, die die Mischintensität angeben. Dies ist nur auf True-Color-Bildschirmen möglich. Es erfordert auch mindestens MUI 3.9. Aufgrund von Einschränkungen in Picasso96 funktioniert das mit AmigaOS 4 noch nicht (erst ab AmigaOS 4.1 Update 6).
- `\33c` Zentriert die (und folgende) Zeile(n). Diese Sequenz ist nur am Anfang einer Zeichenfolge oder nach einem Zeilenvorschubzeichen gültig.
- `\33r` Richtet die aktuelle (und folgende) Zeile(n) rechts aus. Diese Sequenz ist nur am Anfang einer Zeichenfolge oder nach einem Zeilenvorschubzeichen gültig.
- `\33l` Richtet die aktuelle (und nachfolgende) Zeile(n) links aus. Diese Sequenz ist nur am Anfang einer Zeichenfolge oder nach einem Zeilenvorschubzeichen gültig.
- `\33A[s]` Schließt den Hollywood-Pinsel ein, der den Identifikator <s> im Text-Objekt verwendet. Diese Funktion erfordert mindestens MUI 3.9.
- `\33-` Deaktiviert das Text-Modul. Die folgenden Zeichen werden ohne weitere Analyse ausgegeben.
- `\n` Startet eine neue Zeile. Mit diesem Zeichen können Sie z.B. mehrzeilige Schaltflächen erstellen.

Hinweis: Diese Formatierungscodes können nicht nur in Text-Objekten, sondern in allen MUI-Zeichenketten verwendet werden. Sie können z.B. die Spalten einer Listenansicht formatieren oder Bilder in die Einträge eines Auswahl-Listefeld-Gadgets einfügen.

3.10 Online-Hilfe implementieren

MUI bietet mehrere Möglichkeiten, um direkte und sofortige Online-Hilfe in Ihren Anwendungen zu implementieren. Zunächst einmal kann jedes MUI-Gadget eine Blasen-Hilfefunktionalität aufweisen, die angezeigt wird, wenn die Maus für einen benutzerdefinierten Intervall über einem Gadget schwebt. Dies geschieht über das Attribut `Area.ShortHelp`. Da die `Area`-Klasse die Oberklasse für alle MUI-Gadgets ist, können Sie dieses Attribut verwenden, um allen Ihren Gadgets einen Hilfetext hinzuzufügen. Hier ist ein Beispiel:

```
<vgroup>
  <listview shorthelp="List of loaded video files">
    <column/>
  </listview>
  <button shorthelp="Plays a video stream">Play</button>
  <button shorthelp="Stops a video stream">Stop</button>
```

```
<button shorthelp="Exits the program.">Quit</button>
</vgroup>
```

Außerdem können Sie eine externe AmigaGuide-Datei für Ihre Anwendung angeben, die geöffnet wird, wenn der Benutzer die Taste HELP drückt. Dies geschieht durch Festlegen des `Application.HelpFile` Attributs. Hier ist ein Beispiel:

```
<application helpfile="PROGDIR:MyProgram.guide">
...
</application>
```

Um Ihre Online-Hilfefunktion für den Benutzer noch komfortabler zu gestalten, können Sie den Zugriff auf die externe AmigaGuide-Datei optimieren, indem Sie angeben, welcher Knoten oder Zeile der AmigaGuide-Datei angezeigt werden soll, wenn sich die Maus über einem bestimmten Objekt befindet. Dies geschieht mithilfe der Attribute `Notify.HelpNode` oder `Notify.HelpLine`. Angenommen, der Knoten "VideoPlayback" sollte angezeigt werden, wenn sich die Maus über der Schaltfläche "Play" befindet und der Benutzer die Taste HELP drückt. Sie können dies als folgenden XML-Code schreiben:

```
<button helpnode="VideoPlayback">Play</button>
```

Natürlich muss Ihre Anwendung `Application.HelpFile` zuerst definiert werden, wenn Sie `Notify.HelpNode` verwenden möchten.

Wenn Ihr Hilfesystem nicht so detailliert ist, dass Sie für jedes Gadget einen eigenen Knoten haben und Sie nur einen eigenen Knoten für jedes Fenster haben, können Sie `Notify.HelpNode` auch auf Fensterebene verwenden. In etwa so:

```
<window helpnode="VideoPlayback">
...
</window>
```

Wenn dieses Fenster aktiv ist und der Benutzer HELP drückt, öffnet MUI automatisch die in `Application.HelpFile` angegebene AmigaGuide-Datei und zeigt den in `Notify.HelpNode` angegebenen Knoten an. Sie sehen, dass MUI es den Programmierern sehr leicht macht, in ihren Anwendungen ein komfortables Hilfesystem zu implementieren. Alles, was Sie tun müssen, ist es für Ihre eigene Anwendung zu verwenden und Ihre Endnutzer werden dafür sicherlich dankbar sein.

3.11 Kontextmenüs

In MUI-GUIs kann jedes Gadget über ein eigenes Kontextmenü verfügen, das automatisch geöffnet wird, wenn sich der Mauszeiger über dem Gadget befindet und der Benutzer die rechte Maustaste gedrückt hält. Kontextmenüs werden den verschiedenen Gadgets mithilfe des `Area.ContextMenu` Attributs zugewiesen. Da die `Area`-Klasse die Oberklasse für alle MUI-Gadgets ist, können Sie dieses Attribut für jedes MUI-Objekt verwenden, das eine visuelle Darstellung in Form eines Gadgets aufweist.

`Area.ContextMenu` erwartet ein MUI-Menüleisten-Objekt als Argument, sodass Sie zunächst eine Menüleiste für Ihr Kontextmenü in XML erstellen müssen. Es ist sehr wichtig zu beachten, dass Sie Ihre Menüleiste im `<application>` Umfang definieren müssen, da Menüleisten globale Objekte sind und erst später an Fenster oder Gadgets angehängt werden. Aus diesem Grund ist es nicht erlaubt, Menüleisten in einem `<window>` XML-Bereich zu definieren.

Hier ist ein Beispiel, in dem wir ein Kontextmenü für Ausschneiden, Kopieren und Einfügen in ein Objekt vom Typ `Texteditor`-Klasse einfügen:

```
<menustrip id="ctxtmenu">
  <menu title="Context menu">
    <item>Cut</item>
    <item>Copy</item>
    <item>Paste</item>
  </menu>
</menustrip>

<window>
  ...
  <texteditor contextmenu="ctxtmenu"/>
  ...
</window>
```

Bitte beachten Sie, dass Menüleisten, die als Kontextmenüs verwendet werden, nicht mehr als einen `<menu>` Baum enthalten dürfen.

3.12 Drag'n'drop

Leider unterstützt MUI Royale derzeit nicht alle Drag & Drop-Funktionen von MUI. Es ist jedoch möglich, Drag'n'Drop-Operationen zu verwenden, die von der Workbench in Form von App-Nachrichten kommen.

Wenn Ihr Fenster diese Nachrichten akzeptieren kann, müssen Sie zuerst das Attribut `Window.AppWindow` auf `True` festlegen. Danach müssen Sie definieren, welche Ihrer MUI-Gadgets abgelegte Dateien akzeptieren sollen. Dies geschieht, indem Sie eine Benachrichtigung für das Attribut `Notify.AppMessage` für die gewünschten Gadgets einrichten. Da die `Notify`-Klasse die Oberklasse für alle MUI-Gadgets ist, können Sie dieses Attribut für jedes MUI-Objekt verwenden, das eine visuelle Darstellung in Form eines Gadgets aufweist. Wenn Sie keine speziellen Gadgets als Drop-Ziele haben möchten, können Sie auch das gesamte Fenster als Drop-Ziel einrichten, indem Sie einfach eine Benachrichtigung für das Attribut `Notify.AppMessage` für das gesamte Fenster einrichten.

Immer wenn der Benutzer ein Symbol über Ihrem Fenster oder MUI-Gadget ablegt, erhalten Sie eine Benachrichtigung vom Typ `AppMessage`. Siehe [Abschnitt 30.2 \[Notify.AppMessage\]](#), [Seite 181](#), für Details.

Wenn Ihre Anwendung symbolisiert ist und der Benutzer Dateien auf Ihrem App-Symbol ablegt, können Sie das Attribut `Application.DropObject` verwenden, um das MUI-Objekt zu definieren, das die abgelegten Dateien erhalten soll.

3.13 Zeichenkodierung

MUI Royale unterstützt zwei Zeichenkodierungen in den XML-Dateien, die zur Beschreibung des GUI-Layouts verwendet werden: `iso-8859-1` und `utf-8`. Wenn Sie die `utf-8` Kodierung verwenden, wird MUI Royale die Zeichensatzkonvertierung von UTF-8 in den Standard-Zeichensatz des Systems von der `codesets.library` verlangen.

Bitte beachten Sie, dass MUI selbst UTF-8 nicht unterstützt. MUI verwendet immer den Standard-Zeichensatz des Systems. Wenn Sie also UTF-8-Codierung in Ihren XML-Dateien

verwenden, versucht MUI Royale, diese Zeichenketten dem Standard-Zeichensatz des Systems zuzuordnen. Dies kann nicht immer gelingen. Wenn beispielsweise der Standardzeichensatz des Systems ISO-8859-1 ist und die UTF-8-XML-Datei einige osteuropäische Zeichen verwendet, die nicht in ISO-8859-1 enthalten sind, werden sie nicht korrekt angezeigt. Sie werden nur dann korrekt angezeigt, wenn der Standard-Zeichensatz des Systems ebenfalls vorhanden ist.

3.14 Hollywood-Brücke

Ein mächtiges Merkmal von MUI Royale ist, dass es Ihnen ermöglicht, komplette Hollywood-Displays in Ihre MUI-GUIs mit Hollywood-Klasse einzubetten. Immer wenn Sie etwas auf ein Hollywood-Display zeichnen, das an eine Hollywood-Klasse angehängt ist, wird es automatisch auch auf Ihre MUI-GUI gezogen. Sie können sogar das Hollywood-Display ausblenden, so dass der Benutzer nicht einmal bemerkt, dass Hollywood im Hintergrund läuft. Darüber hinaus werden alle Mausklicks und Tastatureingaben, die innerhalb der Hollywood-Klasse passieren, als normale Hollywood-Ereignisse an das entsprechende Hollywood-Display weitergeleitet. Somit können Sie mit der Hollywood-Klasse fast alle leistungsstarken Funktionen von Hollywood auch in einer MUI-GUI nutzen.

Schauen wir uns ein Beispiel an. Der folgende Code verwendet die Hollywood-Klasse zum Einbetten einer Spielanimation der Größe 320x200 in einer MUI-GUI. Dazu wird die Größe des Hollywood-Displays auf 320x200 geändert und dann in die MUI-GUI mit einem MUI-Objekt vom Typ Hollywood-Klasse eingebettet. Hier ist zuerst die XML-GUI-Definition:

```
<?xml version="1.0" encoding="iso-8859-1"?>
<application base="HOLIBRIDGE">
  <window title="Hollywood bridge" muuid="MAIN" notify="closerequest">
    <vgroup>
      <hgroup>
        <rectangle/>
        <hollywood display="1"/>
        <rectangle/>
      </hgroup>
      <hgroup>
        <button id="play" notify="pressed">Play</button>
        <button id="stop" notify="pressed">Stop</button>
      </hgroup>
    </vgroup>
  </window>
</application>
```

Beachten Sie, dass wir MUI-Objekte der Rectangle-Klasse verwenden, um das nicht veränderbare Hollywood-Objekt zu füllen. Dies ist notwendig, damit die GUI in der Größe veränderbar bleibt. Hier ist der Code, der Ihnen zeigt, wie Sie MUI und Hollywood verbinden:

```
@ANIM 1, "amy_walks.anim"
@DISPLAY {Width = 320, Height = 200, Hidden = True}
```

```

Function p_AnimFunc()
    Local numframes = GetAttribute(#ANIM, 1, #ATTRNUMFRAMES)
    curframe = Wrap(curframe + 1, 1, numframes + 1)
    DisplayAnimFrame(1, 0, 0, curframe)
EndFunction

Function p_EventFunc(msg)
    Switch msg.Class
    Case "Window":
        Switch msg.Attribute
        Case "CloseRequest":
            End
        EndSwitch
    Case "Button":
        Switch msg.Attribute
        Case "Pressed":
            Switch msg.ID
            Case "play":
                SetInterval(1, p_AnimFunc, 50)
            Case "stop":
                ClearInterval(1)
            EndSwitch
        EndSwitch
    EndSwitch
EndFunction

InstallEventHandler({MUIRoyale = p_EventFunc})
mui.CreateGUI(FileToString("GUI.xml"))

Repeat
    WaitEvent
Forever

```

Mit etwas mehr Arbeit können wir die Animation auch mit der Maus bewegen. Der Benutzer kann dann auf das Hollywood-Objekt klicken und die Animation mit der Maus ziehen. Hier ist der Code dafür:

```

@ANIM 1, "amy_walks.anim"
@DISPLAY {Width = 320, Height = 200, Hidden = True}

Function p_AnimFunc()
    Local numframes = GetAttribute(#ANIM, 1, #ATTRNUMFRAMES)
    curframe = Wrap(curframe + 1, 1, numframes + 1)
    SelectBrush(1)
    Cls
    DisplayAnimFrame(1, offx, offy, curframe)
    EndSelect
    DisplayBrush(1, 0, 0)

```

```

EndFunction

Function p_MouseFunc()
  If IsLeftMouse() = True
    Local mx, my = MouseX(), MouseY()
    If (grabx = -1) And (graby = -1) Then
      grabx, graby = mx - offx, my - offy
      offx = mx - grabx
      offy = my - graby
    Else
      grabx, graby = -1, -1
    EndIf
  EndFunction

Function p_EventFunc(msg)
  Switch msg.Class
  Case "Window":
    Switch msg.Attribute
    Case "CloseRequest":
      End
    EndSwitch
  Case "Button":
    Switch msg.Attribute
    Case "Pressed":
      Switch msg.ID
      Case "play":
        SetInterval(1, p_AnimFunc, 50)
        SetInterval(2, p_MouseFunc, 20)
      Case "stop":
        ClearInterval(1)
        ClearInterval(2)
      EndSwitch
    EndSwitch
  EndSwitch
EndFunction

CreateBrush(1, 320, 200)

InstallEventHandler({MUIRoyale = p_EventFunc})
mui.CreateGUI(FileToString("GUI8.xml"))

Repeat
  WaitEvent
Forever

```

Schließlich ist es auch möglich, das Hollywood-Objekt in der Größe veränderbar zu machen, indem die Attribute `Hollywood.MinWidth`, `Hollywood.MinHeight`, `Hollywood.MaxWidth`

und `Hollywood.MaxHeight` spezifiziert werden. Wenn der Benutzer die Größe des MUI-Fensters ändert, erhält Ihr Hollywood-Bildschirm ein Ereignis von `SizeWindow`, das Sie mit dem Hollywood-Befehl `InstallEventHandler()` überwachen können. Wenn Sie ein in der Größe veränderbares Hollywood-Objekt verwenden, können Sie die beiden Objekte `<rectangle>` entfernen, die ausschließlich als Füllraum verwendet werden. Der XML-Code sieht dann so aus:

```
<?xml version="1.0" encoding="iso-8859-1"?>
<application base="HOLIBRIDGE">
  <window title="Hollywood bridge" muiid="MAIN" notify="closerequest">
    <vgroup>
      <hollywood display="1" minwidth="32" minheight="32"
        maxwidth="16384" maxheight="16384"/>
      <hgroup>
        <button id="play" notify="pressed">Play</button>
        <button id="stop" notify="pressed">Stop</button>
      </hgroup>
    </vgroup>
  </window>
</application>
```

Beachten Sie, dass wir 16384 für `Hollywood.MaxWidth` und `Hollywood.MaxHeight` angeben. Dies zeigt an, dass unser Hollywood-Objekt nicht wirklich eine maximale Größe hat. Deshalb setzen wir diese beiden Attribute auf die maximal zulässige Größe von 16384 Pixel. Dies sollte für die meisten Zwecke ausreichend groß sein.

Unser Code ist fast derselbe wie zuvor, mit der Ausnahme, dass wir jetzt das Ereignis `SizeWindow` behandeln müssen, um auf die Größenänderung der GUI-Ereignisse achten zu können. Hier ist der angepasste Code von oben:

```
@ANIM 1, "amy_walks.anim"
@DISPLAY {Width = 320, Height = 200, Hidden = True}

Function p_AnimFunc()
  Local numframes = GetAttribute(#ANIM, 1, #ATTRNUMFRAMES)
  curframe = Wrap(curframe + 1, 1, numframes + 1)
  SelectBrush(1)
  Cls
  DisplayAnimFrame(1, offx, offy, curframe,
    {Width = swidth, Height = sheight})
  EndSelect
  DisplayBrush(1, 0, 0)
EndFunction

Function p_MouseFunc()
  If IsLeftMouse() = True
    Local mx, my = MouseX(), MouseY()
    If (grabx = -1) And (graby = -1) Then
      grabx, graby = mx - offx, my - offy
      offx = mx - grabx
```



```

        offy = my - graby
    Else
        grabx, graby = -1, -1
    EndIf
EndFunction

Function p_EventFunc(msg)
    If msg.Action = "SizeWindow"
        swidth = msg.Width
        sheight = msg.Height
        CreateBrush(1, swidth, sheight)
        Return
    EndIf

    Switch msg.Class
    Case "Window":
        Switch msg.Attribute
        Case "CloseRequest":
            End
        EndSwitch
    Case "Button":
        Switch msg.Attribute
        Case "Pressed":
            Switch msg.ID
            Case "play":
                SetInterval(1, p_AnimFunc, 50)
                SetInterval(2, p_MouseFunc, 20)
            Case "stop":
                ClearInterval(1)
                ClearInterval(2)
            EndSwitch
        EndSwitch
    EndSwitch
EndFunction

swidth, sheight = 320, 200
CreateBrush(1, swidth, sheight)

InstallEventHandler({MUIRoyale = p_EventFunc, SizeWindow = p_EventFunc})
mui.CreateGUI(FileToString("GUI8.xml"))

Repeat
    WaitEvent
Forever

```

An diesem Beispiel können Sie sehen, dass Hollywood-Klasse wirklich sehr mächtig ist und viele innovative GUI-Ideen nur durch Ihre Kreativität begrenzt werden kann. Das obige

Beispiel ist übrigens auch in der MUI Royale-Distribution enthalten. Sie finden es in der Schublade `Examples/Hollywood`.

3.15 Styleguide

Hinweis: Diese Themen werden hier nicht nur zum Spaß diskutiert. Sie werden viele Benutzer ärgern, wenn Sie nicht auf sie achten!

Dateidialog:

Selbst wenn MUI eine Dateiliste und ein Datenträgerlisten-Objekt enthält und die Erstellung eines privaten Dateidialogs sehr einfach macht, sollten Sie immer die Möglichkeit bieten, einen Standard-ASL-Dialog für diesen Zweck aufzurufen. Fügen Sie einfach eine kleine Popup-Schaltfläche direkt unter Ihrem Datei-Zeichenketten-Gadget hinzu und alles wird in Ordnung sein. MUI bietet genau zu diesem Zweck ein `Filepopup`-Objekt an. Beachten Sie auch Folgendes: Viele Benutzer (einschließlich mir selbst) verschieben Programme mit nicht standardmäßigen Dateirequestern sofort in den Papierkorb.

Fenstergröße:

Mit MUI ist es sehr einfach, viele Gadgets in einem einzigen Fenster zu haben. Da Sie als Programmierer in der Regel ein leistungsstärkeres System mit höheren Grafikauflösungen als die meisten Ihrer Benutzer haben, neigen die Fenster dazu, zu groß zu werden. Sie sollten immer sicherstellen, dass alles, was Sie entwerfen, auf einen Standard 640x256 Bildschirm mit einer Topaz/8 Schriftart passt. Andernfalls wird MUI versuchen, sehr kleine Schriftarten oder virtuelle Gruppen zu verwenden, um Ihr Fenster anzupassen, sodass Ihre Anwendung schlecht aussieht und sich schlecht anfühlt.

Tastatursteuerung:

Auch wenn Sie ein "Nur-Maus"-Benutzer sind, fügen Sie Tastaturabfolgekette und Gadget-Verknüpfungen zu Ihrer Anwendung hinzu. Es ist nur sehr wenig Arbeit für Sie und hilft vielen Nutzern. Siehe [Abschnitt 3.8 \[Tastaturabfolgekette\]](#), [Seite 14](#), für Details.

Hintergrund:

MUI ermöglicht dem Benutzer, viele verschiedene Hintergründe für Objekte anzupassen. Auch wenn Sie diese Funktion nicht verwenden, sollten Sie Ihr Programm immer mit einer ausgefallenen Hintergrundmusterkonfiguration testen und prüfen, ob alle Ihre Schaltflächen wirklich Schaltflächenhintergründe haben, alle Ihre gerahmten Texte wirklich Texthintergrund haben usw.

Fügen Sie eine Verknüpfung zu den MUI-Einstellungen hinzu:

Jedes MUI-Programm sollte einen Menüpunkt enthalten, der es dem Benutzer ermöglicht, das Erscheinungsbild des Programms mit dem MUI-Programm zu konfigurieren. Dies kann sehr einfach durch Hinzufügen eines Menüelements erfolgen, das einfach die Methode `Application.OpenConfigWindow` für das Anwendungs-Objekt ausführt. Außerdem sollten Sie die Menüoptionen "About MUI" und "About MUI Royale" mit den Fenstern "About MUI" und "About MUI Royale" benutzen, die Sie mit den Methoden `Application.AboutMUI` bzw. `Application.AboutMUIRoyale` öffnen können.

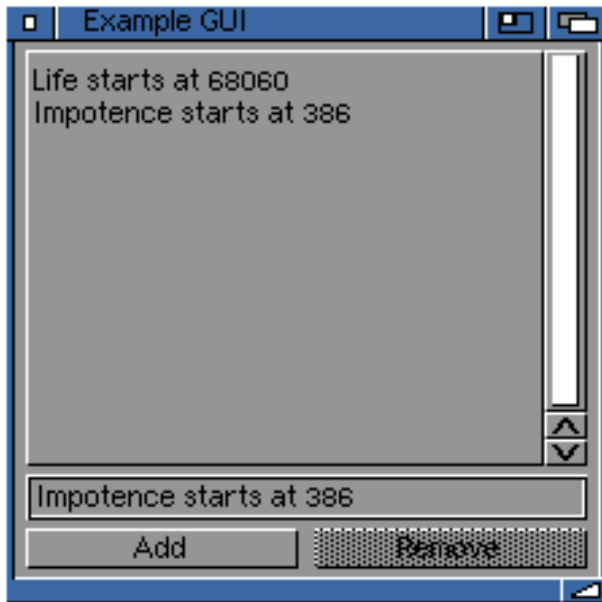
Und zuletzt ...

Vergessen Sie nicht den traditionellen Amiga-Styleguide!

4 Tutorial

4.1 Tutorial

Willkommen beim MUI Royale Tutorial! Dieses kleine Schritt-für-Schritt-Dokument führt Sie durch den Prozess der Erstellung Ihrer ersten GUI mit MUI Royale in wenigen Schritten.



Lassen Sie uns mit den Grundlagen beginnen: In MUI Royale GUIs werden XML-Dateien erstellt, die eine Beschreibung einer Anzahl von Fenstern enthalten, die eine Vielzahl von GUI-Elementen enthalten. Hier ist eine minimale GUI-Beschreibung für eine MUI Royale-GUI, die ein Fenster mit einer Listenansicht, einem Zeichenketten-Gadget und zwei Schaltflächen im XML-Format enthält:

```
<?xml version="1.0" encoding="iso-8859-1"?>
<application base="HELLOWORLD">
  <window title="Example GUI" muuid="MAIN">
    <vgroup>
      <listview>
        <column/>
      </listview>
      <string/>
      <hgroup>
        <button>Add</button>
        <button>Remove</button>
      </hgroup>
    </vgroup>
  </window>
</application>
```

Die Anwendungsbasis ist auf HELLOWORLD eingestellt. Es ist sehr wichtig, dass Sie immer das Attribut `Application.Base` definieren, da die von Ihnen in diesem Attribut angegebene

Zeichenfolge von MUI verwendet wird, um die programmspezifischen Einstellungen in einer externen Datei zu speichern (normalerweise in `ENV:MUI`). Das Objekt `<application>` ist das MUI-Haupt-Objekt für jede Anwendung und darf nur einmal pro Anwendung verwendet werden. Alle anderen MUI-Objekte sind untergeordnete Elemente der Application-Klasse.

Es ist auch wichtig, das Attribut `Window.MuiID` für jedes Fenster festzulegen, da nur Fenster mit einer MUI-ID ihre Position und Größe speichern. Außerdem können nur Fenster mit `Window.MuiID` vom Benutzer im Kontextmenü des Fensters als Momentaufnahme verwendet werden. Daher ist es sehr wichtig, dass Sie dieses Attribut für alle Ihre Fenster festlegen.

Um zu sehen, wie unsere obige XML-Definition als MUI-GUI aussieht, müssen wir sie in einer Datei namens `GUI.xml` speichern und dann den folgenden Code verwenden, um daraus eine vollständige MUI-GUI zu machen:

```
@DISPLAY {Hidden = True}

mui.CreateGUI(FileToString("GUI.xml"))

Repeat
    WaitEvent
Forever
```

Bitte beachten Sie, dass wir den Präprozessorbefehl `@DISPLAY` verwenden, um das Hollywood-Display auszublenden, das standardmäßig eingeblendet wird. Wir möchten dieses Display nicht haben, da wir bereits eine MUI-GUI haben, die die visuelle Darstellung unseres Programms darstellen soll. Denken Sie daran, dass das Hollywood-Display immer noch da ist. Es ist nur versteckt. Hollywoods Konzept besagt, dass es immer ein gültiges Display geben muss. Deshalb können wir das Hollywood-Display nicht komplett entfernen. Alles, was wir tun können, ist es zu verstecken. Sie können jedoch immer noch darauf zurückgreifen.

Sie sollten auch einige Informationen über Ihr Programm hinzufügen, indem Sie die Präprozessorbefehle `@APPAUTHOR`, `@APPCOPYRIGHT`, `@APPDESCRIPTION`, `@APTITLE` und `@APPVERSION` verwenden. MUI benötigt diese Information für verschiedene Zwecke, z.B. die in `@APTITLE` übergebenen Informationen werden vom MUI-Einstellungsfenster verwendet. Hier ist eine Beispieldeklaration dieser Präprozessorbefehle:

```
@APTITLE "Tutorial"
@APPVERSION "$VER: Tutorial 1.0 (27.12.12)"
@APPCOPYRIGHT "Copyright ©2012, Andreas Falkenhahn"
@APPAUTHOR "Andreas Falkenhahn"
@APPDESCRIPTION "The tutorial app from the MUI Royale guide"
```

Wenn Sie den obigen Code ausführen, werden Sie feststellen, dass das Fenster nicht auf Klicks auf das zugehörige Gadget reagiert. Sie müssen `CTRL-C` oder `Exchange` verwenden, um das Programm zu schließen. Natürlich möchten wir, dass unser Programm auch über das Schließ-Gadget des Fensters geschlossen werden kann. Um dies zu erreichen, müssen wir eine Benachrichtigung über das Attribut `Window.CloseRequest` einrichten, das immer dann gesetzt wird, wenn der Benutzer das Schließ-Gadget berührt. Wir können diese Benachrichtigung einrichten, indem wir unseren XML-Code von oben wie folgt ändern:

```
<window title="Example GUI" muiid="MAIN" notify="closerequest">
```

Als nächstes müssen wir eine Callback-Funktion mit dem Befehl `InstallEventHandler()` in Hollywood installieren, da unser Hollywood-Skript jedes Mal, wenn ein MUI Royale-Ereignis eintritt, informiert werden muss. Daher müssen wir in unserem Code folgendes ändern:

```
@DISPLAY {Hidden = True}

Function p_EventFunc(msg)

    Switch msg.Class
    Case "Window":
        Switch msg.Attribute
        Case "CloseRequest":
            End
        EndSwitch
    EndSwitch

EndFunction

InstallEventHandler({MUIRoyale = p_EventFunc})
mui.CreateGUI(FileToString("GUI.xml"))

Repeat
    WaitEvent
Forever
```

Was wir hier getan haben, ist die Installation der Funktion `p_EventFunc` als Ereignis-Handler-Callback-Funktion, der immer dann ausgeführt wird, wenn ein MUI-Royale-Ereignis kommt. Wenn wir ein solches Ereignis bekommen, müssen wir überprüfen, welche MUI-Klasse und welches Attribut es ausgelöst hat. Dies geschieht durch einen Blick in die Felder `msg.Class` und `msg.Attribute` der Ereignismeldung, die unsere Callback-Funktion als ersten Parameter erhält. Um auf das `Window.CloseRequest` Attribut reagieren zu können, müssen wir daher nach einem Ereignis aus der Klasse `Window` und dem Attribut `CloseRequest` suchen. Wenn wir ein solches Ereignis gefunden haben, rufen wir einfach den Befehl `End()` auf, um unser Programm zu beenden.

Als nächstes wollen wir die Funktionalität hinzufügen, die immer dann angezeigt wird, wenn der Benutzer die Schaltfläche "Hinzufügen" drückt. Der Text im Zeichenketten-Gadget sollte als letzter Eintrag zur Listenansicht hinzugefügt werden. Um dies zu tun, müssen wir zuerst einen Weg finden, unsere MUI-Gadgets aus dem Hollywood-Skript zu identifizieren. Dies geschieht durch Angabe von IDs in der XML-Definition. IDs sind einfache Textzeichenfolgen, die zum kommunizieren mit den MUI-Objekten aus Hollywood-Skripten verwendet werden. Fügen wir jetzt einige IDs für alle Gadgets hinzu, mit denen wir kommunizieren müssen. Wir müssen unsere XML-Definition wie folgt ändern:

```
...
    <listview id="mylistview">
        <column/>
    </listview>
    <string id="mystring"/>
```

```

<hgroup>
  <button id="mybt1">Add</button>
  <button id="mybt2">Remove</button>
</hgroup>

```

...

Jetzt müssen wir einige Benachrichtigungen einrichten, so dass unsere Callback-Funktion uns nicht nur über Änderungen vom Attribut `Window.CloseRequest` informiert, sondern auch über das Attribut `Button.Pressed`, welches auf `True` gesetzt wird, wenn der Benutzer auf eine Schaltfläche klickt. Um diese Klicks zu überwachen, müssen wir unseren Code folgendermaßen ändern:

...

```

  <button id="mybt1" notify="pressed">Add</button>
  <button id="mybt2" notify="pressed">Remove</button>

```

...

Nun, da wir dies getan haben, können wir unserer Callback-Funktion für den Ereignis-Handler-Code hinzufügen, der den Inhalt des Zeichenketten-Gadgets aufnimmt und am Ende der Liste in unserem Listenansicht-Objekt hinzugefügt wird. Dies geschieht, indem zuerst `mui.Get()` auf dem Attribut `String.Contents` aufgerufen wird, um den Inhalt des Zeichenketten-Gadgets abzurufen und dann die Methode `Listview.Insert` aus dem Listenansicht-Gadget mit `mui.DoMethod()`, um den Eintrag in die Listenansicht einzufügen. Hier ist der Code, der zu diesem Zweck in `p_EventFunc` eingefügt werden muss:

```

Switch msg.Class
...
Case "Button":
  Switch msg.Attribute
  Case "Pressed":
    Switch msg.ID
    Case "mybt1":      ; "Add" button was pressed
      Local s$ = mui.Get("mystring", "contents")
      mui.DoMethod("mylistview", "insert", "bottom", s$)
    EndSwitch
  EndSwitch
EndSwitch

```

Als nächstes wollen wir die Funktionalität unserer Schaltfläche "Remove" implementieren. Wenn diese Schaltfläche gedrückt wird, möchten wir, dass der aktive Eintrag aus der Listenansicht entfernt wird. Dazu können Sie die Methode `Listview.Remove` in der Listenansicht ausführen. Daher müssen wir unseren Code folgendermaßen ändern:

```

Switch msg.ID
...
Case "mybt2":      ; "Remove" button was pressed
  mui.DoMethod("mylistview", "remove", "active")
EndSwitch

```

Jetzt möchten wir, dass der aktive Eintrag der Listenansicht automatisch im Zeichenketten-Gadget angezeigt wird. Zu diesem Zweck müssen wir eine Benachrichtigung über das At-

tribut `Listview.Active` einrichten, das immer dann ausgelöst wird, wenn sich der aktive Eintrag der Listenansicht ändert. Daher müssen wir unsere XML-Datei wie folgt ändern:

```
...
    <listview id="mylistview" notify="active">
        <column/>
    </listview>
...
```

In unserer Callback-Funktion können wir diese Funktionalität ganz einfach implementieren, indem Sie die Methode `Listview.GetEntry` ausführen und dann den Inhalt des Zeichenketten-Gadgets mit dem Attribut `String.Contents` festlegen. Hier ist der Code dafür:

```
Switch msg.Class
...
Case "Listview":
    Switch msg.Attribute
    Case "Active":
        Local s$ = mui.DoMethod("mylistview", "getentry", "active")
        mui.Set("mystring", "contents", s$)
    EndSwitch
EndSwitch
```

Wenn Sie diesen Code ausprobieren, legen Sie fest, dass das Attribut `Listview.Active` nicht nur ausgelöst wird, wenn der Benutzer einen neuen Listenansichtseintrag mit der Maus auswählt, sondern auch, wenn Einträge aus der Listenansicht entfernt werden und somit ein anderer Eintrag aktiv wird.

Als nächstes wollen wir die Schaltfläche "Remove" deaktivieren, wenn in der Listenansicht kein aktiver Eintrag vorhanden ist. Wir können MUI-Gadgets deaktivieren, indem Sie das Attribut `Area.Disabled` auf `True` setzen. Da in der Listenansicht zunächst keine Einträge vorhanden sind, müssen wir `Area.Disabled` bereits zu Beginn unseres Programms auf `True` setzen. Also müssen Sie diesen Code einfügen:

```
...
mui.CreateGUI(FileToString("GUI.xml"))
mui.Set("mybt2", "disabled", True)
...
```

Nun müssen wir einige Änderungen an unsere Callback-Funktion vornehmen. Wann immer wir die Benachrichtigung aus `Listview.Active` erhalten, müssen wir prüfen, ob sie sich von dem speziellen Wert "Off" unterscheidet. Wenn dies der Fall ist, aktivieren wir das Gadget "Remove". Der spezielle Wert "Off" (-1) wird von `Listview.Active` zurückgegeben, wenn in der Listenansicht kein aktiver Eintrag vorhanden ist. Wir müssen unseren Code folgendermaßen ändern:

```
Switch msg.Class
...
Case "Listview":
    Switch msg.Attribute
    Case "Active":
        Local s$ = mui.DoMethod("mylistview", "getentry", "active")
```

```

        mui.Set("mystring", "contents", s$)
        mui.Set("mybt2", "disabled", IIf(msg.triggervalue = -1,
            True, False))
    EndSwitch
EndSwitch

```

Wir verwenden hier das Feld `msg.TriggerValue`. Dies enthält immer den aktuellen Wert des Attributs, das das Ereignis ausgelöst hat, d.h. in unserem Fall enthält es den aktuellen Wert des Attributs `Listview.Active`. Wir könnten auch `mui.Get()` manuell aus `Listview.Active` aufrufen, aber das ist nicht wirklich erforderlich, da wir einfach die Verknüpfung `msg.TriggerValue` verwenden können.

Das letzte, was wir tun wollen, ist ein Menü zu unserer GUI hinzufügen. Alle MUI-Programme sollten über ein Menüelement verfügen, mit dem der Benutzer das Erscheinungsbild des Programms mithilfe der MUI-Einstellungen anpassen kann. Dies geschieht, indem Sie die Methode `Application.OpenConfigWindow` für das Anwendungs-Objekt ausführen. Außerdem sollte jedes MUI-Programm die Menüpunkte "Über MUI" und "Über MUI Royale" enthalten, die den Benutzer darüber informieren, dass dieses Programm mit MUI und MUI Royale erstellt wurde. Sie können diese Fenster anzeigen, indem Sie die Methoden `Application.AboutMUI` und `Application.AboutMUIRoyale` für das jeweilige Anwendungs-Objekt ausführen. Um diese Menüs zu unserem Programm hinzuzufügen, verwenden wir die `Menustrip`-Klasse. Hier ist der XML-Code, den Sie vor Ihrer Fenster-Definition hinzufügen müssen:

```

...
<application base="HELLOWORLD">
  <menustrip id="mymenustrip">
    <menu title="Project">
      <item id="menabout" notify="selected">About...</item>
      <item id="menaboutmui" notify="selected">About MUI...</item>
      <item id="menaboutmuiroyale" notify="selected">
        About MUI Royale...</item>
      <item/>
      <item id="menquit" notify="selected" shortcut="Q">Quit</item>
    </menu>
    <menu title="Settings">
      <item id="menmuiset" notify="selected">MUI...</item>
    </menu>
  </menustrip>
  ...
</application>

```

Nachdem wir unser Menuleisten-Objekt mit dem obigen XML-Code erstellt haben, müssen wir diese Menuleiste an unser Fenster anhängen. Dies geschieht, indem Sie das Attribut `Window.Menustrip` auf unser Menuleisten-Objekt setzen. Hier ist der XML-Code dafür:

```

<window title="Example GUI" muiid="MAIN" notify="closerequest"
  menustrip="mymenustrip">

```

Wie Sie im obigen XML-Code sehen können, haben wir bereits Benachrichtigungen über das Attribut `MenuItem.Selected` hinzugefügt, um immer informiert zu werden, wenn der Benutzer einen Menüeintrag auswählt. Jetzt müssen wir unseren Code der Callback-Funktion

hinzufügen, der die entsprechende Aktion ausführt, wenn ein Menüelement ausgewählt wird. Bevor wir dies jedoch tun können, müssen wir unserem Anwendungs-Objekt eine ID zuweisen, da wir `mui.DoMethod()` darauf verwenden müssen. So muss der XML-Code angepasst werden:

```
<application base="HELLOWORLD" id="app">
```

Jetzt können wir den Code für unsere Callback-Funktion schreiben, die Menüpunkte behandelt:

```
Switch msg.Class
...
Case "MenuItem":
  Switch msg.Attribute
  Case "Selected":
    Switch msg.id
    Case "menabout":
      mui.Request("Test", "Test program\n" ..
        "© 2012 by Andreas Falkenhahn", "OK")
    Case "menaboutmui":
      mui.DoMethod("app", "aboutmui")
    Case "menaboutmuiroyale":
      mui.DoMethod("app", "aboutmuiroyale")
    Case "menquit":
      End
    Case "menmuiset":
      mui.DoMethod("app", "openconfigwindow")
  EndSwitch
EndSwitch
EndSwitch
```

Ein letzter Schliff für unser Programm könnte darin bestehen, unseren Gadgets eine Online-Hilfe hinzuzufügen, indem wir das Attribut `Area.ShortHelp` verwenden und alle Objekte zur Tastaturabfolgekette hinzufügen, so dass die GUI auch über die Tastatur gesteuert werden kann. Wir müssen die Schaltflächen unserer Tastaturabfolgekette nicht hinzufügen, da dies immer automatisch erfolgt. Wir müssen aber jedoch die Listenansicht und die Zeichenketten-Gadgets manuell hinzufügen. Das Hinzufügen von Objekten zur Tastaturabfolgekette erfolgt mithilfe des Attributs `Area.CycleChain`. Außerdem sollten Sie die Ereignisse `ShowWindow` und `HideWindow` beachten, die aus dem Exchange-Programm in `Commodities` stammen können, und entsprechend vorgehen. So sieht unser finales Programm mit diesen beiden letzten Verbesserungen aus. Zuerst die XML-Datei:

```
<?xml version="1.0" encoding="iso-8859-1"?>
<application base="HELLOWORLD" id="app">
  <menustrip id="mymenustrip">
    <menu title="Project">
      <item id="menabout" notify="selected">About...</item>
      <item id="menaboutmui" notify="selected">About MUI...</item>
      <item id="menaboutmuiroyale" notify="selected">
        About MUI Royale...</item>
    </item/>
```

```

        <item id="menquit" notify="selected" shortcut="Q">Quit</item>
    </menu>
    <menu title="Settings">
        <item id="menmuiset" notify="selected">MUI...</item>
    </menu>
</menustrip>
<window title="Example GUI" muiid="MAIN" notify="closerequest"
        menustrip="mymenustrip">
    <vgroup>
        <listview id="mylistview" notify="active" shorthelp="A listview"
            cyclechain="true">
            <column/>
        </listview>
        <string id="mystring" shorthelp="Enter entry name here"
            cyclechain="true"/>
        <hgroup>
            <button id="mybt1" notify="pressed"
                shorthelp="Add new entry">Add</button>
            <button id="mybt2" notify="pressed"
                shorthelp="Remove entry">Remove</button>
        </hgroup>
    </vgroup>
</window>
</application>

```

Und hier ist der Code für die Programmlogik:

```

@DISPLAY {Hidden = True}

@APTITLE "Tutorial"
@APPVERSION "$VER: Tutorial 1.0 (27.12.12)"
@APPCOPYRIGHT "Copyright ©2012, Andreas Falkenhahn"
@APPAUTHOR "Andreas Falkenhahn"
@APPDESCRIPTION "The tutorial app from the MUI Royale guide"

Function p_EventFunc(msg)

    If msg.Action <> "MUIRoyale"

        Switch msg.Action
        Case "HideWindow":
            mui.Set("app", "iconified", True)
        Case "ShowWindow":
            mui.Set("app", "iconified", False)
        EndSwitch

    Return

```

```

EndIf

Switch msg.Class
Case "Window":
    Switch msg.Attribute
    Case "CloseRequest":
        End
    EndSwitch

Case "Button":
    Switch msg.Attribute
    Case "Pressed":
        Switch msg.ID
        Case "mybt1": ; "Add" button was pressed
            Local s$ = mui.Get("mystring", "contents")
            mui.DoMethod("mylistview", "insert", "bottom", s$)
        Case "mybt2": ; "Remove" button was pressed
            mui.DoMethod("mylistview", "remove", "active")
        EndSwitch
    EndSwitch

Case "Listview":
    Switch msg.Attribute
    Case "Active":
        Local s$ = mui.DoMethod("mylistview", "getentry", "active")
        mui.Set("mystring", "contents", s$)
        mui.Set("mybt2", "disabled", IIf(msg.triggervalue = -1,
            True, False))
    EndSwitch

Case "MenuItem":
    Switch msg.Attribute
    Case "Selected":
        Switch msg.id
        Case "menabout":
            mui.Request("Test", "Test program\n" ..
                "© 2012 by Andreas Falkenhahn", "OK")
        Case "menaboutmui":
            mui.DoMethod("app", "aboutmui")
        Case "menaboutmuiroyale":
            mui.DoMethod("app", "aboutmuiroyale")
        Case "menquit":
            End
        Case "menmuiset":
            mui.DoMethod("app", "openconfigwindow")
        EndSwitch
    EndSwitch

```

```
EndSwitch

EndFunction

InstallEventHandler({MUIRoyale = p_EventFunc, HideWindow = p_EventFunc,
    ShowWindow = p_EventFunc})
mui.CreateGUI(FileToString("GUI.xml"))
mui.Set("mybt2", "disabled", True)

Repeat
    WaitEvent
Forever
```

Das ist es! Jetzt sollten Sie in der Lage sein, MUI-Programme nach Ihrem persönlichen Geschmack zu erstellen. Vielen Dank für das Lesen dieses Tutorials und genießen Sie die Leistung von MUI mit Hollywood und MUI Royale in Ihren Händen!

5 Befehlsreferenz

5.1 mui.CreateGUI

BEZEICHNUNG

`mui.CreateGUI` – erstellt ein Anwendungs-Objekt aus einer XML-Quelle

ÜBERSICHT

`mui.CreateGUI(xml$)`

BESCHREIBUNG

Dieser Befehl erstellt eine MUI-GUI aus der XML-Definition, die im Argument `xml$` übergeben wird, d.h. sie richtet das MUI-Hauptanwendungs-Objekt mit allen seinen untergeordneten Elementen ein. Beachten Sie, dass `xml$` eine Zeichenkette sein muss, die die XML-GUI-Definition und keinen Dateinamen enthält. Wenn Sie eine XML-GUI-Definition aus einer externen Datei verwenden möchten, müssen Sie diese Datei zuerst in eine Zeichenkette konvertieren. Verwenden Sie dafür z.B. den Hollywood-Befehl `FileToString()`.

Sobald dieser Befehl beendet ist, können Sie mit all Ihren MUI-Objekten kommunizieren, die Sie in der XML-GUI-Definition mit den Befehlen `mui.Set()`, `mui.Get()` und `mui.DoMethod()` definiert haben.

Bitte beachten Sie, dass pro Programm nur ein Anwendungs-Objekt vorhanden sein kann, so dass dieser Befehl nur einmal aufgerufen werden kann. Wenn Sie `mui.CreateGUI()` ein zweites Mal aufrufen möchten, müssen Sie zuerst die alte GUI mit dem Aufruf `mui.FreeGUI()` löschen.

Wenn Sie Ihrem Anwendungs-Objekt Objekte wie Fenster oder Schaltflächen dynamisch hinzufügen müssen, können Sie dazu den Befehl `mui.CreateObject()` verwenden.

EINGABEN

`xml$` Eine Zeichenkette, die eine XML-GUI-Definition enthält

BEISPIEL

```
mui.CreateGUI([[
<?xml version="1.0" encoding="iso-8859-1"?>
<application>
  <window title="Test program" notify="closerequest">
    <vgroup>
      <button>Hello World!</button>
    </vgroup>
  </window>
</application>
]])
```

```
InstallEventHandler({MUIRoyale = Function(msg)
  If msg.attribute = "CloseRequest" Then End
EndFunction})
```

```
Repeat
  WaitEvent
Forever
```

Der obige Code erstellt eine minimale GUI mit nur einem Fenster und einer einzigen Schaltfläche.

5.2 mui.CreateObject

BEZEICHNUNG

`mui.CreateObject` – erstellt ein MUI-Objekt aus einer XML-Quelle (V1.2)

ÜBERSICHT

```
mui.CreateObject(xml$)
```

BESCHREIBUNG

Dieser Befehl kann verwendet werden, um ein MUI-Objekt dynamisch aus einer XML-Quelle zu erstellen. Wenn `mui.CreateObject()` beendet ist, wird das neu erstellte MUI-Objekt nicht an ein übergeordnetes Objekt angefügt, sondern es befindet sich in einem losgelösten (isolierten) Zustand von Ihrem Anwendungs-Objekt, das von `mui.CreateGUI()` erstellt wurde. Um diesen losgelösten Zustand zu unterbrechen, müssen Sie das Objekt zuerst an ein übergeordnetes Objekt anhängen, bei dem es sich entweder um ein Gruppen-Objekt, ein Menü-Objekt oder ein Anwendungs-Objekt handeln kann. Wenn das neu zugewiesene MUI-Objekt ein Fenster-Objekt ist, müssen Sie es mit der Methode `Application.AddWindow` an das Anwendungs-Objekt anhängen. Um Menü-Objekte anzuhängen, müssen Sie die Methoden `Menustrip.AddHead`, `Menustrip.AddTail`, `Menustrip.Insert`, `Menu.AddHead`, `Menu.AddTail` oder `Menu.Insert` benutzen. Alle anderen Objekte können mit den Gruppenmethoden `Group.AddHead`, `Group.AddTail` und `Group.Insert` verknüpft werden.

Im Gegensatz zu `mui.CreateGUI()` können Sie `mui.CreateObject()` beliebig oft aufrufen, da es kein Anwendungs-Objekt für Sie erstellt, sondern nur losgelöste MUI-Objekte, von denen Sie so viele haben können, wie Sie wollen.

Es ist wichtig, dass Sie eine ID für Ihr MUI-Objekt in der XML-Definition angeben, da Sie diese ID benötigen, um auf dieses Objekt zu verweisen, wenn Sie es zu einem Anwendungs- oder Gruppen-Objekt hinzufügen möchten.

Sobald dieser Befehl beendet ist, können Sie mit dem neu erstellten MUI-Objekt (und allen seinen untergeordneten) mit den Befehlen `mui.Set()`, `mui.Get()` und `mui.DoMethod()` verwenden.

Losgelöste MUI-Objekte können mit dem Befehl `mui.FreeObject()` gelöscht werden, aber Sie müssen dies nur in bestimmten Fällen aufrufen, z.B. wenn Sie mit vielen dynamisch zugewiesenen MUI-Objekten zu tun haben und Speicher sowie Ressourcen sparen möchten.

EINGABEN

`xml$` eine Zeichenkette, die eine XML-MUI-Objektbeschreibung enthält

BEISPIEL

```
mui.CreateObject([[
```



```

<window id="newwindow" title="A new window" notify="closerequest">
  <vgroup>
    <button>Hello World!</button>
  </vgroup>
</window>
]])

```

```

mui.DoMethod("app", "addwindow", "newwindow")
mui.Set("newwindow", "open", True)

```

Der obige Code erstellt ein neues Fenster, fügt es dem vorhandenen Anwendungs-Objekt hinzu und öffnet es.

```

mui.CreateObject([[
<button id="newbutton">Dynamically created button!</button>
]])

```

```

mui.DoMethod("mygroup", "initchange")
mui.DoMethod("mygroup", "addtail", "newbutton")
mui.DoMethod("mygroup", "exitchange", false)

```

Der obige Code erstellt dynamisch ein neues Schaltflächen-Objekt und fügt es als letztes untergeordnetes Element zu der Gruppe mit der ID "mygroup" hinzu.

5.3 mui.DoMethod

BEZEICHNUNG

mui.DoMethod – wendet eine Methode auf ein MUI-Objekt an

ÜBERSICHT

```
r = mui.DoMethod(id$, method$, ...)
```

BESCHREIBUNG

Dieser Befehl kann verwendet werden, um eine Methode für das angegebene MUI-Objekt auszuführen. Sie müssen den Identifikator des MUI-Objekts im ersten Argument `id$` und den Namen der Methode im zweiten Argument `method$` übergeben. Bei Methoden- und Objekt-IDs wird nicht zwischen Groß- und Kleinschreibung unterschieden. Es spielt also keine Rolle, ob Sie Groß- oder Kleinbuchstaben verwenden.

Die Methoden, die Sie mit diesem Befehl verwenden können, hängen von der Klasse des angegebenen MUI-Objekts ab. Sehen Sie sich die Klassenreferenz an, um zu sehen, welche Methoden von den verschiedenen MUI-Klassen unterstützt werden.

Außerdem hängen die Argumente, die Sie nach dem Methodennamen an diesen Befehl übergeben müssen, von der Methode ab. Sie sind für jede Methode unterschiedlich. Das Gleiche gilt für Rückgabewerte. Einige Methoden geben Werte zurück, andere nicht. In der Klassenreferenz finden Sie, welche Argumente Ihre Methode benötigt und ob es Rückgabewerte für Ihre Methode gibt.

EINGABEN

`id$` ID des MUI-Objekts, für das die Methode ausgeführt werden soll

`method$` Methodenname als Zeichenkette
`...` Weitere Argumente hängen von der Methode ab (siehe Klassenreferenz für Einzelheiten)

RÜCKGABEWERTE

`r` der Rückgabewert hängt vom Methodentyp ab

BEISPIEL

```
mui.DoMethod("my_listview", "insert", "bottom", "Last entry")
```

Der obige Code fügt einen neuen Eintrag namens "Last entry" am Ende der Listenansicht mit der Bezeichnung "my_listview" hinzu. Dies geschieht, indem Sie die Methode `Listview.Insert()` für das Listenansichts-Objekt ausführen.

5.4 mui.FreeGUI

BEZEICHNUNG

`mui.FreeGUI` – löscht das gesamte Anwendungs-Objekt

ÜBERSICHT

```
mui.FreeGUI()
```

BESCHREIBUNG

Verwenden Sie diesen Befehl, um das gesamte GUI-Objekt zu löschen, das mit dem letzten Aufruf von `mui.CreateGUI()` erstellt wurde. `mui.FreeGUI()` löscht das gesamte MUI-Hauptanwendungs-Objekt. Alle Fenster werden geschlossen und ihre untergeordneten Elemente werden gelöscht. Nachdem dieser Aufruf beendet wurde, können Sie eine neue GUI mit dem Befehl `mui.CreateGUI()` erstellen, wenn Sie möchten.

EINGABEN

keiner

5.5 mui.FreeImage

BEZEICHNUNG

`mui.FreeImage` – löscht den Pinsel im Bilder-Cache (V1.7)

ÜBERSICHT

```
mui.FreeImage(id)
```

BESCHREIBUNG

Dieser Befehl kann verwendet werden, um einen Pinsel im internen Bilder-Cache von MUI Royale zu löschen. Sie müssen den Identifikator des Pinsels in `id` übergeben, der gelöscht werden soll. Alternativ können Sie auch `-1` an diesen Befehl übergeben, um alle Pinsel im internen Bilder-Cache von MUI Royale zu löschen.

Sie müssen sicherstellen, dass der angegebene Pinsel nicht mehr von Widgets in Ihrer GUI verwendet wird, bevor Sie diesen Befehl aufrufen. Beachten Sie, dass dieser Befehl unter normalen Bedingungen nicht aufgerufen werden muss, da normalerweise alle Pinsel

von MUI Royale automatisch gelöscht werden. Unter bestimmten Bedingungen kann es jedoch nützlich sein, diesen Befehl aufzurufen.

MUI Royale speichert alle Hollywood-Pinsel, die Sie als Bilder in Ihrer GUI verwenden. Wenn Sie also zum zweiten Mal versuchen, einen Hollywood-Pinsel in Ihrer GUI zu verwenden, wird er aus Leistungsgründen einfach aus dem internen Bilder-Cache von MUI Royale geladen, unabhängig vom aktuellen Inhalt des Pinsels in Hollywood. Dies kann zu unerwünschtem Verhalten führen, wenn Sie die Grafiken Ihres Pinsels in der Zwischenzeit aktualisiert haben und MUI Royale die aktualisierten Grafiken verwenden soll. In diesem Fall müssen Sie zuerst den Pinsel im internen Bilder-Cache von MUI Royale mithilfe von diesem Befehl löschen. Wenn Sie den Pinsel dann wieder an MUI Royale übergeben, wird er aus dem aktuellen Zustand des Pinsels neu erstellt und erneut zwischengespeichert.

EINGABEN

`id` Identifikator des Pinsels, der gelöscht werden soll oder -1 zum Freigeben aller Pinsel

5.6 `mui.FreeObject`

BEZEICHNUNG

`mui.FreeObject` – löscht ein losgelöstes MUI-Objekts (V1.2)

ÜBERSICHT

`mui.FreeObject(id$)`

BESCHREIBUNG

Dieser Befehl kann verwendet werden, um ein losgelöstes MUI-Objekt zu löschen, das entweder von `mui.CreateObject()` oder `mui.CreateGUI()` erstellt wurde. Das MUI-Objekt, das Sie hier angeben, darf nicht länger an ein Anwendungs-, Gruppen- oder Menü-Objekt angehängt sein, da angehängte MUI-Objekte mit ihrem übergeordneten Element gelöscht werden. Stellen Sie daher sicher, dass Sie diesen Befehl nur für MUI-Objekte verwenden, die von ihren übergeordneten Elementen losgelöst wurden und nicht mehr an ein übergeordnetes Objekt gebunden ist.

Um MUI-Objekte von ihren jeweiligen übergeordneten Objekten loszulösen, müssen Sie eine der folgenden Methoden verwenden: `Application.RemoveWindow`, `Menustrip.Remove`, `Menu.Remove` oder `Group.Remove`.

Wenn MUI Royale beendet wird, werden alle gelösten MUI-Objekte automatisch gelöscht. Wenn Ihr Programm also MUI-Objekte zur Laufzeit nicht ständig hinzufügt und entfernt, müssen Sie diesen Befehl normalerweise nicht aufrufen.

EINGABEN

`id$` Identifikator des MUI-Objekts, das gelöscht werden soll

5.7 `mui.Get`

BEZEICHNUNG

`mui.Get` – gibt den Wert eines MUI-Objektattributs zurück

ÜBERSICHT

```
r = mui.Get(id$, attr$)
```

BESCHREIBUNG

Dieser Befehl kann verwendet werden, um den aktuellen Wert des Attributs `attr$` in dem in `id$` angegebenen MUI-Objekt abzurufen. Attributnamen und Objekt-IDs unterscheiden nicht zwischen Groß- und Kleinschreibung, d.h. es spielt keine Rolle für Sie, ob Sie Groß- oder Kleinbuchstaben verwenden.

Die Attribute, die Sie mit diesem Befehl verwenden können, hängen von der Klasse des angegebenen MUI-Objekts ab. Sehen Sie sich die Klassenreferenz an, um zu erfahren, welche Attribute von den verschiedenen MUI-Klassen unterstützt werden. Um ein Attribut mit diesem Befehl zu verwenden, muss es eine Anwendbarkeit von "G" haben. Attribute der Area-Klasse und der Notify-Klasse können für fast alle Klassen verwendet werden, da die Area- und Notify-Klasse für die meisten anderen Klassen als Oberklassen fungieren.

EINGABEN

`id$` Identifikator des MUI-Objekts, das abgefragt werden soll

`attr$` Attribut, dessen Wert abgerufen werden soll

RÜCKGABEWERTE

`r` aktueller Attributwert

BEISPIEL

```
DebugPrint(mui.Get("my_listview", "active"))
```

Der obige Code gibt den Index des gerade aktiven Eintrags in der Listenansicht mit der Kennzeichnung "my_listview" zurück, indem er das Attribut `Listview.Active` abfragt.

5.8 mui.HaveObject

BEZEICHNUNG

`mui.HaveObject` – überprüft, ob das MUI-Objekt existiert (V1.7)

ÜBERSICHT

```
r = mui.HaveObject(id$)
```

BESCHREIBUNG

Dieser Befehl prüft, ob ein MUI-Objekt im angegebenen Identifikator `id$` existiert oder nicht. Wenn es existiert, wird in `r` `True` zurückgegeben, andernfalls `False`.

EINGABEN

`id$` MUI-Objekt-ID, die überprüft wird

RÜCKGABEWERTE

`r` `True` oder `False`, abhängig davon, ob das Objekt existiert oder nicht

5.9 mui.IsVersion4

BEZEICHNUNG

`mui.IsVersion4` – überprüft, ob MUI 4 oder besser installiert ist (V1.2)

ÜBERSICHT

```
r = mui.IsVersion4()
```

BESCHREIBUNG

Mit diesem Befehl kann überprüft werden, ob der Benutzer MUI 4.0 oder höher ausführt. Da einige Funktionen von MUI Royale nur mit MUI 4 verfügbar sind, kann es erforderlich sein, mit diesen Befehl zu prüfen, ob sie verfügbar sind, bevor Sie versuchen, sie zu verwenden.

EINGABEN

keine

RÜCKGABEWERTE

`r` Gibt `True` zurück, wenn MUI 4 oder besser verfügbar ist

5.10 mui.Notify

BEZEICHNUNG

`mui.Notify` – fügt eine Benachrichtigung einem MUI-Objekt hinzu oder entfernt sie

ÜBERSICHT

```
mui.Notify(id$, attr$, enable)
```

BESCHREIBUNG

Dieser Befehl kann verwendet werden, um eine Benachrichtigung für das Attribut `attr$` in dem in `id$` angegebenen MUI-Objekt hinzuzufügen oder zu entfernen. Sie müssen den Attributnamen und ein boolesches Flag übergeben, das angibt, ob Sie Benachrichtigungen für dieses Attribut aktivieren oder deaktivieren möchten. Attributnamen und Objekt-IDs unterscheiden nicht zwischen Groß- und Kleinschreibung, d.h. es spielt keine Rolle, ob Sie Groß- oder Kleinbuchstaben für sie verwenden.

Die Attribute, die Sie mit diesem Befehl verwenden können, hängen von der Klasse des angegebenen MUI-Objekts ab. Sehen Sie sich die Klassenreferenz an, um zu sehen, welche Attribute von den verschiedenen MUI-Klassen unterstützt werden. Um ein Attribut mit diesem Befehl zu verwenden, muss es eine Anwendbarkeit von "N" haben. Attribute der Area-Klasse und Notify-Klasse können für fast alle Klassen verwendet werden, da die Area- und Notify-Klassen für die meisten anderen Klassen als Oberklassen fungieren.

Nachdem Sie eine Benachrichtigung für ein bestimmtes Objektattribut eingerichtet haben, können Sie diese Ereignisse überwachen, indem Sie eine Callback-Funktion mithilfe des Hollywood-Befehles `InstallEventHandler()` benutzen. Siehe [Abschnitt 3.6 \[Benachrichtigungen\]](#), Seite 12, für Details.

Beachten Sie, dass Benachrichtigungen auch mithilfe des Tags `Notify` in der XML-GUI-Definition eingerichtet werden können. Siehe [Abschnitt 3.6 \[Benachrichtigungen\]](#), Seite 12, für Details.

EINGABEN

<code>id\$</code>	Identifikator des zu verwendenden MUI-Objekts
<code>attr\$</code>	Attribut, welches überwacht werden soll oder nicht
<code>enable</code>	<code>True</code> zum Hinzufügen einer Benachrichtigung oder <code>False</code> zum Entfernen der Benachrichtigung von diesem Objekt

BEISPIEL

```
mui.Notify("my_listview", "active", True)
```

Der obige Code installiert eine Benachrichtigung, die ausgelöst wird, wenn sich das Attribut `Listview.Active` in der Listenansicht mit dem Identifikator `"my_listview"` ändert.

5.11 mui.Request

BEZEICHNUNG

`mui.Request` – zeigt ein MUI-System-Dialogfenster

ÜBERSICHT

```
r = mui.Request(title$, body$, gadgets$[, icon$])
```

BESCHREIBUNG

Dieser Befehl öffnet ein Standard-MUI-Dialogfenster, welches die in `body$` angegebene Nachricht anzeigt und es dem Benutzer außerdem erlaubt, eine Auswahl durch die in `gadgets$` definierten Schaltflächen zu treffen. Wenn Sie ein MUI-System-Dialog anstelle eines Standard-System-Dialog verwenden, können Sie Text mit Codes im Text-Modul-Format einfügen.

Trennen Sie die in den `gadgets$` angegebenen Schaltflächen durch ein `"|"`. Durch den Rückgabewert `r` erfahren Sie, welche Schaltfläche der Benutzer gedrückt hat. Beachten Sie, dass die Schaltfläche ganz rechts immer den Wert `False` (0) hat, da es normalerweise als "Abbrechen" verwendet wird. Wenn Sie beispielsweise drei Schaltflächen-Optionen "Eins|Zwei|Drei" haben, hat die Schaltfläche "Drei" den Rückgabewert 0, "Zwei" gibt 2 zurück und "Eins" gibt 1 zurück.

Wenn Sie einem Eintrag in `gadgets$` ein `'*'` voranstellen, wird diese Schaltfläche zum aktiven Objekt. Durch Drücken der Taste `<Return>` wird der Dialog mit dieser Schaltfläche beendet. Ein `'_'` Zeichen gibt die Tastenkombination für diese Schaltfläche an.

Die Zeichenketten, die Sie in `body$` und `gadgets$` angeben, können Textformatierungscodes verwenden. Siehe [Abschnitt 3.9 \[Textformatierungscodes\]](#), Seite 14, für Details.

Beginnend mit MUI Royale 1.4 akzeptiert dieser Befehl ein optionales Argument namens `icon$`. Dadurch können Sie das Symbol angeben, das im Dialogfenster angezeigt werden soll. Beachten Sie, dass dies derzeit nur von MUI 4 und höher auf AmigaOS 3 und AmigaOS 4 unterstützt wird. Das MorphOS MUI unterstützt diese Funktion noch nicht. Die folgenden vordefinierten Werte werden für `icon$` akzeptiert:

```
"None":    kein Symbol
"Information":
            ein Informationszeichen
```

"Error": ein Fehlerzeichen
 "Warning":
 ein Warnzeichen
 "Question":
 ein Fragezeichen

EINGABEN

title\$ Titel des Dialogs; übergeben Sie eine leere Zeichenkette (""), um den Standardtitel zu verwenden
body\$ Text, der im Hauptteil des Dialogs erscheint
gadgets\$ ein oder mehrere Schaltflächen, die der Benutzer drücken kann
icon\$ Optional: Symbol, welches in dem Dialog angezeigt wird (standardmäßig "Information") (V1.4)

RÜCKGABEWERTE

r die Schaltfläche, die vom Benutzer gedrückt wurde

BEISPIEL

```
mui.Request("MUI Royale", "\027c\027b\027uHello!\n\n" ..
  "\027nDo you like MUI Royale?", "*_Yes|_No")
```

Der obige Code demonstriert die Verwendung des Befehls `mui.Request()`.

5.12 mui.Set

BEZEICHNUNG

`mui.Set` – setzt den Wert eines MUI-Objektattributs

ÜBERSICHT

```
mui.Set(id$, attr1$, val1$, ...)
```

BESCHREIBUNG

Dieser Befehl kann verwendet werden, um den aktuellen Wert eines oder mehrerer Attribute in dem in `id$` angegebenen MUI-Objekt festzulegen. Sie müssen den Attributnamen in `attr1$` und den gewünschten neuen Wert in `val1$` für jedes Attribut, das Sie ändern möchten, übergeben. Sie können diese Attribut/Wert-Paare beliebig oft wiederholen, um mehrere Attribute mit nur einem Aufruf von `mui.Set()` zu ändern. Attributnamen und Objekt-IDs unterscheiden nicht zwischen Groß- und Kleinschreibung, d.h. es spielt keine Rolle, ob Sie Groß- oder Kleinbuchstaben für sie verwenden.

Die Attribute, die Sie mit diesem Befehl verwenden können, hängen von der Klasse des angegebenen MUI-Objekts ab. Sehen Sie sich die Klassenreferenz an, um zu sehen, welche Attribute von den verschiedenen MUI-Klassen unterstützt werden. Um ein Attribut mit diesem Befehl zu verwenden, muss es eine Anwendbarkeit von "S" haben. Attribute der Area-Klasse und Notify-Klasse können für fast alle Klassen verwendet werden, da die Area- und Notify-Klasse für die meisten anderen Klassen als Oberklassen fungieren.

Wenn Sie mit diesem Befehl eine Benachrichtigung für das Attribut erstellt haben, das Sie ändern möchten, wird die Benachrichtigung ausgelöst, sobald Sie `mui.Set()` für dieses

Attribut aufrufen. Wenn Sie dieses Verhalten nicht möchten, können Sie das Attribut `Notify.NoNotify` verwenden, um zu verhindern, dass eine Benachrichtigung ausgegeben wird.

EINGABEN

`id$` Identifikator des zu ändernden MUI-Objekts
`attr1$` Attribut, dessen Wert geändert werden soll
`val1$` Neuer Wert für das Attribut
... Sie können die Attribut/Wert-Paare so oft wiederholen, wie Sie möchten

BEISPIEL

```
mui.Set("my_listview", "active", 15)
```

Der obige Code setzt den Eintrag 15 als derzeit aktiven Eintrag in der Listenansicht mit dem Identifikator "my_listview", indem er das Attribut `Listview.Active` setzt.

```
mui.Set("my_listview", "nonotify", True, "active", 15)
```

Dieser Code entspricht dem obigen Code, verhindert jedoch, dass Benachrichtigungen ausgegeben werden, indem das Attribut `Notify.NoNotify` auf `True` gesetzt wird. Dies ist nützlich, wenn Sie zwischen Benutzerauswahlen in der Listenansicht und programmgesteuert mit `mui.Set()` vorgenommenen Auswahlen unterscheiden müssen.

6 Application-Klasse (Anwendung)

6.1 Überblick

Die Application-Klasse (Anwendung) ist die Hauptklasse für alle MUI-Anwendungen. Sie dient als eine Art Anker für alle Eingaben, die entweder vom Benutzer oder irgendwo aus dem System kommen, z.B. Commodities oder ARexx-Nachrichten.

Eine Anwendung kann eine beliebige Anzahl von Unterfenstern haben und diese Fenster sind die untergeordneten Elemente der Anwendung. Jede GUI-Definition muss mit der Application-Klasse beginnen.

6.2 Application.AboutMUI

BEZEICHNUNG

Application.AboutMUI – zeigt das Fenster "Über MUI" an

ÜBERSICHT

```
mui.DoMethod(id, "AboutMUI")
```

BESCHREIBUNG

Zeigt das Fenster MUI über an. Bitte fügen Sie das in all Ihren Anwendungen ein und verknüpfen Sie sie mit einem Menüpunkt namens "Über MUI ...".

EINGABEN

id ID des Anwendungs-Objekts

6.3 Application.AboutMUIRoyale

BEZEICHNUNG

Application.AboutMUIRoyale – zeigt das Fenster "Über MUI Royale" an

ÜBERSICHT

```
mui.DoMethod(id, "AboutMUIRoyale")
```

BESCHREIBUNG

Zeigt das Fenster MUI Royale über an. Bitte fügen Sie das in all Ihren Anwendungen hinzu und verlinken Sie mit einem Menüpunkt namens "Über MUI Royale ...".

EINGABEN

id ID des Anwendungs-Objekts

6.4 Application.AddWindow

BEZEICHNUNG

Application.AddWindow – fügt dem Anwendungs-Objekt ein losgelöstes Fenster hinzu (V1.2)

ÜBERSICHT

```
mui.DoMethod(id, "AddWindow", window)
```

BESCHREIBUNG

Diese Methode kann verwendet werden, um dem Anwendungs-Objekt ein losgelöstes Fenster-Objekt hinzuzufügen. Sobald das Fenster-Objekt an das Anwendungs-Objekt angefügt wurde, können Sie das Fenster öffnen, indem Sie das Attribut `Window.Open` festlegen.

Losgelöste Fenster-Objekte können entweder durch Aufrufen des Befehls `mui.CreateObject()` oder durch explizites Trennen von ihren übergeordneten Elementen mithilfe der Methode `Application.RemoveWindow` erstellt werden.

EINGABEN

`id` ID des Anwendungs-Objekts
`window` ID des hinzuzufügenden Fenster-Objekts

BEISPIEL

Siehe [Abschnitt 5.2 \[mui.CreateObject\]](#), Seite 38.

6.5 Application.Base

BEZEICHNUNG

`Application.Base` – setzt/ermittelt den Basisnamen einer Anwendung

BESCHREIBUNG

Der Basisname für eine Anwendung wird für den integrierten ARexx-Port und für die interne Dateiverwaltung verwendet.

Ein Basisname darf weder Leerzeichen noch Sonderzeichen wie `:/()#?*...` enthalten.

Wenn Ihr Programm eine Einzelanwendung ist (d.h. `Application.SingleTask` ist `True`), wird der Basisname ohne weitere Änderungen verwendet.

Andernfalls wird eine `".1"`, `".2"` usw. angehängt, abhängig davon, wie viele Anwendungen bereits ausgeführt werden. Wenn Sie den Namen Ihres ARexx-Ports kennen müssen, können Sie das Basisnamenattribut abfragen, nachdem die Anwendung erstellt wurde.

TYP

Zeichenkette

ANWENDBARKEIT

IG

6.6 Application.ConfigChange

BEZEICHNUNG

`Application.ConfigChange` – benachrichtigt, wenn die Konfiguration ändert (V1.2)

BESCHREIBUNG

Dieses Attribut wird ausgelöst, wenn der Benutzer die Konfiguration Ihrer Anwendung ändert. Seien Sie gewarnt, dass dies sehr oft ausgelöst werden kann, da es für jede einzelne Änderung des Benutzers eingestellt wird.

Dies wird nur von MUI 4.x unterstützt.

TYP

Boolesch

ANWENDBARKEIT

N

6.7 Application.DoubleStart

BEZEICHNUNG

Application.DoubleStart – benachrichtigt, wenn die Anwendung doppelt gestartet wird

BESCHREIBUNG

Dieses Attribut wird automatisch gesetzt, wenn der Benutzer versucht, eine Anwendung zu starten, für die das Attribut `Application.SingleTask` auf `True` gesetzt wurde. Sie können darauf reagieren und geeignete Maßnahmen ergreifen, z.B. das Öffnen eines Dialogs oder die Anwendung beenden.

TYP

Boolesch

ANWENDBARKEIT

GN

6.8 Application.DropObject

BEZEICHNUNG

Application.DropObject – legt das Standard-Drop-Objekt fest

BESCHREIBUNG

Wenn Ihre Anwendung ikonifiziert ist und der Benutzer Symbole auf dem AppIcon ablegt, erhält das hier angegebene Objekt die Benachrichtigung von `Notify.AppMessage`.

TYP

Zeichenkette

ANWENDBARKEIT

IS

6.9 Application.HelpFile

BEZEICHNUNG

Application.HelpFile – setzt/ermittelt die Hilfe-Datei der Anwendung

BESCHREIBUNG

Dieses Attribut ermöglicht die Definition einer AmigaGuide-Datei, die angezeigt wird, wenn der Benutzer Hilfe anfordert.

Wenn die Taste `HELP` gedrückt wird und die Anwendung ein `Application.HelpFile` definiert hat, versucht MUI, `Notify.HelpNode` vom aktuellen Objekt (das unter dem Mauszeiger) abzurufen. Wenn `Notify.HelpNode` nicht definiert ist, fordert MUI dann das übergeordnete Objekt nach diesem Attribut an (normalerweise eine Gruppe. Aber denken Sie daran: Das übergeordnete Objekte eines Windows-Stamm-Objekts das Fenster selbst ist, das übergeordnete Element eines Fensters ist die Anwendung).

Wenn ein `Notify.HelpNode` gefunden wird, wird dieselbe Prozedur in `Notify.HelpLine` angewendet. Dann versetzt MUI die Anwendung in den Ruhezustand und zeigt die Datei an der mit `Notify.HelpNode` und/oder `Notify.HelpLine` angegebenen Position an.

Mit diesem Verhalten können Sie ein `Application.HelpFile` für Ihr Anwendungs-Objekt und verschiedene Hilfsknoten und -zeilen für Ihre Anwendungsfenster und/oder -gadgets definieren. Siehe [Abschnitt 3.10 \[Online-Hilfe implementieren\]](#), Seite 16, für Details.

TYP

Zeichenkette

ANWENDBARKEIT

ISG

6.10 Application.Icon

BEZEICHNUNG

`Application.Icon` – setzt das Anwendungssymbol

BESCHREIBUNG

Wenn Sie hier den Pfad zu einer Symboldatei angeben, verwendet MUI dieses Objekt für das `AppIcon`, wenn Ihre Anwendung mit einem Icon versehen wird. Andernfalls versucht MUI, `env:sys/dev_mui.info` zu finden und falls nicht vorhanden, auf ein Standardsymbol zurückzugreifen.

TYP

Zeichenkette

ANWENDBARKEIT

I

6.11 Application.Iconified

BEZEICHNUNG

`Application.Iconified` – ikonifiziert die Anwendung oder baut sie wieder auf

BESCHREIBUNG

Wenn dieses Attribut auf `True` gesetzt wird, wird jedes geöffnete Fenster geschlossen und ein (konfigurierbares) `AppIcon` erscheint auf der Workbench.

Das Gleiche passiert, wenn der Benutzer auf das Iconifikations-Gadget im Fensterrahmen klickt oder `Commodities-Exchange` verwendet, um die Anwendungsschnittstelle zu verbergen.

Wenn eine Anwendung ikonifiziert wird und Sie versuchen, ein Fenster zu öffnen, wird dieses nicht sofort geöffnet. Stattdessen erinnert sich MUI an diese Aktion und öffnet das Fenster, sobald die Anwendung wieder aufgebaut wurde.

TYP

Boolesch

ANWENDBARKEIT

SG

6.12 Application.Load**BEZEICHNUNG**

Application.Load – lädt die Programmkonfiguration (V1.7)

ÜBERSICHT

`mui.DoMethod(id, "Load", name$)`

BESCHREIBUNG

Application.Save, Application.Load und Notify.ExportID bietet eine einfache Möglichkeit zum Speichern und Laden einer Programmkonfiguration.

Jedes Gadget mit einem Notify.ExportID sichert seinen Inhalt während dem Speichern mit Application.Save und erhält ihn während dem Wiederherstellen mit Application.Load. Dies macht es sehr einfach, ein Konfigurationsfenster mit den Schaltflächen "Speichern", "Verwenden" und "Abbrechen" zu entwerfen, damit der Benutzer die Einstellungen speichern kann. Wenn die Anwendung gestartet wird, müssen Sie nur Application.Load aufrufen und die gespeicherten Einstellungen werden gelesen und installiert.

Der Parameter `name$` muss auf den Namen der Datei gesetzt werden, von der die Einstellungen geladen werden sollen. Normalerweise müssen Sie nicht an einen echten Namen denken, sondern stattdessen einen der folgenden speziellen Werte verwenden:

`::ENV` Einstellungen von ENV: laden.

`::ENVARC` Einstellungen von ENVARC: laden.

Nicht alle Klassen können ihre Inhalte importieren und exportieren. Derzeit können Sie eine Notify.ExportID definieren für:

String-Klasse

`String.Contents` kann ex-/importieren.

Radio-Klasse

`Radio.Active` kann ex-/importieren.

Cycle-Klasse

`Cycle.Active` kann ex-/importieren.

Listview-Klasse

`Listview.Active` kann ex-/importieren.

Text-Klasse

`Text.Contents` kann ex-/importieren.

Slider-Klasse

`Slider.Level` kann ex-/importieren.

Area-Klasse

`Area.Selected` kann ex-/importieren (z.B. für Auswahlkästchen-Gadgets).

MenuItem-Klasse

`MenuItem.Selected` kann ex-/importieren.

Group-Klasse

`Group.ActivePage` kann ex-/importieren.

EINGABEN

<code>id</code>	ID des Anwendungs-Objekts
<code>name\$</code>	Name der Datei, aus der die Einstellungen geladen werden sollen oder Sonderwert (siehe oben)

6.13 Application.Menustrip

BEZEICHNUNG

`Application.Menustrip` – setzt die Menüleiste für die gesamte Anwendung

BESCHREIBUNG

Geben Sie ein Menüleisten-Objekt für die Anwendung an. Für die Anwendung definierte Menüleisten-Objekte werden als Menü für jedes Fenster der Anwendung verwendet, solange das Fenster sein `private` Menü nicht definiert.

Sie müssen hier den Identifikator einer Menüleiste aus der Klasse `Menustrip` an MUI übergeben.

TYP

MUI Objekt

ANWENDBARKEIT

I

6.14 Application.OpenConfigWindow

BEZEICHNUNG

`Application.OpenConfigWindow` – öffnet das MUI-Einstellungsfenster

ÜBERSICHT

```
mui.DoMethod(id, "OpenConfigWindow")
```

BESCHREIBUNG

Seit MUI 3 können Anwendungen ihr eigenes MUI-Konfigurationsfenster öffnen, damit Benutzer die lokalen Einstellungen anpassen können, ohne dass ein externes Programm benötigt wird. Programmierer sollten einen "Einstellungen/MUI ..." - Menüpunkt einbeziehen, der einfach `Application.OpenConfigWindow` aufruft. MUI zeigt dann automatisch das Einstellungsfenster an, ohne den Rest des Programms zu blockieren.

EINGABEN

`id` ID des Anwendungs-Objekts

6.15 Application.RemoveWindow**BEZEICHNUNG**

`Application.RemoveWindow` – löst das Fenster vom Anwendungs-Objekt (V1.2)

ÜBERSICHT

```
mui.DoMethod(id, "RemoveWindow", window)
```

BESCHREIBUNG

Diese Methode entfernt das angegebene Fenster-Objekt aus dem Anwendungs-Objekt und versetzt das Fenster-Objekt in den Status "losgelöst". Fenster-Objekte im losgelösten Zustand können entweder durch Ausführen der Methode `Application.AddWindow` erneut angefügt oder durch Aufrufen von `mui.FreeObject()` gelöscht werden.

EINGABEN

`id` ID des Anwendungs-Objekts

`window` ID des zu entfernenden Fenster-Objekts

6.16 Application.Save**BEZEICHNUNG**

`Application.Save` – speichert die Programmkonfiguration (V1.7)

ÜBERSICHT

```
mui.DoMethod(id, "Save", name$)
```

BESCHREIBUNG

`Application.Save`, `Application.Load` und `Notify.ExportID` bieten eine einfache Möglichkeit zum Speichern und Laden einer Programmkonfiguration.

Jedes Gadget mit einem `Notify.ExportID` sichert seinen Inhalt während dem Speichern mit `Application.Save` und erhält ihn während dem Wiederherstellen mit `Application.Load`. Dies macht es sehr einfach, ein Konfigurationsfenster mit den Schaltflächen "Speichern", "Verwenden" und "Abbrechen" zu entwerfen, damit der Benutzer die Einstellungen speichern kann. Wenn die Anwendung gestartet wird, müssen Sie nur `Application.Load` aufrufen und die gespeicherten Einstellungen werden gelesen und installiert.

Der Parameter `name$` muss auf den Namen der Datei eingestellt werden, in der die Einstellungen gespeichert werden sollen. Normalerweise müssen Sie nicht an einen echten Namen denken, sondern stattdessen einen der folgenden speziellen Werte verwenden:

`::ENV` Einstellungen unter ENV: speichern.

`::ENVARC` Einstellungen unter ENVARC: speichern.

Nicht alle Klassen können ihre Inhalte importieren und exportieren. Derzeit können Sie eine `Notify.ExportID` definieren für:

String-Klasse

`String.Contents` kann ex-/importieren.

Radio-Klasse

`Radio.Active` kann ex-/importieren.

Cycle-Klasse

`Cycle.Active` kann ex-/importieren.

Listview-Klasse

`Listview.Active` kann ex-/importieren.

Text-Klasse

`Text.Contents` kann ex-/importieren.

Slider-Klasse

`Slider.Level` kann ex-/importieren.

Area-Klasse

`Area.Selected` kann ex-/importieren (z.B. für Auswahlkästchen-Gadgets).

MenuItem-Klasse

`MenuItem.Selected` kann ex-/importieren.

Group-Klasse

`Group.ActivePage` kann ex-/importieren.

EINGABEN

<code>id</code>	ID des Anwendungs-Objekts
<code>name\$</code>	Name der Datei, in der die Einstellungen gespeichert werden sollen oder Sonderwert (siehe oben)

6.17 Application.SingleTask

BEZEICHNUNG

`Application.SingleTask` – setzt den Betriebsmodus der Anwendung

BESCHREIBUNG

Boolescher Wert, um anzugeben, ob Ihre Anwendung ein einzelnes Task-Programm ist oder nicht. Wenn `True` gesetzt ist, verweigert MUI die Erstellung mehrerer Anwendungs-Objekte.

In diesem Fall wird das Attribut `Application.DoubleStart` der bereits ausgeführten Anwendung auf `True` gesetzt. Sie können darauf reagieren und geeignete Maßnahmen ergreifen, z.B. durch Öffnen eines Dialogfensters.

Beispiele für Einzelaufgabenanwendungen sind die Systemvoreinstellungen. Es macht keinen Sinn, dass sie mehr als einmal laufen.

TYP

Boolesch

ANWENDBARKEIT

I

6.18 Application.Sleep**BEZEICHNUNG**

Application.Sleep – bringt die Anwendung in den Ruhezustand

BESCHREIBUNG

Dieses Attribut kann verwendet werden, um eine ganze Anwendung in den Ruhezustand zu versetzen. Alle geöffneten Fenster werden deaktiviert und ein Mauszeiger für den beschäftigten Status wird angezeigt.

Dieses Attribut enthält eine Schachtelungszahl. Wenn Sie Ihrer Anwendung sagen, dass sie zweimal ruhen soll, müssen Sie ihr auch mitteilen, dass sie zweimal erwachen soll.

Wenn Sie zeitraubende Aktionen ausführen müssen, sollten Sie immer dieses Attribut festlegen, um den Benutzer darüber zu informieren, dass Sie derzeit keine Eingaben verarbeiten können.

Die Fenster einer schlafenden Anwendung können nicht in der Größe geändert werden.

TYP

Boolesch

ANWENDBARKEIT

S

6.19 Application.WindowList**BEZEICHNUNG**

Application.WindowList – ermittelt eine Liste aller Anwendungsfenster (V1.2)

BESCHREIBUNG

Mit diesem Attribut können Sie eine Tabelle abrufen, die die Identifizierung aller Fenster enthält, die derzeit dem Anwendungs-Objekt zugeordnet sind.

TYP

Tabelle

ANWENDBARKEIT

G

7 Area-Klasse (Bereich)

7.1 Überblick

Die Area-Klasse (Bereich) ist eine Oberklasse für jede andere MUI-Klasse außer Fenster und Anwendungen. Es enthält Informationen über die aktuelle Position, Größe und die Dicke eines Objekts und verwaltet Rahmen, Schriftarten und Hintergründe.

Da die Area-Klasse die Oberklasse für jedes MUI-Gadget ist, können Sie alle Attribute mit allen anderen MUI-Klassen verwenden, die Gadgets erstellen. Sie können beispielsweise `Area.ShortHelp` verwenden, um einer Schaltfläche eine Blasenhilfe hinzuzufügen, oder Sie können mit `Area.ContextMenu` ein Kontextmenü zu einer Listenansicht hinzufügen, usw.

7.2 Area.Background

BEZEICHNUNG

`Area.Background` – passt den Hintergrund eines Objekts an

BESCHREIBUNG

Passt den Hintergrund eines Objekts an.

Jedes MUI-Objekt hat seine eigene Hintergrundeinstellung. Der Hintergrund wird "hinter" den tatsächlichen Objekthalten angezeigt, z.B. hinter dem Text eines Text-Objekts oder hinter dem Bild eines Bild-Objekts.

Ein Objekt ohne eine bestimmte Hintergrundeinstellung erbt das Muster von seiner Übergeordneten Gruppe. Der Standardhintergrund für ein Fenster und viele andere Hintergrundmuster sind mit dem Voreinstellungsprogramm einstellbar.

Folgende Hintergrundeinstellungen sind möglich:

- WindowBack
- RequesterBack
- ButtonBack
- ListBack
- TextBack
- PropBack
- PopupBack
- SelectedBack
- RegisterBack
- GroupBack
- SliderBack
- PageBack
- ReadListBack
- `pre_Background`
- `pre_Shadow`
- `pre_Shine`

- pre_Fill
- pre_ShadowBack
- pre_ShadowFill
- pre_ShadowShine
- pre_FillBack
- pre_FillShine
- pre_ShineBack
- pre_FillBack2
- pre_HshineBack
- pre_HshadowBack
- pre_HshineShine
- pre_HshadowShadow
- pre_MarkShine
- pre_MarkHalfshine
- pre_MarkBackground

Alle Hintergründe mit "pre" sind MUIs vordefinierte Muster. Diese sind vom Benutzer nicht konfigurierbar und sehen immer gleich aus.

Hinweis: Es ist *wichtig**, dass Sie Ihre Programme mit einer ausgefallenen Musterkonfiguration testen. Mit der Standardeinstellung werden Sie keine Fehler in Ihren Hintergründen bemerken.

TYP

Zeichenkette (siehe oben für mögliche Werte)

ANWENDBARKEIT

IS

7.3 Area.BackgroundBrush

BEZEICHNUNG

Area.BackgroundBrush – legt den Hintergrund für einen MUI-Pinsel fest (V1.5)

BESCHREIBUNG

Verwendet den angegebenen MUI-Pinsel als Hintergrund für das Objekt. Sie müssen den Namen eines externen MUI-Pinsels an dieses Attribut übergeben.

Jedes MUI-Objekt hat seine eigene Hintergrundeinstellung. Der Hintergrund wird "hinter" den tatsächlichen Objekthalten angezeigt, z.B. hinter dem Text eines Text-Objekts oder hinter dem Bild eines Bild-Objekts.

Ein Objekt ohne eine bestimmte Hintergrundeinstellung erbt das Muster von seiner Übergeordneten Gruppe. Der Standardhintergrund für ein Fenster und viele andere Hintergrundmuster sind mit dem Voreinstellungsprogramm einstellbar.

Beachten Sie, dass es in der Regel keine gute Idee ist, den Hintergrund in ein fest codiertes Bild zu ändern, da dies leicht mit den aktuellen Skin-Einstellungen des Benutzers

kollidieren kann. Denken Sie daran, dass alles in MUI vom Benutzer konfigurierbar ist, also sollten Sie diese Einstellungen nicht überschreiben, indem Sie Ihre eigenen Skins definieren.

TYP

Zeichenkette

ANWENDBARKEIT

IS

7.4 Area.BackgroundImage

BEZEICHNUNG

Area.BackgroundImage – legt den Hintergrund für eine Bilddatei fest (V1.2)

BESCHREIBUNG

Verwendet das angegebene Bild als Hintergrund für das Objekt. Sie müssen den Dateinamen von dem Bild in diesem Attribut übergeben. Das Bild wird dann über Datentypen geladen.

Jedes MUI-Objekt hat seine eigene Hintergrundeinstellung. Der Hintergrund wird "hinter" den tatsächlichen Objekthinhalten angezeigt, z.B. hinter dem Text eines Text-Objekts oder hinter dem Bild eines Bild-Objekts.

Ein Objekt ohne eine bestimmte Hintergrundeinstellung erbt das Muster von seiner Übergeordneten Gruppe. Der Standardhintergrund für ein Fenster und viele andere Hintergrundmuster sind mit dem Voreinstellungsprogramm einstellbar.

Beachten Sie, dass es in der Regel keine gute Idee ist, den Hintergrund in ein fest codiertes Bild zu ändern, da dies leicht mit den aktuellen Skin-Einstellungen des Benutzers kollidieren kann. Denken Sie daran, dass alles in MUI vom Benutzer konfigurierbar ist, also sollten Sie diese Einstellungen nicht überschreiben, indem Sie Ihre eigenen Skins definieren.

TYP

Zeichenkette

ANWENDBARKEIT

IS

7.5 Area.BackgroundColor

BEZEICHNUNG

Area.BackgroundColor – stellt den Hintergrund auf eine RGB-Farbe ein (V1.2)

BESCHREIBUNG

Legt den Hintergrund dieses Objekts auf die angegebene RGB-Farbe fest.

Jedes MUI-Objekt hat seine eigene Hintergrundeinstellung. Der Hintergrund wird "hinter" den tatsächlichen Objekthinhalten angezeigt, z.B. hinter dem Text eines Text-Objekts oder hinter dem Bild eines Bild-Objekts.

Ein Objekt ohne eine bestimmte Hintergrundeinstellung erbt das Muster von seiner Übergeordneten Gruppe. Der Standardhintergrund für ein Fenster und viele andere Hintergrundmuster sind mit dem Voreinstellungsprogramm einstellbar.

Beachten Sie, dass es in der Regel keine gute Idee ist, den Hintergrund in einen fest codierten Farbwert zu ändern, da dies leicht mit den aktuellen Skin-Einstellungen des Benutzers kollidieren kann. Denken Sie daran, dass alles in MUI vom Benutzer konfigurierbar ist, also sollten Sie diese Einstellungen nicht überschreiben, indem Sie eigene Farbschemata definieren.

Aus dem Hollywood-Skript wird die Farbe als einfacher numerischer Wert angegeben, der 8 Bits für jede Komponente enthält. Wenn Sie die Farbe in der XML-Datei angeben, muss sie als 6-stellige Zeichenfolge mit dem Präfix # übergeben werden (genau wie in HTML).

TYP

Nummer

ANWENDBARKEIT

IS

7.6 Area.BottomEdge

BEZEICHNUNG

Area.BottomEdge – ermittelt die untere Kante des Objekts

BESCHREIBUNG

Sie können dies verwenden, um die aktuelle Position und Dimension eines Objekts zu ermitteln, wenn Sie z.B. ein Dialog brauchen, der unten geöffnet werden soll.

Dieses Attribut ist natürlich nur gültig, wenn das Übergeordnete Fenster des Objekts gerade geöffnet ist.

TYP

Nummer

ANWENDBARKEIT

G

7.7 Area.ContextMenu

BEZEICHNUNG

Area.ContextMenu – stellt das Kontextmenü für das Objekt ein

BESCHREIBUNG

Gibt ein kontextsensitives Popup-Menü für das aktuelle Objekt an. Für MUI sind Popup-Menüs nichts anderes als Standard-Intuition-Menüs, daher müssen Sie hier die ID eines MUI-Menuleisten-Objekts angeben.

Immer wenn der Benutzer den rechten Mausknopf klickt und sich die Maus über dem übergeordneten Objekt befindet, zeigt MUI anstelle des Fenster-Menüs das Popup-Menü an.

Bitte beachten Sie, dass Menuleisten, die als Kontextmenü verwendet werden, nicht mehr als einen <menu> Baum enthalten dürfen.

In MUI Royale 1.0 konnte dieses Attribut nur in der Initialisierungsphase verwendet werden. Ab MUI Royale 1.1 hat es eine Anwendbarkeit von "IS", das heißt Sie können zur Laufzeit auch Kontextmenüs ändern. Um ein Kontextmenü vollständig zu entfernen, übergeben Sie die Sonderzeichenfolge "(none)" an dieses Attribut. Siehe [Abschnitt 29.1 \[Menustrip-Klasse\], Seite 177](#), für ein Beispiel.

TYP

MUI Objekt

ANWENDBARKEIT

IS

7.8 Area.ContextMenuTrigger

BEZEICHNUNG

Area.ContextMenuTrigger – reagiert auf Kontextmenü-Ereignisse

BESCHREIBUNG

Diese Benachrichtigung gibt die ID des Kontextmenüs zurück, die der Benutzer ausgewählt hat.

TYP

Zeichenkette

ANWENDBARKEIT

N

7.9 Area.ControlChar

BEZEICHNUNG

Area.ControlChar – legt die Verknüpfung für das Objekt fest

BESCHREIBUNG

Das Drücken des Steuerzeichens hat die gleiche Wirkung wie das Drücken von Zurück, wenn das Objekt aktiv war.

Dies kann verwendet werden, um Tastenkombinationen im alten Stil zu erstellen.

Hinweis: Durch die Verwendung eines Großbuchstaben-Steuerzeichens wird der Benutzer gezwungen, die Umschalttaste zu drücken.

TYP

Einzelne Zeichenkette

ANWENDBARKEIT

ISG

7.10 Area.CycleChain

BEZEICHNUNG

Area.CycleChain – setzt/ermittelt den Flag für die Tastaturabfolgekette

BESCHREIBUNG

MUI 3 führt ein neues Tastaturabfolgekette-System ein. Alles was Sie tun müssen, ist `Area.CycleChain` für jedes Objekt, das Sie in Ihrer Tastaturabfolgekette haben möchten, auf `True` zu setzen. MUI erledigt den Rest automatisch. Siehe [Abschnitt 3.8 \[Tastaturabfolgekette\]](#), [Seite 14](#), für Details.

TYP

Boolesch

ANWENDBARKEIT

ISG

7.11 Area.Disabled

BEZEICHNUNG

Area.Disabled – setzt/ermittelt den Deaktivierstatus

BESCHREIBUNG

Deaktivieren oder aktivieren Sie ein Gadget. Wenn dieses Attribut gesetzt wird, wird ein Gadget deaktiviert, es wird ein Geist-Muster angezeigt und reagiert nicht mehr auf Benutzereingaben.

Deaktivierte Gadgets können nicht mit der TAB-Taste aktiviert werden.

Mit `Area.Disabled` für eine Gruppe von Objekten werden alle Objekte innerhalb dieser Gruppe deaktiviert.

TYP

Boolesch

ANWENDBARKEIT

ISG

7.12 Area.FixHeight

BEZEICHNUNG

Area.FixHeight – fixiert die Höhe eines Objektes

BESCHREIBUNG

Geben Sie Ihrem Objekt eine feste Pixelhöhe. Dieser Tag wird in einer allgemeinen MUI-Anwendung absolut nicht benötigt und ist nur in Notfallsituationen vorhanden. Bitte denken Sie zweimal darüber nach, bevor Sie ihn benutzen!

TYP

Nummer

ANWENDBARKEIT

I

7.13 Area.FixHeightTxt**BEZEICHNUNG**

Area.FixHeightTxt – fixiert die Höhe eines Objektes mit einer Zeichenkette

BESCHREIBUNG

Geben Sie Ihrem Objekt eine feste Pixelhöhe. Die Höhe entspricht der Höhe der angegebenen Zeichenkette. Dieser Tag wird in einer allgemeinen MUI-Anwendung absolut nicht benötigt und ist nur in Notfallsituationen vorhanden. Bitte denken Sie zweimal darüber nach, bevor Sie es benutzen!

TYP

Zeichenkette

ANWENDBARKEIT

I

7.14 Area.FixWidth**BEZEICHNUNG**

Area.FixWidth – fixiert die Breite eines Objektes

BESCHREIBUNG

Geben Sie Ihrem Objekt eine feste Pixelbreite. Dieser Tag wird in einer allgemeinen MUI-Anwendung absolut nicht benötigt und ist nur in Notfallsituationen vorhanden. Bitte denken Sie zweimal darüber nach, bevor Sie ihn benutzen!

TYP

Nummer

ANWENDBARKEIT

I

7.15 Area.FixWidthTxt**BEZEICHNUNG**

Area.FixWidthTxt – fixiert die Breite des Objekts mit einer Zeichenkette

BESCHREIBUNG

Geben Sie Ihrem Objekt eine feste Pixelbreite. Die Breite entspricht der Breite der angegebenen Zeichenkette. Dieser Tag wird in einer allgemeinen MUI-Anwendung absolut nicht benötigt und ist nur für Notfallsituationen vorhanden. Bitte denken Sie zweimal darüber nach, bevor Sie ihn benutzen!

TYP

Zeichenkette

ANWENDBARKEIT

I

7.16 Area.Font**BEZEICHNUNG**

Area.Font – passt die Objektschrift an

BESCHREIBUNG

Jedes MUI-Objekt kann eine eigene Schriftart haben, setzen Sie sie einfach mit diesem Tag. Objekte ohne explizite Schriftarteneinstellung erben sie von ihrer übergeordneten Gruppe.

Normalerweise müssen Sie keine Schriftart selbst öffnen, sondern nur einen der vordefinierten Werte verwenden, um eine Schriftart aus den Benutzereinstellungen zu erhalten.

Folgende Schriftarten sind möglich:

- Inherit
- Normal
- List
- Tiny
- Fixed
- Title
- Big
- Button

TYP

Zeichenkette (siehe oben für mögliche Werte)

ANWENDBARKEIT

IG

7.17 Area.Frame**BEZEICHNUNG**

Area.Frame – definiert einen Objektrahmen

BESCHREIBUNG

Definieren Sie einen Rahmen für das aktuelle Objekt. Da die Area-Klasse (Bereich) eine Oberklasse für alle Elemente in einem Fenster ist, können Sie jedem gewünschten Objekt einen Rahmen zuweisen.

Sie passen den Stil Ihres Rahmens nicht direkt an, sondern geben nur einen Typ an:

None Für keinen Rahmen.

Button Für Standardschaltflächen mit Text darin.

ImageButton

Für kleine Schaltflächen mit Bildern, z.B. die Pfeile einer Bildlaufleiste.

Text	Für ein Textfeld, z.B. eine Statuszeilenanzeige.
String	Für ein Zeichenketten-Gadget.
ReadList	Für eine Nur-Lese-Liste.
InputList	Für eine Liste, die Eingaben verarbeitet (hat einen Cursor).
Prop	Für Schieberegler-Gadgets.
Gauge	Für Fortschrittsbalken-Gadgets.
Group	Für Gruppen.
PopUp	Für Dialog-Gadgets.
Virtual	Für virtuelle Gruppen.
Slider	Für Schieberegler-Gadgets.

Wie der Rahmen aussehen wird, ist über das Voreinstellungsprogramm einstellbar.

Zu jedem Rahmen gehören vier Abstandswerte, die MUI mitteilen, wie viele Pixel zwischen dem Rahmen und seinem Inhalt freigelassen werden sollen. Diese Abstandswerte sind auch vom Benutzer einstellbar, solange Sie sie nicht mit einem von `Area.InnerLeft`, `Area.InnerRight`, `Area.InnerTop` oder `Area.InnerBottom` überschreiben.

Hinweis: Das erste Objekt in einem Fenster darf **keinen** Rahmen haben. Wenn Sie dies benötigen, müssen Sie eine Dummy-Gruppe mit nur einem untergeordneten Element erstellen.

TYP

Zeichenkette (siehe oben für mögliche Werte)

ANWENDBARKEIT

I

7.18 Area.FramePhantomHoriz

BEZEICHNUNG

`Area.FramePhantomHoriz` – erstellt einen Phantomrahmen

BESCHREIBUNG

Wenn Sie dies auf `True` setzen, wird der angegebene Rahmen als horizontaler Phantomrahmen verwendet. Der Rahmen wird nicht angezeigt, aber seine vertikalen Komponenten (Rahmenhöhe, innerer oberer und innerer untere Abstand) werden zur Berechnung von Positionen und Abmessungen verwendet (horizontale Komponenten werden als 0 behandelt).

Dies ist äußerst nützlich für eine korrekte Beschriftung von Objekten. Sie würden z.B. ein Zeichenketten-Gadget mithilfe eines Text-Objekts mit einem Phantomzeichenkettenrahmen benennen. Daher befindet sich der Beschriftungstext immer an derselben vertikalen Position wie der Text des Zeichenketten-Gadget, unabhängig davon, welche Abstandswerte der Benutzer konfiguriert hat.

TYP

Boolesch

ANWENDBARKEIT

I

7.19 Area.FrameTitle

BEZEICHNUNG

Area.FrameTitle – legt den Rahmentitel fest

BESCHREIBUNG

Mit diesem Tag legen Sie eine Textzeichenkette fest, die in der oberen Zeile eines Rahmens zentriert angezeigt wird. Dies kann nützlich sein, wenn Sie Objektgruppen benennen möchten.

Sie dürfen `Area.FrameTitle` nicht verwenden, ohne einen `Area.Frame` zu definieren.

TYP

Zeichenkette

ANWENDBARKEIT

I

7.20 Area.Height

BEZEICHNUNG

Area.Height – ermittelt die Höhe eines Objekts

BESCHREIBUNG

Sie können dies verwenden, um die aktuelle Position und Dimension eines Objekts zu ermitteln, wenn Sie z.B. einen Dialog brauchen, der unten geöffnet werden soll.

Dieses Attribut ist natürlich nur gültig, wenn das übergeordnete Fenster des Objekts gerade geöffnet ist.

TYP

Nummer

ANWENDBARKEIT

G

7.21 Area.Hide

BEZEICHNUNG

Area.Hide – blendet ein Objekt ein/aus

BESCHREIBUNG

Objekte mit diesem Attributsatz werden nicht angezeigt. Sie können `Area.Hide` jederzeit einstellen, wodurch Objekte erscheinen oder sofort verschwinden. Ein neues Layout wird

berechnet, sobald einige Objekte ein- oder ausgeblendet werden. Bei Bedarf ändert MUI das übergeordnete Fenster, um Platz für die neuen Objekte zu schaffen.

Momentan führt MUI eine vollständige Aktualisierung des Fensters durch, nachdem Objekte ein-/ausgeblendet wurden. Dieses Verhalten wird möglicherweise in Zukunft verbessert.

TYP

Boolesch

ANWENDBARKEIT

ISG

7.22 Area.HorizWeight

BEZEICHNUNG

Area.HorizWeight – setzt/ermittelt das horizontale Objektgrößenverhältnis

BESCHREIBUNG

Passen Sie das horizontale Größenverhältnis eines Objekts an. Normalerweise können Sie einfach `Area.Weight` anstelle dieses Tags verwenden, aber in einigen zweidimensionalen Gruppen kann es nützlich sein, unterschiedliche horizontale und vertikale Größenverhältnisse zu verwenden.

TYP

Nummer

ANWENDBARKEIT

ISG

7.23 Area.InnerBottom

BEZEICHNUNG

Area.InnerBottom – passt den unteren Abstand zwischen Objekt und Rahmen an

BESCHREIBUNG

Passen Sie den Abstand zwischen einem Objekt und seinem unteren Rahmenrand an. Normalerweise sollten Sie diesen Tag nicht verwenden, da Sie die bevorzugte Einstellung des Benutzers überschreiben werden.

TYP

Nummer

ANWENDBARKEIT

IG

7.24 Area.InnerLeft

BEZEICHNUNG

Area.InnerLeft – passt den linken Abstand zwischen Objekt und Rahmen an

BESCHREIBUNG

Passen Sie den Abstand zwischen einem Objekt und seinem linken Rahmenrand an. Normalerweise sollten Sie diesen Tag nicht verwenden, da Sie die bevorzugte Einstellung des Benutzers überschreiben werden.

TYP

Nummer

ANWENDBARKEIT

IG

7.25 Area.InnerRight

BEZEICHNUNG

Area.InnerRight – passt den rechten Abstand zwischen Objekt und Rahmen an

BESCHREIBUNG

Passen Sie den Abstand zwischen einem Objekt und seinem rechten Rahmenrand an. Normalerweise sollten Sie diesen Tag nicht verwenden, da Sie die bevorzugte Einstellung des Benutzers überschreiben werden.

TYP

Nummer

ANWENDBARKEIT

IG

7.26 Area.InnerTop

BEZEICHNUNG

Area.InnerTop – passt den oberen Abstand zwischen Objekt und Rahmen an

BESCHREIBUNG

Passen Sie den Abstand zwischen einem Objekt und seinem oberen Rahmenrand an. Normalerweise sollten Sie diesen Tag nicht verwenden, da Sie die bevorzugte Einstellung des Benutzers überschreiben werden.

TYP

Nummer

ANWENDBARKEIT

IG

7.27 Area.InputMode

BEZEICHNUNG

Area.InputMode – passt den Eingabemodus für das Objekt an (V1.5)

BESCHREIBUNG

Hiermit können Sie den Eingabemodus für das Objekt anpassen.

MUI hat keine eindeutige Button-Klasse (Schaltflächen). Stattdessen können Sie jedes Objekt (sogar Gruppen) wie eine Schaltfläche behandeln, indem Sie einen Eingabemodus für sie festlegen. Mehrere Eingabemodi sind verfügbar:

None: Keine Eingabe, das ist kein Gadget.

RelVerify:
Für Knöpfe und ähnliche Sachen.

Immediate:
Sie können ihn z.B. in einem Radio-Knopf-Objekt verwenden.

Toggle: Für Dinge wie Auswahlkästchen-Gadgets.

Die Einstellung für den Eingabemodus bestimmt, wie eine Benutzeraktion die Attribute `Area.Selected` und `Area.Pressed` auslöst. Einzelheiten finden Sie in der Dokumentation.

TYP

Zeichenkette (siehe oben für mögliche Werte)

ANWENDBARKEIT

I

7.28 Area.LeftEdge

BEZEICHNUNG

Area.LeftEdge – ermittelt die linke Kante des Objekts

BESCHREIBUNG

Sie können dies verwenden, um die aktuelle Position und Dimension eines Objekts zu ermitteln, wenn Sie z.B. einen Dialog öffnen wollen.

Dieses Attribut ist natürlich nur gültig, wenn das übergeordnete Fenster des Objekts gerade geöffnet ist.

TYP

Nummer

ANWENDBARKEIT

G

7.29 Area.MaxHeight

BEZEICHNUNG

Area.MaxHeight – legt die maximale Höhe des Objekts fest

BESCHREIBUNG

Geben Sie die maximale Höhe für ein Objekt (in Pixeln) an.

TYP

Nummer

ANWENDBARKEIT

I

7.30 Area.MaxWidth

BEZEICHNUNG

Area.MaxWidth – legt die maximale Breite des Objekts fest

BESCHREIBUNG

Geben Sie die maximale Breite für ein Objekt (in Pixeln) an.

TYP

Nummer

ANWENDBARKEIT

I

7.31 Area.Pressed

BEZEICHNUNG

Area.Pressed – benachrichtigt, wenn eine Taste gedrückt/losgelassen wird (V1.5)

BESCHREIBUNG

Benachrichtigt Sie, ob eine Taste gedrückt oder losgelassen wird. Das Attribut `Area.Pressed` eines Gadgets wird abhängig vom Eingabemodus von einer Benutzeraktion ausgelöst:

RelVerify:

Wird eingestellt, wenn der linke Mausknopf gedrückt wird und gelöscht, wenn der linke Mausknopf freigegeben wird und sich die Maus immer noch über dem Gadget befindet (andernfalls wird es ebenfalls gelöscht, ohne jedoch ein Benachrichtigungsereignis auszulösen).

Immediate:

Undefiniert, verwenden Sie dafür `Area.Selected`.

Toggle: Undefiniert, verwenden Sie dafür `Area.Selected`.

Warten auf `Area.Pressed` um `False` zu bekommen, ist die übliche Art, auf Schaltflächengadgets zu reagieren.

TYP

Boolesch

ANWENDBARKEIT

N

7.32 Area.RightEdge

BEZEICHNUNG

Area.RightEdge – ermittelt die rechte Kante des Objekts

BESCHREIBUNG

Sie können dies verwenden, um die aktuelle Position und Dimension eines Objekts zu ermitteln, wenn Sie z.B. einen Dialog öffnen wollen.

Dieses Attribut ist natürlich nur gültig, wenn das übergeordnete Fenster des Objekts gerade geöffnet ist.

TYP

Nummer

ANWENDBARKEIT

G

7.33 Area.Selected

BEZEICHNUNG

Area.Selected – setzt/ermittelt den ausgewählten Status des Gadgets (V1.5)

BESCHREIBUNG

Dieses Attribut ermittelt den ausgewählten Status eines Gadgets oder legt ihn fest. Es kann ausgelöst werden, indem der Benutzer je nach Eingabemodus auf das Gadget (oder über die Tastatur) klickt:

RelVerify:

Wird eingestellt, wenn die linke Maustaste gedrückt wird. Wird gelöscht, wenn die linke Maustaste freigegeben wird und auch gelöscht, wenn das Gadget ausgewählt ist und die Maus das Gadget-Feld verlässt. Wird festgelegt, wenn die Maus das Gadget-Feld erneut aufruft.

Immediate:

Wird eingestellt, wenn die linke Maustaste gedrückt wird.

Toggle: Schaltet um, wenn die linke Maustaste gedrückt wird.

Natürlich können Sie dieses Attribut selbst einstellen, z.B. um den Status eines Auswahlkästchen-Gadgets anzupassen.

Ein ausgewähltes Gadget zeigt seine Umrandung an und ruft den konfigurierten Hintergrund `SelectedBack` auf. Dies kann mit dem Tag `Area.ShowSelState` vermieden werden.

TYP

Boolesch

ANWENDBARKEIT

ISGN

7.34 Area.ShortHelp

BEZEICHNUNG

Area.ShortHelp – setzt/ermittelt die Blasen-Hilfe-Zeichenkette

BESCHREIBUNG

Geben Sie eine Zeichenkette an, die als Blasenhilfe für dieses Objekt verwendet werden soll.

Die Zeichenkette, die Sie hier angeben, kann Textformatierungs-codes verwenden. Siehe [Abschnitt 3.9 \[Textformatierungs-codes\]](#), Seite 14, für Details.

TYP

Zeichenkette

ANWENDBARKEIT

ISG

7.35 Area.ShowSelState

BEZEICHNUNG

Area.ShowSelState – konfiguriert den Hintergrundstatus (V1.5)

BESCHREIBUNG

Normalerweise kehrt ein Gadget seinen Rahmen um (invertiert) und zeigt das konfigurierte Hintergrundmuster `SelectedBack` im ausgewählten Zustand an. Bei einigen Objekten (z.B. Auswahlkästchen) wird dies nicht empfohlen und kann unterdrückt werden, indem `Area.ShowSelState` auf `False` gesetzt wird.

TYP

Boolesch

ANWENDBARKEIT

I

7.36 Area.TopEdge

BEZEICHNUNG

Area.TopEdge – ermittelt die obere Kante des Objekts

BESCHREIBUNG

Sie können dies verwenden, um die aktuelle Position und Dimension eines Objekts zu ermitteln, wenn Sie z.B. einen Dialog öffnen wollen.

Dieses Attribut ist natürlich nur gültig, wenn das übergeordnete Fenster des Objekts gerade geöffnet ist.

TYP

Nummer

ANWENDBARKEIT

G

7.37 Area.VertWeight

BEZEICHNUNG

Area.VertWeight – setzt/ermittelt das vertikale Objektgrößenverhältnis

BESCHREIBUNG

Passen Sie das vertikale Größeverhältnis eines Objekts an. Normalerweise können Sie einfach `Area.Weight` anstelle dieses Tags verwenden, aber in einigen zweidimensionalen Gruppen kann es nützlich sein, unterschiedliche horizontale und vertikale Größenverhältnisse zu verwenden.

TYP

Nummer

ANWENDBARKEIT

ISG

7.38 Area.Weight

BEZEICHNUNG

Area.Weight – setzt/ermittelt das Objektgrößenverhältnis

BESCHREIBUNG

Dieser Tag ist eine Abkürzung für `Area.HorizWeight` und `Area.VertWeight`, es stellt beide Größenverhältnisse gleichzeitig ein.

Die Größe eines Objekts bestimmt, wie viel Platz es während des Layouts erhält. Stellen Sie sich vor, Sie haben eine 100 Pixel breite horizontale Gruppe mit zwei Zeichenketten-Gadgets. Normalerweise erhält jedes Gadget die Hälfte des Raums und ist 50 Pixel breit. Wenn Sie glauben, dass das linke Gadget wichtiger ist und größer sein sollte, können Sie ihm ein Größeverhältnis von 200 (und 100 für das rechte Gadget) geben. Da das linke Gadget zweimal so "groß" ist wie das rechte Gadget, wird es doppelt so groß (etwa 66 Pixel) wie das rechte (34 Pixel).

Das Größenverhältnis ist natürlich nur dann sinnvoll, wenn das Objekt in der Größe veränderbar ist. Ein `Area.VertWeight` für ein Zeichenketten-Gadget (immer feste Höhe) ist nutzlos.

Ein Objekt mit einem Objektgrößenverhältnis von 0 bleibt immer auf seiner minimalen Größe.

Standardmäßig haben alle Objekte ein Objektgrößenverhältnis von 100.

TYP

Nummer

ANWENDBARKEIT

I

7.39 Area.Width**BEZEICHNUNG**

Area.Width – ermittelt die Breite des Objekts

BESCHREIBUNG

Sie können dies verwenden, um die aktuelle Position und Dimension eines Objekts zu lesen, wenn Sie z.B. ein Dialog brauchen, der links oder rechts geöffnet werden soll.

Dieses Attribut ist natürlich nur gültig, wenn das übergeordnete Fenster des Objekts gerade geöffnet ist.

TYP

Nummer

ANWENDBARKEIT

G

8 Busy-Klasse (Beschäftigung)

8.1 Überblick

Die Busy-Klasse (Beschäftigung) erstellt ein unbestimmtes Fortschrittsanzeige-Gadget. Dies ist nützlich, wenn Sie einen Beschäftigungszustand anzeigen müssen, ohne zu wissen, wann die Verarbeitung beendet wird. Für eine lineare Fortschrittsanzeige verwenden Sie stattdessen die Gauge-Klasse (Fortschrittsbalken). Siehe [Abschnitt 16.1 \[Gauge-Klasse\]](#), Seite 99, für Details.

Diese Klasse benötigt mindestens MUI Royale 1.4.

8.2 Busy.Move

BEZEICHNUNG

Busy.Move – verschiebt die Aktivitätsleiste (V1.4)

ÜBERSICHT

```
mui.DoMethod(id, "Move")
```

BESCHREIBUNG

Verschiebt die Aktivitätsleiste in dem unbestimmten Fortschrittsanzeige-Gadget.

EINGABEN

`id` ID des unbestimmte Fortschrittsanzeige-Objekts

8.3 Busy.Speed

BEZEICHNUNG

Busy.Speed – setzt/ermittelt die Geschwindigkeit der Animation (V1.4)

BESCHREIBUNG

Wenn Sie dieses Attribut festlegen, wird die Geschwindigkeit der verschobenen Aktivitätsleiste gesetzt. Gültiger Wertebereich ist 1 bis 250 (Millisekunden).

Zusätzlich gibt es folgende spezielle Werte:

`Off` Stoppt die Aktivitätsleiste.

`User` Verwendet die benutzerdefinierte Geschwindigkeit aus den Einstellungen.

TYP

Nummer oder Zeichenkette (siehe oben für mögliche Werte)

ANWENDBARKEIT

ISG

9 Button-Klasse (Schaltflächen)

9.1 Übersicht

Die Button-Klasse (Schaltflächen) ermöglicht es Ihnen, Schaltflächen sehr einfach zu erstellen. Wenn die Beschriftung der Schaltfläche einen Unterstrich enthält, wird MUI Royale das erste Zeichen nach diesem Unterstrich automatisch als Tastaturkürzel einrichten. Wenn Sie dieses Verhalten nicht möchten, setzen Sie `Button.NoAutoKey` auf `True`.

Beispiel:

```
<button id="ok" notify="pressed">OK</button>
```

Die Beschriftung für die Schaltfläche, die Sie hier angeben, kann Textformatierungs-codes verwenden. Siehe [Abschnitt 3.9 \[Textformatierungs-codes\]](#), [Seite 14](#), für Details.

9.2 Button.HiChar

BEZEICHNUNG

`Button.HiChar` – hebt ein Zeichen in der Schaltflächenbeschriftung hervor

BESCHREIBUNG

Wenn das hier angegebene Zeichen in der angezeigten Zeichenkette vorhanden ist (egal ob Groß- oder Kleinschreibung), wird es unterstrichen. Das Markieren eines Zeichens in der Schaltflächenbeschriftung ist in Verbindung mit `Area.ControlChar` nützlich.

TYP

Einzelne Zeichenkette

ANWENDBARKEIT

I

9.3 Button.Label

BEZEICHNUNG

`Button.Label` – setzt/ermittelt die Schaltflächenbeschriftung

BESCHREIBUNG

Dieses Attribut legt die Beschriftung der Schaltfläche fest oder ermittelt sie.

Die Zeichenfolge, die Sie hier angeben, kann Textformatierungs-codes verwenden. Siehe [Abschnitt 3.9 \[Textformatierungs-codes\]](#), [Seite 14](#), für Details.

TYP

Zeichenkette

ANWENDBARKEIT

SG

9.4 Button.NoAutoKey

BEZEICHNUNG

Button.NoAutoKey – deaktiviert die automatische Tastenkombination

BESCHREIBUNG

Wenn die Schaltflächenbeschriftung einen Unterstrich enthält, wird MUI Royale das erste Zeichen nach diesem Unterstrich automatisch als Tastaturkürzel einrichten. D.h. wenn die Schaltflächenbeschriftung "_OK" lautet, wird "o" automatisch zur Tastenkombination für diese Schaltfläche. Wenn Sie dieses Verhalten nicht wünschen, setzen Sie dieses Attribut auf `True`.

Voreingestellt ist `False`.

TYP

Boolesch

ANWENDBARKEIT

I

9.5 Button.Pressed

BEZEICHNUNG

Button.Pressed – benachrichtigt, ob eine Schaltfläche gedrückt wird

BESCHREIBUNG

Dieses Attribut wird ausgelöst, wenn der Benutzer eine Schaltfläche drückt.

TYP

Boolesch

ANWENDBARKEIT

N

9.6 Button.Selected

BEZEICHNUNG

Button.Selected – schaltet den Auswahlstatus um

BESCHREIBUNG

Verwenden Sie diese Option, um den Auswahlstatus einer Umschalt-Schaltfläche umzuschalten oder sich über ein Benutzerumschaltereignis zu informieren. Wenn Sie dieses Attribut verwenden möchten, müssen Sie `Button.Toggle` zuerst auf `True` setzen.

TYP

Boolesch

ANWENDBARKEIT

ISGN

9.7 Button.Toggle

BEZEICHNUNG

Button.Toggle – erstellt eine Umschalt-Schaltfläche

BESCHREIBUNG

Wenn Sie dies auf `True` setzen, ist die Schaltfläche eine Umschalt-Schaltfläche, die zwei Zustände haben kann: Ein und Aus. Sie können das Attribut `Button.Selected` verwenden, um Zustände umzuschalten und Benachrichtigungen über das Umschalt-Ereignis zu erhalten.

Standardmäßig ist die Schaltfläche eine normale Schaltfläche.

TYP

Boolesch

ANWENDBARKEIT

I

10 Checkmark-Klasse (Auswahlkästchen)

10.1 Übersicht

Die Checkmark-Klasse (Auswahlkästchen) erstellt eine Bildschaltfläche mit einem Häkchen, das zwei verschiedene Status haben kann. Es wird oft zum Umschalten von Optionen verwendet. Diese Klasse erstellt wirklich nur die Häkchen-Schaltfläche. Es wird keine Beschriftung daneben platziert. Wenn Sie eine Beschriftung für Ihr Auswahlkästchen haben möchten, müssen Sie dies manuell tun, indem Sie neben dem Auswahlkästchen ein Text- oder Beschriftungs-Objekt (Label) platzieren.

Hier ein Beispiel, wie Sie ein Auswahlkästchen mit einer angrenzenden Beschriftung erstellen:

```
<hgroup>
  <label>Enable cool option</label>
  <checkmark/>
</hgroup>
```

10.2 Checkmark.Selected

BEZEICHNUNG

Checkmark.Selected – setzt/ermittelt den Zustand vom Auswahlkästchen

BESCHREIBUNG

Ermitteln oder setzen Sie den ausgewählten Status eines Auswahlkästchen. Dieses Attribut kann ausgelöst werden, indem der Benutzer auf das Gadget klickt oder die Tastatur verwendet. Natürlich können Sie dieses Attribut selbst einstellen, z.B. um den Status eines Auswahlkästchen-Gadgets anzupassen.

Sie können auch eine Benachrichtigung für dieses Attribut einrichten, um zu erfahren, wann der Benutzer den Status des Auswahlkästchens wechselt.

TYP

Boolesch

ANWENDBARKEIT

ISGN

11 Coloradjust-Klasse (Farbanpassung)

11.1 Überblick

Die Coloradjust-Klasse (Farbanpassung) erstellt einige Gadgets, mit denen Sie eine einzelne Farbe anpassen können. Je nach Betriebssystem werden verschiedene Arten von Gadgets verwendet. Kickstart 2.x-Benutzer erhalten möglicherweise nur ein RGB-Schieberegister, Kickstart 3.x-Benutzer können ein zusätzliches Farbrad erhalten, falls verfügbar. Allerdings ist das Aussehen dieser Klasse für Sie als Programmierer nicht wichtig.

Hier ist ein Beispiel für einen XML-Auszug zum Erstellen eines Farbanpassungs-Gadgets, das auf ganz Weiß gesetzt ist:

```
<coloradjust rgb="#ffffff"/>
```

11.2 Coloradjust.RGB

BEZEICHNUNG

Coloradjust.RGB – setzt/ermittelt die Farbe

BESCHREIBUNG

Setzen oder ermitteln Sie die eingestellte Farbe. Wenn Sie eine Benachrichtigung für dieses Attribut einrichten, werden Sie informiert, sobald sich die Farbe ändert. Aus dem Hollywood-Skript wird die Farbe als einfacher numerischer Wert angegeben, der 8 Bits für jede Komponente enthält. Wenn Sie die Farbe in der XML-Datei angeben, muss sie als 6-stellige Zeichenkette mit dem Präfix #-Zeichen übergeben werden (genau wie in HTML).

TYP

Nummer

ANWENDBARKEIT

ISGN

12 Colorfield-Klasse (Farbfeld)

12.1 Übersicht

Die Colorfield-Klasse (Farbfeld) erzeugt ein Rechteck, das mit einer bestimmten Farbe gefüllt ist, nützlich z.B. innerhalb eines Paletten-Dialogs. Sie können die Farbe des Feldes jederzeit ändern, indem Sie seine RGB-Attribute festlegen.

Das Feld versucht, einen exklusiven Stift von dem aktuellen Bildschirm zu erhalten. Wenn keiner verfügbar ist, wird nur ein gerasterter Hintergrund angezeigt. (Vielleicht wird es ein bisschen intelligenter wenn ich versuche, die Farbe durch Mischen von anderen Farben anzuzeigen, aber das ist ein Zukunftsthema.)

Unnötig zu sagen, dass das Farbfeld nur mit Kickstart 3.x und höher funktioniert, da niedrigere Betriebssysteme die Stiftfreigabe nicht unterstützen. Wenn Sie diese Klasse mit einem niedrigeren Betriebssystem verwenden, erhalten Sie auch ein (langweiliges) Raster.

Hier finden Sie einen Beispiel-XML-Auszug zum Erstellen eines Farbfeld-Gadgets, das auf ganz Weiß gesetzt ist:

```
<colorfield rgb="#ffffff"/>
```

12.2 Colorfield.RGB

BEZEICHNUNG

Colorfield.RGB – setzt/ermittelt die Farbe

BESCHREIBUNG

Setzen oder ermitteln Sie die aktuelle Farbe. Wenn Sie eine Benachrichtigung für dieses Attribut einrichten, werden Sie benachrichtigt, sobald sich die Farbe ändert. Aus dem Hollywood-Skript wird die Farbe als einfacher numerischer Wert angegeben, der 8 Bits für jede Komponente enthält. Wenn Sie die Farbe in der XML-Datei angeben, muss sie als 6-stellige Zeichenkette mit dem Präfix #-Charakter übergeben werden (genau wie in HTML).

TYP

Nummer

ANWENDBARKEIT

ISGN

13 Cycle-Klasse (Auswahl-Listefeld)

13.1 Übersicht

Die Cycle-Klasse (Auswahl-Listefeld) generiert ein Gadget, welches eine Kombination aus einer Schaltfläche, schreibgeschütztem Textfeld und ein (konfigurierbares) Auswahl-Listefeld-Menü ist. Mit diesem Menü wird ein Druekklicken vieler Einträge vermieden. Das Textfeld kann nur einen der Einträge der Liste annehmen.

Wenn Sie ein Auswahl-Listefeld-Gadget definieren, müssen Sie mit dem Tag `<item>` das Auswahl-Listefeld-Gadget mit Elementen füllen. Jedes Auswahl-Listefeld-Gadget muss mindestens einen Eintrag haben.

Hier ein Beispiel für einen XML-Auszug zum Erstellen eines Auswahl-Listefeld-Gadgets:

```
<cycle id="printer">
  <item>HP Deskjet</item>
  <item>NEC P6</item>
  <item>Okimate 20</item>
</cycle>
```

Die Auswahl-Listefeldeinträge, die Sie mit dem Tag `<item>` angeben, können Textformatierungs-codes verwenden. Siehe [Abschnitt 3.9 \[Textformatierungscodes\]](#), Seite 14, für Details.

13.2 Cycle.Active

BEZEICHNUNG

Cycle.Active – setzt/ermittelt einen aktiven Auswahl-Listefeldeintrag

BESCHREIBUNG

Dieses Attribut definiert die Nummer des aktiven Eintrags in dem Auswahl-Listefeld-Gadgets. Der gültige Bereich reicht von 0 für den ersten Eintrag bis AnzahlEinträge-1 für den letzten Eintrag.

Durch die Einstellung von `Cycle.Active` wird das Gadget aktualisiert. Auf der anderen Seite, wenn der Benutzer mit dem Gadget herum spielt, gibt `Cycle.Active` immer den aktuellen Status wieder.

Übergeben Sie die speziellen Werte `Next` oder `Prev` während des Setzens, so durchläuft das Gadget seine Einträge in der angegebenen Richtung.

TYP

Nummer oder Zeichenkette (siehe oben für mögliche Werte)

ANWENDBARKEIT

ISGN

14 Dirlist-Klasse (Verzeichnislisten)

14.1 Übersicht

Die Dirlist-Klasse (Verzeichnislisten) bietet eine schnelle und einfache Möglichkeit, Einträge von einem Verzeichnis anzuzeigen. Sie verfügt über viele Kontrollattribute, von denen viele aus dem populären ASL-Dateidialog bekannt sind.

Diese Klasse soll **nicht** die ASL.library ersetzen! Niemand möchte, dass jede MUI-Anwendung mit einem anderen selbst erstellten Dateidialog daherkommt. Bitte verwenden Sie ASL weiterhin für echte Dateianfragen!

Manchmal kann es jedoch nützlich sein, eine kleine Verzeichnisliste irgendwo in Ihrer Benutzerschnittstelle zu haben. Stellen Sie sich einen Anrufbeantworter vor, der eingehende Anrufe in einem vorkonfigurierten Verzeichnis speichert. Mit einem Verzeichnislisten-Objekt können Sie die GUI zum Auswählen eines Anrufs in Ihr Fenster mit vielen anderen Gadgets wie "abspielen", "löschen" usw. einfügen.

Die Dirlist-Klasse erstellt eine Listenansicht mit sechs Spalten, die alle Dateiattribute darstellen: Name, Größe, Datum, Uhrzeit, Flags und Kommentar. Wenn Sie nicht möchten, dass alle diese Attribute angezeigt werden, können Sie mit Listviewcolumn-Klasse (Listensichtspalten) Anpassungen vornehmen.

Die Dirlist-Klasse ist eine Unterklasse der Listview-Klasse (Listenansicht). Daher können Sie auch die meisten Attribute und Methoden der Listview-Klasse verwenden. Wenn Sie beispielsweise die Einträge Ihres Verzeichnisses lesen möchten, senden Sie dem Verzeichnislisten-Objekt eine `Listview.GetEntry`, oder wenn Sie Änderungen im aktiven Listeneintrag mitbekommen möchten, richten Sie eine Benachrichtigung auf `Listview.Active` ein.

Wenn Sie `Listview.GetEntry` für diese Klasse verwenden, erhalten Sie sechs Rückgabewerte, die die sechs verschiedenen Spalten der Listenansicht von Verzeichnislisten darstellen. Beachten Sie, dass Sie immer sechs Rückgabewerte erhalten, auch wenn Sie die Spaltenanzeige mit Hilfe von der Listviewcolumn-Klasse angepasst haben. Die Schutzflags werden als eine einzelne Zahl zurückgegeben, die eine Bitfeldkombination der Konstanten `#FILEATTR_XXX` aus Hollywood enthält.

Wenn Sie ein Verzeichnislisten-Objekt im XML-Code erstellen, müssen Sie immer mindestens eine Spalte hinzufügen. Dies geschieht mithilfe der Listviewcolumn-Klasse. Hier ist ein Beispiel für eine minimale Verzeichnislisten-Definition mit nur einer einzigen Spalte (das ist die Namensspalte):

```
<dirlist>
  <column/> <!-- Name -->
</dirlist>
```

Hier ist eine Definition, die alle sechs Spalten enthält:

```
<dirlist>
  <column/> <!-- Name -->
  <column/> <!-- Size -->
  <column/> <!-- Date -->
  <column/> <!-- Time -->
```

```

    <column/> <!-- Flags -->
    <column/> <!-- Comment -->
</dirlist>

```

Wenn Sie nur die Spalte Name und die Spalte Kommentar anzeigen möchten, können Sie das Attribut `Listviewcolumn.Col` verwenden, um dies zu erreichen:

```

<dirlist>
  <column/>          <!-- Name -->
  <column col="5"/> <!-- Comment -->
</dirlist>

```

Um die Reihenfolge der Spalten umzukehren, verwenden Sie:

```

<dirlist>
  <column col="5"/> <!-- Comment -->
  <column col="4"/> <!-- Flags -->
  <column col="3"/> <!-- Time -->
  <column col="2"/> <!-- Date -->
  <column col="1"/> <!-- Size -->
  <column col="0"/> <!-- Name -->
</dirlist>

```

Siehe [Abschnitt 25.1 \[Listview-Klasse\]](#), Seite 149, für Details.

Siehe [Abschnitt 26.1 \[Listviewcolumn-Klasse\]](#), Seite 165, für Details.

14.2 Dirlist.AcceptPattern

BEZEICHNUNG

`Dirlist.AcceptPattern` – legt ein Filtermuster fest

BESCHREIBUNG

Legt ein Filtermuster fest, das im AmigaDOS-Musterformat angegeben ist. Einträge, die diesem Muster nicht entsprechen, werden abgelehnt.

TYP

Zeichenkette

ANWENDBARKEIT

IS

14.3 Dirlist.Directory

BEZEICHNUNG

`Dirlist.Directory` – setzt/ermittelt das aktive Verzeichnis

BESCHREIBUNG

Legt ein neues Verzeichnis für das Verzeichnislisten-Objekt fest. Da das Lesen eines Verzeichnisses sehr lange dauern kann, delegiert MUI diese Arbeit an einen Sub-Task.

Wenn dieses Attribut gesetzt wird, löscht das Objekt das aktuelle Verzeichnis (falls vorhanden) und beginnt mit dem Laden eines neuen. `Dirlist.Status` wird auf `Reading` gesetzt und der Sub-Task wird gestartet.

Wenn Sie `Dirlist.Status` überwachen, können Sie erfahren, ob das Lesen des Verzeichnisses abgeschlossen oder ob etwas schief gelaufen ist.

Wenn Sie hier eine leere Zeichenkette ("") übergeben, wird nur das aktuelle Verzeichnis gelöscht und `Dirlist.Status` auf `Invalid` gesetzt.

Voreingestellt ist `SYS:.`

TYP

Zeichenkette

ANWENDBARKEIT

ISG

14.4 `Dirlist.DrawersOnly`

BEZEICHNUNG

`Dirlist.DrawersOnly` – gibt an, ob nur Schubladen angezeigt werden sollen

BESCHREIBUNG

Hiermit geben Sie an, ob nur Schubladen angezeigt werden sollen.

TYP

Boolesch

ANWENDBARKEIT

IS

14.5 `Dirlist.FilesOnly`

BEZEICHNUNG

`Dirlist.FilesOnly` – gibt an, ob nur Dateien angezeigt werden sollen

BESCHREIBUNG

Mit diesem Attribut geben sie an, ob nur Dateien angezeigt werden sollen.

TYP

Boolesch

ANWENDBARKEIT

IS

14.6 `Dirlist.FilterDrawers`

BEZEICHNUNG

`Dirlist.FilterDrawers` – wendet ein Muster auf die Schubladen an

BESCHREIBUNG

Hier geben Sie an, ob Sie Schubladen wünschen, die mit `Dirlist.RejectPattern` und `Dirlist.AcceptPattern` abgestimmt werden sollen.

Voreingestellt ist `False`.

TYP

Boolesch

ANWENDBARKEIT

IS

14.7 Dirlist.MultiSelDirs

BEZEICHNUNG

Dirlist.MultiSelDirs – erlaubt die mehrfache Auswahl von Verzeichnissen

BESCHREIBUNG

Ermöglicht die Mehrfachauswahl von Verzeichnissen.

Voreingestellt ist `False`.

TYP

Boolesch

ANWENDBARKEIT

IS

14.8 Dirlist.NumBytes

BEZEICHNUNG

Dirlist.NumBytes – ermittelt die Anzahl der Bytes im Verzeichnis

BESCHREIBUNG

Wenn `Dirlist.Status Valid` ist, können Sie die Anzahl der Bytes ermitteln, die das Verzeichnis von diesem Tag belegt.

TYP

Nummer

ANWENDBARKEIT

G

14.9 Dirlist.NumDrawers

BEZEICHNUNG

Dirlist.NumDrawers – ermittelt die Anzahl der Schubladen im Verzeichnis

BESCHREIBUNG

Wenn `Dirlist.Status Valid` ist, können Sie die Anzahl der Schubladen im angezeigten Verzeichnis von diesem Tag abrufen.

TYP

Nummer

ANWENDBARKEIT

G

14.10 Dirlist.NumFiles

BEZEICHNUNG

Dirlist.NumFiles – ermittelt die Anzahl der Dateien im Verzeichnis

BESCHREIBUNG

Wenn `Dirlist.Status Valid` ist, können Sie die Anzahl der Dateien im angezeigten Verzeichnis von diesem Tag abrufen.

TYP

Nummer

ANWENDBARKEIT

G

14.11 Dirlist.Path

BEZEICHNUNG

Dirlist.Path – ermittelt den vollständigen Pfad des aktiven Eintrags

BESCHREIBUNG

Wenn `Dirlist.Status Valid` ist und Sie einen aktiven Eintrag in der Liste haben (wenn `Listview.Active` ungleich `Off` ist), erhalten Sie die vollständige Pfadangabe der ausgewählten Datei. Andernfalls erhalten Sie eine leere Zeichenkette.

TYP

Zeichenkette

ANWENDBARKEIT

G

14.12 Dirlist.RejectIcons

BEZEICHNUNG

Dirlist.RejectIcons – gibt an, ob Symbole nicht angezeigt werden sollen

BESCHREIBUNG

Damit haben Sie die Möglichkeit, dass Symbole (*.info-Dateien) nicht angezeigt werden.

TYP

Boolesch

ANWENDBARKEIT

IS

14.13 Dirlist.RejectPattern

BEZEICHNUNG

Dirlist.RejectPattern – legt das Filtermuster fest

BESCHREIBUNG

Legt ein Filtermuster fest, das im AmigaDOS-Musterformat angegeben ist. Einträge, die diesem Muster entsprechen, werden nicht angezeigt.

TYP

Zeichenkette

ANWENDBARKEIT

IS

14.14 Dirlist.ReRead

BEZEICHNUNG

Dirlist.ReRead – aktualisiert das Verzeichnislisten-Objekt

ÜBERSICHT

```
mui.DoMethod(id, "ReRead")
```

BESCHREIBUNG

Erzwingt, dass das Verzeichnislisten-Objekt das aktuelle Verzeichnis erneut einliest.

EINGABEN

id ID des Verzeichnislisten-Objekts

14.15 Dirlist.SortDirs

BEZEICHNUNG

Dirlist.SortDirs – legt fest, wo die Verzeichnisse angezeigt werden sollen

BESCHREIBUNG

Mit diesem Attribut passen Sie den Ort an, wo die Verzeichnisse in der Liste angezeigt werden sollen.

Folgende Werte sind möglich:

First Zeigt die Verzeichnisse oben an.

Last Zeigt die Verzeichnisse unten an.

Mix Verzeichnisse und Dateien werden verschachtelt angezeigt.

TYP

Zeichenkette (siehe oben für mögliche Werte)

ANWENDBARKEIT

IS

14.16 Dirlist.SortHighLow

BEZEICHNUNG

Dirlist.SortHighLow – kehrt den Sortiermodus um

BESCHREIBUNG

Hiermit geben Sie an, ob das Verzeichnis umgekehrt sortiert werden soll.

TYP

Boolesch

ANWENDBARKEIT

IS

14.17 Dirlist.SortType

BEZEICHNUNG

Dirlist.SortType – legt die Sortierkriterien fest

BESCHREIBUNG

Wenn Sie die Sortierkriterien ändern möchten, können Sie dieses Attribut verwenden und angeben welche Felder sortiert werden sollen.

Folgende Werte sind möglich:

Name Nach Name sortieren.

Date Nach Datum sortieren.

Size Nach Größe sortieren.

TYP

Zeichenkette (siehe oben für mögliche Werte)

ANWENDBARKEIT

IS

14.18 Dirlist.Status

BEZEICHNUNG

Dirlist.Status – ermittelt den Status des Verzeichnislisten-Objekts

BESCHREIBUNG

Hiermit lesen Sie den Status des Verzeichnislisten-Objekts aus. Folgende Ergebnisse sind möglich:

Invalid Objekt enthält kein gültiges Verzeichnis.

Reading Das Objekt liest gerade ein neues Verzeichnis ein.

Valid Objekt enthält ein gültiges Verzeichnis.

TYP

Zeichenkette (siehe oben für mögliche Werte)

ANWENDBARKEIT

GN

14.19 Dirlist.Title**BEZEICHNUNG**

Dirlist.Title – zeigt die Listentitel an oder blendet sie aus

BESCHREIBUNG

Hier können Sie angeben, ob Sie eine Titelleiste für die Listenansicht haben möchten oder nicht. Die Titelleiste wird in der ersten Zeile angezeigt und blättert nicht weg, wenn sich die oberste Position der Liste bewegt.

TYP

Boolesch

ANWENDBARKEIT

ISG

15 Floattext-Klasse (Fließtext)

15.1 Übersicht

Die Floattext-Klasse (Fließtext) ist eine Klasse, die eine große Textzeichenkette als Eingabe verwendet und sie in mehrere Zeilen aufteilt, die angezeigt werden sollen. Zu den Formatierungsmöglichkeiten gehören Absätze und als Blocksatz ausgerichteter Text mit Zeilenumbruch.

Im folgenden finden Sie ein Beispiel für die Verwendung des Befehls `<floattext>`:

```
<floattext>Hello World</floattext>
```

Hier ist das gleiche Beispiel in fett dargestellt:

```
<floattext>\33bHello World</floattext>
```

Die Zeichenkette, die Sie hier angeben, kann Textformatierungscode verwenden. Siehe [Abschnitt 3.9 \[Textformatierungscode\], Seite 14](#), für Details.

15.2 Floattext.Justify

BEZEICHNUNG

Floattext.Justify – legt die Textausrichtung auf Blocksatz fest

BESCHREIBUNG

Geben Sie an, ob der Text an den linken und rechten Rand angepasst werden soll. MUI wird versuchen, Leerzeichen zwischen Wörtern einzufügen, um dieses Ziel zu erreichen.

TYP

Boolesch

ANWENDBARKEIT

ISG

15.3 Floattext.TabSize

BEZEICHNUNG

Floattext.TabSize – passt die Tabulatorgröße an

BESCHREIBUNG

Passt die Tabulatorgröße für einen Text an. Die Tabulatorgröße wird in Leerzeichen gemessen. Wenn Sie Tabulatoren nicht nur am Anfang eines Absatzes verwenden möchten, sollten Sie die Schriftart mit fester Breite verwenden.

Die Tabulatorgröße ist standardmäßig auf 8 eingestellt.

TYP

Nummer

ANWENDBARKEIT

IS

15.4 Floattext.Text

BEZEICHNUNG

Floattext.Text – setzt/ermittelt den Inhalt des Fließtext-Objekts

BESCHREIBUNG

Die angegebene Zeichenkette wird als Fließtext angezeigt. Diese Zeichenkette kann Zeilenumbrüche enthalten, um das Ende von Absätzen oder Tabulatorzeichen für die Einrückung zu markieren.

MUI formatiert den Text automatisch entsprechend der Breite des Fließtext-Objekts. Wenn ein Wort nicht in die aktuelle Zeile passt, wird es umgebrochen.

Wenn Sie Tabulatorzeichen nicht nur am Anfang einer Zeile verwenden möchten, sollten Sie die konfigurierte Schriftart mit fester Breite verwenden.

Bitte beachten Sie, dass Ausrichtung und Zeilenumbruch mit proportionalen Schriftarten eine komplizierte Operation ist und viel Zeit in Anspruch nehmen kann, insbesondere bei langen Texten auf langsamen Maschinen.

Die Zeichenkette, die Sie hier angeben, kann Textformatierungscodes verwenden. Siehe [Abschnitt 3.9 \[Textformatierungscodes\]](#), Seite 14, für Details.

TYP

Zeichenkette

ANWENDBARKEIT

SG

16 Gauge-Klasse (Fortschrittsbalken)

16.1 Übersicht

Die Gauge-Klasse (Fortschrittsbalken) ist ein gut aussehendes Anzeigeelement, das für irgendeine Art von Fortschrittsanzeige nützlich ist. Es wird oft zusammen mit der Scale-Klasse (Skala) verwendet, um nette Fortschrittsbalken zu erstellen.

Hier ist eine Beispiel-XML-Definition:

```
<vgroup>
  <gauge horiz="true" infotext="...%ld %% completed..." />
  <scale />
</vgroup>
```

16.2 Gauge.Current

BEZEICHNUNG

Gauge.Current – setzt/ermittelt den aktuellen Pegel

BESCHREIBUNG

Stellt den aktuellen Pegel des Fortschrittsbalken ein. Der Wert muss zwischen 0 und Gauge.Max liegen.

TYP

Nummer

ANWENDBARKEIT

ISGN

16.3 Gauge.Divide

BEZEICHNUNG

Gauge.Divide – setzt den Divisor

BESCHREIBUNG

Wenn dieses Attribut nicht 0 ist, wird jeder mit Gauge.Current gesetzte Wert vor der weiteren Verarbeitung dividiert.

TYP

Nummer

ANWENDBARKEIT

ISG

16.4 Gauge.Horiz

BEZEICHNUNG

Gauge.Horiz – stellt die Lage des Fortschrittsbalkens ein

BESCHREIBUNG

Bestimmt, ob die Lage des Fortschrittsbalkens horizontal oder vertikal ausgerichtet wird. Voreingestellt ist `False` (vertikal).

TYP

Boolesch

ANWENDBARKEIT

I

16.5 Gauge.InfoText

BEZEICHNUNG

Gauge.InfoText – setzt/ermittelt den Infotext des Fortschrittsbalkens

BESCHREIBUNG

Der hier angegebene Text wird in einem Fortschrittsbalken-Objekt angezeigt und soll normalerweise eine Art von Prozentangabe anzeigen.

Dieser Text ist auf `"\33c\3338"` eingestellt, wodurch er standardmäßig zentriert und hervorgehoben erscheint.

`%ld` ist ein Platzhalter und wird durch den aktuellen Wert von `Gauge.Current` ersetzt.

Beachten Sie, dass wenn Sie ein einzelnes Prozentzeichen (%) anzeigen wollen, müssen Sie zwei Prozentzeichen hintereinander (%%) verwenden, da ein einzelnes Prozentzeichen als Platzhalterzeichen verwendet wird.

Hinweis: Bis zu V18 von `gauge.mui` funktionierte `InfoText` nur für horizontale Anzeigen. Wenn Sie Infotexte verwenden und ändern möchten, sollten Sie zum Zeitpunkt der Objekterstellung ein leeres `Gauge.InfoText` ("") angeben. Dadurch erhält Ihr Objekt eine feste Höhe, die der Höhe des Infotextes entspricht.

Seit Version 19 von `gauge.mui` können Sie auch `Gauge.InfoText` für vertikale Messanzeigen verwenden.

Die Zeichenkette, die Sie hier angeben, kann Textformatierungs-codes verwenden. Siehe [Abschnitt 3.9 \[Textformatierungs-codes\]](#), Seite 14, für Details.

TYP

Zeichenkette

ANWENDBARKEIT

ISG

BEISPIEL

```
mui.Set(obj, "InfoText", "Please wait... (%ld %%)")
```

Der obige Code richtet einen benutzerdefinierten Text ein, der aus der Zeile "Please wait" zusammen mit einer Prozentangabe besteht, die den aktuellen Fortschritt angibt.

16.6 Gauge.Max

BEZEICHNUNG

Gauge.Max – setzt/ermittelt den maximalen Wert des Fortschrittsbalkens

BESCHREIBUNG

Dieses Attribut stellt den Maximalwert für den Fortschrittsbalken ein. Voreingestellt ist 100.

Derzeit sind Gauge.Max und Gauge.Current auf 16 Bit beschränkt.

TYP

Nummer

ANWENDBARKEIT

ISG

17 Group-Klasse (Gruppen)

17.1 Übersicht

Die Group-Klasse (Gruppen) ist für das vollständige Layout eines MUI-Fensters verantwortlich. Eine Gruppe kann eine beliebige Anzahl von untergeordneten Objekten enthalten, möglicherweise Schaltflächen, Auswahl-Listefeld-Gadgets oder sogar andere Gruppen.

Einige Attribute der Group-Klasse definieren, wie die untergeordneten Elemente einer Gruppe angeordnet werden. Sie können z.B. Ihrer Gruppe sagen, dass sie ihre untergeordneten Elemente horizontal (in einer Reihe) oder vertikal (in einer Spalte) platzieren soll. Da jedes MUI-Objekt seine minimalen und maximalen Dimensionen kennt, hat die Group-Klasse alles, was sie für diesen Job benötigt.

Ein ausgefeilteres Layout ist möglich, indem Objekten in einer Gruppe unterschiedliche Größenverhältnisse zugewiesen werden oder indem eine Gruppe zweidimensional gemacht wird.

Unterhalb der Layoutausgabe übergibt ein Gruppen-Objekt Attribute und Methoden an alle untergeordneten Elemente. So können Sie mit jedem untergeordneten Element einer Gruppe kommunizieren und überwachen.

Die folgenden unterschiedlichen Gruppentypen werden von MUI Royale unterstützt:

`<hgroup>` Untergeordnete Gruppen/Elemente werden in einer Reihe (horizontal) angeordnet.

`<vgroup>` Untergeordnete Gruppen/Elemente werden in einer Spalte (vertikal) angeordnet.

`<colgroup>` Untergeordnete Gruppen/Elemente werden in einer Spalte angeordnet.

`<virtgroup>` Eine virtuelle Gruppe wird normalerweise in einer Scroll-Gruppe verwendet. Siehe [Abschnitt 51.1 \[Virtgroup-Klasse\]](#), [Seite 287](#), für Details.

`<scrollgroup>` Eine Gruppe mit Bildlaufleisten. Virtuelle Gruppen sind normalerweise in einer Scroll-Gruppe eingebettet. Siehe [Abschnitt 43.1 \[Scrollgroup-Klasse\]](#), [Seite 223](#), für Details.

Spaltengruppen (Colgroup) sind nützlich, wenn Sie für alle Ihre untergeordneten Elemente in einer Gruppe identische Gadget-Größen für eine angenehmere visuelle Darstellung benötigen. Stellen Sie sich beispielsweise ein Formular vor, das aus Zeichenketten-Gadgets und Text-Objekten besteht. Es wird empfohlen, hier die `<colgroup>` zu verwenden, da dies zu einem klaren und geordneten Aussehen führt. Hier ist ein Beispiel:

```
<colgroup columns="2">
  <text>Name</text>
  <string/>
  <text>Street</text>
  <string/>
```

```

    <text>City</text>
    <string/>
    <text>Zip code</text>
    <string/>
    <text>Country</text>
    <string/>
    <text>Telephone</text>
    <string/>
    <text>Email</text>
    <string/>
  </colgroup>

```

Wenn wir eine `<vgroup>` mit einer `<hgroup>` pro Zeile verwenden, wäre die Darstellung ziemlich schlecht, da die Breite des Zeichenketten-Gadgets für jede Zeile unterschiedlich wäre, was ziemlich unprofessionell aussieht.

17.2 Group.ActivePage

BEZEICHNUNG

Group.ActivePage – setzt/ermittelt die aktive Seite der Seitengruppe

BESCHREIBUNG

Legen Sie die aktive Seite einer Seitengruppe fest (oder ermitteln Sie sie). Nur diese aktive Seite wird angezeigt, alle anderen sind ausgeblendet.

Der Wert kann zwischen 0 (für das erste untergeordnete Element) und AnzahlSeiten-1 (für das letzte untergeordnete Element) liegen. Untergeordnete Elemente werden in der Reihenfolge der Erstellung angesprochen.

Die folgenden speziellen Werte sind beim Setzen der aktiven Seite möglich:

First	Erste Seite.
Last	Letzte Seite.
Prev	Vorherige Seite.
Next	Nächste Seite.
Advance	Seite vorblättern.

Hinweis: Sie dürfen **niemals** einen falschen Seitenwert angeben!

Dieses Attribut kann auch in Verbindung mit der Register-Klasse verwendet werden. Siehe [Abschnitt 40.1 \[Register-Klasse\]](#), [Seite 213](#), für Details.

TYP

Nummer oder Zeichenkette (siehe oben für mögliche Werte)

ANWENDBARKEIT

ISGN

17.3 Group.AddHead

BEZEICHNUNG

Group.AddHead – fügt ein losgelöstes Objekt als erste untergeordnete Gruppe hinzu (V1.2)

ÜBERSICHT

```
mui.DoMethod(id, "AddHead", obj)
```

BESCHREIBUNG

Diese Methode kann verwendet werden, um das in `obj` angegebene losgelöste Objekt zu dem durch `id` angegebene Gruppen-Objekt hinzuzufügen. Das losgelöste Objekt wird als erstes untergeordnetes Objekt der Gruppe hinzugefügt. Nachdem diese Methode beendet wurde, ändert das angegebene Objekt seinen Status von losgelöst in angehängt. Deshalb dürfen Sie jetzt keine Befehle mehr mit diesem Objekt verwenden, die ein losgelöstes Objekt erwarten.

Bevor Sie diese Methode aufrufen können, müssen Sie die Gruppe in einen speziellen Zustand versetzen, in dem untergeordnete Elemente angehängt und gelöst werden können. Dies kann durch Ausführen der Methoden `Group.InitChange` und `Group.ExitChange` für das jeweilige Gruppen-Objekt erfolgen.

Losgelöste MUI-Objekte können entweder durch Aufrufen der Befehle `mui.CreateObject()` oder durch explizites loslösen von ihrem übergeordneten Element mithilfe der Methode `Group.Remove` erstellt werden.

EINGABEN

<code>id</code>	ID des Gruppen-Objekts
<code>obj</code>	ID des anzuhängenden Objekts

BEISPIEL

Siehe [Abschnitt 5.2 \[mui.CreateObject\]](#), Seite 38.

17.4 Group.AddTail

BEZEICHNUNG

Group.AddTail – fügt ein losgelöstes Objekt als letzte untergeordnete Gruppe hinzu (V1.2)

ÜBERSICHT

```
mui.DoMethod(id, "AddTail", obj)
```

BESCHREIBUNG

Diese Methode kann verwendet werden, um das in `obj` angegebene losgelöste Objekt zu dem in `id` angegebene Gruppen-Objekt hinzuzufügen. Das losgelöste Objekt wird als letztes untergeordnetes Objekt der Gruppe hinzugefügt. Nachdem diese Methode beendet wurde, ändert das angegebene Objekt seinen Status von losgelöst in angehängt. Deshalb dürfen Sie jetzt keine Befehle mehr mit diesem Objekt verwenden, die ein losgelöstes Objekt erwarten.

Bevor Sie diese Methode aufrufen können, müssen Sie die Gruppe in einen speziellen Zustand versetzen, in dem untergeordnete Elemente angehängt und gelöst werden können.

Dies kann durch Ausführen der Methoden `Group.InitChange` und `Group.ExitChange` für das jeweilige Gruppen-Objekt erfolgen.

Losgelöste MUI-Objekte können entweder durch Aufrufen der Befehle `mui.CreateObject()` oder durch explizites loslösen von ihrem übergeordneten Element mithilfe der Methode `Group.Remove` erstellt werden.

EINGABEN

`id` ID des Gruppen-Objekts
`obj` ID des anzuhängenden Objekts

BEISPIEL

Siehe [Abschnitt 5.2 \[mui.CreateObject\]](#), Seite 38.

17.5 Group.ChildList

BEZEICHNUNG

`Group.ChildList` – gibt eine Liste aller untergeordneten Gruppen zurück (V1.2)

BESCHREIBUNG

Mit diesem Attribut können Sie eine Tabelle abrufen, die Informationen zu allen untergeordneten Elementen enthält, denen die Gruppe derzeit zugeordnet ist. Für jedes untergeordnete Element in der Gruppe werden die folgenden Tabellenelemente initialisiert:

ID: Enthält die ID der untergeordneten Gruppen
Class: Enthält den Namen der Klasse, zu der dieses Objekt gehört, z.B. "Cycle".
MUIClass: Enthält den internen MUI-Klassennamen für dieses Objekt, z.B. "Cycle.mui". Dies ist wahrscheinlich nicht von großem Interesse für Ihre Anwendung.

TYP

Tabelle

ANWENDBARKEIT

G

BEISPIEL

```
t = mui.Get("mygroup", "childlist")
For Local k = 0 To ListItems(t) - 1
    DebugPrint(t[k].id, t[k].class, t[k].muiclass)
Next
```

Der obige Code speichert alle Objekte, die zu "mygroup" gehören.

17.6 Group.Columns

BEZEICHNUNG

Group.Columns – definiert Gruppenspalten

BESCHREIBUNG

Hier geben Sie die Anzahl der Spalten in einer zweidimensionalen Gruppe an. Wenn Sie diesen Tag verwenden, muss die Gesamtzahl der untergeordneten Elemente durch die Anzahl der Spalten geteilt werden können.

Die untergeordneten Elemente werden in einer zweidimensionalen Anordnung positioniert, z.B. einfache Erstellung von Schaltflächenfeldern (vielleicht für einen Taschenrechner).

Die untergeordneten Elemente ihrer Gruppe werden immer Zeile für Zeile gelesen.

Wenn MUI zweidimensionale Gruppen erstellt, werden tatsächlich zwei Layoutberechnungen durchgeführt, eine für die Zeilen und eine für die Spalten. Parameter wie Größe und Dimensionen werden auf diese Weise behandelt:

- Mindestbreite aller Objekte in dieser Spalte/Zeile.
- Maximale Breite aller Objekte in dieser Spalte/Zeile.
- Die Größe einer Spalte/Zeile ist die Summe aller Objekte.

TYP

Nummer

ANWENDBARKEIT

I

17.7 Group.ExitChange

BEZEICHNUNG

Group.ExitChange – beendet den Gruppenmodifikationsstatus (V1.2)

ÜBERSICHT

```
mui.DoMethod(id, "ExitChange", force)
```

BESCHREIBUNG

Diese Methode beendet den von `Group.InitChange` festgelegten Modifizierstatus. Wenn untergeordnete Elemente hinzugefügt oder entfernt wurden, aktualisiert MUI die Gruppe, wodurch die Änderungen für den Benutzer sichtbar werden. Sie können MUI zwingen, diese Aktualisierung durchzuführen, indem Sie das Argument "force" auf `True` setzen. In diesem Fall aktualisiert MUI immer die gesamte Gruppe, unabhängig davon, ob Objekte hinzugefügt oder entfernt wurden. Das Erzwingen einer Aktualisierung ist nützlich, wenn sich in Ihrer Gruppe ein Objekt befindet, für das eine Aktualisierung erzwungen werden soll.

EINGABEN

<code>id</code>	ID des Gruppen-Objekts
<code>force</code>	Geben Sie hier <code>True</code> an, um eine vollständige Aktualisierung zu erzwingen. Andernfalls aktualisiert MUI die Gruppe nur, wenn Objekte hinzugefügt oder entfernt wurden

BEISPIEL

Siehe [Abschnitt 5.2 \[mui.CreateObject\]](#), Seite 38.

17.8 Group.HorizSpacing

BEZEICHNUNG

Group.HorizSpacing – setzt/ermittelt den horizontalen Abstand

BESCHREIBUNG

Sie geben die Anzahl der Pixel an, die zwischen horizontalen Elementen einer Gruppe eingefügt werden sollen.

Bitte verwenden Sie diesen Tag mit Bedacht. Sie überschreiben die bevorzugte Standard-einstellung des Benutzers!

TYP

Nummer

ANWENDBARKEIT

ISG

17.9 Group.InitChange

BEZEICHNUNG

Group.InitChange – bereitet Gruppen auf das Hinzufügen oder Entfernen von Elementen vor (V1.2)

ÜBERSICHT

```
mui.DoMethod(id, "InitChange")
```

BESCHREIBUNG

Bereitet eine Gruppe zum dynamischen Hinzufügen/Entfernen von Objekten vor. Seit Version 3 bietet MUI die Möglichkeit, untergeordnete Elemente automatisch aus Gruppen hinzuzufügen/zu entfernen, selbst wenn das Fenster, das diese Objekte enthält, gerade geöffnet ist. Um dies tun zu können, müssen Sie die Gruppe mit dieser Methode zuerst in einen speziellen Modifizierstatus versetzen. Dann können Sie untergeordnete Elemente nach belieben hinzufügen/entfernen. Wenn Sie fertig sind, verwenden Sie `Group.ExitChange`, damit MUI die Anzeige neu berechnet.

EINGABEN

id ID des Gruppen-Objekts

BEISPIEL

Siehe [Abschnitt 5.2 \[mui.CreateObject\]](#), Seite 38.

17.10 Group.Insert

BEZEICHNUNG

Group.Insert – fügt ein losgelöstes Objekt nach einem untergeordneten Element ein (V1.2)

ÜBERSICHT

```
mui.DoMethod(id, "Insert", obj, pred)
```

BESCHREIBUNG

Diese Methode kann verwendet werden, um das in `obj` angegebene losgelöste Objekt in das durch `id` angegebene Gruppen-Objekt einzufügen. Das losgelöste Objekt wird nach dem unter `pred` angegebenen untergeordneten Element hinzugefügt. Nachdem diese Methode beendet wurde, ändert das angegebene Objekt seinen Status von losgelöst in angehängt. Deshalb dürfen Sie jetzt keine Befehle mehr verwenden, die ein losgelöstes Objekt mit diesem Objekt erwarten.

Bevor Sie diese Methode aufrufen können, müssen Sie die Gruppe in einen speziellen Zustand versetzen, in dem untergeordnete Elemente hinzugefügt und abgezogen werden können. Dies kann durch Ausführen der Methoden `Group.InitChange` und `Group.ExitChange` für das jeweilige Gruppen-Objekt erfolgen.

Losgelöste MUI-Objekte können entweder durch Aufrufen des Befehls `mui.CreateObject()` oder durch explizites loslösen von ihrem übergeordneten Element mithilfe der Methode `Group.Remove` erstellt werden.

EINGABEN

<code>id</code>	ID des Gruppen-Objekts
<code>obj</code>	ID des einzufügenden Objekts
<code>pred</code>	Das Objekt wird nach diesem Objekt eingefügt

BEISPIEL

Siehe [Abschnitt 5.2 \[mui.CreateObject\]](#), Seite 38.

17.11 Group.PageMode

BEZEICHNUNG

Group.PageMode – setzt die Gruppe in den Seitenmodus

BESCHREIBUNG

Wenn dieses Attribut auf `True` gesetzt wird, wird die aktuelle Gruppe zu einer Seitengruppe. Seitengruppen zeigen immer nur eines ihrer untergeordneten Elemente an, welche mit dem Attribut `Group.ActivePage` angepasst werden kann.

Stellen Sie sich vor, Sie haben ein Einstellungsfenster mit mehreren verschiedenen Seiten, z.B. die MUI-Einstellungen mit Objekt, Rahmen, Bild, Schriftart, Bildschirm, Tastatur und Systemeinstellungen. Statt eines separaten Fensters für jede Gruppe können Sie alle Seiten in eine Seitengruppe einfügen und ein Auswahl-Listefeld-Gadget (Cycle) zum Seitenwechsel verwenden. Dadurch wird Ihr Programm einfacher zu bedienen, da der Benutzer nicht viele Fenster verwalten muss. Er kann jedoch nicht mit mehr als einer Seite gleichzeitig arbeiten.

Größen werden wie folgt berechnet:

Die Mindestbreite/-höhe einer Seitengruppe ist die maximale Mindestbreite/-höhe aller untergeordneten Elemente.

Die maximale Breite/Höhe einer Seitengruppe ist die minimale maximale Breite/Höhe aller ihrer untergeordneten Elemente.

Wenn die maximale Breite/Höhe eines untergeordneten Elements in einer Seitengruppe kleiner als die minimale Breite/Höhe der Seitengruppe ist (da es ein weiteres unterordnetes Element mit einer großen minimalen Breite/Höhe enthält), wird das untergeordnete Element zentriert.

Seitengruppen sind nicht in der Tiefe beschränkt, untergeordnete Elemente einer Seitengruppe können natürlich andere Seitengruppen sein.

Wenn ein Gadget nur unter bestimmten Bedingungen sichtbar sein soll, können Sie eine Seitengruppe erstellen, die dieses Gadget und ein leeres Rechteck-Objekt enthält.

Wenn Sie für die Objekte in einer Seitengruppe einen TAB-Abfolge wünschen, fügen Sie einfach alle Objekte in die Tastaturabfolgekette ein (als wären sie alle gleichzeitig sichtbar).

Eine spezielle Art von Seitengruppe wird von der Register-Klasse erstellt, die eine Reihe von Registerkarten für alle untergeordneten Registerkarten generiert. Siehe [Abschnitt 40.1 \[Register-Klasse\], Seite 213](#), für Details.

TYP

Boolesch

ANWENDBARKEIT

I

17.12 Group.Remove

BEZEICHNUNG

Group.Remove – löst das untergeordnete Element von der Gruppe (V1.2)

ÜBERSICHT

```
mui.DoMethod(id, "Remove", obj)
```

BESCHREIBUNG

Diese Methode kann verwendet werden, um das angegebene Objekt von der angegebenen Gruppe zu lösen. Nachdem diese Methode beendet wurde, ändert das angegebene Objekt seinen Status von angehängt zu losgelöst. Dies bedeutet, dass Sie es nun mit einem Befehl wie `Group.Insert` an eine andere Gruppe anhängen können oder Sie können es mit `mui.FreeObject()` löschen.

Bevor Sie diese Methode aufrufen können, müssen Sie die Gruppe in einen speziellen Zustand versetzen, in dem untergeordnete Elemente hinzugefügt und entfernt werden können. Dies kann durch Ausführen der Methoden `Group.InitChange` und `Group.ExitChange` für das jeweilige Gruppen-Objekt erfolgen.

EINGABEN

id ID des Gruppen-Objekts

obj ID des zu entfernenden Objekts

BEISPIEL

```
mui.DoMethod("mygroup", "initchange")
mui.DoMethod("mygroup", "remove", "mychild")
mui.DoMethod("mygroup", "exitchange", false)
```

Der obige Code entfernt das untergeordnete Element "mychild" aus der Gruppe "mygroup". Sie könnten dann "mychild" an eine andere Gruppe anhängen oder löschen.

17.13 Group.SameHeight

BEZEICHNUNG

Group.SameHeight – stellt dieselbe Höhe für alle untergeordneten Elemente ein

BESCHREIBUNG

Boolescher Wert, um anzugeben, dass alle untergeordneten Elemente dieser Gruppe die gleiche Höhe haben sollen.

TYP

Boolesch

ANWENDBARKEIT

I

17.14 Group.SameSize

BEZEICHNUNG

Group.SameSize – stellt dieselbe Größe für alle untergeordneten Elemente ein

BESCHREIBUNG

Dies ist eine Abkürzung für `Group.SameWidth` und `Group.SameHeight`. Es legt beide Attribute gleichzeitig fest.

Mit `Group.SameSize` müssen Sie nicht darüber nachdenken, ob Ihre Gruppe horizontal oder vertikal ist. Beide Fälle werden automatisch berücksichtigt.

Das Erzwingen, dass alle Objekte einer Gruppe die gleiche Größe haben, ist z.B. für eine Reihe von Schaltflächen nützlich. Es ist optisch ansprechender, wenn diese Schaltflächen gleich groß sind und nicht so groß wie der Text darin ist.

TYP

Boolesch

ANWENDBARKEIT

I

17.15 Group.SameWidth

BEZEICHNUNG

Group.SameWidth – stellt dieselbe Breite für alle untergeordneten Elemente ein

BESCHREIBUNG

Boolescher Wert, um anzugeben, dass alle untergeordneten Elemente dieser Gruppe die gleiche Breite haben sollen.

TYP

Boolesch

ANWENDBARKEIT

I

17.16 Group.Spacing

BEZEICHNUNG

Group.Spacing – setzt/ermittelt den horizontalen und vertikalen Abstand

BESCHREIBUNG

Dies ist eine Abkürzung für `Group.HorizSpacing` und `Group.VertSpacing`, es setzt beide Attribute gleichzeitig.

Mit `Group.Spacing` müssen Sie nicht darüber nachdenken, ob Ihre Gruppe horizontal oder vertikal ist. Beide Fälle werden automatisch berücksichtigt.

Beachten Sie, dass das Festlegen eines Abstandswertes für eine Gruppe die Standardeinstellungen des Benutzers überschreibt. Bitte benutzen Sie es nur, wenn Sie einen guten Grund haben.

TYP

Nummer

ANWENDBARKEIT

ISG

17.17 Group.Title

BEZEICHNUNG

Group.Title – legt den Gruppentitel fest

BESCHREIBUNG

Wenn Sie Gruppen für ein Register definieren, können Sie mit diesem Attribut einen Titel für diese Gruppe angeben, der im Register angezeigt werden soll. Siehe [Abschnitt 40.1 \[Register-Klasse\]](#), [Seite 213](#), für Details.

Ab MUI 3.x kann dieses Attribut nur während der Initialisierung angegeben werden (Anwendbarkeit ist nur I auf MUI 3.x).

Wenn MUI 4 und MUI Royale 1.2 verfügbar sind, können Sie auch `mui.Set()` und `mui.Get()` mit diesem Attribut (ISG-Anwendbarkeit) verwenden.

Wenn MUI 4 und MUI Royale 1.2 verfügbar sind, kann die Zeichenkette, die Sie hier angeben, Textformatierungs-codes verwenden. Siehe [Abschnitt 3.9 \[Textformatierungs-codes\]](#), [Seite 14](#), für Details.

TYP

Zeichenkette

ANWENDBARKEIT

ISG

17.18 Group.VertSpacing

BEZEICHNUNG

Group.VertSpacing – setzt/ermittelt den vertikalen Abstand

BESCHREIBUNG

Anzahl der Pixel, die zwischen vertikalen Elementen einer Gruppe eingefügt werden sollen.

Bitte verwenden Sie diesen Tag mit Bedacht, Sie überschreiben die bevorzugte Standardeinstellung des Benutzers!

TYP

Nummer

ANWENDBARKEIT

ISG

18 HSpace-Klasse (HorizontalAbstand)

18.1 Übersicht

Die HSpace-Klasse (HorizontalAbstand) ist eine Unterklasse der Rectangle-Klasse (Rechteck) und erstellt einfach einen leeren horizontalen Abstand zwischen zwei Objekten.

18.2 HSpace.Width

BEZEICHNUNG

HSpace.Width – stellt den horizontalen Abstand ein

BESCHREIBUNG

Legt den gewünschten horizontalen Abstand für dieses Objekt in Pixeln fest.

TYP

Nummer

ANWENDBARKEIT

I

19 Hollywood Klasse

19.1 Übersicht

Die Hollywood-Klasse ist eine leistungsstarke MUI-Klasse, mit der Sie ein vollständiges Hollywood-Display in Ihre MUI-Benutzeroberfläche einbetten können. Immer wenn Sie etwas auf ein Hollywood-Display zeichnen, das an eine Hollywood-Klasse angehängt ist, wird es automatisch auch auf Ihre MUI-GUI angezeigt. Sie können sogar das Hollywood-Display ausblenden und es wird immer noch funktionieren. Darüber hinaus werden alle Mausklicks und Tastenanschläge, die innerhalb der Hollywood-Klasse passieren, als normale Hollywood-Ereignisse an das entsprechende Hollywood-Display weitergeleitet. Somit können Sie mit der Hollywood-Klasse fast alle leistungsstarken Funktionen von Hollywood auch in einer MUI-GUI nutzen.

Hier sehen Sie ein Beispiel, wie Sie Hollywood Display 1 in Ihre GUI einbetten:

```
<hollywood display="1"/>
```

Siehe [Abschnitt 3.14 \[Hollywood-Brücke\]](#), [Seite 19](#), für Details.

19.2 Hollywood.Display

BEZEICHNUNG

Hollywood.Display – hängt ein Hollywood-Display an das MUI-Objekt an

BESCHREIBUNG

Mit diesem Attribut hängen Sie ein Hollywood-Display an das Objekt an. Sie müssen hier den Identifikator eines Hollywood-Displays angeben.

Dieses Attribut ist obligatorisch und muss jedes Mal angegeben werden, wenn Sie ein Hollywood-Objekt erstellen. Sie können kein leeres Hollywood-Objekt ohne angehängtes Display erstellen.

TYP

Nummer

ANWENDBARKEIT

IS

19.3 Hollywood.MaxHeight

BEZEICHNUNG

Hollywood.MaxHeight – setzt/ermittelt die maximale Höhe

BESCHREIBUNG

Wenn Sie dieses Attribut festlegen, wird Ihr Hollywood-Objekt in der Größe veränderbar. Setzen Sie dieses Attribut auf die maximal zulässige Höhe, die Sie für Ihr Objekt wünschen. Sie können hier nicht mehr als 16384 Pixel angeben.

Um anzugeben, dass Ihr Hollywood-Objekt keine maximale Größe hat, sollten Sie dieses Attribut auf die maximal zulässige Größe von 16384 Pixel setzen. Dies sollte für die meisten Zwecke ausreichend groß sein.

Wenn der Benutzer die Größe Ihres Hollywood-Objekts ändert, erhält das Hollywood-Display, die diesem Hollywood-Objekt zugeordnet ist, ein "SizeWindow"-Ereignis, das Sie mit dem Befehl `InstallEventHandler()` überwachen können.

TYP

Nummer

ANWENDBARKEIT

ISG

19.4 Hollywood.MaxWidth

BEZEICHNUNG

Hollywood.MaxWidth – setzt/ermittelt die maximale Breite

BESCHREIBUNG

Wenn Sie dieses Attribut festlegen, wird Ihr Hollywood-Objekt in der Größe veränderbar. Setzen Sie dieses Attribut auf die maximal zulässige Breite, die Sie für Ihr Objekt wünschen. Sie können hier nicht mehr als 16384 Pixel angeben.

Um anzugeben, dass Ihr Hollywood-Objekt keine maximale Größe hat, sollten Sie dieses Attribut auf die maximal zulässige Größe von 16384 Pixel setzen. Dies sollte für die meisten Zwecke ausreichend groß sein.

Wenn der Benutzer die Größe Ihres Hollywood-Objekts ändert, erhält das Hollywood-Display, die diesem Hollywood-Objekt zugeordnet ist, ein "SizeWindow"-Ereignis, das Sie mit dem Befehl `InstallEventHandler()` überwachen können.

TYP

Nummer

ANWENDBARKEIT

ISG

19.5 Hollywood.MinHeight

BEZEICHNUNG

Hollywood.MinHeight – setzt/ermittelt die minimale Höhe

BESCHREIBUNG

Wenn Sie dieses Attribut festlegen, wird Ihr Hollywood-Objekt in der Größe veränderbar. Setzen Sie dieses Attribut auf die gewünschte Mindesthöhe.

Wenn der Benutzer die Größe Ihres Hollywood-Objekts ändert, erhält das Hollywood-Display, die diesem Hollywood-Objekt zugeordnet ist, ein "SizeWindow"-Ereignis, das Sie mit dem Befehl `InstallEventHandler()` überwachen können.

TYP

Nummer

ANWENDBARKEIT

ISG

19.6 Hollywood.MinWidth

BEZEICHNUNG

Hollywood.MinWidth – setzt/ermittelt die minimale Breite

BESCHREIBUNG

Wenn Sie dieses Attribut festlegen, wird Ihr Hollywood-Objekt in der Größe veränderbar. Setzen Sie dieses Attribut auf die gewünschte Mindestbreite.

Wenn der Benutzer die Größe Ihres Hollywood-Objekts ändert, erhält das Hollywood-Display, die diesem Hollywood-Objekt zugeordnet ist, ein "SizeWindow"-Ereignis, das Sie mit dem Befehl `InstallEventHandler()` überwachen können.

TYP

Nummer

ANWENDBARKEIT

ISG

20 Image-Klasse (Bild)

20.1 Übersicht

Die Image-Klasse (Bild) wird verwendet, um eines der MUI-Standardbilder oder einige selbst erstellte Bilddaten anzuzeigen.

Hier ist eine Beispiel-XML-Definition:

```
<image source="brush:1"/>
```

Der obige XML-Code erstellt ein Bild-Objekt aus dem Hollywood-Pinsel Nummer 1. Die Image-Klasse unterstützt vollständig die Transparenz von Masken und Alpha-Kanälen in Hollywood-Pinseln, so dass Sie auch Bilder mit transparenten Bereichen einbetten können.

20.2 Image.Brush

BEZEICHNUNG

Image.Brush – ändert das Bild in den neuen Pinsel

BESCHREIBUNG

Wenn `Image.Source` auf `Brush` eingestellt ist, können Sie dieses Attribut verwenden, um den Pinsel zu ändern, der vom Bild-Objekt angezeigt werden soll. Übergeben Sie einfach den Identifikator des neuen Hollywood-Pinsels, den Sie angezeigt haben möchten.

Die Image-Klasse (Bild) unterstützt vollständig die Masken- und Alpha-Kanal-Transparenz in Hollywood-Pinseln, so dass Sie auch Bilder mit transparenten Bereichen einbetten können.

TYP

Nummer

ANWENDBARKEIT

S

20.3 Image.FreeHoriz

BEZEICHNUNG

Image.FreeHoriz – erlaubt die horizontale Skalierung

BESCHREIBUNG

Mit diesem Attribut können Sie einstellen, ob es horizontal skaliert werden kann. Voreingestellt ist `False`.

TYP

Boolesch

ANWENDBARKEIT

I

20.4 Image.FreeVert

BEZEICHNUNG

Image.FreeVert – erlaubt die vertikale Skalierung

BESCHREIBUNG

Mit diesem Attribut können Sie einstellen, ob es vertikal skaliert werden kann. Voreingestellt ist `False`.

TYP

Boolesch

ANWENDBARKEIT

I

20.5 Image.Source

BEZEICHNUNG

Image.Source – stellt das anzuzeigende Bild-Symbol ein

BESCHREIBUNG

Hiermit legen Sie das Bild-Symbol fest, welches angezeigt werden soll. Die folgenden vordefinierten MUI-Bilder sind hier verfügbar:

- ArrowDown
- ArrowLeft
- ArrowRight
- ArrowUp
- Assign
- Brush:<id>
- CheckMark
- Chip
- Cycle
- Disk
- Drawer
- HardDisk
- ListCursor
- ListSelCur
- ListSelect
- Network
- PopDrawer
- PopFile
- PopUp
- PropKnob
- RadioButton

- SliderKnob
- TapeDown
- TapePause
- TapePlay
- TapePlayback
- TapeRecord
- TapeStop
- TapeUp
- Volume

Wenn Sie den Typ `Brush` verwenden, müssen Sie `<id>` auf den Identifikator eines Hollywood-Pinsels setzen, den Sie für dieses Bild-Objekt verwenden möchten.

TYP

Zeichenkette (siehe oben für mögliche Werte)

ANWENDBARKEIT

I

20.6 Image.State**BEZEICHNUNG**

`Image.State` – stellt den Bildstatus ein

BESCHREIBUNG

Einige MUI-Bilder bieten verschiedene Zustände. Sie können einen der beiden auswählen, indem Sie dieses Attribut setzen. Verwenden Sie einfach einen der folgenden Werte, um den Status zu wechseln:

`Normal` Normalzustand.

`Selected` Ausgewählter Zustand.

`Disabled` Deaktivierter Zustand.

`Busy` Beschäftigter Zustand.

`Indeterminate`
Unbestimmter Zustand.

`InactiveNormal`
Normaler inaktiver Zustand.

`InactiveSelected`
Ausgewählter inaktiver Zustand.

`InactiveDisabled`
Deaktivierter inaktiver Zustand.

`SelectedDisabled`
Ausgewählter deaktivierter Zustand.

TYP

Zeichenkette (siehe oben für mögliche Werte)

ANWENDBARKEIT

IS

21 Imagebutton-Klasse (Bildschaltflächen)

21.1 Übersicht

Die Imagebutton-Klasse (Bildschaltflächen) ist eine Unterklasse der Image-Klasse (Bild). Es verwandelt ein Bild-Objekt in eine Bildschaltfläche, die vom Benutzer gedrückt werden kann. Die Imagebutton-Klasse erkennt alle Attribute der Image-Klasse (Bild).

Diese Klasse ist wahrscheinlich nicht von großem Nutzen, da Sie Bilder in Schaltflächenbeschriftungen mit Textformatierungs-codes einfügen können. Deshalb sollten Sie stattdessen die normale Button-Klasse (Schaltflächen) verwenden.

Siehe [Abschnitt 9.1 \[Button-Klasse\]](#), [Seite 77](#), für Details.

Siehe [Abschnitt 3.9 \[Textformatierungs-codes\]](#), [Seite 14](#), für Details.

21.2 Imagebutton.Pressed

BEZEICHNUNG

Imagebutton.Pressed – benachrichtigt, ob eine Bildschaltfläche gedrückt wird

BESCHREIBUNG

Dieses Attribut wird ausgelöst, wenn der Benutzer die Bildschaltfläche drückt.

TYP

Boolesch

ANWENDBARKEIT

N

22 Label-Klasse (Beschriftung)

22.1 Übersicht

Die Label-Klasse (Beschriftung) ist eine Oberklasse der Text-Klasse und ermöglicht das einfache Erstellen von Beschriftungen für andere MUI-Objekte, wie z.B. Auswahlkästchen (Checkmark) oder Auswahl-Listefeld-Objekte (Cycle).

Im Folgenden finden Sie ein Beispiel für die Verwendung des Befehls `<label>`:

```
<label>Hello World</label>
```

Die Zeichenkette, die Sie hier angeben, kann Textformatierungs-codes verwenden. Siehe [Abschnitt 3.9 \[Textformatierungs-codes\]](#), Seite 14, für Details.

Beachten Sie, dass Beschriftungen horizontal nicht skalierbar sind. Aus diesem Grund kann es passieren, dass Sie die Größenänderung der gesamten GUI einfach blockieren. Um dieses Problem zu umgehen, können Sie die Beschriftung einfach in einer `<hgroup>` zusammen mit einem leeren `<rectangle>`-Objekt (Rechteck) einfügen. Siehe [Abschnitt 39.1 \[Rectangle-Klasse\]](#), Seite 211, für Details.

22.2 Label.Centered

BEZEICHNUNG

Label.Centered – zentriert den Beschriftungstext

BESCHREIBUNG

Setzen Sie dies auf `True`, um den Beschriftungstext zu zentrieren.

TYP

Boolesch

ANWENDBARKEIT

I

22.3 Label.DoubleFrame

BEZEICHNUNG

Label.DoubleFrame – verwendet einen doppelten hohen Rahmen für die Beschriftung

BESCHREIBUNG

Setzen Sie dies auf `True`, damit die Beschriftung einen doppelten hohen Rahmen verwendet.

TYP

Boolesch

ANWENDBARKEIT

I

22.4 Label.FreeVert

BEZEICHNUNG

Label.FreeVert – lässt die vertikale Größenänderung zu

BESCHREIBUNG

Setzen Sie dies auf `True`, um die vertikale Größenanpassung der Beschriftung zu ermöglichen. Voreingestellt ist `False`.

TYP

Boolesch

ANWENDBARKEIT

I

22.5 Label.Key

BEZEICHNUNG

Label.Key – hebt ein Zeichen in der Beschriftung hervor

BESCHREIBUNG

Wenn das hier angegebene Zeichen in der angezeigten Zeichenkette vorhanden ist (egal ob Groß- oder Kleinschreibung), wird es unterstrichen. Das Markieren eines Zeichens in der Beschriftung ist in Verbindung mit `Area.ControlChar` nützlich.

TYP

Einzelne Zeichenkette

ANWENDBARKEIT

I

22.6 Label.LeftAligned

BEZEICHNUNG

Label.LeftAligned – stellt die linke Textausrichtung ein

BESCHREIBUNG

Setzen Sie dies auf `True`, damit die Beschriftung links ausgerichteten Text verwendet.

TYP

Boolesch

ANWENDBARKEIT

I

22.7 Label.SingleFrame

BEZEICHNUNG

Label.SingleFrame – verwendet den Standardrahmen für die Beschriftung

BESCHREIBUNG

Setzen Sie dies auf `True`, damit die Beschriftung einen Standardrahmen verwendet.

TYP

Boolesch

ANWENDBARKEIT

I

23 Listtree-Klasse (Baumlisten)

23.1 Übersicht

Die Listtree-Klasse (Baumlisten) hat Baumknoten, die als Knoten oder als Blatt definiert werden können. Nur Knoten können eine Liste enthalten, in die andere Baumknoten eingefügt werden können.

Sie können einen sehr komplexen Baum erstellen. Es gibt zwei verschiedene Baumknoten-Listen: Erstens die, die alle von Ihnen eingefügten Baumknoten enthält. Alle Änderungen werden an dieser Liste vorgenommen. Die zweite Liste ist die Anzeigeliste. Da Sie einen Knoten öffnen oder schließen können, werden nicht alle vorhandenen Baumknoten angezeigt.

Die Baumknoten können eingefügt, entfernt, sortiert, verschoben, ausgetauscht oder umbenannt werden. Zum Sortieren können Sie sie auch per Drag & Drop verschieben. Änderungen können in Bezug auf den gesamten Baum, auf eine Ebene, auf einen Unterbaum oder nur auf einen Baumknoten vorgenommen werden, wobei die Sichtbarkeit überprüft wird oder nicht.

Der Benutzer kann die Baumliste über die MUI-Tasten steuern, d.h. ein Knoten wird mit "Rechts" geöffnet und mit "Links" geschlossen. Überprüfen Sie Ihre MUI-Einstellungen für die angegebenen Tasten.

Es kann nur ein Eintrag ausgewählt werden, daher können Sie die Baumliste nicht für die Mehrfachauswahl verwenden.

Wenn Sie `Listtree.DragDropSort` nicht auf `False` setzen, wird die Baumliste für Drag & Drop aktiviert. Das heißt, Sie können einen Eintrag ziehen und ihn erneut auf derselben Baumliste ablegen. Während des Ziehens eines Eintrages wird angezeigt, wo Sie ihn ablegen können. Sie können keinen Eintrag auf sich selbst, noch können Sie einen geöffneten Knoten auf eines seiner Einträge ablegen.

Wenn Sie eine Baumliste im XML-Code erstellen, können Sie `Listtreenode`-Klasse verwenden, um ihn mit Knoten und Blätter zu füllen. Hier ist eine Beispieldefinition:

```
<listtree>
  <node name="CPU">
    <item>Model: Motorola MPC 7447/7457 Apollo V1.1</item>
    <item>CPU speed: 999 Mhz</item>
    <item>FSB speed: 133 Mhz</item>
    <item>Extensions: performancemonitor altivec</item>
  </node>
  <node name="Machine">
    <item>Machine name: Pegasos II</item>
    <item>Memory: 524288 KB</item>
    <item>Extensions: bus.pci bus.agp</item>
  </node>
  <node name="Expansion buses">
    <node name="PCI/AGP">
      <item>Vendor 0x11AB Device 0x6460</item>
    </node>
  </node>
```

```

<node name="Libraries">
  <item>0x6c7d4a58: exec.library V53.34</item>
</node>
<node name="Devices">
  <item>0x6ff8fba4: ramdrive.device V52.6</item>
</node>
<node name="Tasks">
  <node name="input.device">
    <item>Stack: 0x6ff4b000 - 0x6ff5b000</item>
    <item>Signals: SigWait 0x00000000</item>
    <item>State: Task (Waiting)</item>
  </node>
</node>
</listtree>

```

In diesem Beispiel haben wir das Attribut `Listtreenode.Name` verwendet, um jedem unserer Knoten einen Namen hinzuzufügen. Es gibt einige weitere Attribute, mit denen Sie die Darstellung Ihrer Knoten anpassen können. Siehe [Abschnitt 24.1 \[Listtreenode-Klasse\]](#), [Seite 147](#), für Details.

Die Zeichenkette, die Sie mit dem Tag `<item>` angeben und dem Attribut `Listtreenode.Name` verwenden, können Textformatierungs-codes verwenden. Siehe [Abschnitt 3.9 \[Textformatierungs-codes\]](#), [Seite 14](#), für Details.

23.2 Listtree.Active

BEZEICHNUNG

Listtree.Active – setzt/ermittelt einen aktiven Baumeintrag

BESCHREIBUNG

Durch das Setzen dieses Attributs wird der Cursor auf den angegebenen Baumknoten bewegt, sofern dieser sichtbar ist. Wenn sich der Knoten in einem geöffneten Baum befindet, scrollt die Listenansicht in den sichtbaren Bereich. Wenn Sie den speziellen Wert `Off` übergeben, wird der Cursor verschwinden.

Wenn dieses Attribut gelesen wird, wird die ID des aktiven Baumknotens zurückgegeben. Das Ergebnis ist `Off`, wenn kein aktiver Eintrag vorhanden ist.

Sie können eine Benachrichtigung auf `Listtree.Active` erstellen. Der `TriggerValue` ist der aktive Baumknoten.

TYP

Zeichenkette

ANWENDBARKEIT

SGN

23.3 Listtree.AutoLineHeight

BEZEICHNUNG

Listtree.AutoLineHeight – aktiviert die automatische Zeilenhöhenberechnung (V1.4)

BESCHREIBUNG

Aktiviert die automatische Zeilenhöhenberechnung. Nützlich z.B. wenn Sie keine vernünftige Zeilenhöhe vorhersagen können.

Dieses Attribut erfordert MUI 4 oder höher. Bei älteren MUI-Versionen können Sie mit dem Attribut `Listtree.MinLineHeight` die Zeilenhöhe ändern. Siehe [Abschnitt 23.12 \[Listtree.MinLineHeight\]](#), Seite 140, für Details.

TYP

Boolesch

ANWENDBARKEIT

I

23.4 Listtree.Close**BEZEICHNUNG**

Listtree.Close – schließt den Listenknoten

ÜBERSICHT

```
mui.DoMethod(id, "Close", listnode$, treenode$)
```

BESCHREIBUNG

Schließt einen oder mehrere Knoten einer Baumliste. Wenn der aktive Eintrag ein untergeordnetes Element des geschlossenen Knotens war, wird der geschlossene Knoten aktiv.

`listnode$` gibt den Knoten an, dessen Liste verwendet wird, um den Eintrag zu finden. Die Suche wird am Anfang der Liste gestartet. Dies kann der Zeichenketten-Indikator eines Knotens oder einer der folgenden speziellen Werte sein:

Root Die Stamm-Liste.

Active Die Liste des aktiven Knotens.

Parent Übergeordnete Liste

`treenode$` gibt den Knoten an, der geschlossen werden soll. Wenn Unterknoten des Knotens vorhanden sind, werden sie aus der Anzeigeliste entfernt. Dies kann der Zeichenketten-Indikator eines Knotens oder einer der folgenden speziellen Werte sein:

Head Der Kopf der in `listnode$` definierten Liste wird geschlossen.

Tail Schließt das Ende der Liste.

Active Schließt den aktiven Knoten.

All Alle Knoten der Liste, die in `listnode$` angegeben sind, werden geschlossen.

EINGABEN

id ID des Baumlisten-Objekts

listnode\$
ID des zu verwendenden Listenknotens oder Sonderwert (siehe oben)

treenode\$
ID des zu verwendenden Baumknotens oder Sonderwert (siehe oben)

23.5 Listtree.DoubleClick

BEZEICHNUNG

Listtree.DoubleClick – benachrichtigt, wenn doppelt auf Baumknoten geklickt wird

BESCHREIBUNG

Sie können eine Benachrichtigung für dieses Attribut einrichten, um Doppelklicks auf Baumknoten zu überwachen. Der Baumknoten, den der Benutzer doppelt angeklickt hat, wird in `TriggerValue` übergeben.

TYP

Boolesch

ANWENDBARKEIT

N

23.6 Listtree.DragDropSort

BEZEICHNUNG

Listtree.DragDropSort – aktiviert die Baumsortierung per Drag'n'Drop

BESCHREIBUNG

Wenn dieses Attribut auf `False` gesetzt wird, wird die Fähigkeit zum Sortieren der Baumliste durch Drag & Drop deaktiviert. Es ist standardmäßig aktiviert.

TYP

Boolesch

ANWENDBARKEIT

IS

23.7 Listtree.EmptyNodes

BEZEICHNUNG

Listtree.EmptyNodes – blendet die Listenanzeige für leere Knoten aus

BESCHREIBUNG

Wenn dieses Attribut auf `True` gesetzt wird, werden alle leeren Knoten als Blätter angezeigt, d.h. es wird kein Listenindikator angezeigt. Trotzdem wird der Eintrag wie ein Knoten behandelt.

TYP

Boolesch

ANWENDBARKEIT

IS

23.8 Listtree.Exchange

BEZEICHNUNG

Listtree.Exchange – tauscht zwei Baumknoten aus

ÜBERSICHT

```
mui.DoMethod(id, "Exchange", listnd1$, treend1$, listnd2$, treend2$)
```

BESCHREIBUNG

Tauscht die zwei Baumeinträge aus, die in `listnd1$` und `listnd2$` angegeben sind.

`listnd1$` gibt den Knoten an, dessen Liste verwendet wird, um den Eintrag zu finden. Die Suche wird am Anfang dieser Liste gestartet. Dies kann der Zeichenketten-Indikator eines Knotens oder einer der folgenden speziellen Werte sein:

Root Die Stamm-Liste.

Active Die Liste des aktiven Knotens.

`Treend1$` gibt den Knoten an, der ausgetauscht werden soll. Dies kann der Zeichenketten-Indikator eines Knotens oder einer der folgenden speziellen Werte sein:

Head Der Kopf der in `listnd1$` definierten Liste wird ausgetauscht.

Tail Tauscht das Ende der Liste aus.

Active Tauscht den aktiven Knoten aus.

`listnd2$` gibt den Knoten an, dessen Liste verwendet wird, um den Eintrag zu finden, mit dem der erste Eintrag ausgetauscht werden soll. Die Suche wird am Anfang dieser Liste gestartet. Dies kann der Zeichenketten-Indikator eines Knotens oder einer der folgenden speziellen Werte sein:

Root Die Stamm-Liste.

Active Die Liste des aktiven Knotens.

`treend2$` gibt den Knoten an, der ausgetauscht werden soll. Dies kann der Zeichenketten-Indikator eines Knotens oder einer der folgenden speziellen Werte sein:

Head Der Kopf der in `listnd2$` definierten Liste wird ausgetauscht.

Tail Tauscht das Ende der Liste aus.

Active Tauscht den aktiven Knoten aus.

Up Tauscht mit dem nächsten Eintrag.

Down Tauscht mit dem vorherigen Eintrag.

EINGABEN

id ID des Baumlisten-Objekts

listnd1\$ ID des zu verwendenden Listenknotens oder Sonderwert (siehe oben)

treend1\$ ID des zu verwendenden Baumknotens oder Sonderwert (siehe oben)

listnd2\$ ID des zu verwendenden Listenknotens oder Sonderwert (siehe oben)

treend2\$ ID des zu verwendenden Baumknotens oder Sonderwert (siehe oben)

23.9 Listtree.FindName

BEZEICHNUNG

Listtree.FindName – findet den Knoten nach dem Namen

ÜBERSICHT

```
found, table = mui.DoMethod(id, "FindName", listnode$, name$, flags$)
```

BESCHREIBUNG

Sucht einen Knoten, dessen Name mit dem angegebenen übereinstimmt. Die Suche erfolgt ohne Berücksichtigung der Groß-/Kleinschreibung.

`listnode$` gibt den Knoten an, dessen Liste verwendet wird, um den Namen zu finden. Dies kann der Zeichenketten-Indikator eines Knotens oder einer der folgenden speziellen Werte sein:

Root Die Stamm-Liste.

Active Die Liste des aktiven Knotens.

`flags$` kann eine Kombination der folgenden Optionen sein:

SameLevel

Nur Knoten auf derselben Ebene sind betroffen.

Visible Nur sichtbare Einträge werden berücksichtigt.

Wenn Sie mehrere der obigen Flags angeben, müssen Sie sie mit einem Semikolon trennen, z.B. "SameLevel; Visible".

Diese Methode gibt zwei Werte zurück: Der erste Rückgabewert `found` ist ein boolesches Flag, das anzeigt, ob ein Knoten gefunden wurde oder nicht. Wenn der erste Rückgabewert `True` ist, ist der zweite Rückgabewert `table` eine Tabelle mit folgenden initialisierten Feldern:

Name Name des Baumknotens. Da die Suche im Modus ohne Berücksichtigung der Groß-/Kleinschreibung ausgeführt wird, können Sie mit diesem Element die tatsächliche Schreibweise des Eintrags ermitteln.

Node `True`, wenn der gefundene Eintrag ein Knoten ist, `False`, wenn es ein Blatt ist.

ID Zeichenketten-Objekt-ID dieses Baumknotens.

MuiID Interne MUI-ID dieses Baumknotens. Dies kann mit `Listtree.GetEntry` verwendet werden.

EINGABEN

`id` ID des Baumlisten-Objekts

`listnode$`

ID des zu verwendenden Listenknotens oder Sonderwert (siehe oben)

`name$` Name, nachdem gesucht wird

`flags$` Nachschlageflag (siehe oben)

RÜCKGABEWERTE

`found` Boolesches Flag, das anzeigt, ob ein Eintrag gefunden wurde oder nicht

table Tabelle mit Informationen zum gefundenen Eintrag

23.10 Listtree.GetEntry

BEZEICHNUNG

Listtree.GetEntry – gibt den Baumeintrag zurück

ÜBERSICHT

```
found, table = mui.DoMethod(id, "GetEntry", node, position, flags$)
```

BESCHREIBUNG

Ruft einen anderen Knoten in Bezug auf die angegebene Liste oder den angegebenen Knoten ab. Mit dieser Methode können Sie die gesamte Baumliste durchlaufen. Siehe unten für ein Beispiel.

node gibt den Knoten an, mit dem ein anderer Knoten gefunden wird. Dies kann auch ein Listenknoten sein, wenn sich die Position **a** auf eine Liste bezieht. Im Gegensatz zu allen anderen Methoden oder Attributen der Listtree-Klasse (Baumlisten) darf **node** NICHT ein Zeichenketten-Indikator sein, sondern ein spezieller Wert, der von dieser Methode oder von Listtree.FindName im Feld MuiID der Rückgabetable zurückgegeben wird (siehe unten). Alternativ kann es einer der folgenden speziellen Werte sein:

Root Die Stamm-Liste wird verwendet.

Active Die Liste mit dem aktiven Eintrag wird verwendet.

position gibt die Anzahl der Knoten der unter **node** angegebenen Liste oder einen der folgenden Sonderwerte an:

Head Der Kopf der Liste in **node** wird zurückgegeben.

Tail Das Ende der Liste wird zurückgegeben.

Active Der aktive Knoten wird zurückgegeben.

Next Der nächste Knoten nach dem Baumknoten **node** wird zurückgegeben.

Previous Der Knoten vor dem Baumknoten **node** wird zurückgegeben.

Parent Der Listenknoten des **node** ist der übergeordnete Knoten.

flags\$ können eine Kombination der folgenden Optionen sein:

SameLevel

Nur Knoten auf derselben Ebene sind betroffen.

Visible Die Position wird nur für sichtbare Einträge gezählt.

Wenn Sie mehrere der obigen Flags angeben, müssen Sie sie mit einem Semikolon trennen, z.B. "SameLevel; Visible".

Diese Methode gibt zwei Werte zurück: Der erste Rückgabewert **found** ist ein boolesches Flag, das anzeigt, ob ein Knoten gefunden wurde oder nicht. Wenn der erste Rückgabewert **True** ist, ist der zweite Rückgabewert **table** eine Tabelle mit folgenden initialisierten Feldern:

Name Name des Baumknotens.

Node	True , wenn der gefundene Eintrag ein Knoten ist, False , wenn es ein Blatt ist.
ID	Zeichenketten-Objekt-ID dieses Baumknotens.
MuiID	Interne MUI-ID dieses Baumknotens. Dies ist die einzige ID, die Sie im Argument node dieser Methode übergeben dürfen. Das Übergeben von Standard-Zeichenketten-Objekt-IDs ist bei dieser Methode nicht zulässig. Sie können diesen Wert für nachfolgende Aufrufe von Listtree.GetEntry im Argument node verwenden. Siehe oben für weitere Informationen und unten für ein Beispiel.

EINGABEN

id	ID des Baumlisten-Objekts
node	Spezieller Knotenindikator, der von dieser Methode zurückgegeben wird
position	Index des Eintrags
flags\$	Kombination der zu verwendenden Flags (siehe oben)

RÜCKGABEWERTE

found	Boolesches Flag, das anzeigt, ob ein Eintrag gefunden wurde oder nicht
table	Tabelle mit Informationen zum gefundenen Eintrag

BEISPIEL

```
Function p_DumpListTree(id$, node, indent)
  Local found, t = mui.DoMethod(id$, "GetEntry", node, "Head", "")
  While found = True
    If indent > 0
      DebugPrint(RepeatStr(" ", indent) ..
        IIf(t.Node = True, "+", "") .. t.name)
    Else
      DebugPrint(IIf(t.Node = True, "+", "") .. t.name)
    EndIf
    If t.Node = True Then p_DumpListTree(id$, t.muiid, indent + 4)
    found, t = mui.DoMethod(id$, "GetEntry", t.muiid, "Next", "")
  Wend
EndFunction
```

```
p_DumpListTree("mylisttree", "root", 0)
```

Der obige Code zeigt, wie der komplette Inhalt einer Baumliste unter Beibehaltung seiner Struktur ausgegeben wird.

23.11 Listtree.Insert**BEZEICHNUNG**

Listtree.Insert – fügt einen neuen Baumknoten ein

ÜBERSICHT

```
mui.DoMethod(id, "Insert", entry$, id$, listnode$, prevnode$, flags$)
```

BESCHREIBUNG

Fügt den den in `entry$` angegebener Knoten an der Position ein, die mit `listnode$` und `prevnode$` definiert ist. `entry$` enthält den Namen des Eintrags als Zeichenkette. `id$` muss eine eindeutige Zeichenketten-ID sein, die Sie für den neu eingefügten Baumknoten verwenden möchten.

In `listnode$` geben Sie den Knoten an, in dessen Liste der Eintrag eingefügt wird. Dies kann der Zeichenketten-Indikator eines Knotens oder einer der folgenden speziellen Werte sein:

`Root` Die Stamm-Liste.

`Active` Die Liste des aktiven Knotens.

In `prevnode$` müssen Sie den Knoten angeben, der der Vorgänger des einzufügenden Knotens ist. Dies kann der Zeichenketten-Indikator eines Knotens oder einer der folgenden speziellen Werte sein:

`Head` Er wird am Anfang der Liste eingefügt.

`Tail` Er wird am Ende der Liste eingefügt.

`Active` Er wird nach dem aktiven Knoten eingefügt.

`Sorted` Der Knoten wird sortiert eingefügt.

`flags$` können eine Kombination der folgenden Optionen sein:

`List` Der Knoten enthält eine Liste, in die andere Knoten eingefügt werden können.

`Open` Der Listenknoten ist geöffnet, Unterknoten werden angezeigt.

`Frozen` Der Knoten reagiert nicht auf Doppelklick oder auf Öffnen/Schließen durch den Benutzer.

`NoSign` Der Indikator der Listenknoten wird nicht angezeigt.

`Active` Der eingefügte Eintrag wird aktiv gesetzt, d.h. der Cursor wird darauf bewegt.

`NextNode` `prevnode$` gibt den Nachfolger an, nicht den Vorgänger, d.h. der Knoten wird vor `prevnode$`, nicht nach `prevnode$` eingefügt.

Wenn Sie mehrere der obigen Flags angeben, müssen Sie sie mit einem Semikolon trennen, z.B. "Active; NextNode".

EINGABEN

`id` ID des Baumlisten-Objekts

`entry$` Name des einzufügenden Eintrags

`id$` Eindeutige Zeichenketten-ID für den neuen Baumknoten

`listnode$`
ID des zu verwendenden Listenknotens oder Sonderwert (siehe oben)

<code>prevnode\$</code>	ID des zu verwendenden Baumknotens oder Sonderwert (siehe oben)
<code>flags\$</code>	Einfügeflag (siehe oben)

23.12 Listtree.MinLineHeight

BEZEICHNUNG

Listtree.MinLineHeight – stellt die minimale Zeilenhöhe ein (V1.4)

BESCHREIBUNG

Legt die minimale Zeilenhöhe für die Baumliste in Pixel fest. Nützlich z.B. wenn Sie benutzerdefinierte Bilder haben.

Alternativ können Sie auch die Berechnung der automatischen Zeilenhöhe für die Baumliste mit dem Attribut `Listtree.AutoLineHeight` aktivieren. Dies ist bequemer, erfordert jedoch mindestens MUI 4.0. Siehe [Abschnitt 23.3 \[Listtree.AutoLineHeight\]](#), Seite 132, für Details.

TYP

Nummer

ANWENDBARKEIT

I

23.13 Listtree.Move

BEZEICHNUNG

Listtree.Move – verschiebt einen Eintrag an eine neue Position

ÜBERSICHT

```
mui.DoMethod(id, "Move", olistnode$, otreenode$, nlistnode$, ntreenode$)
```

BESCHREIBUNG

Verschieben Sie einen Eintrag an die Position nach einem definierten Knoten.

`olistnode$` gibt den Knoten an, mit dem der Eintrag gefunden wird. Die Suche wird am Anfang der Liste gestartet. Dies kann der Zeichenketten-Indikator eines Knotens oder einer der folgenden speziellen Werte sein:

`Root` Die Stamm-Liste.

`Active` Die Liste des aktiven Knotens.

`otreenode$` gibt den Knoten an, der verschoben werden soll. Dies kann der Zeichenketten-Indikator eines Knotens oder einer der folgenden speziellen Werte sein:

`Head` Der Anfang der in `olistnode$` definierten Liste wird verschoben.

`Tail` Das Ende der Liste wird verschoben.

`Active` Der aktive Knoten wird verschoben.

`nlistnode$` gibt den Knoten an, dessen Liste verwendet wird, um den Eintrag zu finden. Die Suche wird am Anfang der Liste gestartet. Dies kann der Zeichenketten-Indikator eines Knotens oder einer der folgenden speziellen Werte sein:

Root Die Stamm-Liste.

Active Die Liste des aktiven Knotens.

`ntreenode$` gibt den Knoten an, der der Vorgänger des Eintrags ist, der eingefügt wird. Dies kann der Zeichenketten-Indikator eines Knotens oder einer der folgenden speziellen Werte sein:

Head Der Knoten wird an den Anfang der in `nlistnode$` definierten Liste verschoben.

Tail Der Knoten wird an das Ende der Liste verschoben.

Active Der Knoten wird hinter den aktiven Knoten verschoben.

Sorted Der Knoten wird sortiert in der Liste verschoben.

EINGABEN

`id` ID des Baumlisten-Objekts

`olistnode$`
ID des zu verwendenden Listenknotens oder Sonderwert (siehe oben)

`otreenode$`
ID des zu verwendenden Baumknotens oder Sonderwert (siehe oben)

`nlistnode$`
ID des zu verwendenden Listenknotens oder Sonderwert (siehe oben)

`ntreenode$`
ID des zu verwendenden Baumknotens oder Sonderwert (siehe oben)

23.14 Listtree.Open

BEZEICHNUNG

Listtree.Open – öffnet einen Listenknoten

ÜBERSICHT

```
mui.DoMethod(id, "Open", listnode$, treenode$)
```

BESCHREIBUNG

Öffnet einen Knoten in der Baumliste. Um ein untergeordnetes Element zu öffnen, das nicht angezeigt wird, verwenden Sie `Parent` für `listnode$`, um auch alle übergeordneten Elemente zu öffnen. Nur Knoten können geöffnet werden.

`listnode$` gibt den Knoten an, dessen Liste verwendet wird, um den Eintrag zu finden. Die Suche wird am Anfang der Liste gestartet. Dies kann der Zeichenketten-Indikator eines Knotens oder einer der folgenden speziellen Werte sein:

Root Die Stamm-Liste.

Active	Die Liste des aktiven Knotens.
Parent	Flag, um alle übergeordneten Elemente des Knotens ebenfalls zu öffnen.
treenode\$	gibt den Knoten an, der geöffnet werden soll. Dies kann der Zeichenketten-Indikator eines Knotens oder einer der folgenden speziellen Werte sein:
Head	Öffnet den Knoten am Anfang der Liste.
Tail	Öffnet den Knoten am Ende der Liste.
Active	Der aktive Knoten wird geöffnet.
All	Alle Knoten der Liste werden geöffnet.

EINGABEN

id	ID des Baumlisten-Objekts
listnode\$	ID des zu verwendenden Listenknotens oder Sonderwert (siehe oben)
treenode\$	ID des zu verwendenden Baumknotens oder Sonderwert (siehe oben)

23.15 Listtree.Quiet**BEZEICHNUNG**

Listtree.Quiet – deaktiviert die Baumlistenaktualisierung (V1.4)

BESCHREIBUNG

Wenn Sie viele Einträge zu einer aktuell sichtbaren Listenstruktur hinzufügen/entfernen, führt dies zu vielen Bildschirmaktionen und verlangsamt den Vorgang. Wenn Listtree.Quiet auf True gesetzt wird, wird vorübergehend verhindert, dass die Baumliste aktualisiert wird. Diese Aktualisierung wird nur einmal durchgeführt, wenn Sie Listtree.Quiet erneut auf False setzen.

TYP

Boolesch

ANWENDBARKEIT

S

23.16 Listtree.Remove**BEZEICHNUNG**

Listtree.Remove – entfernt den Baumknoten

ÜBERSICHT

```
mui.DoMethod(id, "Remove", listnode$, treenode$)
```

BESCHREIBUNG

Entfernt einen oder mehrere Knoten aus einer Baumliste. Wenn der aktive Eintrag entfernt wird, wird der folgende Eintrag aktiv.

`listnode$` gibt den Knoten an, dessen Liste verwendet wird, um den Eintrag zu finden. Die Suche wird am Anfang dieser Liste gestartet. Dies kann der Zeichenketten-Indikator eines Knotens oder einer der folgenden speziellen Werte sein:

`Root` Die Stamm-Liste.

`Active` Die Liste des aktiven Knotens.

`treenode$` gibt den Knoten an, der entfernt werden soll. Wenn untergeordnete Knoten vorhanden sind, werden sie ebenfalls entfernt. Dies kann der Zeichenketten-Indikator eines Knotens oder einer der folgenden speziellen Werte sein:

`Head` Der Anfang der in `listnode$` definierten Liste wird entfernt.

`Tail` Entfernt das Ende der Liste.

`Active` Entfernt den aktiven Knoten.

`All` Alle Knoten der Liste, die in `listnode$` angegeben sind, werden entfernt. Andere Knoten der Übergeordnetenlisten sind nicht betroffen.

EINGABEN

`id` ID des Baumlisten-Objekts

`listnode$`
ID des zu verwendenden Listenknotens oder Sonderwert (siehe oben)

`treenode$`
ID des zu verwendenden Baumknotens oder Sonderwert (siehe oben)

23.17 Listtree.Rename

BEZEICHNUNG

Listtree.Rename – benennt den Baumknoten um

ÜBERSICHT

```
mui.DoMethod(id, "Rename", treenode$, newname$)
```

BESCHREIBUNG

Den Namen des angegebenen Knotens umbenennen.

`treenode$` gibt den Knoten an, der umbenannt werden soll. Dies kann der Zeichenketten-Indikator eines Knotens oder einer der folgenden speziellen Werte sein:

`Active` Der aktive Baumknoten wird verwendet.

EINGABEN

`id` ID des Baumlisten-Objekts

`treenode$`
ID des zu verwendenden Baumknotens oder Sonderwert (siehe oben)

`newname$` Neuer Name für den Eintrag

23.18 Listtree.Sort

BEZEICHNUNG

Listtree.Sort – sortiert einen Listenknoten

ÜBERSICHT

```
mui.DoMethod(id, "Sort", listnode$)
```

BESCHREIBUNG

Sortiert einen Listenknoten mit dem in `Listtree.SortHook` angegebenen Modus.

`listnode$` gibt den Knoten an, dessen Liste sortiert werden soll. Dies kann der Zeichenketten-Indikator eines Knotens oder einer der folgenden speziellen Werte sein:

`Root` Die Stamm-Liste.
`Active` Die Liste des aktiven Knotens.

EINGABEN

`id` ID des Baumlisten-Objekts
`listnode$` ID des zu verwendenden Listenknotens oder Sonderwert (siehe oben)

23.19 Listtree.SortHook

BEZEICHNUNG

Listtree.SortHook – stellt den Sortiermodus ein

BESCHREIBUNG

Setzen Sie dieses Attribut auf den gewünschten Sortiermodus. Wenn Sie `Listtree.Insert` mit `Sorted` verwenden oder einen Eintrag auf einem geschlossenen Knoten löschen, wird dieser Sortiermodus verwendet.

Folgende Werte sind hier möglich:

`Head` Alle Einträge werden am Anfang der Liste eingefügt.
`Tail` Alle Einträge werden am Ende der Liste eingefügt.
`LeavesTop` Blätter werden oben in der Liste eingefügt, Knoten unten. Sie sind alphabetisch sortiert.
`LeavesMixed` Die Einträge sind nur alphabetisch sortiert.
`LeavesBottom` Blätter werden am Ende der Liste eingefügt, Knoten oben. Sie sind alphabetisch sortiert. Dies ist der Standardwert.

TYP

Zeichenkette (siehe oben für mögliche Werte)

ANWENDBARKEIT

IS

23.20 Listtree.Title

BEZEICHNUNG

Listtree.Title – setzt den Baumlistentitel

BESCHREIBUNG

Hiermit geben Sie einen Titel für den aktuellen Listenbaum an.

Die Zeichenkette, die Sie hier angeben, kann Textformatierungs-codes verwenden. Siehe [Abschnitt 3.9 \[Textformatierungs-codes\]](#), [Seite 14](#), für Details.

TYP

Zeichenkette

ANWENDBARKEIT

I

24 Listtreenode-Klasse (Baumlistenknoten)

24.1 Übersicht

Die Listtreenode-Klasse (Baumlistenknoten) wird beim Erstellen von Baumlisten benötigt. Sie können die einzelnen Knoten der Baumliste definieren und unterschiedliche Attribute für sie festlegen.

Die Listtreenode-Klasse muss immer in eine `<listtree>`-Definition eingebettet sein. Siehe [Abschnitt 23.1 \[Listtree-Klasse\], Seite 131](#), für Details.

24.2 Listtreenode.Frozen

BEZEICHNUNG

Listtreenode.Frozen – erstellt einen eingefrorenen Knoten

BESCHREIBUNG

Wenn Sie dies auf `True` setzen, reagiert der Knoten nicht auf Doppelklick oder Öffnen/Schließen durch den Benutzer.

TYP

Boolesch

ANWENDBARKEIT

I

24.3 Listtreenode.Name

BEZEICHNUNG

Listtreenode.Name – legt den Knotennamen fest

BESCHREIBUNG

Gibt eine Zeichenkette an, die als Knotenname verwendet werden soll. Dieses Attribut ist obligatorisch und muss immer festgelegt werden.

Die Zeichenkette, die Sie hier angeben, kann Textformatierungscodes verwenden. Siehe [Abschnitt 3.9 \[Textformatierungscodes\], Seite 14](#), für Details.

TYP

Zeichenkette

ANWENDBARKEIT

I

24.4 Listtreenode.NoSign

BEZEICHNUNG

Listtreenode.NoSign – blendet die Listenanzeige aus

BESCHREIBUNG

Wenn Sie dies auf `True` setzen, wird der Indikator der Listenknoten nicht angezeigt.

TYP

Boolesch

ANWENDBARKEIT

I

24.5 Listtreenode.Open

BEZEICHNUNG

Listtreenode.Open – legt den Knotenstatus fest

BESCHREIBUNG

Setzen Sie dies auf `True`, um den Knoten standardmäßig zu öffnen. Alle Unterknoten werden dann angezeigt.

TYP

Boolesch

ANWENDBARKEIT

I

25 Listview-Klasse (Listenansicht)

25.1 Übersicht

Diese Klasse erstellt eine Listenansicht (Listview). MUIs Listview-Klasse ist sehr mächtig. Sie verarbeitet verschiedene Arten von Einträgen und es werden mehrere Spalten unterstützt. Das Format für eine Spalte ist einstellbar.

Listenansichten unterstützen jede Art von Sortierung, Mehrfachauswahl und einen aktiven Eintrag, der mit der Maus oder den Cursortasten gesteuert werden kann.

Wenn Sie eine Listenansicht im XML-Code erstellen, müssen Sie immer mindestens eine Spalte hinzufügen. Dies geschieht mithilfe der Listviewcolumn-Klasse. Hier ist ein Beispiel für eine minimale Listenansicht-Definition mit nur einer einzigen Spalte:

```
<listview>
  <column/>
</listview>
```

Es ist auch möglich, einige Einträge zur Definitionszeit zur Listenansicht hinzuzufügen. Dies kann mithilfe des Tags `<item>` erfolgen:

```
<listview>
  <column>
    <item>Entry 1</item>
    <item>Entry 2</item>
    <item>Entry 3</item>
  </column>
</listview>
```

Die Zeichenketten, die Sie mit dem Tag `<item>` angeben, können Textformatierungs-codes verwenden. Siehe [Abschnitt 3.9 \[Textformatierungs-codes\]](#), Seite 14, für Details.

Wenn Sie eine mehrspaltige Liste haben möchten, müssen Sie den Tag `<column>` mehrmals verwenden. Hier ist ein Beispiel:

```
<listview>
  <column title="Column 1">
    <item>Entry 1</item>
    <item>Entry 2</item>
    <item>Entry 3</item>
  </column>
  <column title="Column 2">
    <item>Entry 1</item>
    <item>Entry 2</item>
    <item>Entry 3</item>
  </column>
  <column title="Column 3">
    <item>Entry 1</item>
    <item>Entry 2</item>
    <item>Entry 3</item>
  </column>
```

```
</listview>
```

In diesem Beispiel haben wir auch das Attribut `Listviewcolumn.Title` verwendet, um jeder Spalte eine Titelleiste hinzuzufügen. Es gibt einige weitere Attribute, mit denen Sie die Darstellung Ihrer Spalten anpassen können. Siehe [Abschnitt 26.1 \[Listviewcolumn-Klasse\]](#), [Seite 165](#), für Details.

25.2 Listview.Active

BEZEICHNUNG

`Listview.Active` – setzt/ermittelt den aktiven Listeneintrag

BESCHREIBUNG

Das Lesen dieses Attributs gibt die Nummer des aktiven Eintrags zurück (der mit dem Cursor darauf). Das Ergebnis liegt zwischen 0 und `Listview.Entries-1` oder `Off`. In diesem Fall gibt es derzeit keinen aktiven Eintrag.

Wenn Sie das Attribut setzen, verschiebt die Liste den Cursor an die neue Position und scrollt diese Position in den sichtbaren Bereich.

Folgende spezielle Werte sind beim Setzen dieses Attributs möglich:

<code>Off</code>	Löscht die Auswahl
<code>Top</code>	Wählt den obersten Eintrag aus.
<code>Bottom</code>	Wählt den unteren Eintrag aus.
<code>Up</code>	Wählt den vorherigen Eintrag aus.
<code>Down</code>	Wählt den nächsten Eintrag aus.
<code>PageUp</code>	Verschiebt den Cursor um eine Seite nach oben.
<code>PageDown</code>	Verschiebt den Cursor um eine Seite nach unten.

TYP

Nummer oder Zeichenkette (siehe oben für mögliche Werte)

ANWENDBARKEIT

ISGN

25.3 Listview.AdjustHeight

BEZEICHNUNG

`Listview.AdjustHeight` – korrigiert die Höhe der Listenansicht

BESCHREIBUNG

Eine Liste mit `Listview.AdjustHeight`, die auf `True` gesetzt ist, ist genau so hoch wie alle ihre Einträge und ist nicht skalierbar. Dies ist nur möglich, wenn die Liste *vorher* das Fenster geöffnet hat.

TYP

Boolesch

ANWENDBARKEIT

I

25.4 Listview.AdjustWidth

BEZEICHNUNG

Listview.AdjustWidth – korrigiert die Breite der Listenansicht

BESCHREIBUNG

Eine Liste mit Listview.AdjustWidth, die auf True gesetzt ist, ist genau so breit wie der breiteste Eintrag und ist nicht skalierbar. Dies ist nur möglich, wenn die Liste *vorher* das Fenster geöffnet hat.

TYP

Boolesch

ANWENDBARKEIT

I

25.5 Listview.AutoLineHeight

BEZEICHNUNG

Listview.AutoLineHeight – aktiviert die automatische Zeilenhöhenberechnung (V1.4)

BESCHREIBUNG

Aktiviert die automatische Zeilenhöhenberechnung. Nützlich z.B. wenn Sie keine vernünftige Zeilenhöhe vorhersagen können.

Dieses Attribut erfordert MUI 4 oder höher. Bei älteren MUI-Versionen können Sie mit dem Attribut Listview.MinLineHeight die Zeilenhöhe ändern. Siehe [Abschnitt 25.20 \[Listview.MinLineHeight\]](#), Seite 157, für Details.

TYP

Boolesch

ANWENDBARKEIT

I

25.6 Listview.AutoVisible

BEZEICHNUNG

Listview.AutoVisible – springt automatisch zum aktiven Eintrag

BESCHREIBUNG

Setzen Sie dies, damit Ihre Liste automatisch zum aktiven Eintrag springt, wenn sie angezeigt wird.

TYP

Boolesch

ANWENDBARKEIT

ISG

25.7 Listview.Clear

BEZEICHNUNG

Listview.Clear – löscht die Listenansicht

ÜBERSICHT

```
mui.DoMethod(id, "Clear")
```

BESCHREIBUNG

Löscht die Liste, womit alle Einträge entfernt werden.

EINGABEN

id ID des Listenansichts-Objekts

25.8 Listview.ClickColumn

BEZEICHNUNG

Listview.ClickColumn – benachrichtigt, wenn auf eine Spalte geklickt wird

BESCHREIBUNG

Bei Verwendung einer mehrspaltigen Liste enthält dieses Attribut die Nummer der Spalte, auf die der Benutzer geklickt hat.

TYP

Nummer

ANWENDBARKEIT

GN

25.9 Listview.DefClickColumn

BEZEICHNUNG

Listview.DefClickColumn – setzt/ermittelt die Standardspalte

BESCHREIBUNG

Wenn die Listenansicht über die Tastatur gesteuert wird und der Benutzer RETURN drückt, wird der hier angegebene Wert als Standard für Listview.ClickColumn verwendet.

TYP

Nummer

ANWENDBARKEIT

ISG

25.10 Listview.DoubleClick

BEZEICHNUNG

Listview.DoubleClick – benachrichtigt, wenn auf die Listenansichten doppelgeklickt wird

BESCHREIBUNG

Dieses Attribut wird auf True gesetzt, wenn der Benutzer auf einen Eintrag in der Liste doppelklickt.

TYP

Boolesch

ANWENDBARKEIT

GN

25.11 Listview.Entries

BEZEICHNUNG

Listview.Entries – gibt die Anzahl der Listeneinträge zurück

BESCHREIBUNG

Ermittelt die aktuelle Anzahl der Einträge in der Liste.

TYP

Nummer

ANWENDBARKEIT

G

25.12 Listview.Exchange

BEZEICHNUNG

Listview.Exchange – tauscht zwei Einträge aus

ÜBERSICHT

```
mui.DoMethod(id, "Exchange", pos1, pos2)
```

BESCHREIBUNG

Tauscht zwei Einträge in einer Liste aus. Positionen müssen als absolute Werte von 0 bis Listview.Entries-1 oder einen der folgenden speziellen Werte übergeben werden:

- Top Verwendet den obersten Eintrag.
- Active Verwendet den aktiven Eintrag.
- Bottom Verwendet den untersten Eintrag.
- Next Verwendet den nächsten Eintrag. Dies gilt nur für den zweiten Parameter.
- Previous Verwendet den vorherigen Eintrag. Dies gilt nur für den zweiten Parameter.

EINGABEN

id ID des Listenansichts-Objekts

<code>pos1</code>	Nummer des ersten Eintrags
<code>pos2</code>	Nummer des zweiten Eintrags

25.13 `Listview.First`

BEZEICHNUNG

`Listview.First` – ermittelt die Nummer des ersten sichtbaren Eintrags

BESCHREIBUNG

Ermittelt die Nummer des Eintrags, der oben in der Liste angezeigt wird. Sie müssen darauf vorbereitet sein, ein Ergebnis von -1 zu erhalten, was bedeutet, dass die Liste überhaupt nicht sichtbar ist (z.B. wenn das Fenster ikonifiziert ist).

TYP

Nummer

ANWENDBARKEIT

G

25.14 `Listview.GetEntry`

BEZEICHNUNG

`Listview.GetEntry` – gibt einen Listeneintrag zurück

ÜBERSICHT

```
column1$, ... = mui.DoMethod(id, "GetEntry", pos)
```

BESCHREIBUNG

Ermittelt einen Eintrag der Liste. Übergeben Sie `Active` in `pos`, um den aktiven Eintrag zu erhalten. `Listview.GetEntry` gibt die Einträge aller Spalten in der durch `pos` angegebenen Zeile zurück. Sie erhalten so viele Rückgabewerte, wie Spalten in der Listenansicht vorhanden sind.

EINGABEN

<code>id</code>	ID des Listenansichts-Objekts
<code>pos</code>	Index der Listenansichtszeile oder "Aktiv"

RÜCKGABEWERTE

<code>column1\$</code>	Eingabedaten der ersten Spalte
<code>...</code>	Weitere Daten, wenn die Listenansicht mehrere Spalten enthält

25.15 `Listview.GetSelection`

BEZEICHNUNG

`Listview.GetSelection` – gibt die ausgewählten Einträge zurück

ÜBERSICHT

```
t = mui.DoMethod(id, "GetSelection")
```

BESCHREIBUNG

Diese Methode durchläuft den Inhalt einer (Mehrfachauswahl) Listenansicht und gibt jeden aktuell ausgewählten Eintrag zurück. Wenn kein Eintrag ausgewählt ist, aber ein Eintrag aktiv ist, wird nur der aktive Eintrag zurückgegeben.

Dieses Verhalten führt dazu, dass der aktive Eintrag nicht zurückgegeben wird, wenn Sie einige andere ausgewählte Einträge irgendwo in Ihrer Liste haben. Da der aktive Eintrag nur als eine Art Cursor-Markierung fungiert, scheint dies die einzige vernünftige Möglichkeit zu sein, die Mehrfachauswahl zusammen mit der Tastatursteuerung zu handhaben.

EINGABEN

id ID des Listenansichts-Objekts

RÜCKGABEWERTE

t Tabelle mit ausgewählten Einträgen

25.16 Listview.Input**BEZEICHNUNG**

Listview.Input – konfiguriert den Eingabemodus der Listenansicht

BESCHREIBUNG

Wenn Sie dies auf `False` setzen, wird eine schreibgeschützte Listenansicht angezeigt. Der Standardwert ist `True`.

TYP

Boolesch

ANWENDBARKEIT

I

25.17 Listview.Insert**BEZEICHNUNG**

Listview.Insert – fügt einen neuen Eintrag ein

ÜBERSICHT

```
mui.DoMethod(id, "Insert", pos, column1$, ...)
```

BESCHREIBUNG

Fügt einen neuen Eintrag in eine Liste ein. Wenn die Listenansicht mehrere Spalten enthält, müssen Sie so viele Argumente übergeben, wie Spalten in der Listenansicht vorhanden sind.

Die Einfügeposition wird im Argument `pos` angegeben. Der neue Eintrag wird vor dem mit `pos` angegebenen Eintrag hinzugefügt. Dies kann eine absolute Indexposition sein, die bei 0 für den ersten Eintrag oder einen der folgenden speziellen Werte beginnt:

Top Als ersten Eintrag einfügen.

Active	Vor dem aktiven Eintrag einfügen.
Sorted	Sortiert einfügen
Bottom	Als letzten Eintrag einfügen.

Wenn `pos` größer oder gleich der Anzahl der Einträge in der Liste ist, wird es wie `Bottom` behandelt.

Alle Einträge, die Sie mit dieser Methode einfügen, können Textformatierungs-codes verwenden. Siehe [Abschnitt 3.9 \[Textformatierungs-codes\]](#), Seite 14, für Details.

EINGABEN

<code>id</code>	ID des Listenansichts-Objekts
<code>pos</code>	Position als absolute Zahl oder Sonderwert einfügen (siehe oben)
<code>column1\$</code>	Eintrag zum Einfügen in die erste Spalte
<code>...</code>	Weitere Einträge, wenn die Listenansicht mehrere Spalten enthält

25.18 Listview.InsertPosition

BEZEICHNUNG

`Listview.InsertPosition` – fragt die Position des neu eingefügten Eintrags ab

BESCHREIBUNG

Nach dem Einfügen eines Elements mit `Listview.Insert` können Sie die Position des neuen Eintrags abfragen, indem Sie dieses Attribut abrufen.

TYP

Nummer

ANWENDBARKEIT

G

25.19 Listview.Jump

BEZEICHNUNG

`Listview.Jump` – blättert zu einem Eintrag

ÜBERSICHT

```
mui.DoMethod(id, "Jump", pos)
```

BESCHREIBUNG

Scrollt einen Eintrag in den sichtbaren Teil einer Liste.

Hinweis: Wenn Sie zu einem Eintrag springen, bedeutet dies nicht, dass dieser Eintrag zum aktiven Eintrag wird. Dies kann durch Festlegen des Attributs `Listview.Active` erfolgen.

Ab MUI Royale 1.4 kann `pos` auch einer der folgenden Sonderwerte sein:

Active Springt zum aktiven Eintrag.

Top	Springt ganz nach oben.
Bottom	Springt ganz nach unten.
Up	Springt eine Seite hoch.
Down	Springt eine Seite runter.

EINGABEN

id	ID des Listenansichts-Objekts
pos	Nummer des Eintrags, der sichtbar gemacht werden soll. Übergeben Sie "Aktiv", um zum aktiven Eintrag zu springen.

25.20 Listview.MinLineHeight**BEZEICHNUNG**

Listview.MinLineHeight – stellt die minimale Zeilenhöhe ein

BESCHREIBUNG

Legt die minimale Zeilenhöhe für Listen in Pixeln fest. Nützlich z.B. wenn Sie benutzerdefinierte Bilder haben.

Alternativ können Sie auch die Berechnung der automatischen Zeilenhöhe für die Listenansicht mit dem Attribut `Listview.AutoLineHeight` aktivieren. Dies ist bequemer, erfordert jedoch mindestens MUI 4.0. Siehe [Abschnitt 25.5 \[Listview.AutoLineHeight\]](#), [Seite 151](#), für Details.

TYP

Nummer

ANWENDBARKEIT

I

25.21 Listview.Move**BEZEICHNUNG**

Listview.Move – verschiebt den Eintrag an eine neue Position

ÜBERSICHT

```
mui.DoMethod(id, "Move", from, to)
```

BESCHREIBUNG

Verschiebt einen Eintrag von einer Position zu einer anderen. Positionen müssen als absolute Werte von 0 bis `Listview.Entries-1` oder einen der folgenden speziellen Werte übergeben werden:

Top	Verwendet den obersten Eintrag.
Active	Verwendet den Aktiven Eintrag.
Bottom	Verwendet den untersten Eintrag.

- Next** Verwendet den nächsten Eintrag. Dies gilt nur für den zweiten Parameter.
- Previous** Verwendet den vorherigen Eintrag. Dies gilt nur für den zweiten Parameter.

EINGABEN

- id** ID des Listenansichts-Objekts
- from** Nummer des ersten Eintrags
- to** Nummer des zweiten Eintrags

25.22 Listview.MultiSelect**BEZEICHNUNG**

Listview.MultiSelect – stellt den Mehrfachauswahlmodus für die Listenansicht ein

BESCHREIBUNG

Es gibt vier Möglichkeiten für eine Listenansicht mit mehreren Auswahlmöglichkeiten:

- None** Die Listenansicht kann nicht mehrfach ausgewählt werden.
- Default** Der Mehrfach-Auswahl-Typ (mit oder ohne Umschalttaste) hängt von den Benutzereinstellungen ab.
- Shifted** Überschreibt die Benutzereinstellungen, Mehrfachauswahl nur zusammen mit Umschalttaste.
- Always** Überschreibt die Benutzereinstellungen, Mehrfachauswahl ohne Umschalttaste.

Überschreiben Sie die Benutzereinstellungen nicht, es sei denn, Sie haben einen guten Grund!

TYP

Zeichenkette (siehe oben für mögliche Werte)

ANWENDBARKEIT

I

25.23 Listview.Quiet**BEZEICHNUNG**

Listview.Quiet – deaktiviert die Aktualisierung der Listenansicht (V1.2)

BESCHREIBUNG

Wenn Sie viele Einträge zu/von einer aktuell sichtbaren Liste hinzufügen/entfernen, führt dies zu einer großen Anzahl von Bildschirmaktionen und verlangsamt den Vorgang. Wenn Sie Listview.Quiet auf True setzen, wird vorübergehend verhindert, dass die Liste sich ständig aktualisiert. Diese Aktualisierung wird nur einmal durchgeführt, wenn Sie Listview.Quiet wieder auf False setzen.

TYP

Boolesch

ANWENDBARKEIT

S

25.24 Listview.Remove**BEZEICHNUNG**

Listview.Remove – entfernt einen Eintrag aus der Liste

ÜBERSICHT`mui.DoMethod(id, "Remove", pos)`**BESCHREIBUNG**

Entfernt einen Eintrag aus einer Liste. Die Position kann als absoluter Indexwert oder als einer der folgenden speziellen Werte angegeben werden:

First Entfernt den ersten Eintrag.

Active Entfernt den aktiven Eintrag.

Last Entfernt den letzten Eintrag.

Selected Entfernt den ausgewählten Eintrag.

Wenn der aktive Eintrag entfernt wird, wird der folgende Eintrag aktiv.

EINGABEN

id ID des Listenansichts-Objekts

pos Index des zu entfernenden Eintrags oder eines der Sonderwerte (siehe oben)

25.25 Listview.Rename**BEZEICHNUNG**

Listview.Rename – benennt einen Eintrag um (V1.1)

ÜBERSICHT`mui.DoMethod(id, "Rename", pos, column1$, ...)`**BESCHREIBUNG**

Benennt den Listenansichtseintrag an der angegebenen Position um. Wenn die Listenansicht mehrere Spalten enthält, müssen Sie so viele Argumente übergeben, wie Spalten in der Listenansicht vorhanden sind. Es ist nicht möglich, nur einen einzelnen Spalteneintrag umzubenennen. Diese Methode wirkt sich immer auf die gesamte Zeile aus, so dass Sie so viele Zeichenketten übergeben müssen, wie Spalten in Ihrer Listenansicht vorhanden sind.

Die Einstiegsposition wird im Argument `pos` angegeben. Dies kann eine absolute Indexposition sein, die bei 0 für den ersten Eintrag oder einen der folgenden speziellen Werte beginnt:

Active Benennt den aktiven Eintrag um.

Alle Zeichenketten, die Sie an diese Methode übergeben, können Textformatierungscodes verwenden. Siehe [Abschnitt 3.9 \[Textformatierungscodes\]](#), [Seite 14](#), für Details.

EINGABEN

<code>id</code>	ID des Listenansichts-Objekts
<code>pos</code>	Einstiegsposition als absolute Zahl oder Sonderwert (siehe oben)
<code>column1\$</code>	Neuer Text für den Eintrag in der ersten Spalte
<code>...</code>	Weitere Einträge, wenn die Listenansicht mehrere Spalten enthält

25.26 Listview.Select

BEZEICHNUNG

Listview.Select – wählt den Listeneintrag ab

ÜBERSICHT

```
mui.DoMethod(id, "Select", pos, seltype)
```

BESCHREIBUNG

Aktiviert/deaktiviert einen Listeneintrag oder ermittelt, ob dieser Eintrag ausgewählt ist.

`pos` kann entweder die Nummer des Eintrags oder eines der folgenden Sonderwerte sein:

`Active` Für den aktiven Eintrag.

`All` Für alle Einträge.

`seltype` kann einer der folgenden sein:

`Off` Wählt den Eintrag ab.

`On` Wählt den Eintrag.

`Toggle` Schaltet den Eintrag um.

EINGABEN

<code>id</code>	ID des Listenansichts-Objekts
<code>pos</code>	Eintrittsindex oder Sonderwert (siehe oben)
<code>seltype</code>	Auswahltyp (siehe oben)

25.27 Listview.ScrollerPos

BEZEICHNUNG

Listview.ScrollerPos – stellt die Bildlaufleistenposition für die Listenansicht ein

BESCHREIBUNG

Gibt die Position einer Listenansicht-Bildlaufleiste an. Verwenden Sie diesen Tag nicht, es sei denn, es ist absolut erforderlich!

Wenn Sie hier `None` angeben, erhält Ihre Listenansicht überhaupt keinen Scroller und sieht fast wie ein Listen-Objekt aus. Listenansichten ohne Scroller sind jedoch immer noch leistungsfähiger als Listen-Objekte, da sie z.B. Drag & Drop Möglichkeiten haben. Das Erstellen von Listenansichten ohne Bildlaufleiste ist sinnvoll, wenn Sie die Bildlaufleiste woanders haben möchten, z.B. außerhalb einer horizontalen virtuellen Gruppe, in der sich die Listenansicht befindet. Diese Technik ermöglicht die Erstellung horizontal scrollbarer Listenansichten.

Folgende Optionen sind möglich:

- Default** Positioniert die Bildlaufleiste gemäß den Benutzereinstellungen.
- Left** Positioniert die Bildlaufleiste links von der Listenansicht.
- Right** Positioniert die Bildlaufleiste rechts neben der Listenansicht.
- None** Keine Bildlaufleiste.

TYP

Zeichenkette (siehe oben für mögliche Werte)

ANWENDBARKEIT

I

25.28 Listview.SelectChange**BEZEICHNUNG**

Listview.SelectChange – benachrichtigt, wenn Auswahländerungen erfolgen

BESCHREIBUNG

Dieses Attribut wird auf `True` gesetzt, wenn sich der Auswahlstatus eines oder mehrerer Elemente in der Liste ändert. Sie können dies benutzen z.B. wenn Sie die Anzahl ausgewählter Artikel in einer Statuszeile anzeigen möchten.

TYP

Boolesch

ANWENDBARKEIT

GN

25.29 Listview.Sort**BEZEICHNUNG**

Listview.Sort – sortiert die Liste

ÜBERSICHT

```
mui.DoMethod(id, "Sort", column)
```

BESCHREIBUNG

Sortiert die Liste. MUI verwendet einen iterativen Quicksort-Algorithmus, somit treten keine Stapelprobleme auf. Sie müssen den Index der Spalte übergeben, nach der die

Liste im dritten Argument `column` sortiert werden soll. Spalten werden von 0 bis zur Anzahl der Spalten minus 1 gezählt, d.h. um die Liste ab der ersten Spalte zu sortieren, übergeben Sie 0 als Spaltenindex.

EINGABEN

`id` ID des Listenansichts-Objekts
`column` Index der Spalte, die sortiert wird (V1.1)

25.30 Listview.SortFunc

BEZEICHNUNG

Listview.SortFunc – bestimmt, wie Einträge sortiert werden sollen (V1.2)

BESCHREIBUNG

Dieses Attribut kann verwendet werden, um eine Callback-Funktion einzurichten, die immer dann aufgerufen wird, wenn MUI Royale die Listeneinträge sortieren muss. Die Callback-Funktion erhält zwei Einträge als Argumente und muss bestimmen, welcher Eintrag zuerst eingefügt werden soll.

Die Callback-Funktion, die Sie hier angeben, wird wie ein Standard-MUI-Royale-Ereignis-Callback aufgerufen, jedoch mit den folgenden zusätzlichen Argumenten:

Entry1: Dies ist eine Tabelle, die alle Elemente einer einzelnen Listenansichtszeile enthält. Da Listenansichten mehr als eine Spalte enthalten können, enthält diese Tabelle so viele Zeichenkettenelemente wie Spalten in Ihrer Listenansicht vorhanden sind.

Entry2: Dies ist eine Tabelle, die alle Elemente einer einzelnen Listenansichtszeile enthält. Da Listenansichten mehr als eine Spalte enthalten können, enthält diese Tabelle so viele Zeichenkettenelemente wie Spalten in Ihrer Listenansicht vorhanden sind.

SortColumn:

Dieser enthält den Spaltenindex, nach dem die Liste sortiert werden soll. Spalten werden von 0 bis zur Anzahl der Spalten minus 1 gezählt. Siehe [Abschnitt 25.29 \[Listview.Sort\]](#), [Seite 161](#), für Details.

Ihre Callback-Funktion muss dann einen Wert zurückgeben, der angibt, wie die zwei Einträge in der Listenansicht ausgerichtet werden sollen. Wenn Eintrag 1 vor 2 platziert werden soll, muss die Callback-Funktion -1 zurückgeben. Wenn Eintrag 1 nach Eintrag 2 platziert werden soll, muss Ihre Callback-Funktion 1 zurückgeben. Wenn die beiden Einträge identisch sind, müssen Sie 0 zurück geben.

Beachten Sie, dass Sie die Funktion niemals direkt, sondern immer als eine Zeichenkette übergeben müssen, die den Namen einer Hollywood-Funktion für dieses Attribut angibt.

Beachten Sie auch, dass Sie für dieses Attribut eine Benachrichtigung einrichten müssen. Andernfalls wird die Callback-Funktion niemals aufgerufen. Siehe [Abschnitt 3.6 \[Benachrichtigungen\]](#), [Seite 12](#), für Details.

TYP

Zeichenkette

ANWENDBARKEIT

ISGN

25.31 Listview.TitleClick**BEZEICHNUNG**

Listview.TitleClick – benachrichtigt, wenn in die Titelleiste geklickt wird (V1.1)

BESCHREIBUNG

Dieses Attribut wird auf die Indexnummer gesetzt, auf dessen Spalte der Benutzer in der Spaltentitelleiste klickt.

Dieses Attribut wird nur auf AmigaOS 4 und MorphOS unterstützt.

TYP

Nummer

ANWENDBARKEIT

N

25.32 Listview.Visible**BEZEICHNUNG**

Listview.Visible – ermittelt die Anzahl der sichtbaren Einträge

BESCHREIBUNG

Ermitteln Sie die aktuelle Anzahl der sichtbaren Einträge in der Liste. Sie müssen darauf vorbereitet sein, ein Ergebnis von -1 zu erhalten, was bedeutet, dass die Liste überhaupt nicht sichtbar ist (z.B. wenn das Fenster ikonifiziert ist).

TYP

Nummer

ANWENDBARKEIT

G

26 Listviewcolumn-Klasse (Listenansichtspalten)

26.1 Übersicht

Die Listviewcolumn-Klasse (Listenansichtspalten) wird beim Erstellen von Listenansichten benötigt. Sie können verschiedene Attribute für die Spalten Ihrer Listenansichten festlegen.

Die Listviewcolumn-Klasse muss immer in eine `<listview>`-Definition eingebettet sein. Siehe [Abschnitt 25.1 \[Listview-Klasse\]](#), Seite 149, für Details.

26.2 Listviewcolumn.Bar

BEZEICHNUNG

Listviewcolumn.Bar – verwendet eine Trennleiste zwischen den Spalten

BESCHREIBUNG

Seit `mui-master.library V11` können Sie mit diesem Attribut einen vertikalen Balken zwischen dieser und der nächsten Spalte aktivieren.

TYP

Boolesch

ANWENDBARKEIT

I

26.3 Listviewcolumn.Col

BEZEICHNUNG

Listviewcolumn.Col – legt die Spaltenreihenfolge fest

BESCHREIBUNG

Dieser Wert passt die Nummer der aktuellen Spalte an. Auf diese Weise können Sie die Reihenfolge Ihrer Spalten anpassen.

Standardeinstellung ist die aktuelle Eintragsnummer (0,1,2, ...).

Dieses Attribut ist besonders nützlich in Verbindung mit der `Dirlist`-Klasse (Verzeichnislisten) und der `Volumelist`-Klasse (Datenträgerlisten), um die Anzeigereihenfolge der verschiedenen Attribute zu ändern. Siehe [Abschnitt 14.1 \[Dirlist-Klasse\]](#), Seite 89, für Details.

Beginnend mit MUI Royale 1.1 hat dieses Attribut eine Anwendbarkeit von ISG. In MUI Royale 1.0 war die Anwendbarkeit dieses Attributs nur I.

TYP

Nummer

ANWENDBARKEIT

ISG

26.4 Listviewcolumn.Delta

BEZEICHNUNG

Listviewcolumn.Delta – legt die Spaltenbreite fest

BESCHREIBUNG

Legt den Abstand in Pixel zwischen dieser und der nächsten Spalte fest. Die letzte angezeigte Spalte ignoriert diese Einstellung. Der Standardwert ist 4.

Beginnend mit MUI Royale 1.1 hat dieses Attribut eine Anwendbarkeit von ISG. In MUI Royale 1.0 war die Anwendbarkeit dieses Attributs nur I.

TYP

Nummer

ANWENDBARKEIT

ISG

26.5 Listviewcolumn.Hidden

BEZEICHNUNG

Listviewcolumn.Hidden – blendet Listenansichtsspalten ein/aus (V1.1)

BESCHREIBUNG

Mit diesem Attribut können Sie einzelne Listenansichtsspalten ein- oder ausblenden. Beachten Sie, dass die Spalten immer noch da sind, sie werden nur unsichtbar. Daher dürfen Sie versteckte Spalten nicht vergessen, wenn Sie Einträge in der Listenansicht mit `Listview.Insert` oder ähnlichen Methoden ändern.

Dieses Attribut wird nur auf AmigaOS 4 und MorphOS unterstützt.

TYP

Boolesch

ANWENDBARKEIT

ISG

26.6 Listviewcolumn.MinWidth

BEZEICHNUNG

Listviewcolumn.MinWidth – setzt/ermittelt die minimale Spaltenbreite

BESCHREIBUNG

Stellt die minimale prozentuale Breite für die aktuelle Spalte ein. Wenn Ihre Liste 200 Pixel breit ist und Sie dieses Attribut auf 25 setzen, beträgt Ihre Spalte mindestens 50 Pixel. Der spezielle Wert -1 für diesen Parameter bedeutet, dass die Mindestbreite so breit wie der breiteste Eintrag in dieser Spalte ist. Dies stellt sicher, dass jeder Eintrag vollständig sichtbar ist (solange die Liste breit genug ist). Der Standardwert ist -1.

Beginnend mit MUI Royale 1.1 hat dieses Attribut eine Anwendbarkeit von ISG. In MUI Royale 1.0 war die Anwendbarkeit dieses Attributs nur I.

TYP

Nummer

ANWENDBARKEIT

ISG

26.7 ListViewcolumn.MaxWidth

BEZEICHNUNG

ListViewcolumn.MaxWidth – setzt/ermittelt die maximale Spaltenbreite

BESCHREIBUNG

Stellt die maximale prozentuale Breite für die aktuelle Spalte ein. Wenn Ihre Liste 200 Pixel breit ist und Sie diese auf 25 setzen, wird Ihre Spalte nicht breiter als 50 Pixel. Der spezielle Wert -1 für diesen Parameter bedeutet, dass die maximale Breite so breit wie der breiteste Eintrag in dieser Spalte ist. Der Standardwert ist -1.

Beginnend mit MUI Royale 1.1 hat dieses Attribut eine Anwendbarkeit von ISG. In MUI Royale 1.0 war die Anwendbarkeit dieses Attributs nur I.

TYP

Nummer

ANWENDBARKEIT

ISG

26.8 ListViewcolumn.PreParse

BEZEICHNUNG

ListViewcolumn.PreParse – legt die vorbereitende Textzeichenkette für die Spalte fest

BESCHREIBUNG

Ein Vorbereitungswert für diese Spalte. Die Einstellung von z.B. "\33c" würde die Spalte zentrieren.

Die Zeichenkette, die Sie hier angeben, kann Textformatierungscodes verwenden. Siehe [Abschnitt 3.9 \[Textformatierungscodes\]](#), [Seite 14](#), für Details.

Beginnend mit MUI Royale 1.1 hat dieses Attribut eine Anwendbarkeit von ISG. In MUI Royale 1.0 war die Anwendbarkeit dieses Attributs nur I.

TYP

Zeichenkette

ANWENDBARKEIT

ISG

26.9 Listviewcolumn.Title

BEZEICHNUNG

Listviewcolumn.Title – legt den Spaltentitel fest

BESCHREIBUNG

Geben Sie einen Titel für die aktuelle Spalte an. Der Titel wird in der ersten Zeile angezeigt und blättert nicht weg, wenn sich die oberste Position der Liste bewegt.

Die Zeichenkette, die Sie hier angeben, kann Textformatierungscode verwenden. Siehe [Abschnitt 3.9 \[Textformatierungscode\]](#), Seite 14, für Details.

Beginnend mit MUI Royale 1.1 hat dieses Attribut eine Anwendbarkeit von ISG. In MUI Royale 1.0 war die Anwendbarkeit dieses Attributs nur I.

TYP

Zeichenkette

ANWENDBARKEIT

ISG

26.10 Listviewcolumn.Weight

BEZEICHNUNG

Listviewcolumn.Weight – legt das Größenverhältnis der Spalte fest

BESCHREIBUNG

Hiermit legen Sie das Größenverhältnis der Spalte fest. Wie bei der MUI-Gruppenklasse werden die Spalten mit einer Mindestgröße, einer maximalen Größe und einem Größenverhältnis ausgelegt. Eine Spalte mit einem Größenverhältnis von 200 würde doppelt so viel Platz bekommen wie eine Spalte mit einem Größenverhältnis von 100. Der Standardwert ist 100.

Beginnend mit MUI Royale 1.1 hat dieses Attribut eine Anwendbarkeit von ISG. In MUI Royale 1.0 war die Anwendbarkeit dieses Attributs nur I.

TYP

Nummer

ANWENDBARKEIT

ISG

27 Menu-Klasse (Menü)

27.1 Übersicht

Objekte der Menu-Klasse (Menü) beschreiben genau ein Abroll-Menü. Sie bietet selbst nicht viele Optionen, aber fungiert als übergeordnete Klasse für ihre verschiedenen Menüelement-Objekte, die Sie mit der MenuItem-Klasse zum Menü hinzufügen können. Siehe [Abschnitt 28.1 \[MenuItem-Klasse\]](#), [Seite 173](#), für Details.

Menüs müssen immer in ein `<menustrip>`-Objekt eingebettet werden. Siehe [Abschnitt 29.1 \[Menustrip Klasse\]](#), [Seite 177](#), für Details.

27.2 Menu.AddHead

BEZEICHNUNG

Menu.AddHead – fügt ein losgelöstes Objekt als erste Menügruppe hinzu (V1.2)

ÜBERSICHT

```
mui.DoMethod(id, "AddHead", obj)
```

BESCHREIBUNG

Diese Methode kann verwendet werden, um das durch `obj` angegebene losgelöste Objekt zu dem durch `id` angegebene Abroll-Menü hinzuzufügen. Das losgelöste Objekt wird als erste Menügruppe dem Abroll-Menü hinzugefügt. Nachdem diese Methode beendet wurde, ändert das angegebene Objekt seinen Status von losgelöst in angehängt. Deshalb dürfen Sie keine Befehle mehr verwenden, die jetzt ein losgelöstes Objekt erwartet.

Losgelöste MUI-Objekte können entweder durch Aufrufen des Befehls `mui.CreateObject()` oder durch explizites Trennen von ihren übergeordneten Elementen mithilfe der Methode `Menu.Remove` erstellt werden.

EINGABEN

<code>id</code>	ID des Abroll-Menüs
<code>obj</code>	ID des anzuhängenden Objekts

27.3 Menu.AddTail

BEZEICHNUNG

Menu.AddTail – fügt ein losgelöstes Objekt als letzte Menügruppe hinzu (V1.2)

ÜBERSICHT

```
mui.DoMethod(id, "AddTail", obj)
```

BESCHREIBUNG

Diese Methode kann verwendet werden, um das durch `obj` angegebene losgelöste Objekt zu dem in `id` angegebene Abroll-Menü hinzuzufügen. Das losgelöste Objekt wird als letzte Menügruppe dem Abroll-Menü hinzugefügt. Nachdem diese Methode beendet wurde, ändert das angegebene Objekt seinen Status von losgelöst in angehängt. Deshalb dürfen Sie keine Befehle mehr verwenden, die jetzt ein losgelöstes Objekt erwartet.

Losgelöste MUI-Objekte können entweder durch Aufrufen des Befehls `mui.CreateObject()` oder durch explizites Trennen von ihren übergeordneten Elementen mithilfe der Methode `Menu.Remove` erstellt werden.

EINGABEN

`id` ID des Abroll-Menüs
`obj` ID des anzuhängenden Objekts

27.4 Menu.Disabled

BEZEICHNUNG

`Menu.Disabled` – aktiviert/deaktiviert das komplette Menü

BESCHREIBUNG

Aktiviert oder deaktiviert das ganze Menü.

TYP

Boolesch

ANWENDBARKEIT

ISG

27.5 Menu.Insert

BEZEICHNUNG

`Menu.Insert` – fügt ein losgelöstes Objekt der Menügruppe ein (V1.2)

ÜBERSICHT

```
mui.DoMethod(id, "Insert", obj, pred)
```

BESCHREIBUNG

Diese Methode kann verwendet werden, um das in `obj` angegebene losgelöste Objekt in das durch `id` angegebene Abroll-Menü einzufügen. Das losgelöste Objekt wird nach der unter `pred` angegebene Menügruppe hinzugefügt. Nachdem diese Methode beendet wurde, ändert das angegebene Objekt seinen Status von losgelöst in angehängt. Deshalb dürfen Sie keine Befehle mehr verwenden, die jetzt ein losgelöstes Objekt erwartet.

Losgelöste MUI-Objekte können entweder durch Aufrufen des Befehls `mui.CreateObject()` oder durch explizites Trennen von ihren übergeordneten Elementen mithilfe der Methode `Menu.Remove` erstellt werden.

EINGABEN

`id` ID des Abroll-Menüs
`obj` ID des einzufügenden Objekts
`pred` Das Objekt wird nach diesem Objekt eingefügt

27.6 Menu.Remove

BEZEICHNUNG

Menu.Remove – löst die Menügruppe vom Abroll-Menü (V1.2)

ÜBERSICHT

```
mui.DoMethod(id, "Remove", obj)
```

BESCHREIBUNG

Diese Methode kann verwendet werden, um das angegebene Objekt vom Abroll-Menü zu trennen. Nachdem diese Methode beendet wurde, ändert das angegebene Objekt seinen Status von angehängt zu losgelöst. Das bedeutet, dass Sie es nun mit einem Befehl wie `Menu.Insert` an ein anderes Abroll-Menü anhängen können oder Sie können es mit `mui.FreeObject()` löschen.

EINGABEN

id	ID des Abroll-Menüs
obj	ID des zu entfernenden Objekts

27.7 Menu.Title

BEZEICHNUNG

Menu.Title – setzt/ermittelt den Menütitel

BESCHREIBUNG

Hiermit können Sie den Titel des Menüs setzen oder ermitteln.

TYP

Zeichenkette

ANWENDBARKEIT

ISG

28 MenuItem-Klasse (Menüelement)

28.1 Übersicht

Die MenuItem-Klasse (Menüelement) beschreibt einen einzelnen Menüeintrag. Sie können alle Gadgetools-Menüfunktionen verwenden, um hier Bildmenüs zu bearbeiten.

Da die MenuItem-Klasse eine Unterklasse der Menu-Klasse ist, können Sie andere Menüelemente als untergeordnete Elemente eines Menüelements hinzufügen, um Untermenüs anzuzeigen. MUI beschränkt nicht die Ebene von Untermenüs, aber das Betriebssystem erlaubt derzeit eine maximale Verschachtelungsebene von eins. Aus diesem Grund sollten untergeordnete Elemente von Menüelementen vorerst keine anderen Menüelemente enthalten. Die Ergebnisse sind unvorhersehbar.

Menüelemente müssen immer in ein `<menu>`-Tag eingebettet werden, das wiederum in ein `<menustrip>`-Tag eingebettet werden muss. Siehe [Abschnitt 29.1 \[Menustrip-Klasse\]](#), Seite 177, für Details.

Beachten Sie, dass der XML-Tag für die MenuItem-Klasse nur `<item>` und nicht `<menuItem>` ist.

Siehe [Abschnitt 29.1 \[Menustrip-Klasse\]](#), Seite 177, für ein Beispiel.

28.2 MenuItem.CommandString

BEZEICHNUNG

`MenuItem.CommandString` – setzt/ermittelt die Tastenkürzelzeichenkette

BESCHREIBUNG

Setzen Sie dieses Attribut auf `True`, wenn `MenuItem.Shortcut` eine Tastenkürzelzeichenkette (z.B. "shift alt q") anstelle eines einfachen Buchstabens verwendet. Beachten Sie, dass MUI nicht prüft, ob diese Tasten gedrückt sind (genau wie Intuition). Sie müssen dies selbst tun.

TYP

Boolesch

ANWENDBARKEIT

ISG

28.3 MenuItem.Disabled

BEZEICHNUNG

`MenuItem.Disabled` – setzt/ermittelt den deaktivierten Zustand des Menüpunktes

BESCHREIBUNG

Mit diesem Attribut aktiviere/deaktivieren Sie den Menüpunkt.

TYP

Boolesch

ANWENDBARKEIT

ISG

28.4 MenuItem.Exclude**BEZEICHNUNG**

MenuItem.Exclude – setzt/ermittelt die Ausschlussmaske für Radio-Menüelemente

BESCHREIBUNG

Hier übergeben Sie die Bitmaske der Menüelementnummern, die bei Auswahl dieser Menüoption deaktiviert werden sollen.

TYP

Nummer

ANWENDBARKEIT

ISG

28.5 MenuItem.Selected**BEZEICHNUNG**

MenuItem.Selected – setzt/ermittelt/benachrichtigt den Auswahlstatus der Menüelemente

BESCHREIBUNG

Legt den ausgewählten Status eines Menüelements fest. Wenn Sie eine Benachrichtigung für dieses Attribut einrichten, können Sie sofort auf Menüaktionen reagieren.

TYP

Boolesch

ANWENDBARKEIT

ISGN

28.6 MenuItem.Shortcut**BEZEICHNUNG**

MenuItem.Shortcut – setzt/ermittelt die Tastaturverknüpfung für Menüelement

BESCHREIBUNG

Definiert die Tastaturverknüpfung für einen Menüeintrag.

TYP

Zeichenkette

ANWENDBARKEIT

ISG

28.7 MenuItem.Title

BEZEICHNUNG

MenuItem.Title – setzt/ermittelt den Menüelementtitel

BESCHREIBUNG

Definiert den Titel des Elements.

Wenn MUI Royale 1.3 und MUI 4.0 oder höher installiert ist und dieser Menüeintrag in einem Kontextmenü angezeigt wird, können Sie hier Textformatierungscode verwenden. Siehe [Abschnitt 3.9 \[Textformatierungscode\]](#), [Seite 14](#), für Details.

TYP

Zeichenkette

ANWENDBARKEIT

ISG

28.8 MenuItem.Type

BEZEICHNUNG

MenuItem.Type – setzt/ermittelt den Menüelementtyp

BESCHREIBUNG

Mit diesem Attribut kann der Typ für diesen Menüeintrag festgelegt werden. Folgende Arten sind derzeit möglich:

Normal Normaler Menüpunkt.

Toggle Der Menüeintrag wird mit einem Häkchen umgeschaltet.

Radio Der Menüpunkt ist Teil einer Gruppe von Radio-Objekten. Mit `MenuItem.Exclude` können mögliche Kombinationen der Radio-Objekte definiert werden.

TYP

Zeichenkette (siehe oben für mögliche Werte)

ANWENDBARKEIT

IG

29 Menustrip-Klasse (Menüleisten)

29.1 Übersicht

Die Menustrip-Klasse (Menüleisten) ist die Basisklasse für die objektorientierten Menüs von MUI. Seine untergeordneten Elemente sind die Objekte der Menu-Klasse, jedes von ihnen beschreibt genau ein Abroll-Menü.

Ein Menüleisten-Objekt bietet selbst nicht viele Optionen, sondern fungiert als übergeordnetes Element für mehrere Menü-Objekte.

Das Menüleisten-Objekt wird normalerweise als untergeordnetes Objekt angegeben, entweder der Application-Klasse (Anwendung) oder der Window-Klasse (Fenster) mit den Attributen `Application.Menustrip` oder `Window.Menustrip`. Darüber hinaus kann es auch als Kontextmenü für andere MUI-Objekte verwendet werden, die das Attribut `Area.ContextMenu` verwenden.

In einer XML-Datei wird eine Menüstruktur mit dem Tag `<menustrip>`, `<menu>` und `<item>` definiert. Hier ist eine Beispieldefinition für eine einfache Menüleiste:

```
<menustrip id="mymenustrip">
  <menu title="Project">
    <item>New...</item>
    <item>Open...</item>
    <item/>
    <item>Save</item>
    <item>Save as...</item>
    <item/>
    <item>Quit</item>
  </menu>
  <menu title="Edit">
    <item shortcut="X">Cut</item>
    <item shortcut="C">Copy</item>
    <item shortcut="V">Paste</item>
  </menu>
  <menu title="?">
    <item>Settings...</item>
    <item>MUI settings...</item>
    <item/>
    <item>About...</item>
    <item>About MUI...</item>
  </menu>
</menustrip>
```

Beachten Sie die leeren `<item/>` Definitionen: Diese werden einen Trennbalken in den Menübaum einfügen. Durch die Verwendung von Trennbalken wird Ihr Menü für den Endbenutzer besser lesbar. Nachdem Sie die obige XML-Definition geschrieben haben, können Sie die Menüleiste mit dem Attribut `Window.Menustrip` wie folgt zu einem Ihrer Fenster hinzufügen:

```
<window menustrip="mymenustrip">
```

```
...
</window>
```

Es ist auch möglich, eine Menüleiste als Kontextmenü zu einem der Gadgets Ihres Fensters hinzuzufügen, indem Sie das Attribut `Area.ContextMenu` verwenden. Das Kontextmenü erscheint anschließend immer dann, wenn der Benutzer die rechte Maustaste über das Gadget mit einem Kontextmenü drückt. Hier ein Beispiel, wie Sie einen Menübaum als Kontextmenü zu einer Listenansicht hinzufügen:

```
<listview contextmenu="mymenustrip">
...
</listview>
```

Bitte beachten Sie, dass Menüleisten, die als Kontextmenüs verwendet werden, nicht mehr als ein `<menu>` enthalten dürfen.

Es ist sehr wichtig zu beachten, dass Sie Ihre Menüleisten im `<application>`-Bereich definieren müssen, da Menüleisten globale Objekte sind und erst später an Fenster oder Gadgets angehängt werden. Aus diesem Grund ist es nicht erlaubt, Menüleisten in einem `<window>` XML-Bereich zu definieren.

29.2 Menustrip.AddHead

BEZEICHNUNG

Menustrip.AddHead – fügt ein losgelöstes Objekt als erstes Element hinzu (V1.2)

ÜBERSICHT

```
mui.DoMethod(id, "AddHead", obj)
```

BESCHREIBUNG

Diese Methode kann verwendet werden, um das durch `obj` angegebene losgelöste Objekt zu dem in `id` angegebene übergeordneten Objekt (Application-Klasse (Anwendung) oder Window-Klasse (Fenster)) hinzuzufügen. Das losgelöste Objekt wird als erstes untergeordnetes Element hinzugefügt. Nachdem diese Methode beendet wurde, ändert das angegebene Objekt seinen Status von losgelöst in angehängt. Deshalb dürfen Sie keine Befehle mehr verwenden, die jetzt ein losgelöstes Objekt erwartet.

Losgelöste MUI-Objekte können entweder durch Aufrufen des Befehls `mui.CreateObject()` oder durch explizites Trennen von ihren übergeordneten Elementen mithilfe der Methode `Menustrip.Remove` erstellt werden.

EINGABEN

<code>id</code>	ID des übergeordneten Objekts
<code>obj</code>	ID des anzuhängenden Objekts

29.3 Menustrip.AddTail

BEZEICHNUNG

Menustrip.AddTail – fügt ein losgelöstes Objekt als letztes Element hinzu (V1.2)

ÜBERSICHT

```
mui.DoMethod(id, "AddTail", obj)
```

BESCHREIBUNG

Diese Methode kann verwendet werden, um das durch `obj` angegebene losgelöste Objekt zu dem in `id` angegebene übergeordneten Objekt (Application-Klasse (Anwendung) oder Window-Klasse (Fenster)) hinzuzufügen. Das losgelöste Objekt wird als letztes untergeordnetes Element hinzugefügt. Nachdem diese Methode beendet wurde, ändert das angegebene Objekt seinen Status von losgelöst in angehängt. Deshalb dürfen Sie keine Befehle mehr verwenden, die jetzt ein losgelöstes Objekt erwartet.

Losgelöste MUI-Objekte können entweder durch Aufrufen des Befehls `mui.CreateObject()` oder durch explizites Trennen von ihren übergeordneten Elementen mithilfe der Methode `Menustrip.Remove` erstellt werden.

EINGABEN

<code>id</code>	ID des übergeordneten Objekts
<code>obj</code>	ID des anzuhängenden Objekts

29.4 Menustrip.Insert**BEZEICHNUNG**

`Menustrip.Insert` – fügt ein losgelöstes Objekt nach dem angegebenen Element ein

ÜBERSICHT

```
mui.DoMethod(id, "Insert", obj, pred)
```

BESCHREIBUNG

Diese Methode kann verwendet werden, um das durch `obj` angegebene losgelöste Objekt zu dem in `id` angegebene übergeordneten Objekt (Application-Klasse (Anwendung) oder Window-Klasse (Fenster)) hinzuzufügen. Das losgelöste Objekt wird nach dem unter `pred` angegebenen untergeordneten Element hinzugefügt. Nachdem diese Methode beendet wurde, ändert das angegebene Objekt seinen Status von losgelöst in angehängt. Deshalb dürfen Sie keine Befehle mehr verwenden, die jetzt ein losgelöstes Objekt erwartet.

Losgelöste MUI-Objekte können entweder durch Aufrufen des Befehls `mui.CreateObject()` oder durch explizites Trennen von ihren übergeordneten Elementen mithilfe der Methode `Menustrip.Remove` erstellt werden.

EINGABEN

<code>id</code>	ID des übergeordneten Objekts
<code>obj</code>	ID des einzufügenden Objekts
<code>pred</code>	Das Objekt wird nach diesem Objekt eingefügt

29.5 Menustrip.Remove**BEZEICHNUNG**

`Menustrip.Remove` – löst das Objekt vom übergeordneten Objekt

ÜBERSICHT

```
mui.DoMethod(id, "Remove", obj)
```

BESCHREIBUNG

Diese Methode kann verwendet werden, um das durch `obj` angegebene Objekt von dem in `id` angegebene übergeordneten Objekt (Application-Klasse (Anwendung) oder Window-Klasse (Fenster)) zu trennen. Nachdem diese Methode beendet wurde, ändert das angegebene Objekt seinen Status von angehängt in losgelöst. Das bedeutet, dass Sie es nun mit einem Befehl wie `Menustrip.Insert` an ein anderes übergeordnete Objekt anhängen können. Sie können es auch mit `mui.FreeObject()` löschen.

EINGABEN

<code>id</code>	ID des übergeordneten Objekts
<code>obj</code>	ID des zu entfernenden Objekts

30 Notify-Klasse (Benachrichtigung)

30.1 Übersicht

Die Notify-Klasse (Benachrichtigung) ist die Oberklasse aller anderen MUI-Klassen. Ihr Hauptzweck besteht darin, den Benachrichtigungsmechanismus von MUI zu handhaben, aber sie enthält auch einige andere Methoden und Attribute, die für jedes Objekt nützlich sind.

Da die Notify-Klasse die Oberklasse für alle anderen MUI-Klassen ist, können Sie alle Benachrichtigungs-Attribute mit allen anderen MUI-Klassen verwenden. Sie können beispielsweise `Notify.HelpNode` verwenden, um einen Online-Hilfeverweis zu MUI-Gadgets hinzuzufügen. Oder Sie können `Notify.AppMessage` verwenden, um MUI-Objekte zum App-Meldungssystem hinzuzufügen.

30.2 Notify.AppMessage

BEZEICHNUNG

`Notify.AppMessage` – benachrichtigt bei App-Nachrichten

BESCHREIBUNG

Wenn Ihr Fenster ein `AppWindow` ist, d.h. Sie haben das Attribut `Window.AppWindow` auf `True` gesetzt, können Sie App-Nachrichten abrufen, indem Sie `Notify.AppMessage` überwachen. Immer wenn eine App-Nachricht ankommt, löst dieses Attribut eine Benachrichtigung aus.

Bei Benachrichtigungenvon App-Nachrichten erhält der standardmäßige Ereignis-Handler-Callback von MUI Royale zwei zusätzliche Einträge in der Tabelle, die als Parameter an Ihre Callback-Funktion übergeben werden:

`NumDropFiles`:

Die Anzahl der Dateien, die der Benutzer über Ihr Objekt fallen gelassen hat. Dies ist normalerweise 1.

`DropFiles`:

Eine Tabelle mit der Liste der Dateien, die über das Objekt fallengelassen wurden. Diese Tabelle enthält genau 'NumDropFiles' Einträge.

Siehe [Abschnitt 3.6 \[Benachrichtigungen\]](#), Seite 12, für Details.

`Notify.AppMessage` ist objektspezifisch. Sie können z.B. verschiedene Benachrichtigungen für verschiedene Objekte in Ihrem Fenster einrichten. Sie werden nur ausgeführt, wenn Symbole (Icons) über das spezifische Objekt fallengelassen werden.

Wenn Sie auf `Notify.AppMessage` mit einem Fenster-Objekt verwenden, wird Ihre Benachrichtigung immer ausgeführt, wenn Symbole im Fenster gelöscht werden.

Beachten Sie, dass `AppWindows` nur auf dem Workbench-Bildschirm möglich sind.

TYP

Zeichenkette

ANWENDBARKEIT

N

30.3 Notify.Class

BEZEICHNUNG

Notify.Class – ermittelt den Klassennamen des Objekts

BESCHREIBUNG

Hiermit ermitteln Sie den MUI Royale-Klassennamen eines Objekts.

TYP

Zeichenkette

ANWENDBARKEIT

G

30.4 Notify.ExportID

BEZEICHNUNG

Notify.ExportID – legt die ID für den Export von Einstellungen fest (V1.7)

BESCHREIBUNG

Objekte mit `Notify.ExportID` exportieren ihren Inhalt mit `Application.Save` und importieren ihn mit `Application.Load`.

Sie müssen unterschiedliche Export-IDs für Ihre Objekte verwenden!

TYP

Vierzeichen-Zeichenkette

ANWENDBARKEIT

ISG

30.5 Notify.HelpLine

BEZEICHNUNG

Notify.HelpLine – definiert die Zeile in der Hilfedatei

BESCHREIBUNG

Definiert eine Zeile in einer Hilfedatei, die mit `Application.HelpFile` angegeben wurde.

Siehe [Abschnitt 3.10 \[Online-Hilfe implementieren\]](#), Seite 16, für Details.

TYP

Nummer

ANWENDBARKEIT

ISG

30.6 Notify.HelpNode

BEZEICHNUNG

Notify.HelpNode – definiert die Seite in der Hilfedatei

BESCHREIBUNG

Definiert die Seite in einer Hilfedatei, die mit `Application.HelpFile` angegeben wurde.

Siehe [Abschnitt 3.10 \[Online-Hilfe implementieren\]](#), Seite 16, für Details.

TYP

Zeichenkette

ANWENDBARKEIT

ISG

30.7 Notify.ID

BEZEICHNUNG

Notify.ID – legt die Objekt-ID fest

BESCHREIBUNG

Mit diesem Attribut kann die ID für ein MUI-Objekt festgelegt werden. Sie müssen Ihren Objekten eindeutige IDs geben, damit Sie mit den Befehlen `mui.Set()`, `mui.Get()` und `mui.DoMethod()` darauf zugreifen können.

TYP

Zeichenkette

ANWENDBARKEIT

I

30.8 Notify.MUIClass

BEZEICHNUNG

Notify.MUIClass – ermittelt den MUI-Klassenname des Objekts

BESCHREIBUNG

Ruft den MUI-Klassennamen eines Objekts ab.

TYP

Zeichenkette

ANWENDBARKEIT

G

30.9 Notify.NoNotify

BEZEICHNUNG

Notify.NoNotify – schaltet die Benachrichtigungen aus

BESCHREIBUNG

Wenn Sie eine Benachrichtigung über ein Attribut einrichten, um auf Benutzereingaben zu reagieren, werden Sie auch Ereignisse erkennen, wenn Sie dieses Attribut programmgesteuert mit `mui.Set()` ändern. Wenn Sie `Notify.NoNotify` zusammen mit Ihrem Attribut einstellen, wird verhindert, dass diese Benachrichtigung ausgelöst wird.

Wenn Sie eine Benachrichtigung einrichten, um auf Benutzereingaben zu reagieren, werden auch Ereignisse erkannt, wenn Sie Attribute mit `mui.Set()` ändern. Durch das Festlegen von `Notify.NoNotify` zusammen mit Ihrem Attribut wird verhindert, dass diese Benachrichtigung mit `mui.Set()` ausgelöst wird.

`Notify.NoNotify` ist ein "einmaliges" Attribut. Es ist nur während des aktuellen `mui.Set()` Aufrufs gültig!

TYP

Boolesch

ANWENDBARKEIT

S

30.10 Notify.NotifyData

BEZEICHNUNG

Notify.NotifyData – setzt/ermittelt ereignisspezifische Benutzerdaten

BESCHREIBUNG

Mit diesem Attribut können Sie benachrichtigungsspezifische Benutzerdaten in einem Objekt definieren. Sie müssen hier eine Zeichenkette übergeben, die eine oder mehrere Benachrichtigungen und Benutzerdaten für jede Benachrichtigung in der Zeichenkette enthält. Wenn eine Benachrichtigung ausgelöst wird, die in der Zeichenkette angegeben ist, empfängt die Callback-Funktion die Benutzerdaten, die in `Notify.NotifyData` im Feld `NotifyData` der Ereignismeldung angegeben sind.

Die Zeichenkette, die Sie an dieses Attribut übergeben müssen, muss folgendermaßen formatiert sein: Name des Benachrichtigungsattributs, gefolgt von einem Doppelpunkt, gefolgt von einer Benutzerdatenfolge, gefolgt von einem Semikolon. Die Sequenz kann dann so oft wiederholt werden, wie es erforderlich ist.

Zum Beispiel: "Aktiv: foo; DoubleClick: bar;". Wenn die Benachrichtigung "Aktiv" ausgelöst wird, wird "foo" an die Callback-Funktion gesendet. Wenn das Attribut "DoubleClick" ausgelöst wird, wird "bar" gesendet.

Siehe [Abschnitt 3.6 \[Benachrichtigungen\]](#), Seite 12, für Details.

TYP

Alle

ANWENDBARKEIT

ISG

30.11 Notify.Revision**BEZEICHNUNG**

Notify.Revision – ermittelt die Revisionsnummer der Objektklasse

BESCHREIBUNG

Ruft die Revisionsnummer einer Objektklasse ab. Obwohl `Notify.Revision` in der Notify-Klasse (Benachrichtigung) dokumentiert ist, erhalten Sie natürlich die Revisionsnummer der richtigen Klasse der Objekte.

TYP

Nummer

ANWENDBARKEIT

G

30.12 Notify.UserData**BEZEICHNUNG**

Notify.UserData – setzt/ermittelt die Benutzerdaten des Objekts

BESCHREIBUNG

Hier setzen Sie einen allgemeinen Wert jeglicher Art von Informationen oder rufen ihn ab. Sie können diesen Wert später direkt vom Objekt abrufen. Dies ist ein guter Mechanismus, um die Verwendung globaler Variablen zu vermeiden.

Die Benutzerdaten, die Sie hier angeben, werden auch an Ihre Callback-Funktion übergeben, die Sie mit `InstallEventHandler()` installiert haben. Siehe [Abschnitt 3.6 \[Benachrichtigungen\]](#), [Seite 12](#), für Details.

TYP

Alle

ANWENDBARKEIT

ISG

30.13 Notify.Version**BEZEICHNUNG**

Notify.Version – ermittelt die Versionsnummer der Objektklasse

BESCHREIBUNG

Ruft die Versionsnummer einer Objektklasse ab. Obwohl `Notify.Version` in der Notify-Klasse (Benachrichtigung) dokumentiert ist, erhalten Sie natürlich die Versionsnummer der richtigen Klasse der Objekte.

TYP

Nummer

ANWENDBARKEIT

G

31 Numericbutton-Klasse (Nummerschaltflächen)

31.1 Übersicht

Die Numericbutton-Klasse (Nummerschaltflächen) ist eine Art platzsparender Schieberegler. Sie hat keine zusätzlichen Attribute, aber Sie können einfach die Attribute der Slider-Klasse (Schieberegler) mit dieser Klasse verwenden. Siehe [Abschnitt 44.1 \[Slider-Klasse\]](#), [Seite 225](#), für Details.

Diese Klasse benötigt mindestens MUI Royale 1.5.

Hier folgt nun ein XML-Beispiel für eine Nummerschaltflächen-Schieberegler, mit dem der Benutzer eine Zahl zwischen 0 und 50 konfigurieren kann:

```
<numericbutton min="0" max="50"/>
```


32 Popdrawer-Klasse (Verzeichnisdialog)

32.1 Übersicht

Die Popdrawer-Klasse (Verzeichnisdialog) kann verwendet werden, um einen Standard ASL-Verzeichnisdialog zu benutzen. Eine separater Task wird erstellt, um mit dem Dialog zu arbeiten. Die Anwendung wird weiterhin ausgeführt.

Die Popdrawer-Klasse erstellt ein Zeichenketten-Gadget und eine Dialog-Schaltfläche für Sie. Sie müssen sich nicht um die Handhabung von einem ASL-Dialog kümmern. MUI öffnet automatisch einen, wenn die entsprechende Dialog-Schaltfläche gedrückt wird und aktualisiert das entsprechende Zeichenketten-Gadget, wenn der Benutzer den Dialog beendet. Aus der Sicht des Programmierers müssen Sie nur den Inhalt der Zeichenketten-Gadgets verarbeiten.

32.2 Popdrawer.Acknowledge

BEZEICHNUNG

Popdrawer.Acknowledge – benachrichtigt, wenn der Benutzer auf RETURN klickt

BESCHREIBUNG

Dieses Attribut wird immer dann auf `True` gesetzt, wenn der Benutzer im Zeichenketten-Gadget auf RETURN klickt. Eine Anwendung kann diese Benachrichtigung überwachen und die entsprechenden Maßnahmen ergreifen.

Die Verwendung der TAB-Taste oder ein Mausklick zum Deaktivieren des Gadgets wird `Popdrawer.Acknowledge` nicht auslösen.

TYP

Boolesch

ANWENDBARKEIT

N

32.3 Popdrawer.Active

BEZEICHNUNG

Popdrawer.Active – überprüft, ob der Dialog noch geöffnet ist

BESCHREIBUNG

Die Popdrawer-Klasse (Verzeichnisdialog) erstellt asynchrone Dialoge. So werden Dialoge in einem separat erstellten Task geöffnet und behindern den Rest der Anwendung nicht. Sie können nach dem Status eines Dialogs fragen, indem Sie das Attribut `Popdrawer.Active` abfragen. Es wird `True` zurückgegeben, wenn der Dialog gerade geöffnet ist, andernfalls `False`.

Dieses Attribut wird häufig verwendet, um zu verhindern, dass eine Anwendung beendet wird, während ein Dialog geöffnet ist. Wenn Sie versuchen, das Verzeichnisdialog-Objekt mit einem aktuell geöffneten Dialog zu beenden, wird MUI Ihren Task einfrieren, solange der Dialog offen bleibt.

TYP

Boolesch

ANWENDBARKEIT

G

32.4 Popdrawer.AdvanceOnCR

BEZEICHNUNG

Popdrawer.AdvanceOnCR – aktiviert das nächste Objekt in der Tastaturabfolgekette (V1.1)

BESCHREIBUNG

Hiermit legen Sie fest, ob sich der Zeilenumbruch im Zeichenketten-Gadget wie die TAB-Taste verhalten soll. Wenn Sie also Return oder Enter drücken, wird das nächste/vorherige Gadget in der Tastaturabfolgekette aktiviert.

TYP

Boolesch

ANWENDBARKEIT

ISG

32.5 Popdrawer.Contents

BEZEICHNUNG

Popdrawer.Contents – setzt/ermittelt den aktuellen Verzeichnisdialog-Pfad

BESCHREIBUNG

Mit diesem Attribut setzen Sie den aktuellen Pfad dieses Verzeichnisdialog-Objekts, wie es im Inhalt des Zeichenketten-Gadgets angezeigt wird. Ausserdem können Sie auch den aktuellen Pfad ermitteln.

Popdrawer.Contents wird jedes Mal aktualisiert, wenn sich der Inhalt des Zeichenketten-Gadgets ändert. Wenn Sie eine Benachrichtigung für dieses Attribut einrichten, werden Sie über jeden Tastendruck informiert.

TYP

Zeichenkette

ANWENDBARKEIT

ISGN

32.6 Popdrawer.SaveMode

BEZEICHNUNG

Popdrawer.SaveMode – aktiviert den Speichermodus

BESCHREIBUNG

Setzen Sie diese Variable auf `True`, wenn der Verzeichnisdialog zum Speichern verwendet wird. Der Standardwert ist `False`.

TYP

Boolesch

ANWENDBARKEIT

I

32.7 Popdrawer.Title

BEZEICHNUNG

Popdrawer.Title – setzt den Dialogtitel

BESCHREIBUNG

Legt die Titelzeichenkette für das ASL-Dialogfenster fest.

TYP

Zeichenkette

ANWENDBARKEIT

I

33 Popfile-Klasse (Dateidialog)

33.1 Übersicht

Die Popfile-Klasse (Dateidialog) kann verwendet werden, um einen Standard ASL-Dateidialog aufzurufen. Ein separater Task wird erstellt, um mit dem Dialog zu arbeiten. Die Anwendung wird weiterhin ausgeführt.

Die Popfile-Klasse erstellt für Sie ein Zeichenketten-Gadget und eine Dialog-Schaltfläche. Sie müssen sich nicht um die Handhabung von einem ASL-Dialog kümmern. MUI öffnet automatisch einen, wenn die Dialog-Schaltfläche gedrückt wird, und aktualisiert das entsprechende Zeichenketten-Gadget, wenn der Benutzer den Dialog beendet. Aus der Sicht des Programmierers müssen Sie nur den Inhalt des Zeichenketten-Gadgets verarbeiten.

33.2 Popfile.AcceptPattern

BEZEICHNUNG

Popfile.AcceptPattern – setzt das Datei-Muster

BESCHREIBUNG

Gibt ein AmigaDOS-Muster an, das zum Filtern von Dateien verwendet wird. Das heißt, nur Dateien mit Namen, die diesem Muster entsprechen, werden in die Dateiliste aufgenommen. Standard ist #?, womit jede Datei aufgelistet wird.

TYP

Zeichenkette

ANWENDBARKEIT

I

33.3 Popfile.Acknowledge

BEZEICHNUNG

Popfile.Acknowledge – benachrichtigt, wenn der Benutzer auf RETURN klickt

BESCHREIBUNG

Dieses Attribut wird immer dann auf `True` gesetzt, wenn der Benutzer im Zeichenketten-Gadget auf RETURN klickt. Eine Anwendung kann diese Benachrichtigung überwachen und die entsprechenden Maßnahmen ergreifen.

Die Verwendung der TAB-Taste oder ein Mausklick zum Deaktivieren des Gadgets wird `Popdrawer.Acknowledge` nicht auslösen.

TYP

Boolesch

ANWENDBARKEIT

N

33.4 Popfile.Active

BEZEICHNUNG

Popfile.Active – überprüft, ob der Dialog noch geöffnet ist

BESCHREIBUNG

Die Popfile-Klasse (Dateidialog) erstellt asynchrone Dialoge. So werden Dialoge in einem separat erstellten Task geöffnet und stören den Rest der Anwendung nicht. Sie können nach dem Status eines Dialogs fragen, indem Sie das Attribut `Popfile.Active` abfragen. Es wird `True` zurückgegeben, wenn der Dialog gerade geöffnet ist, andernfalls `False`.

Dieses Attribut wird häufig verwendet, um zu verhindern, dass eine Anwendung beendet wird, während ein Dialog geöffnet ist. Wenn Sie versuchen, das Dateidialog-Objekt mit einem aktuell geöffneten Dialog zu beenden, wird MUI Ihren Task einfrieren, solange der Dialog offen bleibt.

TYP

Boolesch

ANWENDBARKEIT

G

33.5 Popfile.AdvanceOnCR

BEZEICHNUNG

Popfile.AdvanceOnCR – aktiviert das nächste Objekt in der Tastaturabfolgekette (V1.1)

BESCHREIBUNG

Hiermit legen Sie fest, ob sich der Zeilenumbruch im Zeichenketten-Gadget wie die TAB-Taste verhalten soll. Wenn Sie also Return oder Enter drücken, wird das nächste/vorherige Gadget in der Tastaturabfolgekette aktiviert.

TYP

Boolesch

ANWENDBARKEIT

ISG

33.6 Popfile.Contents

BEZEICHNUNG

Popfile.Contents – setzt/ermittelt den aktuellen Dateidialog-Pfad

BESCHREIBUNG

Ermittelt den aktuellen Pfad dieses Dateidialog-Objekts, wie im Inhalt des Zeichenketten-Gadgets angezeigt wird oder legt ihn fest.

`Popfile.Contents` wird jedes Mal aktualisiert, wenn sich der Inhalt des Zeichenketten-Gadgets ändert. Wenn Sie eine Benachrichtigung für dieses Attribut einrichten, werden Sie über jeden Tastendruck informiert.

TYP

Zeichenkette

ANWENDBARKEIT

ISGN

33.7 Popfile.Pattern

BEZEICHNUNG

Popfile.Pattern – legt das Filtermuster für den Dateidialog fest

BESCHREIBUNG

Legt das Filtermuster für den Dateidialog fest. Voreingestellt ist #?.

TYP

Zeichenkette

ANWENDBARKEIT

I

33.8 Popfile.RejectIcons

BEZEICHNUNG

Popfile.RejectIcons – zeigt/verbirgt .info-Dateien

BESCHREIBUNGSetzen Sie diesen Tag auf `True`, damit der Dialog keine Workbench-Symbole anzeigt. Der Standardwert ist `False`.**TYP**

Boolesch

ANWENDBARKEIT

I

33.9 Popfile.RejectPattern

BEZEICHNUNG

Popfile.RejectPattern – legt das Zurückweisungsmuster fest

BESCHREIBUNGGibt ein AmigaDOS-Muster an, mit dem Dateien abgelehnt werden. Das heißt, alle Dateien mit Namen, die mit diesem Muster übereinstimmen, sind nicht in der Dateiliste enthalten. Standard ist `~(#?)`, womit keine Dateien abgelehnt werden.**TYP**

Zeichenkette

ANWENDBARKEIT

I

33.10 Popfile.SaveMode

BEZEICHNUNG

Popfile.SaveMode – aktiviert den Speichermodus

BESCHREIBUNG

Setzen Sie diese Variable auf **True**, wenn der Datei-Dialog zum Speichern verwendet wird. Der Standardwert ist **False**.

TYP

Boolesch

ANWENDBARKEIT

I

33.11 Popfile.ShowPattern

BEZEICHNUNG

Popfile.ShowPattern – zeigt/verbirgt das Muster-Gadget

BESCHREIBUNG

Setzen Sie diesen Tag auf **True**, damit ein Muster-Gadget angezeigt wird. Der Standardwert ist **False**.

TYP

Boolesch

ANWENDBARKEIT

I

33.12 Popfile.Title

BEZEICHNUNG

Popfile.Title – setzt den Dialogtitel

BESCHREIBUNG

Legt die Titelzeichenkette für das ASL-Dialogfenster fest.

TYP

Zeichenkette

ANWENDBARKEIT

I

34 Popfont-Klasse (Schriftdialog)

34.1 Übersicht

Die Popfont-Klasse (Schriftdialog) kann verwendet werden, um einen Standard ASL-Schriftartendialog zu öffnen. Ein separater Task wird erstellt, um mit dem Dialog zu arbeiten. Die Anwendung wird weiterhin ausgeführt.

Die Popfont-Klasse erstellt für Sie ein Zeichenketten-Gadget und eine Dialog-Schaltfläche. Sie müssen sich nicht um die Handhabung von einem ASL-Dialog kümmern. MUI öffnet automatisch einen, wenn die Dialog-Schaltfläche gedrückt wird, und aktualisiert das entsprechende Zeichenketten-Gadget, wenn der Benutzer den Dialog beendet. Aus der Sicht des Programmierers müssen Sie nur den Inhalt des Zeichenketten-Gadgets verarbeiten.

Die Popfont-Klasse zeigt den Namen und die Größe der Schriftart getrennt durch einen Schrägstrich im Zeichenketten-Gadget an, z.B. `topaz/8`.

34.2 Popfont.Acknowledge

BEZEICHNUNG

Popfont.Acknowledge – benachrichtigt, wenn der Benutzer auf RETURN klickt

BESCHREIBUNG

Dieses Attribut wird immer dann auf `True` gesetzt, wenn der Benutzer im Zeichenketten-Gadget auf RETURN klickt. Eine Anwendung kann diese Benachrichtigung überwachen und die entsprechenden Maßnahmen ergreifen.

Die Verwendung der TAB-Taste oder ein Mausklick zum Deaktivieren des Gadgets wird `Popfont.Acknowledge` nicht auslösen.

TYP

Boolesch

ANWENDBARKEIT

N

34.3 Popfont.Active

BEZEICHNUNG

Popfont.Active – überprüft, ob der Dialog noch geöffnet ist

BESCHREIBUNG

Die Popfont-Klasse (Schriftdialog) erstellt asynchrone Dialoge. So werden Dialoge in einem separat erstellten Task geöffnet und behindern den Rest der Anwendung nicht. Sie können nach dem Status eines Dialogs fragen, indem Sie das Attribut `Popfont.Active` abfragen. Es wird `True` zurückgegeben, wenn der Dialog gerade geöffnet ist, andernfalls `False`.

Dieses Attribut wird häufig verwendet, um zu verhindern, dass eine Anwendung beendet wird, während ein Dialog geöffnet ist. Wenn Sie versuchen, das Schriftdialog-Objekt mit einem aktuell geöffneten Dialog zu beenden, wird MUI Ihren Task einfrieren, solange der Dialog offen bleibt.

TYP

Boolesch

ANWENDBARKEIT

G

34.4 Popfont.AdvanceOnCR

BEZEICHNUNG

Popfont.AdvanceOnCR – aktiviert das nächste Objekt in der Tastaturabfolgekette (V1.1)

BESCHREIBUNG

Hiermit legen Sie fest, ob sich der Zeilenumbruch im Zeichenketten-Gadget wie die TAB-Taste verhalten soll. Wenn Sie also Return oder Enter drücken, wird das nächste/vorherige Gadget in der Tastaturabfolgekette aktiviert.

TYP

Boolesch

ANWENDBARKEIT

ISG

34.5 Popfont.Contents

BEZEICHNUNG

Popfont.Contents – setzt/ermittelt den Inhalt des Zeichenketten-Gadgets

BESCHREIBUNG

Setzt oder ermittelt den Inhalt eines Zeichenketten-Gadgets. Die Popfont-Klasse (Schrift-dialog) verwendet einen Schrägstrich, um den Namen und die Größe der Schriftart zu trennen, z.B. `topaz/8`.

`Popfont.Contents` wird jedes Mal aktualisiert, wenn sich der Inhalt des Zeichenketten-Gadgets ändert. Wenn Sie eine Benachrichtigung für dieses Attribut einrichten, werden Sie über jeden Tastendruck informiert.

TYP

Zeichenkette

ANWENDBARKEIT

ISGN

34.6 Popfont.FixedWidthOnly

BEZEICHNUNG

Popfont.FixedWidthOnly – zeigt nur Schriften mit fester Breite an

BESCHREIBUNG

Setzen Sie diesen Tag auf `True`, damit der Dialog nur Schriftarten mit fester Breite (nichtproportionale) anzeigt. Der Standardwert ist `False`.

TYP

Boolesch

ANWENDBARKEIT

I

34.7 Popfont.MaxHeight

BEZEICHNUNG

Popfont.MaxHeight – legt die maximale Schrifthöhe fest

BESCHREIBUNG

Die maximale Schrifthöhe, die der Benutzer auswählen kann. Der Standardwert ist 24.

TYP

Nummer

ANWENDBARKEIT

I

34.8 Popfont.MinHeight

BEZEICHNUNG

Popfont.MinHeight – legt die minimale Schrifthöhe fest

BESCHREIBUNG

Die Mindestschrifthöhe, die der Benutzer auswählen kann. Der Standardwert ist 5.

TYP

Nummer

ANWENDBARKEIT

I

34.9 Popfont.Title

BEZEICHNUNG

Popfont.Title – setzt den Dialogtitel

BESCHREIBUNG

Legt die Titelzeichenkette für das ASL-Dialogfenster fest.

TYP

Zeichenkette

ANWENDBARKEIT

I

35 Poplist-Klasse (Listendialog)

35.1 Übersicht

Die Poplist-Klasse (Listendialog) vereinfacht die Erstellung von Dialogen, die nur eine einfache Liste vordefinierter Gadget-Inhalte enthalten und richtet ein Zeichenketten-Gadget und eine Dialog-Schaltfläche für Sie ein. Immer wenn der Benutzer auf die Dialog-Schaltfläche klickt, wird ein Fenster mit einer Liste vordefinierter Einträge geöffnet, so dass der Benutzer einen aus der Liste auswählen kann.

Wenn Sie ein Listendialog-Objekt erstellen, müssen Sie den Tag `<item>` verwenden, um es mit Einträgen zu füllen. Hier ist ein Beispiel:

```
<poplist>
  <item>The</item>
  <item>quick</item>
  <item>brown</item>
  <item>fox</item>
  <item>jumps</item>
  <item>over</item>
  <item>the</item>
  <item>lazy</item>
  <item>dog</item>
</poplist>
```

Die Listeneinträge, die Sie mit dem Tag `<item>` angeben, können Textformatierungs-codes verwenden. Siehe [Abschnitt 3.9 \[Textformatierungs-codes\]](#), Seite 14, für Details.

35.2 Poplist.Acknowledge

BEZEICHNUNG

Poplist.Acknowledge – benachrichtigt, wenn der Benutzer auf RETURN klickt

BESCHREIBUNG

Dieses Attribut wird immer dann auf `True` gesetzt, wenn der Benutzer im Zeichenketten-Gadget auf RETURN klickt. Eine Anwendung kann diese Benachrichtigung überwachen und die entsprechenden Maßnahmen ergreifen.

Die Verwendung der TAB-Taste oder ein Mausklick zum Deaktivieren des Gadgets wird Poplist.Acknowledge nicht auslösen.

TYP

Boolesch

ANWENDBARKEIT

N

35.3 Poplist.AdvanceOnCR

BEZEICHNUNG

Poplist.AdvanceOnCR – aktiviert das nächste Objekt in der Tastaturabfolgekette (V1.1)

BESCHREIBUNG

Hiermit legen Sie fest, ob sich der Zeilenumbruch im Zeichenketten-Gadget wie die TAB-Taste verhalten soll. Wenn Sie also Return oder Enter drücken, wird das nächste/vorherige Gadget in der Tastaturabfolgekette aktiviert.

TYP

Boolesch

ANWENDBARKEIT

ISG

35.4 Poplist.Contents

BEZEICHNUNG

Poplist.Contents – setzt/ermittelt den aktuellen Inhalt des Listendialogs

BESCHREIBUNG

Ermittelt den aktuellen Inhalt dieses Listendialog-Objekts, wie im Inhalt des Zeichenketten-Gadgets gezeigt wird oder setzt ihn.

`Poplist.Contents` wird jedes Mal aktualisiert, wenn sich der Inhalt des Zeichenketten-Gadgets ändert. Wenn Sie eine Benachrichtigung für dieses Attribut einrichten, werden Sie über jeden Tastendruck informiert.

TYP

Zeichenkette

ANWENDBARKEIT

ISGN

36 Poppen-Klasse (Stiftdialog)

36.1 Übersicht

Die Poppen-Klasse (Stiftdialog) erweitert die Stiftanzeige um Eingabefunktionen. Sie sollte verwendet werden, wenn Ihre Anwendung es Benutzern ermöglicht, einige benutzerdefinierte Stifte für die Wiedergabe zu konfigurieren.

Ein Farbdialog-Objekt wird als eine Art Schaltfläche angezeigt, die die aktuell ausgewählte Farbe anzeigt. Wenn der Benutzer auf die Schaltfläche klickt, öffnet sich ein Dialog-Fenster, das ein Stiftanpassungs-Objekt enthält und der Benutzer kann die Farbe ändern.

Sie können den Fenstertitel des Dialog-Fensters mit dem Attribut `Poppen.Title` im Farbdialog-Objekt steuern. Es wird sich an seinen Wert erinnern und es beim Erstellen des Dialog-Fensters verwenden.

Wie bei den meisten MUI-Dialogen wird das Farbdialog-Fenster asynchron ausgeführt und bleibt dort, bis der Benutzer es mit "OK" oder "Abbrechen" beendet. Außerdem, wird das Dialog-Fenster automatisch geschlossen, wenn das Haupt-Fenster geschlossen wird.

Sie können die aktuelle Farbe von einem Farbdialog-Objekt mithilfe des Attributs `Poppen.RGB` abrufen oder festlegen.

Sie finden einen Beispielcode für die Verwendung dieser Klasse in der Class2-Demo der MUI-Distribution.

36.2 Poppen.RGB

BEZEICHNUNG

`Poppen.RGB` – setzt/ermittelt die Farbe

BESCHREIBUNG

Setzen oder ermitteln Sie die Farbe des Stiftes. Wenn Sie eine Benachrichtigung für dieses Attribut einrichten, werden Sie benachrichtigt, sobald sich die Farbe ändert. Aus dem Hollywood-Skript wird die Farbe als einfacher numerischer Wert angegeben, der 8 Bits für jede Komponente enthält. Wenn Sie die Farbe in der XML-Datei angeben, muss sie als 6-stellige Zeichenkette mit dem Präfix `#` übergeben werden (genau wie in HTML).

TYP

Nummer

ANWENDBARKEIT

ISGN

36.3 Poppen.Title

BEZEICHNUNG

`Poppen.Title` – setzt/ermittelt den Dialogtitel

BESCHREIBUNG

Setzen oder ermitteln Sie den Fenstertitel des Dialog-Fensters.

TYP

Zeichenkette

ANWENDBARKEIT

ISG

37 Prop-Klasse (Proportional)

37.1 Übersicht

Die Prop-Klasse generiert die bekannten Proportional-Gadgets. Es bietet dieselben Attribute wie ein gewöhnliches Boopsi-Gadget der Prop-Klasse. Die Proportional-Gadgets von MUI ermöglichen jedoch die Verwendung beliebiger Bilder für den Regler und für den Hintergrund.

Beachten Sie, dass diese Klasse keine Navigationsschaltflächen für die Bildlaufleiste erstellt. Wenn Sie zwei Navigationsschaltflächen verwenden möchten, die mit der Bildlaufleiste verbunden werden sollen, verwenden Sie stattdessen Scrollbar-Klasse. Siehe [Abschnitt 42.1 \[Scrollbar-Klasse\]](#), [Seite 221](#), für Details.

37.2 Prop.Decrease

BEZEICHNUNG

Prop.Decrease – verringert den Wert des Proportional-Gadgets

ÜBERSICHT

```
mui.DoMethod(id, "Decrease", amount)
```

BESCHREIBUNG

Diese Methode verringert den Wert eines proportionalen Gadgets um den angegebenen Betrag. Negative Werte sind in erlaubt. Die Bereichsüberprüfung wird automatisch durchgeführt.

EINGABEN

`id` ID des Proportional-Gadgets

`amount` Betrag, der von der aktuellen Position des Gadgets subtrahiert wird

37.3 Prop.Entries

BEZEICHNUNG

Prop.Entries – setzt/ermittelt die Anzahl der Einträge

BESCHREIBUNG

Legt die Gesamtzahl der Einträge fest oder ermittelt sie.

TYP

Nummer

ANWENDBARKEIT

ISG

37.4 Prop.First

BEZEICHNUNG

Prop.First – setzt/ermittelt die Nummer des ersten Eintrags

BESCHREIBUNG

Setzt oder ermittelt die Nummer des ersten Eintrags.

TYP

Nummer

ANWENDBARKEIT

ISGN

37.5 Prop.Horiz

BEZEICHNUNG

Prop.Horiz – setzt/ermittelt die Richtung des Proportional-Objekts

BESCHREIBUNG

Bestimmt, ob Sie ein horizontales oder ein vertikales Proportional-Gadget verwenden.

Der Standardwert ist `False`, d.h. vertikal.

TYP

Boolesch

ANWENDBARKEIT

IG

37.6 Prop.Increase

BEZEICHNUNG

Prop.Increase – erhöht den Wert des Proportional-Gadgets

ÜBERSICHT

```
mui.DoMethod(id, "Increase", amount)
```

BESCHREIBUNG

Diese Methode erhöht den Wert eines Proportional-Gadgets um den angegebenen Betrag. Negative Werte sind erlaubt. Die Bereichsüberprüfung wird automatisch durchgeführt.

EINGABEN

`id` ID des Proportional-Gadgets

`amount` Betrag, der der aktuellen Position des Gadgets hinzugefügt wird

37.7 Prop.Slider

BEZEICHNUNG

Prop.Slider – setzt ein Proportional-Objekt in den Schieberegler-Modus

BESCHREIBUNG

Hiermit geben Sie an, dass dieses Proportional-Gadget als Schieberegler (Slider) verwendet wird. MUI verwendet dann möglicherweise andere Bilder. Da Sie beim Erstellen von Schieberegler die Slider-Klasse verwenden sollten, müssen Sie sich normalerweise nicht um dieses Attribut kümmern.

TYP

Boolesch

ANWENDBARKEIT

ISG

37.8 Prop.UseWinBorder

BEZEICHNUNG

Prop.UseWinBorder – setzt das Proportional-Objekt in den Fensterrahmen

BESCHREIBUNG

Wenn Sie dieses Attribut setzen, werden die Gadgets-Steuerung und -Anzeige mit einer der Bildlaufleisten im Fensterrahmen verknüpft.

Es gibt keinen Unterschied beim kommunizieren mit dem Objekt. Sie können alle Proportional-Gadget-Attribute verwenden, unabhängig davon, ob es sich um ein reales Proportional-Gadget oder nur um eine Bildlaufleiste im Fensterrahmen handelt. Natürlich verwenden Gadgets in Fensterrahmen das Intuition-Aussehen, unabhängig davon, was der Benutzer konfiguriert hat.

Sie **müssen** Rahmen-Bildlaufleisten für das übergeordnete Fenster mit den entsprechenden Attributen aktivieren, bevor Sie `Prop.UseWinBorder` verwenden.

Offensichtlich können Sie nur genau ein Proportional-Gadget mit einer Fenster-Bildlaufleiste verknüpfen. Das Verknüpfen von mehr als einem Gadget mit demselben Fenster-Bildlaufleiste führt zu großer Verwirrung.

Die folgenden Werte werden von diesem Attribut erkannt:

- `Left` Verwendet den linken Fensterrahmen.
- `Right` Verwendet den rechten Fensterrahmen.
- `Bottom` Verwendet den unteren Fensterrahmen.

TYP

Zeichenkette (siehe oben für mögliche Werte)

ANWENDBARKEIT

I

37.9 Prop.Visible

BEZEICHNUNG

Prop.Visible – setzt/ermittelt die Anzahl der sichtbaren Einträge

BESCHREIBUNG

Setzt oder ermittelt die Anzahl der sichtbaren Einträge.

TYP

Nummer

ANWENDBARKEIT

ISG

38 Radio-Klasse

38.1 Übersicht

Die Radio-Klasse generiert Radio-Schaltflächen-Gadgets. Sie machen die gleiche Arbeit wie Auswahl-Listefeld-Gadgets (Cycle), verbrauchen aber mehr Platz im Fenster. Vielleicht ist das der Grund, warum so wenige von ihnen in bestehenden Anwendungen gefunden werden. Wenn Sie ein Radio-Gadget definieren, müssen Sie mit dem Tag `<item>` das Radio-Gadget mit Elementen füllen. Jedes Radio-Gadget muss mindestens eine Schaltfläche haben.

Hier ein Beispiel für einen XML-Auszug zum Erstellen eines Radio-Gadgets:

```
<radio id="printer">
  <item>HP Deskjet</item>
  <item>NEC P6</item>
  <item>Okimate 20</item>
</radio>
```

Die Radioeinträge, die Sie mit dem Tag `<item>` angeben, können Textformatierungs-codes verwenden. Siehe [Abschnitt 3.9 \[Textformatierungs-codes\]](#), Seite 14, für Details.

38.2 Radio.Active

BEZEICHNUNG

Radio.Active – setzt/ermittelt die aktive Radio-Schaltfläche

BESCHREIBUNG

Dieses Attribut definiert die Nummer der aktiven Schaltfläche in den Radio-Gadgets. Der gültige Bereich reicht von 0 für den ersten Eintrag bis AnzahlEinträge-1 für den letzten Eintrag.

Durch die Einstellung von `Radio.Active` wird das Gadget aktualisiert. Auf der anderen Seite, wenn der Benutzer mit dem Gadget herumspielt, wird `Radio.Active` immer den aktuellen Zustand wiedergeben.

TYP

Nummer

ANWENDBARKEIT

ISGN

39 Rectangle-Klasse (Rechteck)

39.1 Übersicht

Die Rectangle-Klasse (Rechteck) scheint nutzlos, da sie nicht viele Attribute oder Methoden selbst definiert. Objekte dieses Typs werden jedoch häufig in jeder Anwendung verwendet. Sie ermöglichen das Einfügen von dehnbaren Bereichen, um den MUI-Layoutprozess zu steuern.

Sie können die Attribute der Oberklasse verwenden, um die Größe und andere Attribute des Rechtecks zu steuern. Z.B. können Sie `Area.FixWidth` und `Area.FixHeight` verwenden, um Rechtecke mit einer festen Größe zu erstellen.

Oft sind auch Rechteck-Objekte als Füllraum neben Objekten erforderlich, die selbst nicht veränderbar sind. Das Objekt `<label>` ist z.B. nicht veränderbar. Um zu verhindern, dass diese statischen Beschriftungs-Objekte die Größenänderungsfunktion Ihrer gesamten GUI blockieren, können Sie sie einfach mit einem Objekt `<rectangle>` auffüllen. Hier ist ein Beispiel für eine Beschriftung neben einem Auswahlkästchen-Objekt, das mit einem unsichtbaren Rechteck gefüllt ist:

```
<hgroup>
  <label>Enable hardware acceleration</label>
  <checkbox/>
  <rectangle/>
</hgroup>
```

39.2 Rectangle.BarTitle

BEZEICHNUNG

Rectangle.BarTitle – definiert den Titel für den horizontalen Balken

BESCHREIBUNG

Dieses Attribut beschreibt eine Textzeichenkette, die im Gruppentitelstil zentriert im Rechteck angezeigt wird. Wirklich nur sinnvoll, wenn `Rectangle.HBar` auf `True` gesetzt ist.

TYP

Zeichenkette

ANWENDBARKEIT

IG

39.3 Rectangle.HBar

BEZEICHNUNG

Rectangle.HBar – fügt einen horizontalen Balken zum Rechteck hinzu

BESCHREIBUNG

Wenn `True` gesetzt ist, zeichnet MUI einen horizontalen Balken in der Mitte des Rechtecks. Solche Balken können anstelle von Gruppenrahmen verwendet werden, um Objekte in einem Fenster zu trennen.

TYP

Boolesch

ANWENDBARKEIT

IG

39.4 Rectangle.VBar

BEZEICHNUNG

Rectangle.VBar – fügt einen vertikalen Balken zum Rechteck hinzu

BESCHREIBUNG

Wenn `True` festgelegt ist, zeichnet MUI einen vertikalen Balken in der Mitte des Rechtecks. Solche Balken können anstelle von Gruppenrahmen verwendet werden, um Objekte in einem Fenster zu trennen.

TYP

Boolesch

ANWENDBARKEIT

IG

40 Register-Klasse

40.1 Übersicht

Die Register-Klasse ist eine spezielle Klasse für das Arbeiten mit mehrseitigen Gruppen. Mit dieser Klasse müssen Sie nur ein Feld von Gruppen bereitstellen, die die untergeordneten Elemente des Registers beschreiben. Wie diese untergeordneten Elemente visualisiert werden, entweder mit einem Auswahl-Listefeld-Gadget (*Cycle*) oder einer registerartigen Gruppe, ist die Wahl des Benutzers. Sie müssen das Attribut `Group.Title` verwenden, um Ihren Registerkarten einen Titel zu geben.

Hier ist ein Beispiel für eine dreiseitige Registergruppe:

```
<register>
  <vgroup title="Page 1">
    <listview>
      <column>
        <item>Entry</item>
      </column>
    </listview>
  </vgroup>
  <vgroup title="Page 2">
    <texteditor/>
  </vgroup>
  <vgroup title="Page 3">
    <button>Click me</button>
  </vgroup>
</register>
```

40.2 Register.ActivePage

BEZEICHNUNG

`Register.ActivePage` – setzt/ermittelt die aktive Seite der Registergruppe

BESCHREIBUNG

Stellen Sie die aktive Seite einer Registergruppe ein (oder rufen Sie sie ab). Nur diese aktive Seite wird angezeigt, alle anderen sind ausgeblendet.

Der Wert kann zwischen 0 (für das erste untergeordnete Element und `AnzahlSeiten-1` (für das letzte untergeordnete Element) liegen. Untergeordnete Elemente werden in der Reihenfolge der Erschaffung angesprochen.

Die folgenden speziellen Werte sind beim Setzen der aktiven Seite möglich:

<code>First</code>	Erste Seite.
<code>Last</code>	Letzte Seite.
<code>Prev</code>	Vorherige Seite.
<code>Next</code>	Nächste Seite.
<code>Advance</code>	Seite vorblättern.

Hinweis: Sie dürfen **niemals** einen falschen Seitenwert angeben!

TYP

Nummer oder Zeichenkette (siehe oben für mögliche Werte)

ANWENDBARKEIT

ISGN

40.3 Register.AddPage

BEZEICHNUNG

Register.AddPage – fügt der Registergruppe eine neue Seite hinzu (V1.2)

ÜBERSICHT

```
mui.DoMethod(id, "AddPage", obj)
```

BESCHREIBUNG

Diese Methode kann verwendet werden, um einer Registergruppe eine neue Seite hinzuzufügen. Die mit `obj` angegebene neue Seite muss ein losgelöstes Gruppen-Objekt sein. Dieses losgelöste Gruppen-Objekt wird dann als letzte Seite des Registers hinzugefügt. Nachdem diese Methode beendet wurde, ändert das angegebene Gruppen-Objekt seinen Status von losgelöst zu angehängt. Deshalb dürfen Sie keine Befehle mehr mit diesem Objekt verwenden, die jetzt ein losgelöstes Objekt erwarten.

Losgelöste MUI-Objekte können entweder durch Aufrufen des Befehls `mui.CreateObject()` oder durch explizites Trennen von ihren übergeordneten Elementen mithilfe der Methode `Register.ClosePage` erstellt werden.

Dieser Befehl erfordert mindestens MUI 4.0.

EINGABEN

<code>id</code>	ID des Register-Objekts
<code>obj</code>	ID des Gruppen-Objekts, das als neue Seite hinzugefügt werden soll

40.4 Register.Closable

BEZEICHNUNG

Register.Closable – fügt ein Schließ-Gadget der Registerkarte hinzu (V1.2)

BESCHREIBUNG

Wenn Sie dieses Attribut auf `True` setzen, fügt MUI jeder einzelnen Registerkarte Schließ-Gadgets hinzu. Wenn der Benutzer auf das Schließ-Gadget klickt, wird `Register.CloseRequest` ausgelöst.

Dieser Befehl erfordert mindestens MUI 4.0.

TYP

Boolesch

ANWENDBARKEIT

I

40.5 Register.ClosePage

BEZEICHNUNG

Register.ClosePage – entfernt die Seite aus der Registergruppe (V1.2)

ÜBERSICHT

```
mui.DoMethod(id, "ClosePage", obj)
```

BESCHREIBUNG

Diese Methode schließt die durch `obj` angegebene Seitenregisterkarte. Das Gruppen-Objekt, welches die Seite darstellt, die geschlossen wird, ändert dann seinen Status von angehängt in losgelöst. Das bedeutet, dass Sie es jetzt mit einem Befehl wie `Group.Insert` und `Register.AddPage` an ein anderes Register oder eine andere Gruppe anhängen können oder Sie können es mit `mui.FreeObject()` löschen.

Dieser Befehl erfordert mindestens MUI 4.0.

EINGABEN

<code>id</code>	ID des Register-Objekts
<code>obj</code>	ID der zu entfernenden Seite

40.6 Register.CloseRequest

BEZEICHNUNG

Register.CloseRequest – reagiert auf die Schließenanfrage der Seite (V1.2)

BESCHREIBUNG

Wenn der Benutzer auf das Schließ-Gadget einer Registerseite klickt, wird die Seite nicht sofort geschlossen. Stattdessen setzt MUI dieses Attribut nur auf `True`, damit Ihre Anwendung darauf reagieren kann.

In der Regel werden Sie eine Benachrichtigung einrichten, die die Seite automatisch schließt, wenn eine Schließenanforderung anliegt. Sie können jedoch z.B. einen Bestätigungsdialog öffnen oder zuerst einige andere Schritte ausführen.

Dieser Befehl erfordert mindestens MUI 4.0.

TYP

Boolesch

ANWENDBARKEIT

N

40.7 Register.GetPageID

BEZEICHNUNG

Register.GetPageID – gibt die ID einer Seite innerhalb der Registergruppe zurück (V1.2)

ÜBERSICHT

```
id$ = mui.DoMethod(id, "GetPageID", idx)
```

BESCHREIBUNG

Diese Methode gibt die ID der Seite im angegebenen Index `idx` innerhalb der Registergruppe zurück. `idx` kann eine absolute Zahl sein, die von 0 bis zur Anzahl der Seiten im Register minus 1 reicht oder es kann einer der folgenden Sonderwerte sein:

First Erste Seite.
Last Letzte Seite.
Active Aktive Seite.

Dieser Befehl erfordert mindestens MUI 4.0.

EINGABEN

`id` ID des Register-Objekts
`idx` Absoluter Index der gewünschten Seite oder spezielle Zeichenkettenkonstante (siehe oben)

RÜCKGABEWERTE

`id$` ID des Seiten-Objekts im angegebenen Registerindex

40.8 Register.InsertPage

BEZEICHNUNG

`Register.InsertPage` – fügt eine neue Seite in die Registergruppe ein (V1.4)

ÜBERSICHT

```
mui.DoMethod(id, "InsertPage", obj, pos)
```

BESCHREIBUNG

Diese Methode kann verwendet werden, um eine neue Seite in eine Registergruppe einzufügen. Die mit `obj` angegebene neue Seite muss ein losgelöstes Gruppen-Objekt sein. Dieses losgelöste Gruppen-Objekt wird dann in die mit `pos` angegebene Registerposition eingefügt. Dies ist eine ganzzahlige Position, beginnend bei 0 für die erste Seite. Nachdem diese Methode beendet wurde, ändert das angegebene Gruppen-Objekt seinen Status von losgelöst zu angehängt. Deshalb dürfen Sie keine Befehle mehr mit diesem Objekt verwenden, die jetzt ein losgelöstes Objekt erwarten.

Losgelöste MUI-Objekte können entweder durch Aufrufen des Befehls `mui.CreateObject()` oder durch explizites Trennen von ihren übergeordneten Elementen mithilfe der Methode `Register.ClosePage` erstellt werden.

Dieser Befehl erfordert mindestens MUI 4.0.

EINGABEN

`id` ID des Register-Objekts
`obj` ID des Gruppen-Objekts, das als neue Seite eingefügt werden soll
`pos` Einfügeposition beginnend mit 0 für die erste Seite

40.9 Register.Pages

BEZEICHNUNG

Register.Pages – ermittelt die Anzahl der Seiten im Register-Objekt (V1.2)

BESCHREIBUNG

Sie können dieses Attribut abfragen, um die aktuelle Anzahl der Seiten im Register-Objekt zu ermitteln.

Dieser Befehl erfordert mindestens MUI 4.0.

TYP

Nummer

ANWENDBARKEIT

G

40.10 Register.Position

BEZEICHNUNG

Register.Position – legt die Position der Seitenregisterkarten fest (V1.2)

BESCHREIBUNG

Mit diesem Attribut können Sie konfigurieren, ob der Seitenregisterkarten-Browser oben, links, rechts oder unten auf den Seiten angezeigt werden soll.

Die folgenden Werte werden von diesem Attribut erkannt:

- Left** Setzt den Registerkarten-Browser links von den Seiten.
- Right** Setzt den Registerkarten-Browser rechts von den Seiten.
- Bottom** Setzt den Registerkarten-Browser an den unteren Rand der Seiten.
- Top** Setzt den Registerkarten-Browser an den oberen Rand der Seiten.

Dieser Befehl erfordert mindestens MUI 4.0.

TYP

Zeichenkette (siehe oben für mögliche Werte)

ANWENDBARKEIT

I

41 Scale-Klasse (Skala)

41.1 Übersicht

Ein Skala-Objekt (Scale) generiert eine Prozentskala von 0% bis 100%. Ein guter Platz für ein solches Objekt ist z.B. unter einer Füllanzeige.

Abhängig davon, wie viel Platz verfügbar ist, wird die Skala mehr oder weniger detailliert sein.

Dank des automatischen Gestaltungs-Systems von MUI müssen Sie sich keine Gedanken über die Größe machen. Wenn es in einer vertikalen Gruppe direkt unter dem Objekt platziert wird, auf das sich die Skala bezieht, ist alles in Ordnung.

Die Scale-Klasse wird oft zusammen mit der Gauge-Klasse (Fortschrittsbalken) verwendet. Siehe [Abschnitt 16.1 \[Gauge-Klasse\]](#), [Seite 99](#), für Details.

41.2 Scale.Horiz

BEZEICHNUNG

Scale.Horiz – setzt/ermittelt die Ausrichtung der Skala

BESCHREIBUNG

Geben Sie an, ob Sie eine horizontale (**True**) oder eine vertikale (**False**) Skala wünschen.

Derzeit ist nur die horizontale Ausrichtung implementiert.

TYP

Boolesch

ANWENDBARKEIT

ISG

42 Scrollbar-Klasse (Bildlaufleiste)

42.1 Übersicht

Die Scrollbar-Klasse (Bildlaufleiste) erstellt ein Proportional- und zwei Schaltflächen-Gadgets mit geeigneten Bildern, um eine Bildlaufleiste zu erstellen. Da die Scrollbar-Klasse eine Unterklasse von Prop-Klasse ist, können Sie mit einer Bildlaufleiste kommunizieren und überwachen, als wäre es nur ein einzelnes Proportional-Gadget. Siehe [Abschnitt 37.1 \[Prop-Klasse\]](#), [Seite 205](#), für Details.

Diese Klasse benötigt mindestens MUI Royale 1.4.

42.2 Scrollbar.IncDecSize

BEZEICHNUNG

Scrollbar.IncDecSize – setzt/ermittelt den Verschiebewert beim Schaltflächenklick (V1.4)

BESCHREIBUNG

Stellen Sie den Betrag ein, um den die Bildlaufleiste bei jedem Klicken auf eine der Pfeiltasten vergrößert oder verkleinert wird. Der Standardwert ist 1.

TYP

Nummer

ANWENDBARKEIT

ISG

42.3 Scrollbar.Type

BEZEICHNUNG

Scrollbar.Type – stellt den Bildlaufleisten-Typ ein (V1.4)

BESCHREIBUNG

Geben Sie einen bestimmten Bildlaufleisten-Typ an. Normalerweise sollten Sie die Benutzerauswahl respektieren und vermeiden, dieses Attribut zu verwenden.

Folgende Eingaben werden akzeptiert:

Default	Standardtyp
Bottom	Unterer Typ.
Top	Oberer Typ.
Sym	Symbol Typ.
None	Kein Typ.

TYP

Zeichenkette (siehe oben für mögliche Werte)

ANWENDBARKEIT

I

43 Scrollgroup-Klasse (Bildlaufgruppen)

43.1 Übersicht

Scrollgroup-Klasse (Bildlaufgruppen) können verwendet werden, um virtuelle Gruppen mit Bildlaufleisten zu versehen. Diese Bildlaufleisten passen sich automatisch an die virtuelle Größe und die Größe der Anzeige der zugrunde liegenden virtuellen Gruppe an. Wenn das Scrollen unnötig ist (d.h. die virtuelle Gruppe ist vollständig sichtbar), können die Scroller deaktiviert werden oder sogar vollständig verschwinden, abhängig von den Benutzereinstellungen.

Es ist wichtig zu beachten, dass Bildlaufgruppen-Objekte nicht mit normalen Gruppen arbeiten. Stattdessen müssen Sie ihnen eine virtuelle Gruppe als untergeordnetes Element zuweisen, d.h. Sie müssen den Tag `<virtgroup>` verwenden, um ein MUI-Objekt vom Typ Virtgroup-Klasse zu erstellen. Siehe [Abschnitt 51.1 \[Virtgroup-Klasse\], Seite 287](#), für Details.

Hier ist ein XML-Beispiel:

```
<scrollgroup>
  <virtgroup>
    <radio>
      <item>Amiga 500</item>
      <item>Amiga 1200</item>
      <item>Amiga 4000</item>
    </radio>
    <listview>
      <column/>
    </listview>
  </virtgroup>
</scrollgroup>
```

Der obige XML-Code bettet ein Radio- und ein Listenansichts-Objekt in eine virtuelle Gruppe ein, die wiederum in eine Bildlaufgruppe eingebettet ist.

43.2 Scrollgroup.FreeHoriz

BEZEICHNUNG

Scrollgroup.FreeHoriz – lässt horizontales Scrollen zu

BESCHREIBUNG

Geben Sie an, ob eine Bildlaufgruppe horizontal beweglich sein soll. Der Standardwert ist `True`.

TYP

Boolesch

ANWENDBARKEIT

I

43.3 Scrollgroup.FreeVert

BEZEICHNUNG

Scrollgroup.FreeVert – lässt vertikales Scrollen zu

BESCHREIBUNG

Geben Sie an, ob eine Bildlaufgruppe vertikal beweglich sein soll. Der Standardwert ist `True`.

TYP

Boolesch

ANWENDBARKEIT

I

43.4 Scrollgroup.UseWinBorder

BEZEICHNUNG

Scrollgroup.UseWinBorder – verwendet Fensterlaufleisten für diese Bildlaufgruppe

BESCHREIBUNG

Wenn Sie dies auf `True` setzen, wird MUI diese Bildlaufgruppe automatisch über Gadgets im Fensterrahmen kontrollieren können. MUI verwendet die rechte Fensterlaufleiste, wenn sich die virtuellen Gruppen vertikal bewegen dürfen und die untere Fensterlaufleiste, wenn sich die virtuelle Gruppe horizontal bewegen darf.

Sie müssen die entsprechenden Fensterattribute festlegen, z.B. muss das Attribut `Window.UseRightBorderScroller` für das übergeordnete Fenster festgelegt werden, um diese Funktion zu verwenden.

TYP

Boolesch

ANWENDBARKEIT

I

44 Slider-Klasse (Schieberegler)

44.1 Übersicht

Die Slider-Klasse (Schieberegler) generiert ein GUI-Element, mit dem ein Benutzer einen numerischen Wert anpassen kann. Der Programmierer hat keinen großen Einfluss auf das Outfit des Schieberegler, es sind nur sehr wenige Tags verfügbar. Zukünftige Versionen von MUI werden wahrscheinlich einige Vorzugsoptionen enthalten, um es dem Benutzer (*nicht* dem Programmierer) zu ermöglichen, dieses Outfit zu konfigurieren.

Beachten Sie, dass Schieberegler eine Unterklasse der Group-Klasse (Gruppen) ist. Sie können horizontale oder vertikale Schieberegler verwenden, indem Sie einfach das Attribut `Slider.Horiz` verwenden. Standard ist ein horizontaler Schieberegler.

Hier ist ein XML-Beispiel für einen Schieberegler, mit dem der Benutzer eine Zahl zwischen 0 und 100 konfigurieren kann:

```
<slider min="0" max="100"/>
```

44.2 Slider.Format

BEZEICHNUNG

`Slider.Format` – passt den Text des Schiebereglers an

BESCHREIBUNG

Eine Vorlagenzeichenkette zum Beschreiben des Formats der Schieberegleranzeige. Immer wenn `Slider.mui` einen neuen Wert wiedergeben wird, schreibt `Slider.mui` ihn nicht einfach in eine Zeichenkette. Stattdessen wird die in diesem Attribut angegebene Formatmaske verwendet, um die neue Zeichenkette zu erstellen, die dann auf der Schieberegler-Darstellung wiedergegeben wird. Jedes `%ld` in der Formatzeichenkette wird durch die aktuelle Schieberegler-Ebene ersetzt.

Beachten Sie: Die maximale Länge der Ergebniszeichenkette für `Slider.Format` ist auf 32 Zeichen begrenzt. Wenn Sie mehr benötigen, müssen Sie `Slider.Stringify` verwenden.

`Slider.Format` Standardeinstellung ist `%ld`.

Die Zeichenkette, die Sie hier angeben, kann Textformatierungs-codes verwenden. Siehe [Abschnitt 3.9 \[Textformatierungs-codes\]](#), [Seite 14](#), für Details.

TYP

Zeichenkette

ANWENDBARKEIT

ISG

BEISPIEL

```
mui.Set(obj, "Format", "Custom text: %ld")
```

Der obige Code setzt den Schieberegler-Text auf "Custom text" gefolgt von der aktuellen Schieberegler-Ebene.

44.3 Slider.Horiz

BEZEICHNUNG

Slider.Horiz – setzt/ermittelt die Ausrichtung des Schiebereglers

BESCHREIBUNG

Geben Sie an, ob Sie einen horizontalen oder vertikalen Schieberegler wünschen.

TYP

Boolesch

ANWENDBARKEIT

ISG

44.4 Slider.Level

BEZEICHNUNG

Slider.Level – setzt/ermittelt die aktuelle Position des Schiebereglers

BESCHREIBUNG

Setzt die aktuelle Position des Schiebereglers oder ermittelt sie. Dieser Wert liegt garantiert zwischen `Slider.Min` und `Slider.Max`.

TYP

Nummer

ANWENDBARKEIT

ISGN

44.5 Slider.Max

BEZEICHNUNG

Slider.Max – setzt/ermittelt die maximale Position des Schiebereglers

BESCHREIBUNG

Passt den maximalen Wert für ein Schieberegler-Objekt an.

TYP

Nummer

ANWENDBARKEIT

ISG

44.6 Slider.Min

BEZEICHNUNG

Slider.Min – setzt/ermittelt die minimale Position des Schiebereglers

BESCHREIBUNG

Passt den Mindestwert für ein Schieberegler-Objekt an. Natürlich können Sie negative Zahlen verwenden, z.B. für einen Schieberegler zum Anpassen der Aufgabenpriorität.

TYP

Nummer

ANWENDBARKEIT

ISG

44.7 Slider.Pressed**BEZEICHNUNG**

Slider.Pressed – benachrichtigt, wenn der Schieberegler gedrückt wird (V1.5)

BESCHREIBUNG

Dieses Attribut wird ausgelöst, wenn der Benutzer den Schieberegler drückt.

TYP

Boolesch

ANWENDBARKEIT

N

44.8 Slider.Quiet**BEZEICHNUNG**

Slider.Quiet – versteckt die aktuelle Position des Schiebereglers

BESCHREIBUNG

Wenn True gesetzt ist, zeigt der Schieberegler nicht die aktuelle Position in einem Text-Objekt an.

TYP

Boolesch

ANWENDBARKEIT

I

44.9 Slider.Reverse**BEZEICHNUNG**

Slider.Reverse – kehrt die Richtung des Schiebereglers um

BESCHREIBUNG

Wenn Sie dieses Attribut auf True setzen, wird die Richtung des Schiebereglers umgekehrt.

TYP

Boolesch

ANWENDBARKEIT

ISG

44.10 Slider.Stringify

BEZEICHNUNG

Slider.Stringify – ruft den Callback für benutzerdefinierten Schieberegler-Texte auf

BESCHREIBUNG

Dieses Attribut kann verwendet werden, um eine Callback-Funktion einzurichten, die einen Schieberegler in einen benutzerdefinierten Schieberegler-Text umwandelt. Die Callback-Funktion, die Sie hier angeben, wird wie ein Standard-MUI-Royale-Ereignis-Callback aufgerufen, jedoch mit dem folgenden zusätzlichen Argument:

Value: Enthält den aktuellen Schieberegler.

Ihre Callback-Funktion muss dann eine Zeichenkette zurückgeben, die im Schieberegler-Gadget angezeigt werden soll. Sie könnten z.B. eine schön formatierte Zeitkette (hh:mm:ss) in einem Schieberegler anzeigen, die eine Anzahl von Sekunden anpasst. Oder Sie müssen eine Baudrate von Hand vordefinierten Werte anpassen. Verwenden Sie einfach `Slider.Stringify` und Sie haben die Wahl, wie der Schieberegler-Wert in eine Zeichenkette übersetzt wird.

Beachten Sie, dass Sie die Funktion niemals direkt, sondern immer als eine Zeichenkette übergeben müssen, die den Namen einer Hollywood-Funktion für dieses Attribut angibt.

Beachten Sie auch, dass Sie für dieses Attribut eine Benachrichtigung einrichten müssen. Andernfalls wird die Callback-Funktion niemals aufgerufen. Siehe [Abschnitt 3.6 \[Benachrichtigungen\]](#), [Seite 12](#), für Details.

TYP

Zeichenkette

ANWENDBARKEIT

ISGN

BEISPIEL

```
<slider notify="stringify" stringify="p_Callback" min="0" max="64"
  level="32" format="--Full--"/>
```

```
Function p_Callback(msg)
  If msg.value = 0 Then Return("Mute")
  If msg.value = 64 Then Return("Full")
  Return(msg.value)
EndFunction
```

Der obige Code erstellt einen Schieberegler zur Steuerung der Lautstärke. Wenn der Schieberegler ganz links ist, wird "Mute", hingegen wenn er sich ganz rechts befindet, wird "Full" angezeigt. Beachten Sie, dass wir den Tag "Format" in der XML-Datei verwenden, um eine anfängliche Breite für den Schieberegler festzulegen, sodass er groß genug für alle Zeichenketten ist, die von unserer Callback-Funktion zurückgegeben werden.

45 String-Klasse (Zeichenketten)

45.1 Übersicht

Die String-Klasse (Zeichenketten) generiert Standard-Zeichenketten-Gadgets mit allen Bearbeitungsfunktionen (Löschen, Rückgängigmachen usw.). Standardmäßig haben Zeichenketten-Gadgets keine Beschriftung neben sich. Wenn Sie eine Beschriftung neben Ihrem Zeichenketten-Gadget haben möchten, müssen Sie sie in eine `<hgroup>` einfügen und dann die Label-Klasse oder Text-Klasse verwenden, um eine Beschriftung daneben zu platzieren.

Hier ist ein XML-Beispiel für ein Zeichenketten-Gadget:

```
<string id="mystring" contents="Enter something!"/>
```

45.2 String.Accept

BEZEICHNUNG

String.Accept – setzt/ermittelt Zeichen, die von Zeichenketten Gadgets akzeptiert werden

BESCHREIBUNG

Eine Zeichenkette mit Zeichen, die als Eingaben für das Zeichenketten-Gadget zulässig sind. Immer wenn der Benutzer ein Zeichen eintippt, das nicht in `String.Accept` gefunden wurde, hört er einen Signalton und der Gadget-Inhalt ändert sich nicht.

TYP

Zeichenkette

ANWENDBARKEIT

ISG

BEISPIEL

```
mui.Set(obj, "Accept", "0123456789-")
```

Der obige Code setzt ein Zeichenketten-Gadget so, dass es nur Zahlen und Bindestriche akzeptiert.

45.3 String.Acknowledge

BEZEICHNUNG

String.Acknowledge – benachrichtigt, wenn der Benutzer auf RETURN klickt

BESCHREIBUNG

Dieses Attribut wird immer dann auf `True` gesetzt, wenn der Benutzer im Gadget auf RETURN klickt. Eine Anwendung kann diese Benachrichtigung überwachen und die entsprechenden Maßnahmen ergreifen.

Die Verwendung der TAB-Taste oder ein Mausklick zum Deaktivieren des Gadgets wird `String.Acknowledge` nicht ausgelöst.

TYP

Boolesch

ANWENDBARKEIT

N

45.4 String.AdvanceOnCR

BEZEICHNUNG

String.AdvanceOnCR – aktiviert das nächste Objekt in der Tastaturabfolgekette

BESCHREIBUNG

Legt fest, wenn die Zeilenumbrüche in Zeichenketten-Gadgets sich wie die TAB-Taste verhalten sollen. Wenn Sie also CR drücken, wird das nächste/vorherige Gadget in der Tastaturabfolgekette aktiviert.

TYP

Boolesch

ANWENDBARKEIT

ISG

45.5 String.Contents

BEZEICHNUNG

String.Contents – setzt/ermittelt den Inhalt des Zeichenketten-Gadgets

BESCHREIBUNG

Setzt oder ermittelt den Inhalt eines Zeichenketten-Gadgets.

`String.Contents` wird jedes Mal aktualisiert, wenn sich der Inhalt des Zeichenketten-Gadgets ändert. Wenn Sie eine Benachrichtigung für dieses Attribut einrichten, werden Sie über jeden Tastendruck informiert.

Wenn Sie versuchen einen längeren Inhalt als `String.MaxLen` zu setzen, wird MUI die zusätzlichen Zeichen automatisch entfernen.

TYP

Zeichenkette

ANWENDBARKEIT

ISGN

45.6 String.Copy

BEZEICHNUNG

String.Copy – kopiert den markierten Text (V1.3)

ÜBERSICHT

```
mui.DoMethod(id, "Copy")
```


BESCHREIBUNG

Kopiert den aktuell ausgewählten Text in die Zwischenablage.

Diese Methode benötigt MUI 4.0 oder besser.

EINGABEN

id ID des Zeichenketten-Objekts

45.7 String.CursorPos

BEZEICHNUNG

String.CursorPos – setzt/ermittelt die Cursorposition (V1.3)

BESCHREIBUNG

Setzt die aktuelle Cursorposition oder ermittelt sie.

Dieses Attribut erfordert MUI 4.0 oder höher.

TYP

Nummer

ANWENDBARKEIT

SG

45.8 String.Cut

BEZEICHNUNG

String.Cut – schneidet den markierten Text aus (V1.3)

ÜBERSICHT

```
mui.DoMethod(id, "Cut")
```

BESCHREIBUNG

Schneidet den ausgewählten Text aus und legt ihn in die Zwischenablage.

Diese Methode benötigt MUI 4.0 oder besser.

EINGABEN

id ID des Zeichenketten-Objekts

45.9 String.Insert

BEZEICHNUNG

String.Insert – fügt den Text ein (V1.3)

ÜBERSICHT

```
mui.DoMethod(id, "Insert", t$)
```

BESCHREIBUNG

Damit wird der in `t$` angegebene Text an der aktuellen Cursorposition eingefügt.

Diese Methode benötigt MUI 4.0 oder besser.

EINGABEN

id	ID des Zeichenketten-Objekts
t\$	Text zum Einfügen

45.10 String.MarkEnd

BEZEICHNUNG

String.MarkEnd – setzt/ermittelt die Endposition des markierten Textes (V1.3)

BESCHREIBUNG

Setzt oder ermittelt die Endposition des markierten Textes. Setzen Sie dies auf -1, um eine Textmarkierung zu entfernen.

Dieses Attribut erfordert MUI 4.0 oder höher.

TYP

Nummer

ANWENDBARKEIT

ISG

45.11 String.MarkStart

BEZEICHNUNG

String.MarkStart – setzt/ermittelt die Startposition des markierten Textes (V1.3)

BESCHREIBUNG

Setzt oder ermittelt die Startposition des markierten Textes. Setzen Sie dies auf -1, um eine Textmarkierung zu entfernen.

Dieses Attribut erfordert MUI 4.0 oder höher.

TYP

Nummer

ANWENDBARKEIT

ISG

45.12 String.MaxLen

BEZEICHNUNG

String.MaxLen – setzt/ermittelt die maximale Länge der Zeichenkette

BESCHREIBUNG

Richtet die maximale Länge für das Zeichenketten-Gadget ein oder ermittelt sie. Dieses Attribut ist nur zum Zeitpunkt der Objekterstellung gültig.

Die maximale Standardlänge beträgt 80.

HINWEIS: Die maximale Länge enthält das 0-Byte am Ende der Zeichenkette. Um den Benutzer z.B. 10 Zeichen eingeben zu lassen, müssten Sie eine maxlen von 11 angeben.

TYP

Nummer

ANWENDBARKEIT

IG

45.13 String.Paste

BEZEICHNUNG

String.Paste – fügt den Text aus der Zwischenablage ein (V1.3)

ÜBERSICHT

```
mui.DoMethod(id, "Paste")
```

BESCHREIBUNG

Fügt den Text aus der Zwischenablage in das Zeichenketten-Gadget ein.

Diese Methode benötigt MUI 4.0 oder besser.

EINGABEN

id ID des Zeichenketten-Objekts

45.14 String.Redo

BEZEICHNUNG

String.Redo – hebt ein vorangehendes Undo auf (V1.3)

ÜBERSICHT

```
mui.DoMethod(id, "Redo")
```

BESCHREIBUNG

Mit Redo können Sie ein vorangegangenes Undo des Zeichenketten-Gadgets wieder aufheben.

Diese Methode benötigt MUI 4.0 oder besser.

EINGABEN

id ID des Zeichenketten-Objekts

45.15 String.Reject

BEZEICHNUNG

String.Reject – setzt/ermittelt Zeichen, die von Zeichenketten-Gadgets abgelehnt werden

BESCHREIBUNG

Eine Zeichenkette, die Zeichen enthält, die nicht als Eingabe für das Zeichenketten-Gadget akzeptiert werden sollen. Immer wenn der Benutzer ein solches Zeichen eintippt, hört er einen Signalton und der Gadget-Inhalt ändert sich nicht.

TYP

Zeichenkette

ANWENDBARKEIT

ISG

BEISPIEL

```
mui.Set(obj, "Reject", "0123456789")
```

Der obige Code wird ein Zeichenketten-Gadget zum ablehnen von Zahlen festlegen.

45.16 String.Secret

BEZEICHNUNG

String.Secret – blendet Benutzereingaben aus

BESCHREIBUNG

Dieses Attribut bewirkt, dass das Zeichenketten-Gadget nur Punkte anstelle des tatsächlichen Inhalts anzeigt. Nützlich für das Eintippen von Passwörtern.

TYP

Boolesch

ANWENDBARKEIT

IG

45.17 String.Undo

BEZEICHNUNG

String.Undo – macht die letzte Operation rückgängig (V1.3)

ÜBERSICHT

```
mui.DoMethod(id, "Undo")
```

BESCHREIBUNG

Macht die letzte Operation des Zeichenketten-Gadgets rückgängig.

Diese Methode benötigt MUI 4.0 oder besser.

EINGABEN

id ID des Zeichenketten-Objekts

46 Text-Klasse

46.1 Übersicht

Die Text-Klasse ermöglicht das Erzeugen von Objekten, die irgendeine Art von Text enthalten. Sie können das Outfit Ihres Textes mit einigen speziellen Steuerzeichen steuern, einschließlich Kursiv-, Fett-, Unterstreichungs- und Farbcodes. Format-Codes richten Text entweder nach links, zentriert oder rechts aus, Zeilenumbrüche ermöglichen mehrzeilige Textfelder.

Bitte beachten Sie, dass die Text-Klasse keinen automatischen Wortumbruch bietet. Wenn Sie den Wortumbruch on-the-fly haben wollen, müssen Sie stattdessen die Floattext-Klasse (Fließtext) verwenden. Siehe [Abschnitt 15.1 \[Fließtext\]](#), [Seite 97](#), für Details.

Im Folgenden finden Sie ein Beispiel, wie Sie `<text>` als XML-Definition verwenden:

```
<text>Hello World</text>
```

Hier ist das gleiche Beispiel in fett:

```
<text>\33bHello World</text>
```

Die Zeichenkette, die Sie hier angeben, kann Textformatierungs-codes verwenden. Siehe [Abschnitt 3.9 \[Textformatierungs-codes\]](#), [Seite 14](#), für Details.

46.2 Text.Contents

BEZEICHNUNG

Text.Contents – setzt/ermittelt den Inhalt des Text-Objekts

BESCHREIBUNG

Zeichenkette, die in einem Text-Objekt angezeigt werden soll oder ermittelt sie.

Wenn die Zeichenkette größer als der verfügbare Anzeigebereich ist, wird sie abgeschnitten. Das Setzen von `Text.Contents` auf "" führt zu einem leeren Text-Objekt.

Die Zeichenkette, die Sie hier angeben, kann Textformatierungs-codes verwenden. Siehe [Abschnitt 3.9 \[Textformatierungs-codes\]](#), [Seite 14](#), für Details.

TYP

Zeichenkette

ANWENDBARKEIT

SG

46.3 Text.HiChar

BEZEICHNUNG

Text.HiChar – hebt ein einzelnes Zeichen hervor

BESCHREIBUNG

Wenn das hier angegebene Zeichen in der angezeigten Zeichenkette vorhanden ist (egal ob Groß- oder Kleinschreibung), wird es unterstrichen. Dies ist nützlich in Verbindung mit `Area.ControlChar`.

TYP

Einzelne Zeichenkette

ANWENDBARKEIT

I

46.4 Text.PreParse

BEZEICHNUNG

Text.PreParse – setzt/ermittelt die Vorbereitungs-Zeichenkette für das Text-Objekt

BESCHREIBUNG

Diese Zeichenkette enthält die Formatdefinitionen, die analysiert werden sollen, bevor der Text von `Text.Contents` gedruckt wird.

Mit diesem Tag können Sie problemlos verschiedene Formate, Farben und Stile definieren, ohne die ursprüngliche Zeichenkette zu ändern. Wenn Sie beispielsweise `Text.PreParse` auf `"\33c"` setzen, wird der Text im Text-Objekt immer zentriert.

Die Zeichenkette, die Sie hier angeben, kann Textformatierungscodes verwenden. Siehe [Abschnitt 3.9 \[Textformatierungscodes\]](#), [Seite 14](#), für Details.

TYP

Zeichenkette

ANWENDBARKEIT

ISG

46.5 Text.SetMax

BEZEICHNUNG

Text.SetMax – begrenzt die maximale Breite

BESCHREIBUNG

Boolescher Wert, der angibt, ob die maximale Breite des Objekts so berechnet werden soll, dass sie zu der mit `Text.Contents` angegebenen Zeichenkette passt.

Bei `False` ist die maximale Breite nicht begrenzt.

Für ein Text-Objekt, das aktualisiert werden muss (z.B. einige Informationen über den Status Ihres Programms), würden Sie wahrscheinlich `Text.SetMax` auf `False` setzen, um die Größenänderung dieses Objekts zu erlauben.

Für eine Beschriftung für eines Ihrer Gadgets möchten Sie diesen Tag möglicherweise den Wert `True` zuweisen, um zu verhindern, dass MUI zusätzlichen Layoutbereich einfügt.

Standardeinstellung ist `False`.

TYP

Boolesch

ANWENDBARKEIT

I

46.6 Text.SetMin

BEZEICHNUNG

Text.SetMin – legt eine Mindestbreite für das Text-Objekt fest

BESCHREIBUNG

Boolescher Wert, der angibt, ob die minimale Breite des Objekts so berechnet werden soll, dass sie zu der mit `Text.Contents` angegebenen Zeichenkette passt.

Bei der Einstellung `False` wird die minimale Breite auf 0 gesetzt und die angezeigte Zeichenkette wird möglicherweise abgeschnitten.

Standardeinstellung ist `True`.

TYP

Boolesch

ANWENDBARKEIT

I

46.7 Text.SetVMax

BEZEICHNUNG

Text.SetVMax – begrenzt die Objekthöhe

BESCHREIBUNG

Wenn Sie dies auf `False` setzen, wird die Text-Objekt-Y-Größe unbegrenzt. Der Standardwert ist `True`, was bedeutet, dass die Höhe der Objekte festgelegt ist.

TYP

Boolesch

ANWENDBARKEIT

I

47 Texteditor-Klasse

47.1 Übersicht

`TextEditor.mcc` ist ein mehrzeiliges Zeichenketten-Gadget, das die meisten Funktionen eines normalen Texteditors enthält, einschließlich einiger speziellen Funktionen zum Importieren/Exportieren von Text aus/in spezielle Formate.

Das Gadget wurde ursprünglich im Jahr 1997 von Allan Odgaard geschrieben. Ab der Version 15.10, die im April 2005 veröffentlicht wurde, wird das `TextEditor.mcc` Gadget vom Open Source-Team verwaltet.

Es wird veröffentlicht und unter den Bedingungen der GNU Lesser General Public Licence (LGPL) veröffentlicht und ist kostenlos erhältlich.

Bitte besuchen Sie <http://www.sf.net/projects/texteditor-mcc/> für die neueste Version und Informationen bezüglich des `TextEditor.mcc`.

Im Folgenden finden Sie ein XML-Beispiel für die Einbindung eines Texteditor-Objekts in Ihre GUI:

```
<texteditor scrollbar="true" contents="Enter your text here!"/>
```

47.2 Texteditor.ActiveObjectOnClick

BEZEICHNUNG

`Texteditor.ActiveObjectOnClick` – wird automatisch zum aktiven Fenster-Objekt

BESCHREIBUNG

Mit diesem Tag können Sie festlegen, ob das Texteditor-Objekt automatisch das Objekt `Window.ActiveObject` des Fensters setzt, wenn der Benutzer im Texteditor eine Mausektion ausführt (klicken/auswählen). Als Konsequenz erhält der Texteditor dann den vollen Tastaturfokus und signalisiert schließlich seinen aktiven Status, indem er einen markierten Bereich mit der aktiven Farbe markiert, anstatt die inaktiv markierte Farbe im inaktiven Zustand zu verwenden.

Bitte beachten Sie, dass dieses Attribut abhängig von der Einstellung `Texteditor.ReadOnly` andere Standardwerte hat. Wenn der Texteditor auf `Texteditor.ReadOnly` eingestellt ist, wird dieses Attribut standardmäßig auf `False` gesetzt. Wenn sich das Objekt dagegen im vollständigen Schreibmodus befindet, wird es stattdessen auf `True` gesetzt. Während des Betriebs kann dieses Verhalten jedoch außer Kraft gesetzt werden, indem dieses Attribut auf einen anderen Wert gesetzt wird.

TYP

Boolesch

ANWENDBARKEIT

ISG

47.3 Texteditor.Align

BEZEICHNUNG

Texteditor.Align – setzt/ermittelt die aktuelle Ausrichtung des Absatzes

BESCHREIBUNG

Legt die aktuelle Ausrichtung des Absatzes fest oder ermittelt sie.

Wenn beim Festlegen dieses Attributs ein Bereich markiert ist, wird die neue Ausrichtung für den gesamten Bereich festgelegt.

Folgende Werte sind möglich:

Left Links ausgerichtet

Right Rechts ausgerichtet

Center Zentriert ausgerichtet

Justified

Als Blocksatz ausgerichtet (noch nicht implementiert).

TYP

Zeichenkette (siehe oben für mögliche Werte)

ANWENDBARKEIT

SGN

47.4 Texteditor.AreaMarked

BEZEICHNUNG

Texteditor.AreaMarked – ermittelt Informationen über markierte Bereiche

BESCHREIBUNG

Dieser Tag wird auf **True** gesetzt, wenn Text markiert ist, und wieder auf **False**, wenn nichts markiert ist. Sie können mit diesem Tag ein Benachrichtigungs-Ereignis erstellen und Ihre Ausschneiden-/Kopieren-Schaltflächen deaktiviert darstellen lassen, wenn nichts markiert ist.

TYP

Boolesch

ANWENDBARKEIT

GN

47.5 Texteditor.AutoClip

BEZEICHNUNG

Texteditor.AutoClip – aktiviert/deaktiviert die automatische Kopie in die Zwischenablage

BESCHREIBUNG

Wenn sich das Gadget im schreibgeschützten Modus befindet und der Benutzer einen Text markiert, wird er automatisch in die Zwischenablage kopiert. Mit diesem Tag können Sie dieses Verhalten deaktivieren. Aber denken Sie zweimal darüber nach, denn die konfigurierte Taste 'Kopieren' (mit der der Benutzer normalerweise Text kopiert) funktioniert nur, wenn das Gadget aktiv ist (oder voreingestellt), das nicht automatisch im Nur-Lese-Modus (beim Anklicken) erscheint.

TYP

Boolesch

ANWENDBARKEIT

ISG

47.6 Texteditor.Clear

BEZEICHNUNG

Texteditor.Clear – löscht den Text

ÜBERSICHT

```
mui.DoMethod(id, "Clear")
```

BESCHREIBUNG

Dadurch wird der gesamte Text im Gadget gelöscht.

EINGABEN

id ID des Text-Editor-Objekts

47.7 Texteditor.Color

BEZEICHNUNG

Texteditor.Color – setzt/ermittelt die aktuelle Textfarbe (V1.2)

BESCHREIBUNG

Mit diesem Attribut kann die aktuelle Textfarbe festgelegt/abgerufen werden. Jeder Text, der nach dem Festlegen dieses Attributs eingegeben wird, wird in der angegebenen Farbe angezeigt. Um die Farbe des vorhandenen Textes zu ändern, verwenden Sie stattdessen die Methode `Texteditor.SetColor`.

Sie können für dieses Attribut auch eine Benachrichtigung einrichten, um zu erfahren, wann der Cursor über einen Text in einer anderen Farbe verschoben wurde.

Aus dem Hollywood-Skript wird die Farbe als einfacher numerischer Wert angegeben, der 8 Bits für jede Komponente enthält. Wenn Sie die Farbe in der XML-Datei angeben, muss sie als 6-stellige Zeichenkette mit dem Präfix `#` übergeben werden (genau wie in HTML).

TYP

Nummer

ANWENDBARKEIT

SGN

47.8 Texteditor.ColorMap**BEZEICHNUNG**

Texteditor.ColorMap – ermittelt die aktuelle Farbpalette des Texteditors (V1.2)

BESCHREIBUNG

Dieses Attribut kann verwendet werden, um die aktuelle Farbpalette des Editors zu erhalten. Da die Texteditor-Klasse immer noch palettenbasiert ist, wird immer eine Tabelle mit 256 Einträgen zurückgegeben, die beschreiben, welche Farben den Stiften des Texteditors zugeordnet sind. Dies ist erforderlich, wenn Sie Farben identifizieren möchten, die in einem von `Texteditor.Contents` zurückgegebenen Text verwendet werden. Wenn ein Text unterschiedliche Farben verwendet, erhalten Sie nur die Informationen über die verschiedenen Stifte, die vom Text verwendet werden. Sie können diese Tabelle dann verwenden, um den Stiftinformationen RGB-Farben zuzuordnen.

TYP

Tabelle

ANWENDBARKEIT

G

47.9 Texteditor.Columns**BEZEICHNUNG**

Texteditor.Columns – stellt die gewünschte Gadget-Breite ein

BESCHREIBUNG

Stellt die gewünschte Breite in Zeichen ein.

TYP

Nummer

ANWENDBARKEIT

I

47.10 Texteditor.Contents**BEZEICHNUNG**

Texteditor.Contents – setzt/ermittelt den Inhalt des Texteditors

BESCHREIBUNG

Verwenden Sie dieses Attribut, um den Inhalt des Texteditor-Objekts festzulegen oder abzurufen.

Die Zeichenkette, die Sie hier angeben, kann Textformatierungscodes verwenden. Siehe [Abschnitt 3.9 \[Textformatierungscodes\]](#), [Seite 14](#), für Details.

TYP

Zeichenkette

ANWENDBARKEIT

ISG

47.11 Texteditor.ConvertTabs

BEZEICHNUNG

Texteditor.ConvertTabs – konvertiert Tabulatoren in Leerzeichen

BESCHREIBUNG

Wenn **True** (Standard), wird TextEditor Tabulatoren (\t) in die Anzahl der durch die Benutzerkonfigurationseinstellung angegebenen Leerzeichen konvertieren. Wenn **False**, setzt TextEditor stattdessen reine \t Zeichen und zeigt dem Benutzer nur Leerzeichen an.

Bitte beachten Sie, dass eine Änderung dieses Attributs von **True** auf **False** zu einer Aktualisierung des Editor-Gadgets führt (die Cursorkoordinaten werden auf Null gesetzt).

TYP

Boolesch

ANWENDBARKEIT

ISG

47.12 Texteditor.Copy

BEZEICHNUNG

Texteditor.Copy – kopiert den markierten Text

ÜBERSICHT

```
mui.DoMethod(id, "Copy")
```

BESCHREIBUNG

Kopiert den aktuell ausgewählten Text in die Zwischenablage.

EINGABEN

id ID des Text-Editor-Objekts

47.13 Texteditor.CursorX

BEZEICHNUNG

Texteditor.CursorX – setzt/ermittelt die x-Position des Cursors

BESCHREIBUNG

Sie können die x-Position des Cursors mit diesem Tag ermitteln oder setzen. Das erste Zeichen in einer Zeile hat die Position 0. Die Position wird nicht durch die automatische Zeilenumbruchfunktion des Gadgets beeinflusst. Wenn Sie einen Wert festlegen, der höher ist als die Länge der aktuellen Zeile, wird sie automatisch abgeschnitten.

TYP

Nummer

ANWENDBARKEIT

ISGN

47.14 Texteditor.CursorY**BEZEICHNUNG**

Texteditor.CursorY – setzt/ermittelt die y-Position des Cursors

BESCHREIBUNG

Sie können die y-Position des Cursors mit diesem Tag ermitteln oder setzen. Die erste Zeile hat die Position 0. Die Position wird nicht durch die automatische Zeilenumbruchfunktion des Gadgets beeinflusst. Wenn Sie einen höheren Wert als die Anzahl der Zeilen festlegen, wird diese automatisch abgeschnitten.

TYP

Nummer

ANWENDBARKEIT

ISGN

47.15 Texteditor.Cut**BEZEICHNUNG**

Texteditor.Cut – schneidet den markierten Text aus

ÜBERSICHT

```
mui.DoMethod(id, "Cut")
```

BESCHREIBUNG

Schneidet den ausgewählten Text aus und legt ihn in die Zwischenablage.

EINGABEN

id	ID des Text-Editor-Objekts
----	----------------------------

47.16 Texteditor.Erase**BEZEICHNUNG**

Texteditor.Erase – löscht den gesamten Text

ÜBERSICHT

```
mui.DoMethod(id, "Erase")
```

BESCHREIBUNG

Löscht den gesamten Text im Texteditor-Gadget.

EINGABEN

id	ID des Text-Editor-Objekts
----	----------------------------

47.17 Texteditor.ExportHook

BEZEICHNUNG

Texteditor.ExportHook – setzt das Exportverfahren des Texteditors (V1.4)

BESCHREIBUNG

Abhängig von den Eingaben wird ein anderes Exportverfahren ausgelöst, sobald Text aus dem Texteditor-Gadget exportiert wird. Standardmäßig werden die aktuell angezeigten Daten unverändert exportiert. Das bedeutet, dass der gesamte Text, einschließlich der Escape-Sequenzen zum Anzeigen diverser Stile, exportiert wird.

Sie können dieses Verhalten ändern, indem Sie dieses Attribut festlegen. Die folgenden Exportverfahren werden erkannt:

Plain	Exportiert den gesamten angezeigten Text einschließlich der Escape-Sequenzen (<code>'\33'</code>). Dies ist der Standardmodus.												
EMail	Exportiert den gesamten Text, aber konvertiert die stilrelevanten Escape-Sequenzen in die Pseudo-Standard-Textsequenzen wie: <table style="margin-left: 2em;"> <tr> <td><code>*bold*</code></td> <td>: für fetten Text</td> </tr> <tr> <td><code>/italic/</code></td> <td>: für kursiven Text</td> </tr> <tr> <td><code>_underline_</code></td> <td>: für unterstrichenen Text</td> </tr> <tr> <td><code>#colored#</code></td> <td>: für farbigen/hervorgehobenen Text</td> </tr> <tr> <td><code><tsb></code></td> <td>: für eine dicke Trennleiste</td> </tr> <tr> <td><code><sb></code></td> <td>: für eine dünne Trennleiste</td> </tr> </table>	<code>*bold*</code>	: für fetten Text	<code>/italic/</code>	: für kursiven Text	<code>_underline_</code>	: für unterstrichenen Text	<code>#colored#</code>	: für farbigen/hervorgehobenen Text	<code><tsb></code>	: für eine dicke Trennleiste	<code><sb></code>	: für eine dünne Trennleiste
<code>*bold*</code>	: für fetten Text												
<code>/italic/</code>	: für kursiven Text												
<code>_underline_</code>	: für unterstrichenen Text												
<code>#colored#</code>	: für farbigen/hervorgehobenen Text												
<code><tsb></code>	: für eine dicke Trennleiste												
<code><sb></code>	: für eine dünne Trennleiste												
NoStyle	Exportiert alle Texte wie die Plain Variante, aber entfernt alle stilrelevanten Escape-Sequenzen. Konvertiert auch die Escape-Sequenzen für den dicken und dünnen Trennbalken in <code><tsb></code> und <code><sb></code> wie das EMail -Verfahren.												

Siehe [Abschnitt 47.23 \[Texteditor.ImportHook\]](#), Seite 248, für Details.

TYP

Zeichenkette (siehe oben für mögliche Werte)

ANWENDBARKEIT

IS

47.18 Texteditor.ExportWrap

BEZEICHNUNG

Texteditor.ExportWrap – verwendet den harten Wortumbruch

BESCHREIBUNG

Dieses Attribut ermöglicht es dem integrierten Exportverfahren, während des Exportierens von Text einen harten Wortumbruch auszuführen. Null bedeutet kein Umbruch (Standardwert). Siehe [Abschnitt 47.51 \[Texteditor.WrapMode\]](#), Seite 260, für Informationen zu den Umbruchsarten.

Bitte beachten Sie, dass `Texteditor.WrapMode` keine Auswirkungen auf die Funktionsweise der Exportfunktion hat. Das bedeutet, dass das Setzen von `Texteditor.ExportWrap` immer mit einem harten Zeilenumbruch endet.

TYP

Nummer

ANWENDBARKEIT

ISG

47.19 Texteditor.FixedFont**BEZEICHNUNG**

Texteditor.FixedFont – verwendet eine Schriftart mit fester Breite

BESCHREIBUNG

Legt fest, dass der Editor eine Schriftart mit fester Breite verwenden soll.

TYP

Boolesch

ANWENDBARKEIT

IG

47.20 Texteditor.GetSelection**BEZEICHNUNG**

Texteditor.GetSelection – ermittelt die Blockauswahl

ÜBERSICHT

```
x1, y1, x2, y2 = mui.DoMethod(id, "GetSelection")
```

BESCHREIBUNG

Diese Methode gibt die Start- und Stoppposition des aktuell markierten Blocks zurück.
Wenn es keinen markierten Bereich gibt, wird -1 in allen vier Werten zurückgegeben.

EINGABEN

id ID des Text-Editor-Objekts

RÜCKGABEWERTE

x1 Start-x-Position des markierten Blocks
y1 Start-y-Position des markierten Blocks
x2 Stopp-x-Position des markierten Blocks
y2 Stopp-y-Position des markierten Blocks

47.21 Texteditor.GetText**BEZEICHNUNG**

Texteditor.GetText – exportiert Teile des aktuellen Textes (V1.1)

ÜBERSICHT

```
t$ = mui.DoMethod(id, "GetText", x1, y1, x2, y2)
```

BESCHREIBUNG

Diese Methode exportiert Teile des aktuellen Textes und gibt sie zurück. Die Methode akzeptiert vier Koordinaten, mit denen Sie einen Textblock abrufen können, ohne dass er markiert ist. Dies ist nützlich, um einen Block zu analysieren, ohne dass der Benutzer ein ähnliches Verhalten wie beim Markieren eines Blocks sieht.

EINGABEN

id	ID des Text-Editor-Objekts
x1	Start-x-Position des gewünschten Blocks
y1	Start-y-Position des gewünschten Blocks
x2	Stopp-x-Position des gewünschten Blocks
y2	Stopp-y-Position des gewünschten Blocks

RÜCKGABEWERTE

t\$	Text an den angegebenen Blockkoordinaten
-----	--

47.22 Texteditor.HasChanged

BEZEICHNUNG

Texteditor.HasChanged – benachrichtigt über Inhaltsänderungen

BESCHREIBUNG

Dieser Tag zeigt an, wenn sich der Inhalt des Gadgets geändert hat. Sie können eine Benachrichtigung für diesen Tag verwenden, um es mit einem Auswahlkästchen- oder einem Text-Objekt zu verbinden.

Sie sollten diesen Tag auf **False** setzen, wenn Sie den Inhalt des Gadgets exportieren oder mit etwas Neuem überschreiben.

Selbst wenn Sie eine Benachrichtigung für diesen Tag eingerichtet haben, sollten Sie diese immer noch überprüfen, bevor Sie den Text löschen. Dies ermöglicht es Ihnen, einige erweiterte Tests durchzuführen, um zu sehen, ob der Text tatsächlich geändert wurde, z.B. durch Überprüfen des Undo-Puffers, Vergleichen von Prüfsummen oder die Überprüfung, ob der Textpuffer leer ist oder nicht (nichts davon wird derzeit durchgeführt, aber es ist vielleicht zukünftig möglich).

TYP

Boolesch

ANWENDBARKEIT

ISGN

47.23 Texteditor.ImportHook

BEZEICHNUNG

Texteditor.ImportHook – legt das Importverfahren des Texteditors fest (V1.4)

BESCHREIBUNG

Da dieses Gadget verschiedene Textstile zulässt, können Sie ein Importverfahren wählen, um den Text korrekt zu analysieren.

Das Standard-Importverfahren versteht die folgenden Escape-Sequenzen. Sie können an jeder Position innerhalb der Zeile erscheinen:

`\33u` Stellt den Stil zur Unterstreichung ein.

`\33b` Stellt den Stil auf fett ein.

`\33i` Stellt den Stil auf kursiv ein.

`\33n` Stellt den Stil wieder auf normal ein.

`\33h` Hebt die aktuellen Zeile hervor.

`\33p[x]` Wechselt zu Farbe *x*, wobei *x* aus der Farbpalette übernommen wird. 0 bedeutet normal. Die Farbe wird für jede neue Zeile zurückgesetzt.

`\33P [RRGGBB]`

Ändert die Frontfarbe in die angegebene RGB-Farbe. Die RGB-Farbe muss in Form von sechs Hexadezimalziffern RRGGBB angegeben werden. Dies ist nur auf True-Color-Bildschirmen möglich. Die Farbe wird für jede neue Zeile zurückgesetzt.

`\33P [AARRGGBB]`

Ändert die Frontfarbe in die angegebene RGB-Farbe und wendet Alpha-Mischung mit der angegebenen Intensität an. Die RGB-Farbe muss in Form von sechs Hexadezimalziffern RRGGBB angegeben werden, denen zwei Alphakanalziffern AA vorangestellt werden müssen, die die Mischintensität angeben. Dies ist nur auf True-Color-Bildschirmen möglich. Die Farbe wird für jede neue Zeile zurückgesetzt.

Die folgenden Sequenzen sind nur am Anfang einer Zeile gültig. Wenn sie an einer anderen Stelle platziert werden, ist das Ergebnis nicht definiert (sie werden möglicherweise ignoriert oder auch nicht):

`\33l` Linke Ausrichtung der aktuellen und folgenden Zeilen.

`\33r` Rechte Ausrichtung der aktuellen und folgenden Zeilen.

`\33c` Zentriert die aktuelle und folgende Zeilen.

`\33[s:x]` Erstellt ein Trennzeichen. *x* ist eine Bitkombination von Flags:

Bit 0 set Obere Platzierung des Trennzeichen.

Bit 1 set Mittlere Platzierung des Trennzeichen.

Bit 2 set Untere Platzierung des Trennzeichen.

Bit 3 set zieht das Trennzeichen über den Text (durchstreichen).

Bit 4 set Trennzeichen wird extra dick dargestellt.

Die folgenden Werte können an dieses Attribut übergeben werden:

Plain Den Standard-Importverfahren verwenden (siehe oben).

MIME Dieses integrierte Verfahren wandelt zitierte Ausdrücke (z.B. "=E5") in die ASCII-Darstellung um, verbindet Zeilen mit einem "=", schreibt den Text (unter Verwendung des mit `TextEditor.ImportWrap` gesetzten Wertes), hebt alle Zeilen die mit ">" beginnen hervor, macht wirklich **fetten**, */kursiven/*, *_unterstrichenen_* und *#farbigen#* Text und ersetzt `<sb>` oder `<tsb>` durch eine echte Trennlinie. Dieses Verfahren beendet das Analysieren, wenn ein NULL-Byte erreicht wird.

Die Farbe für *#farbigen#* Text ist der Farbpalette-Eintrag 6, der standardmäßig auf `MPEN_FILL` gesetzt wird. Um ihn zu überschreiben, geben Sie einfach eine Farbpalette mit dem Eintrag 6 in der gewünschten Farbe ein.

MIMEQuoted Wie das MIME-Importverfahren, aber jede Zeile wird zitiert und hervorgehoben.

Email Wie das MIME-Importverfahren, konvertiert jedoch keine zitierten Ausdrücke.

Beachten Sie, dass die letzten drei Verfahren auch die Escape-Sequenzen auswerten, die für das Import-Verfahren `Plain` beschrieben wurden. Während es nie dokumentiert wurde, gibt es einige Programme, die auf diesem Nebeneffekt einer alten Implementierung bestehen. Daher musste dieses Verhalten ab Version 51.13 offiziell unterstützt werden.

TYP

Zeichenkette (siehe oben für mögliche Werte)

ANWENDBARKEIT

IS

47.24 Texteditor.ImportWrap

BEZEICHNUNG

`Texteditor.ImportWrap` – führt beim Importieren einen automatischen Zeilenumbruch durch

BESCHREIBUNG

Mit diesem Attribut können die integrierten Import-Verfahren beim Importieren von Text den automatischen Zeilenumbruch ausführen.

Die integrierten Verfahren akzeptieren einen Wert zwischen 4 und 1024. Der Standardwert ist 1023.

Siehe [Abschnitt 47.51 \[Texteditor.WrapMode\]](#), Seite 260, für Informationen zu den Umbruchsarten.

Bitte beachten Sie, dass `Texteditor.WrapMode` keine Auswirkungen auf die Funktionsweise der Importfunktion hat. Das bedeutet, dass das Setzen von `Texteditor.ImportWrap` immer mit einem harten Zeilenumbruch endet.

TYP

Nummer

ANWENDBARKEIT

ISG

47.25 Texteditor.Insert**BEZEICHNUNG**

`Texteditor.Insert` – fügt Text ein

ÜBERSICHT

```
mui.DoMethod(id, "Insert", t$, pos$)
```

BESCHREIBUNG

Dadurch wird der angegebene Text `t$` an der angegebenen Position eingefügt. Die Position des eingefügten Textes kann einer der folgenden sein:

`Cursor` Fügt ihn an der aktuellen Cursorposition ein.

`Top` Fügt ihn am Anfang ein.

`Bottom` Fügt ihn am Schluss ein.

EINGABEN

`id` ID des Text-Editor-Objekts

`t$` Text zum Einfügen

`pos$` Einfügeposition (siehe oben für mögliche Werte)

47.26 Texteditor.Mark**BEZEICHNUNG**

`Texteditor.Mark` – markiert den Text

ÜBERSICHT

```
mui.DoMethod(id, "Mark", startx, starty, stopx, stopy)
```

BESCHREIBUNG

Diese Methode markiert (wählt) den angegebenen Bereich, der durch das Rechteck `startx/stopx, starty/stopy` angegeben wird.

EINGABEN

`id` ID des Text-Editor-Objekts

`startx` Start-x-Position

`starty` Start-y-Position

`stopx` Stop-x-Position
`stopy` Stop-y-Position

47.27 Texteditor.MarkAll

BEZEICHNUNG

`Texteditor.MarkAll` – markiert den gesamten Text

ÜBERSICHT

```
mui.DoMethod(id, "MarkAll")
```

BESCHREIBUNG

Markiert den gesamten Text im Gadget.

EINGABEN

`id` ID des Text-Editor-Objekts

47.28 Texteditor.MarkNone

BEZEICHNUNG

`Texteditor.MarkNone` – hebt die Textmarkierung auf

ÜBERSICHT

```
mui.DoMethod(id, "MarkNone")
```

BESCHREIBUNG

Hebt die Markierung des Textes auf.

EINGABEN

`id` ID des Text-Editor-Objekts

47.29 Texteditor.Paste

BEZEICHNUNG

`Texteditor.Paste` – fügt den Text aus der Zwischenablage ein

ÜBERSICHT

```
mui.DoMethod(id, "Paste")
```

BESCHREIBUNG

Fügt Text aus der Zwischenablage in das Texteditor-Gadget ein.

EINGABEN

`id` ID des Text-Editor-Objekts

47.30 Texteditor.PasteColors

BEZEICHNUNG

Texteditor.PasteColors – ignoriert beim Einfügen die Farbinformationen

BESCHREIBUNG

Wenn Sie diesen Tag auf `False` setzen, werden alle Farbinformationen beim Einfügen eines Textausschnittes aus der Zwischenablage ignoriert.

TYP

Boolesch

ANWENDBARKEIT

ISG

47.31 Texteditor.PasteStyles

BEZEICHNUNG

Texteditor.PasteStyles – ignoriert beim Einfügen die Stilinformationen

BESCHREIBUNG

Wenn Sie diesen Tag auf `False` setzen, werden alle Stilinformationen beim Einfügen eines Textausschnittes aus der Zwischenablage ignoriert.

TYP

Boolesch

ANWENDBARKEIT

ISG

47.32 Texteditor.ReadOnly

BEZEICHNUNG

Texteditor.ReadOnly – setzt/ermittelt den Schreibschutzmodus des Texteditors

BESCHREIBUNG

Wenn Sie diesen Tag auf `True` setzen, wird der Text schreibgeschützt. Dies ist sehr ähnlich zu dem, was der `Floattext.mui` bietet, außer dass dieses Gadget das Blockieren anbietet.

Im schreibgeschützten Modus:

- Es wird keinen Cursor geben (nur einen normalen TAB-Rahmen),
- TAB aktiviert das nächste Gadget (anstelle von `RCOMMAND+TAB`),
- Der Rahmen wird auf einen Nur-Lese-Liste-Rahmen gesetzt (kann sich ändern),
- Es gibt keine `ARexx`-Unterstützung, außer "Kopieren".

TYP

Boolesch

ANWENDBARKEIT

ISG

47.33 Texteditor.Redo

BEZEICHNUNG

Texteditor.Redo – hebt ein vorangehendes Undo auf

ÜBERSICHT

```
mui.DoMethod(id, "Redo")
```

BESCHREIBUNG

Diese Methode hebt ein vorangehendes Undo in einem Texteditor-Gadget auf.

EINGABEN

id ID des Text-Editor-Objekts

47.34 Texteditor.RedoAvailable

BEZEICHNUNG

Texteditor.RedoAvailable – benachrichtigt, wenn Redo verfügbar ist

BESCHREIBUNG

Dieser Tag wird auf `True` gesetzt, wenn der Benutzer in der Lage ist, seine Aktion (normalerweise nach einem Undo) wieder herzustellen. Sie können eine Benachrichtigung für diesen Tag erstellen und Ihre Redo-Taste deaktivieren lassen, wenn nichts wieder herzustellen ist.

TYP

Boolesch

ANWENDBARKEIT

GN

47.35 Texteditor.Replace

BEZEICHNUNG

Texteditor.Replace – ersetzt den markierten Text durch eine Zeichenkette

ÜBERSICHT

```
mui.DoMethod(id, "Replace", r$)
```

BESCHREIBUNG

Diese Methode ersetzt den markierten Bereich durch die angegebene Zeichenkette `r$`.

EINGABEN

id ID des Text-Editor-Objekts

r\$ Ersatzzeichenkette

47.36 Texteditor.Rows

BEZEICHNUNG

Texteditor.Rows – stellt die gewünschte Gadget-Höhe ein

BESCHREIBUNG

Stellt die gewünschte Höhe in Anzahl Zeilen ein.

TYP

Nummer

ANWENDBARKEIT

I

47.37 Texteditor.Scrollbar

BEZEICHNUNG

Texteditor.Scrollbar – fügt dem Editor eine Bildlaufleiste hinzu

BESCHREIBUNG

Wenn Sie dieses Attribut auf `True` setzen, erhält Ihr Texteditor-Objekt eine Bildlaufleiste. Standardeinstellung ist `False`.

TYP

Boolesch

ANWENDBARKEIT

I

47.38 Texteditor.Search

BEZEICHNUNG

Texteditor.Search – sucht im Text die Zeichenkette

ÜBERSICHT

```
r = mui.DoMethod(id, "Search", s$, flags$)
```

BESCHREIBUNG

Sucht im Text nach der angegebenen Zeichenkette `s$`. Die Zeichenkette darf nicht länger als 120 Zeichen sein.

`flags$` kann eine Kombination von folgenden Flags sein:

`FromTop` Normalerweise beginnt die Suche an der Cursor-Position. Dieser Flag bewirkt, dass sie am Anfang des Textes beginnt.

`CaseSensitive`

Wenn die Groß-/Kleinschreibung beachtet werden soll, setzen Sie dieses Flag.

`Backwards`

Mit diesem Flag führt der TextEditor eine Rückwärtssuche von der Cursor-Position aus durch.

Wenn Sie mehrere der obigen Flags angeben, müssen Sie sie mit einem Semikolon trennen, z.B. "FromTop; CaseSensitive".

Wenn die Zeichenkette gefunden wird, wird sie automatisch markiert. Wenn Sie sie also ersetzen möchten, löschen Sie einfach die markierte Zeichenkette und fügen Sie die neue Zeichenkette ein.

EINGABEN

`id` ID des Text-Editor-Objekts
`s$` Zeichenkette, nach der gesucht werden soll
`flags$` eine oder mehrere der oben aufgeführten Flags

RÜCKGABEWERTE

`r` True, wenn die Zeichenkette gefunden wurde, andernfalls False

47.39 Texteditor.SetBold

BEZEICHNUNG

`Texteditor.SetBold` – schaltet den Fettstil des Textblocks um

ÜBERSICHT

`mui.DoMethod(id, "SetBold", startx, starty, stopx, stopy, flag)`

BESCHREIBUNG

Diese Methode schaltet den Fettstil in einem Textblock um, der durch vier Koordinaten (`startx`, `starty`, `stopx`, `stopy`) definiert ist. Wenn das Argument `flag` auf `True` gesetzt ist, wird dem angegebenen Textblock ein fett formatierter Stil hinzugefügt, andernfalls wird der fett formatierte Stil aus dem Textblock entfernt.

`startx`, `starty`, `stopx` und `stopy` kann entweder eine absolute Koordinate oder einer der folgenden Sonderwerte sein:

`Min` Gibt die erste Zeile oder Spalte an.
`Max` Gibt die letzte Zeile oder Spalte an.

EINGABEN

`id` ID des Text-Editor-Objekts
`startx` Start-x-Position
`starty` Start-y-Position
`stopx` Stopp-x-Position
`stopy` Stopp-y-Position
`flag` Boolesches Flag, das den Umschaltstatus festlegt

47.40 Texteditor.SetColor

BEZEICHNUNG

Texteditor.SetColor – ändert die Farbe des Textblocks (V1.2)

ÜBERSICHT

```
mui.DoMethod(id, "SetColor", startx, starty, stopx, stopy, color)
```

BESCHREIBUNG

Diese Methode ändert die Farbe eines Textblocks, der durch vier Koordinaten definiert ist (`startx`, `starty`, `stopx`, `stopy`).

`startx`, `starty`, `stopx` und `stopy` kann entweder eine absolute Koordinaten oder einer der folgenden Sonderwerte sein:

Min Gibt die erste Zeile oder Spalte an.

Max Gibt die letzte Zeile oder Spalte an.

Aus dem Hollywood-Skript wird die Farbe als einfacher numerischer Wert angegeben, der 8 Bits für jede Komponente enthält. Wenn Sie die Farbe in der XML-Datei angeben, muss sie als 6-stellige Zeichenkette mit dem Präfix `#` übergeben werden (genau wie in HTML).

Um die Farbe zu ändern, die für neuen Text verwendet werden soll, verwenden Sie stattdessen das Attribut `Texteditor.Color`.

EINGABEN

<code>id</code>	ID des Text-Editor-Objekts
<code>startx</code>	Start-x-Position
<code>starty</code>	Start-y-Position
<code>stopx</code>	Stopp-x-Position
<code>stopy</code>	Stopp-y-Position
<code>color</code>	gewünschte Farbe für den Text

47.41 Texteditor.SetItalic

BEZEICHNUNG

Texteditor.SetItalic – schaltet den Kursivstil des Textblocks um

ÜBERSICHT

```
mui.DoMethod(id, "SetItalic", startx, starty, stopx, stopy, flag)
```

BESCHREIBUNG

Diese Methode schaltet den Kursivstil auf einen Textblock um, der durch vier Koordinaten (`startx`, `starty`, `stopx`, `stopy`) definiert ist. Wenn das Argument `flag` auf `True` gesetzt ist, wird der Kursivstil zum angegebenen Textblock hinzugefügt, andernfalls wird die Kursivschrift aus dem Textblock entfernt.

`startx`, `starty`, `stopx` und `stopy` können entweder eine absolute Koordinaten oder einer der folgenden Sonderwerte haben:

Min Gibt die erste Zeile oder Spalte an.

Max Gibt die letzte Zeile oder Spalte an.

EINGABEN

id ID des Text-Editor-Objekts

startx Start-x-Position

starty Start-y-Position

stopx Stopp-x-Position

stopy Stopp-y-Position

flag Boolesches Flag, das den Umschaltstatus anzeigt

47.42 Texteditor.SetUnderline

BEZEICHNUNG

Texteditor.SetUnderline – schaltet den Unterstreichungsstil des Textblocks um

ÜBERSICHT

```
mui.DoMethod(id, "SetUnderline", startx, starty, stopx, stopy, flag)
```

BESCHREIBUNG

Diese Methode schaltet den Unterstreichungsstil in einem Textblock ein, der durch vier Koordinaten (**startx**, **starty**, **stopx**, **stopy**) definiert ist. Wenn das Argument **flag** auf **True** gesetzt ist, wird dem angegebenen Textblock ein Unterstreichungsstil hinzugefügt, andernfalls wird der Unterstreichungsstil aus dem Textblock entfernt.

startx, **starty**, **stopx** und **stopy** können entweder eine absolute Koordinaten oder einer der folgenden Sonderwerte haben:

Min Gibt die erste Zeile oder Spalte an.

Max Gibt die letzte Zeile oder Spalte an.

EINGABEN

id ID des Text-Editor-Objekts

startx Start-x-Position

starty Start-y-Position

stopx Stopp-x-Position

stopy Stopp-y-Position

flag Boolesches Flag, das den Umschaltstatus anzeigt

47.43 Texteditor.StyleBold

BEZEICHNUNG

Texteditor.StyleBold – setzt/ermittelt den Fettstil

BESCHREIBUNG

Dieser Tag zeigt an, ob der Cursor oder der Textblock fett formatiert ist oder nicht. Sie können eine Benachrichtigung für diesen Tag einrichten, um sich über Stiländerungen zu informieren. Sie können diesen Tag auf **True** oder **False** setzen, wenn Sie den Stil ändern möchten.

TYP

Boolesch

ANWENDBARKEIT

SGN

47.44 Texteditor.StyleItalic

BEZEICHNUNG

Texteditor.StyleItalic – setzt/ermittelt den Kursivstil

BESCHREIBUNG

Dieser Tag zeigt an, ob der Cursor oder Textblock kursiv steht oder nicht. Sie können eine Benachrichtigung für diesen Tag einrichten, um sich über Stiländerungen zu informieren. Sie können diesen Tag auf **True** oder **False** setzen, wenn Sie den Stil ändern möchten.

TYP

Boolesch

ANWENDBARKEIT

SGN

47.45 Texteditor.StyleUnderline

BEZEICHNUNG

Texteditor.StyleUnderline – setzt/ermittelt den Unterstreichungsstil

BESCHREIBUNG

Dieser Tag zeigt an, ob der Cursor oder Textblock unterstrichen ist oder nicht. Sie können eine Benachrichtigung für diesen Tag einrichten, um sich über Stiländerungen zu informieren. Sie können diesen Tag auf **True** oder **False** setzen, wenn Sie den Stil ändern möchten.

TYP

Boolesch

ANWENDBARKEIT

SGN

47.46 Texteditor.TabSize

BEZEICHNUNG

Texteditor.TabSize – setzt/ermittelt die TAB-Größe

BESCHREIBUNG

Dieser Tag überschreibt die globale TAB-Größe, die durch MUI-Einstellungen konfiguriert wird, wenn eine bestimmte Anzahl von Leerzeichen pro TAB durchgesetzt werden muss. Gültige Werte liegen zwischen 2 und 12 Leerzeichen pro TAB-Zeichen. Sie können den speziellen Wert `Default` übergeben, um den Wert auf den vom Benutzer konfigurierten Wert zurückzusetzen.

Verwenden Sie dieses Attribut mit Bedacht. Es widerspricht der MUI-Philosophie, wenn der Benutzer die Einstellungen nicht selber wählen kann.

TYP

Nummer oder Zeichenkette (siehe oben)

ANWENDBARKEIT

ISG

47.47 Texteditor.Undo

BEZEICHNUNG

Texteditor.Undo – Undo macht die letzte Operation rückgängig

ÜBERSICHT

```
mui.DoMethod(id, "Undo")
```

BESCHREIBUNG

Macht die letzte Operation des Texteditor-Gadgets rückgängig.

EINGABEN

id ID des Text-Editor-Objekts

47.48 Texteditor.UndoAvailable

BEZEICHNUNG

Texteditor.UndoAvailable – benachrichtigt, wenn Undo verfügbar ist

BESCHREIBUNG

Dieser Tag wird auf `True` gesetzt, wenn der Benutzer seine Aktion(en) rückgängig machen kann. Sie können eine Benachrichtigung für diesen Tag erstellen und Ihre Schaltfläche zum Rückgängigmachen deaktivieren, wenn nichts rückgängig gemacht werden kann.

TYP

Boolesch

ANWENDBARKEIT

GN

47.49 Texteditor.UndoLevels

BEZEICHNUNG

Texteditor.UndoLevels – setzt/ermittelt die verfügbaren Undo-Ebenen

BESCHREIBUNG

Mit diesem Tag kann eine Anwendung die globale Einstellung der Undo-Ebenen (Schritte/Stufen) überschreiben. Ein Wert von Null deaktiviert das Rückgängigmachen/Wiederherstellen vollständig. Beim Abrufen wird die aktuelle Anzahl der Undo-Ebenen zurückgegeben.

TYP

Nummer

ANWENDBARKEIT

ISG

47.50 Texteditor.WrapBorder

BEZEICHNUNG

Texteditor.WrapBorder – legt den automatischen Zeilenumbruch fest

BESCHREIBUNG

Dieses Attribut ermöglicht es, die Anzahl der Zeichen im Texteditor zu definieren, bei der der Text in die nächste Zeile (d.h. "Autowrap") umgebrochen werden soll.

Die Art des Umbruchs hängt jedoch vom tatsächlichen Umbruchmodus ab, der von `Texteditor.WrapMode` angegeben wurde. Darüber hinaus erlaubt `Texteditor.WrapWords` festzulegen, ob der Texteditor an Wortgrenzen umgebrochen wird oder nicht.

Standardmäßig ist der WrapBorder auf 0 gesetzt, wodurch der Umbruch deaktiviert wird. Abhängig vom eingestellten Umbruch-Modus wird der Umbruch entweder komplett ignoriert oder der Umbruch wird an den Fenstergrenzen vorgenommen, was der Standard ist.

TYP

Nummer

ANWENDBARKEIT

ISG

47.51 Texteditor.WrapMode

BEZEICHNUNG

Texteditor.WrapMode – setzt/ermittelt den Zeilenumbruchmodus

BESCHREIBUNG

Mit diesem Attribut kann die Art und Weise, wie Zeilenumbrüche ausgeführt werden, direkt gesteuert werden, wenn `Texteditor.WrapBorder` auf >0 gesetzt wurde. Es gibt

zwei wichtige Möglichkeiten, wie der Umbruch ausgeführt werden kann: "Soft wrapping" und "hard wrapping". Der weiche Umbruch bezieht sich auf den Vorgang, bei dem Zeilen nur grafisch in die nächste Zeile eingearbeitet werden, aber immer noch als einzelne Zeile betrachtet werden. Der Harte Umbruch fügt sofort ein Zeilenumbruchzeichen ein, sobald der Cursor die angegebene Anzahl von Zeichen in einer Zeile passiert hat und sie daher auf eine "harte" direkte Weise trennt.

Die folgenden Modi werden unterstützt:

- NoWrap** Der Umbruch ist vollständig deaktiviert, das Festlegen von `WrapBorder >0` hat keine Auswirkungen. Das bedeutet, dass die Größe des Fenster vom Benutzer auf z.B. sehen oder bearbeiten des gesamten Textes geändert wird.
- SoftWrap** Aktiviert den weichen Umbruch von Zeilen, wenn `WrapBorder >0` ist. Das bedeutet, dass Zeilen an der angegebenen Umrandung so umgebrochen werden, dass sie immer noch logisch eine Zeile sind. Dies ermöglicht z.B. Zeilen dynamisch neu zu verknüpfen. Dies sollte als neue bevorzugte Einstellung für neue/überarbeitete Anwendungen betrachtet werden. Wenn `WrapBorder == 0` ist, wird der weiche Umbruch an den aktuellen Fenstergrenzen ausgeführt.
- HardWrap** Ermöglicht den harten Umbruch von Zeilen, wenn `WrapBorder >0` ist. Das bedeutet, dass Zeilen an der angegebenen Umrandung umgebrochen werden, indem Sie direkt Zeilenumbrüche einfügen, sobald der Cursor die angegebene Umrandung passiert. Obwohl dieser Modus aus historischen Gründen der Standard ist, erlaubt er nicht, Zeilen direkt neu zu verketteten, wenn der Benutzer z.B. Zeichen entfernt. Wenn `WrapBorder == 0` ist, wird der weiche Umbruch an den aktuellen Fenstergrenzen ausgeführt. Dies ist der Standardwert.

Aus historischen Gründen besteht der Standardwert darin, eine Zeile umzubrechen, wenn das `WrapBorder`-Attribut auf `>0` gesetzt ist und `WrapMode` nicht geändert wird. Für neue Anwendungen sollte der Soft-Umbruch jedoch als die bevorzugte Einstellung angesehen werden, da dieser Modus intuitiver ist und es Benutzern des Texteditors ermöglicht, Zeilen direkt zu verketteten, wenn sie z.B. Zeichen entfernen, während sie schreiben.

TYP

Zeichenkette (siehe oben für mögliche Werte)

ANWENDBARKEIT

ISG

47.52 Texteditor.WrapWords

BEZEICHNUNG

`Texteditor.WrapWords` – setzt den Zeilenumbruch nur zwischen Wörtern

BESCHREIBUNG

Wenn der Wert `True` (Standard) ist, erfolgt der Zeilenumbruch an Wortgrenzen (z.B. Leer- oder Tabulatorzeichen). Bei `False` wird die Zeile des Texteditors an einem beliebigen Zeichen umgebrochen.

Die Änderung dieses Attributs führt zur Aktualisierung des Editor-Gadgets (die Cursorkoordinaten werden auf Null gesetzt).

TYP

Boolesch

ANWENDBARKEIT

ISG

48 Toolbar-Klasse (Symbolleiste)

48.1 Übersicht

Mit der Toolbar-Klasse können Sie Symbolleisten-Gadgets für Ihre Fenster erstellen. Symbolleisten können entweder als Bilder, Bilder und Text oder nur als Text angezeigt werden. Diese Klasse benötigt die benutzerdefinierte Klasse `TheBar.mcc`.

`TheBar.mcc` ist Copyright (C) 2003-2005 Alfonso Ranieri und Copyright (C) 2005-2009 TheBar Open Source Team

Sie wird unter den Bedingungen der GNU Lesser General Public Licence (LGPL) veröffentlicht und ist kostenlos erhältlich.

Bitte besuchen Sie <http://www.sf.net/projects/thebar/> für die neueste Version und Informationen zu `TheBar.mcc`.

Wenn Sie ein Symbolleisten-Gadget im XML-Code definieren, müssen Sie immer mindestens eine Symbolleistenschaltfläche hinzufügen. Dies geschieht mit `Toolbarbutton`-Klasse. Hier ist eine Beispiel-XML-Definition einer Symbolleiste mit drei Schaltflächen:

```
<toolbar horiz="true">
  <button image="1">Cut</button>
  <button image="2">Copy</button>
  <button image="3">Paste</button>
</toolbar>
```

In der obigen XML-Definition verwendet der Symbolleistenschaltfläche 1 den Hollywood-Pinsel 1 als Bild, der Knopf 2 den Pinsel 2 und so weiter. Symbolleistenschaltflächen können viele weitere Optionen wie spezielle Bilder für ausgewählte und deaktivierte Zustände, Hilfe-Sprechblasen und mehr verwenden. Siehe [Abschnitt 49.1 \[Toolbarbutton-Klasse\]](#), [Seite 279](#), für Details.

48.2 Toolbar.Active

BEZEICHNUNG

`Toolbar.Active` – setzt/ermittelt die aktive Schaltfläche in gegenseitigen Ausschlussgruppen

BESCHREIBUNG

Mit diesen Attributen kann die ID der aktuell aktiven Schaltfläche in der Symbolleiste abgefragt oder gesetzt werden.

Beachten Sie, dass Sie viele gegenseitige Ausschlussgruppen haben können. Dieses Attribut enthält immer die ID der zuletzt ausgewählten Schaltfläche.

TYP

Nummer

ANWENDBARKEIT

ISGN

48.3 `Toolbar.BarPos`

BEZEICHNUNG

`Toolbar.BarPos` – setzt/ermittelt die Schaltflächenausrichtung in der Symbolleiste

BESCHREIBUNG

Dieses Attribut definiert die Ausrichtung der Schaltflächen in einer Leiste. Dies kann eine von folgenden sein:

`Left` Linke Ausrichtung. Dies ist auch die Standardeinstellung.

`Center` Zentrierte Ausrichtung.

`Right` Rechte Ausrichtung.

TYP

Zeichenkette (siehe oben für mögliche Werte)

ANWENDBARKEIT

ISG

48.4 `Toolbar.BarSpacer`

BEZEICHNUNG

`Toolbar.BarSpacer` – stellt die Zwischenräume als Balkenabstandshalter dar

BESCHREIBUNG

Wenn dieses Attribut `True` ist, wird jeder leerer Abstandshalter, der Gruppen von Schaltflächen voneinander optisch trennt, zu einem Balkenabstandshalter. Standardeinstellung ist `False`.

TYP

Boolesch

ANWENDBARKEIT

ISG

48.5 `Toolbar.BarSpacerSpacing`

BEZEICHNUNG

`Toolbar.BarSpacerSpacing` – stellt den Abstand zwischen Balken und Schaltfläche ein

BESCHREIBUNG

Definiert die Pixel zwischen einem Balkenabstandshalter und den Schaltflächen links/rechts. Akzeptiert wird der Bereich von $0 \leq x \leq 16$.

Dies überschreibt die Benutzereinstellungen und muss daher mit Bedacht verwendet werden. Im Zweifel benutzen Sie dies nicht.

TYP

Nummer

ANWENDBARKEIT

I

48.6 Toolbar.Borderless**BEZEICHNUNG**

Toolbar.Borderless – gestaltet die Symbolleiste randlos

BESCHREIBUNG

Wenn dieses Attribut `True` ist, erhalten Sie randlose Schaltflächen. Standardeinstellung ist `False`.

Dies überschreibt die Benutzereinstellungen und muss daher mit Bedacht verwendet werden. Im Zweifel benutzen Sie dies nicht.

TYP

Boolesch

ANWENDBARKEIT

ISG

48.7 Toolbar.BottomBarFrameSpacing**BEZEICHNUNG**

Toolbar.BottomBarFrameSpacing – stellt den Abstand zum unteren Rahmen ein

BESCHREIBUNG

Definiert die Pixel zwischen dem Balken und dem unteren Rahmen. Akzeptiert wird der Bereich von $0 \leq x \leq 16$.

Dies überschreibt die Benutzereinstellungen und muss daher mit Bedacht verwendet werden. Im Zweifel benutzen Sie dies nicht.

TYP

Nummer

ANWENDBARKEIT

I

48.8 Toolbar.BottomInnerSpacing**BEZEICHNUNG**

Toolbar.BottomInnerSpacing – stellt den Abstand vom Inhalt zum unteren Rahmen ein

BESCHREIBUNG

Definiert die Pixel zwischen dem Inhalt einer Schaltfläche und ihrem unteren Rahmen. Akzeptiert wird der Bereich von $0 < x \leq 16$.

Dies überschreibt die Benutzereinstellungen und muss daher mit Bedacht verwendet werden. Im Zweifel benutzen Sie dies nicht.

TYP

Boolesch

ANWENDBARKEIT

I

48.9 Toolbar.Columns

BEZEICHNUNG

Toolbar.Columns – setzt/ermittelt die Symbolleistenspalten

BESCHREIBUNG

Dieses Attribut definiert die Anzahl der Spalten der Symbolleiste. Wenn dieses Attribut auf einen Wert größer als 0 gesetzt wird, wird die Symbolleiste in einem sogenannten "Spaltenmodus" ausgeführt. Der Standardwert ist 0.

TYP

Nummer

ANWENDBARKEIT

ISG

48.10 Toolbar.DisMode

BEZEICHNUNG

Toolbar.DisMode – stellt den deaktivierten Modus ein oder ermittelt ihn

BESCHREIBUNG

Ermöglicht festzulegen, wie deaktivierte Schaltflächen gezeigt werden sollen, wenn kein eigener Bildsatz dafür verfügbar ist.

Folgende Werte sind möglich:

Shape Zeichnet nur eine S/W-Form, um den deaktivierten Status anzuzeigen.

Grid Zeichnet ein Lichtgitter vor dem Bild.

FullGrid Zeichnet ein komplettes Gitter.

Sunny Konvertiert das Bild in ein Graustufenbild und zeigt es an.

Blend Das deaktivierte Bild wird irgendwie gemischt. Es erfordert ein farbgetreues Bild mit einem aktiven Alphakanal und einem System, das es darstellen kann.

BlendGray Wie oben, aber als grau dargestellt.

Dies überschreibt die Benutzereinstellungen und muss daher mit Bedacht verwendet werden. Im Zweifel benutzen Sie dies nicht.

TYP

Zeichenkette (siehe oben für mögliche Werte)

ANWENDBARKEIT

ISG

48.11 Toolbar.DontMove**BEZEICHNUNG**

Toolbar.DontMove – erstellt eine statische Symbolleiste

BESCHREIBUNG

Wenn dieses Attribut **True** ist, wird der Inhalt der Schaltflächen nicht verschoben, wenn sie aktiv sind. Standardeinstellung ist **False**.

Dies überschreibt die Benutzereinstellungen und muss daher mit Bedacht verwendet werden. Im Zweifel benutzen Sie dies nicht.

TYP

Boolesch

ANWENDBARKEIT

I

48.12 Toolbar.EnableKeys**BEZEICHNUNG**

Toolbar.EnableKeys – aktiviert die Tastaturkürzel

BESCHREIBUNG

Wenn dieses Attribut auf **True** gesetzt ist, sind die Tastaturkürzel aktiviert. Tastaturkürzel werden durch Voranstellen der Taste mit einem "-" im Schaltflächentext definiert. Der Standardwert ist **True**.

Dies überschreibt die Benutzereinstellungen und muss daher mit Bedacht verwendet werden. Im Zweifel benutzen Sie dies nicht.

TYP

Boolesch

ANWENDBARKEIT

ISG

48.13 Toolbar.Frame**BEZEICHNUNG**

Toolbar.Frame – zeichnet einen Rahmen um die Symbolleiste

BESCHREIBUNG

Wenn dieses Attribut **True** ist, wird ein Rahmen um die Symbolleiste gezogen. Standardeinstellung ist **False**.

Dies überschreibt die Benutzereinstellungen und muss daher mit Bedacht verwendet werden. Im Zweifel benutzen Sie dies nicht.

TYP

Boolesch

ANWENDBARKEIT

ISG

48.14 **Toolbar.Free**

BEZEICHNUNG

Toolbar.Free – ermöglicht das Ändern der Größe der Symbolleiste

BESCHREIBUNG

Wenn dieses Attribut `True` ist, darf die Symbolleiste sowohl in vertikaler als auch in horizontaler Richtung die Größe verändern.

Der Standardwert ist:

- Wenn `Toolbar.Horiz True` ist, kann die Symbolleiste in horizontaler Richtung die Größe verändern, ist jedoch vertikal begrenzt.
- Wenn `Toolbar.Horiz False` ist, kann die Symbolleiste in vertikaler Richtung die Größe ändern, ist jedoch horizontal begrenzt.

TYP

Boolesch

ANWENDBARKEIT

ISG

48.15 **Toolbar.FreeHoriz**

BEZEICHNUNG

Toolbar.FreeHoriz – ermöglicht die horizontale Größenanpassung der Symbolleiste

BESCHREIBUNG

Wenn dieses Attribut `True` ist, darf die Symbolleiste in horizontaler Richtung die Größe verändern.

Der Standardwert ist:

- Wenn `Toolbar.Horiz True` ist, kann die Symbolleiste in horizontaler Richtung die Größe verändern, ist jedoch vertikal begrenzt.
- Wenn `Toolbar.Horiz False` ist, kann die Symbolleiste in vertikaler Richtung die Größe ändern, ist jedoch horizontal begrenzt.

TYP

Boolesch

ANWENDBARKEIT

ISG

48.16 `Toolbar.FreeVert`

BEZEICHNUNG

`Toolbar.FreeVert` – ermöglicht die vertikale Größenanpassung der Symbolleiste

BESCHREIBUNG

Wenn dieses Attribut `True` ist, darf die Symbolleiste in vertikaler Richtung die Größe verändern.

Der Standardwert ist:

- Wenn `Toolbar.Horiz True` ist, kann die Symbolleiste in horizontaler Richtung die Größe verändern, ist jedoch vertikal begrenzt.
- Wenn `Toolbar.Horiz False` ist, kann die Symbolleiste in vertikaler Richtung die Größe ändern, ist jedoch horizontal begrenzt.

TYP

Boolesch

ANWENDBARKEIT

ISG

48.17 `Toolbar.Horiz`

BEZEICHNUNG

`Toolbar.Horiz` – legt die Ausrichtung der Symbolleisten fest

BESCHREIBUNG

Boolescher Wert, um anzugeben, ob die Symbolleistenschaltflächen horizontal oder vertikal ausgerichtet werden sollen. Standardeinstellung ist `False`.

TYP

Boolesch

ANWENDBARKEIT

I

48.18 `Toolbar.HorizInnerSpacing`

BEZEICHNUNG

`Toolbar.HorizInnerSpacing` – stellt den horizontalen Abstand vom Inhalt zum Rahmen ein

BESCHREIBUNG

Definiert die Pixel zwischen einem Schaltflächeninhalt und seinen linken sowie rechten Rahmen. Akzeptiert wird der Bereich von $0 < x \leq 16$.

Dies überschreibt die Benutzereinstellungen und muss daher mit Bedacht verwendet werden. Im Zweifel benutzen Sie dies nicht.

TYP

Nummer

ANWENDBARKEIT

I

48.19 Toolbar.HorizSpacing**BEZEICHNUNG**

Toolbar.HorizSpacing – stellt den horizontalen Abstand zwischen zwei Spalten ein

BESCHREIBUNG

Definiert die Pixel zwischen zwei Symbolleistenpalten. Akzeptiert wird der Bereich von $0 < x \leq 16$.

Dies überschreibt die Benutzereinstellungen und muss daher mit Bedacht verwendet werden. Im Zweifel benutzen Sie dies nicht.

TYP

Nummer

ANWENDBARKEIT

I

48.20 Toolbar.HorizTextGfxSpacing**BEZEICHNUNG**

Toolbar.HorizTextGfxSpacing – stellt den horizontalen Text/GFX-Abstand ein

BESCHREIBUNG

Definiert die Pixel zwischen dem Bild und der Links/Rechts-Beschriftung in den Text/Gfx-Schaltflächen. Akzeptiert wird der Bereich von $0 < x \leq 16$.

Dies überschreibt die Benutzereinstellungen und muss daher mit Bedacht verwendet werden. Im Zweifel benutzen Sie dies nicht.

TYP

Nummer

ANWENDBARKEIT

I

48.21 Toolbar.IgnoreAppearance**BEZEICHNUNG**

Toolbar.IgnoreAppearance – ignoriert die Benutzereinstellungen

BESCHREIBUNG

Der Benutzer kann das allgemeine Erscheinungsbild des Objekts in den MUI-Einstellungen ändern.

Wenn dieses Attribut **True** ist, werden MCP-Erscheinungsbildeinstellungen ignoriert. Wenn dieses Attribut **False** ist, werden MCP-Erscheinungsbildeinstellungen verwendet.

Standarteinstellung ist **False**.

TYP

Boolesch

ANWENDBARKEIT

ISG

48.22 `Toolbar.LabelPos`

BEZEICHNUNG`Toolbar.LabelPos` – legt die Position der Textbeschriftung fest**BESCHREIBUNG**

Steuert die Position des Textes in einer `Text/Gfx`-Symbolleiste. Folgende Positionen werden unterstützt:

- | | |
|---------------------|--|
| <code>Bottom</code> | Der Text erscheint unter dem Bild der Schaltfläche. Dies ist der Standardwert. |
| <code>Top</code> | Der Text wird über dem Schaltflächenbild angezeigt. |
| <code>Right</code> | Der Text erscheint rechts neben dem Schaltflächenbild. |
| <code>Left</code> | Der Text erscheint links neben dem Schaltflächenbild. |

Dies überschreibt die Benutzereinstellungen und muss daher mit Bedacht verwendet werden. Im Zweifel benutzen Sie dies nicht.

TYP

Zeichenkette (siehe oben für mögliche Werte)

ANWENDBARKEIT

ISG

48.23 `Toolbar.LeftBarFrameSpacing`

BEZEICHNUNG`Toolbar.LeftBarFrameSpacing` – stellt den linken Rahmenabstand zum Balken ein**BESCHREIBUNG**

Definiert die Pixel zwischen dem Balken und seinem linken Rahmen. Akzeptiert wird der Bereich von $0 < x \leq 16$.

Dies überschreibt die Benutzereinstellungen und muss daher mit Bedacht verwendet werden. Im Zweifel benutzen Sie dies nicht.

TYP

Nummer

ANWENDBARKEIT

I

48.24 `Toolbar.MouseOver`

BEZEICHNUNG

`Toolbar.MouseOver` – gibt Informationen über das Maus-Über-Ereignis zurück

BESCHREIBUNG

Wenn `Toolbar.Raised` oder `Toolbar.Sunny True` ist, enthält dieses Attribut die ID der letzten Schaltfläche, über der die Maus war.

TYP

Nummer

ANWENDBARKEIT

N

48.25 `Toolbar.NtRaiseActive`

BEZEICHNUNG

`Toolbar.NtRaiseActive` – entfernt den Rahmen von erhöhten Schaltflächen

BESCHREIBUNG

Wenn `False`, wird ein Rahmen um randlose oder erhöhten Schaltflächen gezeichnet. Wenn `True`, wird kein Rahmen verwendet.

Nützlich, um eine Schaltfläche zu aktivieren, wenn die Maus darüber ist und so die Seite in einer Seitenmodusgruppe zu ändern.

Standardeinstellung ist `False`.

TYP

Boolesch

ANWENDBARKEIT

ISG

48.26 `Toolbar.Raised`

BEZEICHNUNG

`Toolbar.Raised` – verwendet den erhöhten Modus

BESCHREIBUNG

Wenn `True`, wird ein Rahmen um eine randlose Schaltfläche gezeichnet, wenn sich die Maus darüber befindet. Es sollte auf `False` gesetzt werden, wenn `Toolbar.Borderless False` ist. Standardeinstellung ist `False`.

Dies überschreibt die Benutzereinstellungen und muss daher mit Bedacht verwendet werden. Im Zweifel benutzen Sie dies nicht.

TYP

Boolesch

ANWENDBARKEIT

ISG

48.27 `Toolbar.RightBarFrameSpacing`

BEZEICHNUNG

`Toolbar.RightBarFrameSpacing` – stellt den rechten Rahmenabstand ein

BESCHREIBUNG

Definiert die Pixel zwischen dem Balken und seinem rechten Rahmen. Akzeptiert wird der Bereich von $0 < x \leq 16$.

Dies überschreibt die Benutzereinstellungen und muss daher mit Bedacht verwendet werden. Im Zweifel benutzen Sie dies nicht.

TYP

Nummer

ANWENDBARKEIT

I

48.28 `Toolbar.Rows`

BEZEICHNUNG

`Toolbar.Rows` – setzt/ermittelt die Anzahl der Symbolleistenzeilen

BESCHREIBUNG

Dieses Attribut definiert die Anzahl der Zeilen, mit denen die Symbolleiste dargestellt werden soll. Wenn Sie dieses Attribut auf einen Wert größer als 0 setzen, wird die Symbolleiste in einem sogenannten Zeilenmodus ausgeführt. Vertikal ist die Anzahl unbegrenzt. Der Standardwert ist 0.

TYP

Nummer

ANWENDBARKEIT

ISG

48.29 `Toolbar.Scale`

BEZEICHNUNG

`Toolbar.Scale` – setzt/ermittelt das Skalierungsverhältnis

BESCHREIBUNG

Definiert das Skalierungsverhältnis für skalierte Schaltflächen. Akzeptiert wird der Bereich von $50 \leq x \leq 200$.

Dies überschreibt die Benutzereinstellungen und muss daher mit Bedacht verwendet werden. Im Zweifel benutzen Sie dies nicht.

TYP

Nummer

ANWENDBARKEIT

ISG

48.30 `Toolbar.Scaled`

BEZEICHNUNG

`Toolbar.Scaled` – aktiviert die Skalierung der Schaltflächen

BESCHREIBUNG

Wenn `True`, werden alle Schaltflächen entsprechend dem definierten Skalierungsverhältnis skaliert. Standardeinstellung ist `False`.

Dies überschreibt die Benutzereinstellungen und muss daher mit Bedacht verwendet werden. Im Zweifel benutzen Sie dies nicht.

TYP

Boolesch

ANWENDBARKEIT

ISG

48.31 `Toolbar.SpacersSize`

BEZEICHNUNG

`Toolbar.SpacersSize` – definiert die Größe der Abstandhalter

BESCHREIBUNG

Definiert die verwendete Größe der Abstandhalter relativ zur Größe einer Schaltfläche. Folgende Werte sind möglich:

`Quarter` Ein Viertel einer Schaltfläche. Dies ist der Standardwert.

`Half` Die Hälfte einer Schaltfläche.

`One` Gleich groß wie eine Schaltfläche.

Dies überschreibt die Benutzereinstellungen und muss daher mit Bedacht verwendet werden. Im Zweifel benutzen Sie dies nicht.

TYP

Zeichenkette (siehe oben für mögliche Werte)

ANWENDBARKEIT

ISG

48.32 `Toolbar.SpecialSelect`

BEZEICHNUNG

`Toolbar.SpecialSelect` – verschiebt den Inhalt bei Auswahl der Schaltfläche

BESCHREIBUNG

Wenn `True`, wird der Inhalt der Schaltfläche nach rechts/unten verschoben, wenn sich die Maus darüber befindet. Standardeinstellung ist `False`.

Dies überschreibt die Benutzereinstellungen und muss daher mit Bedacht verwendet werden. Im Zweifel benutzen Sie dies nicht.

TYP

Boolesch

ANWENDBARKEIT

I

48.33 `Toolbar.Sunny`

BEZEICHNUNG`Toolbar.Sunny` – verwendet den Schwarz/Weiß-Modus**BESCHREIBUNG**

Wenn `True`, werden Schaltflächen in Schwarz/Weiß wiedergegeben, aber wenn sich die Maus über ihnen befindet, werden sie farbig dargestellt. Standardeinstellung ist `False`.

Dies überschreibt die Benutzereinstellungen und muss daher mit Bedacht verwendet werden. Im Zweifel benutzen Sie dies nicht.

TYP

Boolesch

ANWENDBARKEIT

ISG

48.34 `Toolbar.TextOverUseShine`

BEZEICHNUNG`Toolbar.TextOverUseShine` – verwendet den Glanzstift für die hervorgehobenen Beschriftung**BESCHREIBUNG**

Wenn `True`, werden die Beschriftungen der Schaltflächen mit dem Glanz des Textstiftes dargestellt, wenn sich die Maus darüber befindet.

Dies überschreibt die Benutzereinstellungen und muss daher mit Bedacht verwendet werden. Im Zweifel benutzen Sie dies nicht.

TYP

Boolesch

ANWENDBARKEIT

I

48.35 `Toolbar.TopBarFrameSpacing`

BEZEICHNUNG`Toolbar.TopBarFrameSpacing` – stellt den oberen Abstand zum Balken ein

BESCHREIBUNG

Definiert die Pixel zwischen dem Balken und seinem oberen Rahmen. Akzeptiert wird der Bereich von $0 < x \leq 16$.

Dies überschreibt die Benutzereinstellungen und muss daher mit Bedacht verwendet werden. Im Zweifel benutzen Sie dies nicht.

TYP

Nummer

ANWENDBARKEIT

I

48.36 `Toolbar.TopInnerSpacing`

BEZEICHNUNG

`Toolbar.TopInnerSpacing` – stellt den Abstand zwischen Inhalt und oberem Rahmen ein

BESCHREIBUNG

Definiert die Pixel zwischen dem Inhalt einer Schaltfläche und ihrem oberen Rahmen. Akzeptiert wird der Bereich von $0 < x \leq 16$.

Dies überschreibt die Benutzereinstellungen und muss daher mit Bedacht verwendet werden. Im Zweifel benutzen Sie dies nicht.

TYP

Boolesch

ANWENDBARKEIT

I

48.37 `Toolbar.VertSpacing`

BEZEICHNUNG

`Toolbar.VertSpacing` – stellt den vertikalen Abstand zwischen zwei Balkenreihen ein

BESCHREIBUNG

Definiert die Pixel zwischen zwei Balkenreihen. Akzeptiert wird der Bereich von $0 < x \leq 16$.

Dies überschreibt die Benutzereinstellungen und muss daher mit Bedacht verwendet werden. Im Zweifel benutzen Sie dies nicht.

TYP

Nummer

ANWENDBARKEIT

I

48.38 `Toolbar.VertTextGfxSpacing`

BEZEICHNUNG

`Toolbar.VertTextGfxSpacing` – legt den vertikalen Text-GFX-Abstand fest

BESCHREIBUNG

Definiert die Pixel zwischen dem Bild und der obersten/untersten Beschriftung in den Text/Gfx-Schaltflächen. Akzeptiert wird der Bereich von $0 < x \leq 16$.

Dies überschreibt die Benutzereinstellungen und muss daher mit Bedacht verwendet werden. Im Zweifel benutzen Sie dies nicht.

TYP

Nummer

ANWENDBARKEIT

I

48.39 `Toolbar.ViewMode`

BEZEICHNUNG

`Toolbar.ViewMode` – legt den Ansichtsmodus der Symbolleiste fest

BESCHREIBUNG

Legt den Stil der Symbolleiste fest. Folgende Stile sind möglich:

`TextGfx` Symbolleiste wird als Text mit Bilder angezeigt. Dies ist auch die Standard-einstellung.

`Gfx` Die Symbolleiste wird nur mit Bilder angezeigt.

`Text` Die Symbolleiste wird nur mit Texten angezeigt.

Dies überschreibt die Benutzereinstellungen und muss daher mit Bedacht verwendet werden. Im Zweifel benutzen Sie dies nicht.

TYP

Zeichenkette (siehe oben für mögliche Werte)

ANWENDBARKEIT

ISG

49 Toolbarbutton-Klasse (Symbolleistenschaltfläche)

49.1 Übersicht

Die Toolbarbutton-Klasse (Symbolleistenschaltfläche) ist eine Unterklasse der Toolbar-Klasse (Symbolleiste). Sie kann nicht alleine verwendet werden, sondern muss immer in einer Symbolleisten-Klassendefinition gekapselt sein. Die Toolbarbutton-Klasse erstellt eine einzelne Schaltfläche für ihre Toolbar-Klasse. Bitte beachten Sie, dass der XML-Tag für diese Klasse nur `<button>` ist, nicht `<toolbarbutton>`. Siehe [Abschnitt 48.1 \[Toolbar-Klasse\], Seite 263](#), für Details.

49.2 Toolbarbutton.Disabled

BEZEICHNUNG

Toolbarbutton.Disabled – aktiviert/deaktiviert die Schaltfläche

BESCHREIBUNG

Aktiviert/deaktiviert die Schaltfläche.

TYP

Boolesch

ANWENDBARKEIT

ISG

49.3 Toolbarbutton.DisImage

BEZEICHNUNG

Toolbarbutton.DisImage – stellt das deaktivierte Schaltflächenbild ein

BESCHREIBUNG

Legt das Bild für diese Symbolleistenschaltfläche fest, das angezeigt werden soll, wenn sich die Schaltfläche im deaktivierten Zustand befindet. Dies muss auf den Identifikator eines Hollywood-Pinsels gesetzt werden.

TYP

Nummer

ANWENDBARKEIT

I

49.4 Toolbarbutton.Exclude

BEZEICHNUNG

Toolbarbutton.Exclude – setzt die Ausschlussmaske

BESCHREIBUNG

Ermöglicht die Angabe einer gemeinsamen exklusiven Maske, um bis zu 24 Schaltflächen miteinander zu verbinden. Die Maske muss als eine Bitoperation des Typs $1 \ll x$ angegeben werden, wobei $x = 0, \dots, 23$ die ID der Schaltfläche ist, die die gegenseitige Anschlussmaske ist.

TYP

Nummer

ANWENDBARKEIT

I

49.5 Toolbarbutton.Hide

BEZEICHNUNG

Toolbarbutton.Hide – zeigt die Schaltfläche an oder verbergt sie

BESCHREIBUNG

Verwenden Sie dieses Attribut zum Anzeigen/Ausblenden von Symbolleistenschaltflächen.

TYP

Boolesch

ANWENDBARKEIT

ISG

49.6 Toolbarbutton.Image

BEZEICHNUNG

Toolbarbutton.Image – legt das Bild der Schaltfläche fest

BESCHREIBUNG

Legt das Bild für diese Symbolleistenschaltfläche fest. Dies muss auf den Identifikator eines Hollywood-Pinsels oder einen der folgenden speziellen Werte gesetzt werden:

BarSpacer

Erstellt einen Balkenabstandshalter.

ButtonSpacer

Erstellt einen Abstandshalter zwischen zwei Schaltflächen.

TYP

Nummer oder Zeichenkette (siehe oben für mögliche Werte)

ANWENDBARKEIT

I

49.7 Toolbarbutton.Immediate

BEZEICHNUNG

Toolbarbutton.Immediate – erstellt umgehend eine Schaltfläche

BESCHREIBUNG

Wenn Sie dies auf `True` setzen, wird die Schaltfläche sofort angezeigt.

TYP

Boolesch

ANWENDBARKEIT

I

49.8 Toolbarbutton.NoClick

BEZEICHNUNG

Toolbarbutton.NoClick – erstellt eine nicht reagierende Schaltfläche

BESCHREIBUNG

Wenn Sie dies auf `True` setzen, reagiert die Schaltfläche nicht auf Benutzereingaben.

TYP

Boolesch

ANWENDBARKEIT

I

49.9 Toolbarbutton.Pressed

BEZEICHNUNG

Toolbarbutton.Pressed – benachrichtigt, ob eine Schaltfläche gedrückt wird

BESCHREIBUNG

Dieses Attribut wird ausgelöst, wenn der Benutzer die Schaltfläche drückt.

TYP

Boolesch

ANWENDBARKEIT

N

49.10 Toolbarbutton.Selected

BEZEICHNUNG

Toolbarbutton.Selected – schaltet den Auswahlstatus um oder benachrichtigt

BESCHREIBUNG

Verwenden Sie diese Option, um den Auswahlstatus der Schaltfläche umzuschalten oder um über ein Umschalteignis des Benutzers benachrichtigt zu werden.

TYP

Boolesch

ANWENDBARKEIT

ISGN

49.11 `Toolbarbutton.SelImage`

BEZEICHNUNG

`Toolbarbutton.SelImage` – stellt das ausgewählte Schaltflächenbild ein

BESCHREIBUNG

Legt das Bild für diese Symbolleistenschaltfläche fest, die angezeigt werden soll, wenn sich die Schaltfläche im ausgewählten Status befindet. Dies muss auf den Identifikator eines Hollywood-Pinsels gesetzt werden.

TYP

Nummer

ANWENDBARKEIT

I

49.12 `Toolbarbutton.ShortHelp`

BEZEICHNUNG

`Toolbarbutton.ShortHelp` – legt die Blasenhilfe für die Schaltfläche fest

BESCHREIBUNG

Legt die Blasenhilfe für die Schaltfläche fest.

Die Zeichenkette, die Sie hier angeben, kann Textformatierungs-codes verwenden. Siehe [Abschnitt 3.9 \[Textformatierungs-codes\]](#), Seite 14, für Details.

TYP

Zeichenkette

ANWENDBARKEIT

I

49.13 `Toolbarbutton.Sleep`

BEZEICHNUNG

`Toolbarbutton.Sleep` – erstellt eine schlafende Schaltfläche

BESCHREIBUNG

Wenn dies auf `True` gesetzt ist, wird die Schaltfläche überhaupt nicht erstellt.

TYP

Boolesch

ANWENDBARKEIT

ISG

49.14 Toolbarbutton.Toggle

BEZEICHNUNG

Toolbarbutton.Toggle – erstellt eine Umschaltfläche

BESCHREIBUNG

Wenn Sie dies auf `True` setzen, ist die Schaltfläche eine Umschaltfläche.

TYP

Boolesch

ANWENDBARKEIT

I

50 VSpace-Klasse (VertikalAbstand)

50.1 Übersicht

Die VSpace-Klasse (VertikalAbstand) ist eine Unterklasse der Rectangle-Klasse (Rechteck) und erstellt einfach einen leeren vertikalen Abstand zwischen zwei Objekten.

```
<vspace height="2"/>
```

50.2 VSpace.Height

BEZEICHNUNG

VSpace.Height – stellt den vertikalen Abstand ein

BESCHREIBUNG

Legt den gewünschten vertikalen Abstand für dieses Objekt in Pixeln fest.

TYP

Nummer

ANWENDBARKEIT

I

51 Virtgroup-Klasse (Virtuelle-Gruppen)

51.1 Übersicht

Die Virtgroup-Klasse (Virtuelle-Gruppen) generiert spezielle Arten von Gruppen-Objekten, deren untergeordnete Elemente viel größer als die tatsächliche Gruppe sein können. Die Gruppe fungiert als (kleines) Fenster, durch das ein rechteckiger Bereich ihres Inhalts sichtbar ist.

Während des Layouts versucht MUI, die untergeordneten Elemente einer virtuellen Gruppe im sichtbaren Teil zu platzieren. Wenn dies nicht möglich ist, wird der Raum erweitert, bis alle untergeordneten Elemente passen.

Virtuelle Gruppen selbst bieten keine Bildlaufleisten, um Benutzerinteraktionen zu ermöglichen. Diese Dinge werden von der Scrollgroup-Klasse (Bildlaufgruppen) behandelt. Normalerweise würden Sie keine virtuelle Gruppe ohne Bildlaufgruppe verwenden. Siehe [Abschnitt 43.1 \[Scrollgroup-Klasse\]](#), [Seite 223](#), für Details.

51.2 Virtgroup.Height

BEZEICHNUNG

Virtgroup.Height – ermittelt die Höhe der virtuellen Gruppe

BESCHREIBUNG

Ermittelt die virtuelle Höhe einer virtuellen Gruppe. Dieses Attribut ist ziemlich sinnlos. Verwenden Sie besser ein Bildlaufgruppen-Objekt (Scrollgroup), um die virtuelle Gruppe zu steuern. Siehe [Abschnitt 43.1 \[Scrollgroup-Klasse\]](#), [Seite 223](#), für Details.

TYP

Nummer

ANWENDBARKEIT

G

51.3 Virtgroup.Horiz

BEZEICHNUNG

Virtgroup.Horiz – legt die Ausrichtung der virtuellen Gruppe fest

BESCHREIBUNG

Boolescher Wert, um anzugeben, ob die Objekte in dieser Gruppe horizontal (`True`) oder vertikal (`False`) ausgerichtet werden sollen. Standardeinstellung ist `False`.

Dies ist der einfache Weg, Ihre Gruppe auszurichten. Wenn Sie zweidimensionale Gruppen möchten, müssen Sie `Group.Columns` verwenden.

TYP

Boolesch

ANWENDBARKEIT

I

51.4 Virtgroup.Input

BEZEICHNUNG

Virtgroup.Input – stellt den Eingabemodus der virtuellen Gruppe ein

BESCHREIBUNG

Geben Sie an, ob eine virtuelle Gruppe beweglich sein soll, indem Sie darauf klicken und die Maus ziehen. Der Standardwert ist `True`.

TYP

Boolesch

ANWENDBARKEIT

I

51.5 Virtgroup.Left

BEZEICHNUNG

Virtgroup.Left – setzt/ermittelt den linken Rand der virtuellen Gruppe

BESCHREIBUNG

Ermittelt/setzt den linken Rand einer virtuellen Gruppe. Der linke Rand wird automatisch zwischen 0 und (`VirtualWidth-DisplayWidth`) abgeschnitten.

Dieses Attribut ist ziemlich sinnlos. Verwenden Sie besser ein Bildlaufgruppen-Objekt, um die virtuelle Gruppe zu steuern. Siehe [Abschnitt 43.1 \[Scrollgroup-Klasse\]](#), Seite 223, für Details.

TYP

Nummer

ANWENDBARKEIT

ISG

51.6 Virtgroup.Top

BEZEICHNUNG

Virtgroup.Top – setzt/ermittelt den oberen Rand der virtuellen Gruppe

BESCHREIBUNG

Ermittelt/setzt den oberen Rand einer virtuellen Gruppe. Der obere Rand wird automatisch zwischen 0 und (`VirtualTop-DisplayTop`) abgeschnitten.

Dieses Attribut ist ziemlich sinnlos. Verwenden Sie besser ein Bildlaufgruppen-Objekt, um die virtuelle Gruppe zu steuern. Siehe [Abschnitt 43.1 \[Scrollgroup-Klasse\]](#), Seite 223, für Details.

TYP

Nummer

ANWENDBARKEIT

ISG

51.7 Virtgroup.Width

BEZEICHNUNG

Virtgroup.Width – setzt/ermittelt die Breite der virtuellen Gruppe

BESCHREIBUNG

Ermittelt die Breite einer virtuellen Gruppe.

Dieses Attribut ist ziemlich sinnlos. Verwenden Sie besser ein Bildlaufgruppen-Objekt, um die virtuelle Gruppe zu steuern. Siehe [Abschnitt 43.1 \[Scrollgroup-Klasse\]](#), Seite 223, für Details.

TYP

Nummer

ANWENDBARKEIT

G

52 Volumelist-Klasse (Datenträgerlisten)

52.1 Übersicht

Die Volumelist-Klasse (Datenträgerlisten) generiert eine Liste aller verfügbaren Datenträger. Da Sie in keiner Anwendung Ihre eigenen Dateidialog verwenden sollten, ist diese Klasse wahrscheinlich nicht von großem Nutzen.

Die Volumelist-Klasse erstellt eine Listenansicht mit den folgenden fünf Spalten: Symbol, Name, Füllstatus in Prozent, freier Speicherplatz und belegter Speicherplatz. Wenn Sie nicht möchten, dass alle diese Attribute angezeigt werden, können Sie mit der Listviewcolumn-Klasse Anpassungen vornehmen.

Die Volumelist-Klasse ist eine Unterklasse der Listview-Klasse (Listenansicht). Daher können Sie auch die meisten Attribute und Methoden der Listview-Klasse verwenden. Wenn Sie zum Beispiel die Einträge Ihrer Datenträgerliste lesen möchten, benutzen Sie einfach für das Datenträgerlisten-Objekt die Methode `Listview.GetEntry`, oder wenn Sie Änderungen im aktiven Listeneintrag überwachen möchten, richten Sie eine Benachrichtigung mit `Listview.Active` ein.

Wenn Sie ein Datenträgerlisten-Objekt im XML-Code erstellen, müssen Sie immer mindestens eine Spalte hinzufügen. Dies geschieht mithilfe der Listviewcolumn-Klasse. Hier ist ein Beispiel für eine minimale Datenträgerlisten-Definition mit nur einer einzigen Spalte (das ist die Symbol-Spalte):

```
<volumelist>
  <column/> <!-- Icon -->
</volumelist>
```

Hier ist eine Definition, die alle fünf Spalten enthält:

```
<volumelist>
  <column/> <!-- Icon -->
  <column/> <!-- Name -->
  <column/> <!-- Percent -->
  <column/> <!-- Free -->
  <column/> <!-- Used -->
</volumelist>
```

Wenn Sie nur die Namen- und die Prozentspalte anzeigen möchten, können Sie das Attribut `Listviewcolumn.Col` verwenden, um dies zu erreichen:

```
<volumelist>
  <column col="1"/> <!-- Name -->
  <column col="2"/> <!-- Percent -->
</volumelist>
```

Siehe [Abschnitt 25.1 \[Listview-Klasse\]](#), Seite 149, für Details.

Siehe [Abschnitt 26.1 \[Listviewcolumn-Klasse\]](#), Seite 165, für Details.

52.2 Volumelist.Title

BEZEICHNUNG

Volumelist.Title – zeigt den Listentitel an oder blendet ihn aus

BESCHREIBUNG

Geben Sie an, ob Sie eine Titelleiste für die Listenansicht haben möchten (**True**). Der Titel wird in der ersten Zeile angezeigt und blättert nicht weg, wenn sich die oberste Position der Liste bewegt.

TYP

Boolesch

ANWENDBARKEIT

ISG

53 Window-Klasse (Fenster)

53.1 Übersicht

Objekte der Window-Klasse (Fenster) werden verwendet, um Fenster zu erzeugen und einen Ort bereitzustellen, an dem Sie MUI-Gadgets plazieren können. Sie übernimmt die komplizierte Aufgabe der vollautomatischen Fenstergrößenänderung. Sie müssen sich nicht darum kümmern.

Fenster sind untergeordnete Elemente einer Anwendung. Sie können kein Fenster-Objekt verwenden, ohne ein übergeordnetes Anwendungs-Objekt zu haben. Auf der anderen Seite sind die Gadgets in einem Fenster untergeordnete Elemente des Fensters. Sie können MUI-Gadgets nicht ohne ein übergeordnetes MUI-Fenster verwenden.

Das Erstellen eines Fenster-Objekts bedeutet nicht, es sofort zu öffnen. Dies geschieht später, indem das Attribut `Window.Open` des Fensters gesetzt wird. Wenn Ihre Anwendung mehrere Fenster hat, besteht die übliche Vorgehensweise darin, sie alle gleichzeitig beim Start zu erstellen und später zu öffnen/zu schließen, indem Sie einfach `Window.Open` setzen.

Es gibt keinen Unterschied, um mit Gadgets zu arbeiten, egal ob das übergeordnete Fenster offen ist oder nicht. Wenn Sie z.B. den Inhalt eines Zeichenketten-Gadgets in einem geöffneten Fenster festlegen, wird das Gadget sofort aktualisiert. Wenn das Fenster geschlossen ist, merkt sich das Gadget die neue Einstellung und zeigt sie später an.

53.2 Window.Activate

BEZEICHNUNG

`Window.Activate` – ändert den Aktivierungsstatus des Fensters

BESCHREIBUNG

Wenn Sie dies auf `True` setzen, wird das Fenster aktiviert. Das Setzen auf `False` hat keine Auswirkung. Das Attribut ändert sich, wenn der Benutzer das Fenster aktiviert/deaktiviert.

Wenn Sie zum Zeitpunkt der Objekterstellung `False` angeben, wird das Fenster in einem inaktiven Zustand geöffnet.

TYP

Boolesch

ANWENDBARKEIT

ISGN

53.3 Window.ActiveObject

BEZEICHNUNG

`Window.ActiveObject` – setzt/ermittelt ein aktives Fenster-Objekt

BESCHREIBUNG

Setzen Sie das aktive Objekt in einem Fenster so, als ob der Benutzer es mit der Tabulatortaste aktiviert hätte. Das Objekt muss in der Tastaturabfolgekette sein, damit dieser

Befehl funktioniert. Ausserdem kann das aktive Objekt ermittelt und eine Benachrichtigung eingerichtet werden.

Beginnend mit MUI Royale 1.1 ist die Anwendbarkeit dieses Attributs SGN. Vor Version 1.1 war es nur SG.

TYP

MUI Objekt

ANWENDBARKEIT

SGN

53.4 Window.AppWindow

BEZEICHNUNG

Window.AppWindow – macht ein Fenster zum App-Fenster

BESCHREIBUNG

Wenn dieses Attribut auf `True` gesetzt wird, wird dieses Fenster zu einem App-Fenster, worauf der Benutzer Symbole ablegen kann. Sie können von den Ereignissen erfahren, indem Sie das Attribut `Notify.AppMessage` überwachen.

TYP

Boolesch

ANWENDBARKEIT

I

53.5 Window.Borderless

BEZEICHNUNG

Window.Borderless – blendet die Fensterrahmen ein oder aus

BESCHREIBUNG

Macht das Fenster randlos (`True`).

TYP

Boolesch

ANWENDBARKEIT

I

53.6 Window.CloseGadget

BEZEICHNUNG

Window.CloseGadget – konfiguriert das Schließ-Gadget des Fensters

BESCHREIBUNG

Setzen Sie dies auf `False` und Ihr Fenster wird kein Schließ-Gadget haben.

TYP

Boolesch

ANWENDBARKEIT

I

53.7 Window.CloseRequest

BEZEICHNUNG

Window.CloseRequest – benachrichtigt, wenn auf das Schließ-Gadget geklickt wird

BESCHREIBUNG

Wenn der Benutzer auf das Schließ-Gadget des Fensters klickt, wird es nicht sofort geschlossen. Stattdessen setzt MUI dieses Attribut nur auf `True`, damit Ihre Anwendung reagieren kann.

Normalerweise werden Sie eine Benachrichtigung einrichten, die das Fenster automatisch schließt, wenn auf das Schließ-Gadget geklickt wird. Aber Sie könnten stattdessen z.B. einen Bestätigungsdialog öffnen oder führen zuerst einige andere Schritte aus.

TYP

Boolesch

ANWENDBARKEIT

N

53.8 Window.DefaultObject

BEZEICHNUNG

Window.DefaultObject – setzt/ermittelt das Standard-Objekt des Fensters

BESCHREIBUNG

Das Standard-Objekt in einem Fenster empfängt Tastatureingaben, solange kein anderes Objekt aktiv ist. Gute Kandidaten für Standard-Objekte sind z.B. einsame Listenansichten (Listview). Wenn eine solche Listenansicht zum Standard-Objekt gemacht wird, kann der Benutzer sie sofort steuern, ohne mehrere male die TAB-Taste für die Aktivierung zu betätigen.

TYP

MUI Objekt

ANWENDBARKEIT

ISG

53.9 Window.DepthGadget

BEZEICHNUNG

Window.DepthGadget – konfiguriert das Vorder-/Hintergrund-Gadget des Fensters

BESCHREIBUNG

Aktivieren oder deaktivieren Sie das Vorder-/Hintergrund-Gadget. Der Standardwert ist `True`. Es gibt keinen Grund, diesen Tag zu verwenden.

TYP

Boolesch

ANWENDBARKEIT

I

53.10 Window.DragBar

BEZEICHNUNG

Window.DragBar – konfiguriert die Ziehleiste des Fensters

BESCHREIBUNG

Bei `True` erhält das Fenster eine Ziehleiste.

Standardeinstellung ist `True`.

Es gibt keinen Grund, die Ziehleiste zu deaktivieren!

TYP

Boolesch

ANWENDBARKEIT

I

53.11 Window.Height

BEZEICHNUNG

Window.Height – setzt/ermittelt die Fensterhöhe

BESCHREIBUNG

Geben Sie die Höhe eines Fensters an. Normalerweise geben Sie hier keinen Pixelwert, sondern stattdessen eines der folgenden Tags an:

Default Berechnet die Höhe aus den Standardobjektgrößen.

MinMax:<0..100>

Irgendwo zwischen der minimalen Höhe (0) und der maximalen Höhe (100) Ihres Fensters.

Visible:<1..100>

Prozentsatz der sichtbaren Höhe des Bildschirms.

Screen:<1..100>

Prozentsatz der Gesamthöhe des Bildschirms.

Scaled Die Höhe wird so eingestellt, dass $\text{Breite} : \text{Höhe} == \text{minimale Breite} : \text{minimale Höhe}$ ist. Beachten Sie, dass die Fensterbreite und -höhe nicht beide skaliert werden können.

Standard für diesen Tag ist **Default**.

Solange Ihr Fenster eine MUI-Fenster-ID über das Attribut `Window.MuiID` hat, ist die Auswahl einer Größe nicht so wichtig. MUI wird sich immer an die letzte Position und Größe eines Fensters erinnern und diese Werte überschreiben Ihre Einstellungen einfach. Positionierung und Dimensionierung sollten vollständig vom Benutzer kontrolliert werden. Ein Programmierer muss sich nicht darum kümmern.

TYP

Nummer oder vordefinierter Tag

ANWENDBARKEIT

IG

53.12 Window.LeftEdge

BEZEICHNUNG

`Window.LeftEdge` – setzt/ermittelt den linken Rand des Fensters

BESCHREIBUNG

Geben Sie den linken Rand eines Fensters an. Normalerweise sollten Sie hier keinen Pixelwert definieren, sondern stattdessen eines der folgenden Tags verwenden:

Centered Das Fenster erscheint zentriert im sichtbaren Bereich des Bildschirms.

Moused Das Fenster erscheint zentriert unter dem Mauszeiger.

Standard für diesen Tag ist **Centered**.

Solange Ihr Fenster eine Fenster-ID über das Attribut `Window.MuiID` hat, ist die Auswahl einer Position nicht so wichtig. MUI wird sich immer an die letzte Position und Größe eines Fensters erinnern und diese Werte Ihrer Einstellungen einfach überschreiben. Positionierung und Dimensionierung sollten vollständig vom Benutzer kontrolliert werden. Ein Programmierer muss sich nicht darum kümmern.

TYP

Nummer oder vordefiniertes Makro

ANWENDBARKEIT

IG

53.13 Window.Menustrip

BEZEICHNUNG

`Window.Menustrip` – definiert die Menüleiste des Fensters

BESCHREIBUNG

Geben Sie ein Menüleisten-Objekt für dieses Fenster an. Das Objekt wird als untergeordnetes Element des Fensters behandelt und bei der Anordnung des Fensters eingestellt.

Menüleisten-Objekte, die für ein Fenster definiert sind, überschreiben ein Anwendungsmenüleisten-Objekt.

Wenn Sie ein globales Menü für alle Ihre Anwendungsfenster haben, aber einige Fenster kein Menü haben möchten, verwenden Sie den Tag `Window.NoMenus`.

TYP

MUI Objekt

ANWENDBARKEIT

I

53.14 Window.MouseObject

BEZEICHNUNG

`Window.MouseObject` – benachrichtigt, über welchem Objekt die Maus gerade ist

BESCHREIBUNG

Wenn `Window.NeedsMouseObject` für dieses Fenster aktiviert ist, können Sie unter `Window.MouseObject` Benachrichtigungen einrichten, um herauszufinden, auf welchem Objekt sich der Mauszeiger befindet.

TYP

MUI Objekt

ANWENDBARKEIT

N

53.15 Window.MuiID

BEZEICHNUNG

`Window.MuiID` – setzt/ermittelt den Fensteridentifikator

BESCHREIBUNG

Für die meisten Ihrer Fenster sollten Sie eine vierstellige Zeichenkette als ID-Wert definieren. Nur ein Fenster mit einer ID kann sich an die Größe und Position erinnern.

Wenn Sie zusätzlich eine ASCII-ID (z.B. "MAIN") verwenden, kann Ihr Fenster von ARexx aus gesteuert werden.

Natürlich müssen alle Fenster Ihrer Anwendung eindeutige IDs haben.

Verwechseln Sie `Window.MuiID` nicht mit `Notify.ID`. Die ID, die Sie in `Notify.ID` angeben, ist diejenige, mit der Sie mit MUI-Objekten kommunizieren, indem Sie Befehle wie `mui.Set()` oder `mui.Get()` verwenden. Es hat nichts mit der ID zu tun, die Sie in `Window.MuiID` festgelegt haben!

TYP

Vierzeichen-Zeichenkette

ANWENDBARKEIT

ISG

53.16 Window.NeedsMouseObject

BEZEICHNUNG

Window.NeedsMouseObject – aktiviert die Mausverfolgung

BESCHREIBUNG

Wenn Sie auf Änderungen des Attributs `Window.MouseObject` reagieren möchten, müssen Sie `Window.NeedsMouseObject` beim Erstellen des Fensters auf `True` setzen.

TYP

Boolesch

ANWENDBARKEIT

I

53.17 Window.NoMenus

BEZEICHNUNG

Window.NoMenus – deaktiviert das Fenstermenü

BESCHREIBUNG

Deaktiviert vorübergehend die Menüleiste eines Fensters.

TYP

Boolesch

ANWENDBARKEIT

IS

53.18 Window.Open

BEZEICHNUNG

Window.Open – benachrichtigt, öffnet/schließt ein Fenster

BESCHREIBUNG

Dieses kleine Attribut kann zum Öffnen und Schließen eines Fensters verwendet werden. Beim Öffnen eines Fensters berechnet MUI mit vielen Schritten die Größen und Positionen aller Gadgets. Minimale und maximale Fenstergrößen werden automatisch angepasst.

Wenn die Mindestgröße eines Fensters zu groß ist, um auf den Bildschirm zu passen, versucht MUI, die Schriftgrößen zu reduzieren und führt eine neue Berechnung durch. Sie sollten Ihre Fenster immer so entwerfen, dass sie auf einen Bildschirm mit 640*200 passen, wobei alle Zeichensätze auf `topaz/8` eingestellt sind.

Wenn ein Fenster geschlossen wird (und Sie ein `Window.MuiID` angegeben haben, merkt sich MUI seine Position und Größe und verwendet diese Werte bei der nächsten Öffnung. Nachdem Sie `Window.Open` auf `True` gesetzt haben, sollten Sie testen, ob MUI das Fenster öffnen konnte, indem Sie das Attribut erneut abrufen. Wenn Sie dies nicht tun und dies das einzige Fenster Ihrer Anwendung ist, kann der Benutzer keine Eingaben vornehmen und Ihre Anwendung scheint hängen zu bleiben.

TYP

Boolesch

ANWENDBARKEIT

ISG

53.19 Window.PubScreen

BEZEICHNUNG

Window.PubScreen – setzt/ermittelt den Bildschirm des Fensters (V1.1)

BESCHREIBUNG

Mit diesem Attribut können Sie den Namen eines öffentlichen Bildschirms angeben, auf dem das Fenster geöffnet werden soll. Bitte verwenden Sie dieses Attribut nur, wenn es wirklich notwendig ist, da normalerweise der Benutzer Ihrer Anwendung derjenige sein sollte, der entscheidet, auf welchem Bildschirm er Ihre Anwendung mit den MUI-Einstellungen ausführen möchte.

TYP

Zeichenkette

ANWENDBARKEIT

ISG

53.20 Window.ScreenTitle

BEZEICHNUNG

Window.ScreenTitle – setzt/ermittelt den Bildschirmtitel des Fensters

BESCHREIBUNG

Dieser Text wird in der Titelleiste des Bildschirms angezeigt, wenn das Fenster aktiv ist.

TYP

Zeichenkette

ANWENDBARKEIT

ISG

53.21 Window.ScreenToBack

BEZEICHNUNG

Window.ScreenToBack – stellt den Fenster-Bildschirm nach hinten

ÜBERSICHT

```
mui.DoMethod(id, "ScreenToBack")
```

BESCHREIBUNG

Setzt den Bildschirm des Fensters nach hinten. Dieser Befehl ist nur gültig, wenn das Fenster geöffnet ist.

EINGABEN

id ID des Fenster-Objekts

53.22 Window.ScreenToFront

BEZEICHNUNG

Window.ScreenToFront – stellt den Fenster-Bildschirm nach vorne

ÜBERSICHT

```
mui.DoMethod(id, "ScreenToFront")
```

BESCHREIBUNG

Setzt den Bildschirm des Fensters nach vorne. Dieser Befehl ist nur gültig, wenn das Fenster geöffnet ist.

EINGABEN

id ID des Fenster-Objekts

53.23 Window.Sleep

BEZEICHNUNG

Window.Sleep – versetzt das Fenster in den Ruhezustand

BESCHREIBUNG

Dieses Attribut kann verwendet werden, um ein Fenster in den Ruhezustand zu versetzen. Das Fenster wird deaktiviert und ein Beschäftigungsmauszeiger erscheint.

Das Attribut enthält eine Anzahl von Verschachtelungen. Wenn Sie Ihrem Fenster sagen, dass es zweimal ruhen soll, müssen Sie ihm ebenfalls mitteilen, dass es zweimal aktiviert werden soll.

Ein ruhendes Fenster kann nicht in der Größe geändert werden.

TYP

Boolesch

ANWENDBARKEIT

SG

53.24 Window.SizeGadget

BEZEICHNUNG

Window.SizeGadget – konfiguriert das Größen-Gadget des Fensters

BESCHREIBUNG

Mit diesem Attribut legen sie fest, ob Sie ein Größen-Gadget für dieses Fenster haben oder nicht. Normalerweise benötigen Sie dieses Attribut nicht, da MUI das Größen-Gadget automatisch deaktiviert, wenn Ihr Fenster aufgrund Ihres Gadget-Layouts nicht in der Größe veränderbar ist.

TYP

Boolesch

ANWENDBARKEIT

I

53.25 Window.Snapshot**BEZEICHNUNG**

Window.Snapshot – merkt sich die Fensterposition und -grösse

ÜBERSICHT

`mui.DoMethod(id, "Snapshot", flags)`

BESCHREIBUNG

Window.Snapshot ist die Schnittstelle des Programmierers zur MUI-Fensterpositions-Erinnerungsfunktion. Schnappschüsse eines Fensters sind nur möglich, wenn Window.MuiID für dieses Fenster festgelegt wurde.

EINGABEN

`id` ID des Fenster-Objekts

`flags` Verwenden Sie 0, um nichts zu merken, 1, um die Position und Größe des Fensters zu merken

53.26 Window.ToBack**BEZEICHNUNG**

Window.ToBack – setzt das Fenster nach hinten

ÜBERSICHT

`mui.DoMethod(id, "ToBack")`

BESCHREIBUNG

Setzt das Fenster nach hinten. Wenn das Fenster gerade nicht geöffnet ist, führt dieser Befehl einfach nichts aus.

EINGABEN

`id` ID des Fenster-Objekts

53.27 Window.ToFront**BEZEICHNUNG**

Window.ToFront – setzt das Fenster nach vorne

ÜBERSICHT

`mui.DoMethod(id, "ToFront")`

BESCHREIBUNG

Setzt das Fenster nach vorne. Wenn das Fenster gerade nicht geöffnet ist, führt dieser Befehl einfach nichts aus.

EINGABEN

`id` ID des Fenster-Objekts

53.28 Window.TopEdge**BEZEICHNUNG**

`Window.TopEdge` – setzt/ermittelt den oberen Rand des Fensters

BESCHREIBUNG

Geben Sie den oberen Rand eines Fensters an. Normalerweise sollten Sie hier keinen Pixelwert definieren, sondern stattdessen eines der folgenden Tags verwenden:

`Centered` Das Fenster erscheint zentriert im sichtbaren Bereich des Bildschirms.

`Moused` Das Fenster erscheint zentriert unter dem Mauszeiger.

`Delta:<p>`

Das Fenster erscheint <p> Pixel unter der Titelleiste des Bildschirms.

Standard für diesen Tag ist `Centered`.

Solange Ihr Fenster eine Fenster-ID über das Attribut `Window.MuiID` hat, ist die Auswahl einer Position nicht so wichtig. MUI wird sich immer an die letzte Position und Größe eines Fensters erinnern und diese Werte Ihrer Einstellungen einfach überschreiben. Positionierung und Dimensionierung sollten vollständig vom Benutzer kontrolliert werden, ein Programmierer muss sich nicht darum kümmern.

TYP

Nummer oder vordefiniertes Makro

ANWENDBARKEIT

IG

53.29 Window.UseBottomBorderScroller**BEZEICHNUNG**

`Window.UseBottomBorderScroller` – aktiviert die untere Rahmen-Bildlaufleiste

BESCHREIBUNG

Wenn `True` gesetzt ist, wird das Fenster am unteren Rand eine Bildlaufleiste enthalten. Sie müssen dies für das Fenster-Objekt festlegen, wenn untergeordnete Elemente diese Fensterbildlaufleiste verwenden, z.B. `Proportional-Gadget` mit dem Attribut `Prop.UseWinBorder`.

Offensichtlich sehen Bildlaufleisten-Gadgets in randlosen Fensterrahmen oder nicht veränderbaren Fenstern nicht gut aus.

TYP

Boolesch

ANWENDBARKEIT

I

53.30 Window.UseLeftBorderScroller

BEZEICHNUNG

Window.UseLeftBorderScroller – aktiviert die linke Rahmen-Bildlaufleiste

BESCHREIBUNG

Wenn `True` gesetzt ist, wird das Fenster am linken Rand eine Bildlaufleiste enthalten. Sie müssen dies für das Fenster-Objekt festlegen, wenn untergeordnete Elemente diese Fensterbildlaufleiste verwenden, z.B. Proportional-Gadget mit dem Attribut `Prop.UseWinBorder`.

Offensichtlich sehen Bildlaufleisten-Gadgets in randlosen Fensterrahmen oder nicht veränderbaren Fenstern nicht gut aus.

TYP

Boolesch

ANWENDBARKEIT

I

53.31 Window.UseRightBorderScroller

BEZEICHNUNG

Window.UseRightBorderScroller – aktiviert die rechte Rahmen-Bildlaufleiste

BESCHREIBUNG

Wenn `True` gesetzt ist, wird das Fenster am rechten Rand eine Bildlaufleiste enthalten. Sie müssen dies für das Fenster-Objekt festlegen, wenn untergeordnete Elemente diese Fensterbildlaufleiste verwenden, z.B. Proportional-Gadget mit dem Attribut `Prop.UseWinBorder`.

Offensichtlich sehen Bildlaufleisten-Gadgets in randlosen Fensterrahmen oder nicht veränderbaren Fenstern nicht gut aus.

TYP

Boolesch

ANWENDBARKEIT

I

53.32 Window.Title

BEZEICHNUNG

Window.Title – setzt/ermittelt den Titel des Fensters

BESCHREIBUNG

Gibt den Titel eines Fensters zurück oder setzt ihn.

TYP

Zeichenkette

ANWENDBARKEIT

ISG

53.33 Window.Width

BEZEICHNUNG

Window.Width – setzt/ermittelt die Fensterbreite

BESCHREIBUNG

Gibt die Breite eines Fensters an. Normalerweise geben Sie hier keinen Pixelwert an, sondern stattdessen eines der folgenden Tags:

Default Berechnet die Breite aus den Standardobjektgrößen.

MinMax:<0..100>

Irgendwo zwischen der minimalen Breite (0) und der maximalen Breite (100) Ihres Fensters.

Visible:<1..100>

Prozentsatz der sichtbaren Breite des Bildschirms.

Screen:<1..100>

Prozentsatz der Gesamtbreite des Bildschirms.

Scaled Die Breite wird so eingestellt, dass Breite : Höhe == minimale Breite : minimale Höhe ist. Beachten Sie, dass die Fensterbreite und -höhe nicht beide skaliert werden können.

Standard für diesen Tag ist **Default**.

Solange Ihr Fenster eine Fenster-ID über das Attribut `Window.MuiID` hat, ist die Auswahl einer Größe nicht so wichtig. MUI wird sich immer an die letzte Position und Größe eines Fensters erinnern und diese Werte Ihrer Einstellungen einfach überschreiben. Positionierung und Dimensionierung sollten vollständig vom Benutzer kontrolliert werden, ein Programmierer muss sich nicht darum kümmern.

TYP

Nummer oder vordefiniertes Makro

ANWENDBARKEIT

IG

Anhang A Lizenzen

A.1 MUI Lizenzen

This application uses MUI - MagicUserInterface (c) Copyright 1992-97 by Stefan Stuntz. MUI is a system to generate and maintain graphical user interfaces. With the aid of a preferences program, the user of an application has the ability to customize the outfit according to his personal taste.

MUI is distributed as shareware. To obtain a complete package containing lots of examples and more information about registration please look for a file called "muiXXusr.lha" (XX means the latest version number) on your local bulletin boards or on public domain disks.

If you want to register directly, feel free to send DM 30.- or US\$ 20.- to

Stefan Stuntz
Eduard-Spranger-Straße 7
80935 München
GERMANY

Support and online registration is available at <http://www.sasg.com/>

A.2 Expat Lizenzen

Copyright (c) 1998, 1999, 2000 Thai Open Source Software Center Ltd and Clark Cooper
Copyright (c) 2001, 2002, 2003, 2004, 2005, 2006 Expat maintainers.

Permission is hereby granted, free of charge, to any person obtaining a copy of this software and associated documentation files (the "Software"), to deal in the Software without restriction, including without limitation the rights to use, copy, modify, merge, publish, distribute, sublicense, and/or sell copies of the Software, and to permit persons to whom the Software is furnished to do so, subject to the following conditions:

The above copyright notice and this permission notice shall be included in all copies or substantial portions of the Software.

THE SOFTWARE IS PROVIDED "AS IS", WITHOUT WARRANTY OF ANY KIND, EXPRESS OR IMPLIED, INCLUDING BUT NOT LIMITED TO THE WARRANTIES OF MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE AND NONINFRINGEMENT. IN NO EVENT SHALL THE AUTHORS OR COPYRIGHT HOLDERS BE LIABLE FOR ANY CLAIM, DAMAGES OR OTHER LIABILITY, WHETHER IN AN ACTION OF CONTRACT, TORT OR OTHERWISE, ARISING FROM, OUT OF OR IN CONNECTION WITH THE SOFTWARE OR THE USE OR OTHER DEALINGS IN THE SOFTWARE.

A.3 LGPL Lizenzen

GNU Lesser General Public License Version 2.1, February 1999

Copyright (C) 1991, 1999 Free Software Foundation, Inc. 59 Temple Place, Suite 330, Boston, MA 02111-1307 USA Everyone is permitted to copy and distribute verbatim copies of this license document, but changing it is not allowed.

[This is the first released version of the Lesser GPL. It also counts as the successor of the GNU Library Public License, version 2, hence the version number 2.1.]

Preamble

The licenses for most software are designed to take away your freedom to share and change it. By contrast, the GNU General Public Licenses are intended to guarantee your freedom to share and change free software—to make sure the software is free for all its users.

This license, the Lesser General Public License, applies to some specially designated software packages—typically libraries—of the Free Software Foundation and other authors who decide to use it. You can use it too, but we suggest you first think carefully about whether this license or the ordinary General Public License is the better strategy to use in any particular case, based on the explanations below.

When we speak of free software, we are referring to freedom of use, not price. Our General Public Licenses are designed to make sure that you have the freedom to distribute copies of free software (and charge for this service if you wish); that you receive source code or can get it if you want it; that you can change the software and use pieces of it in new free programs; and that you are informed that you can do these things.

To protect your rights, we need to make restrictions that forbid distributors to deny you these rights or to ask you to surrender these rights. These restrictions translate to certain responsibilities for you if you distribute copies of the library or if you modify it.

For example, if you distribute copies of the library, whether gratis or for a fee, you must give the recipients all the rights that we gave you. You must make sure that they, too, receive or can get the source code. If you link other code with the library, you must provide complete object files to the recipients, so that they can relink them with the library after making changes to the library and recompiling it. And you must show them these terms so they know their rights.

We protect your rights with a two-step method: (1) we copyright the library, and (2) we offer you this license, which gives you legal permission to copy, distribute and/or modify the library.

To protect each distributor, we want to make it very clear that there is no warranty for the free library. Also, if the library is modified by someone else and passed on, the recipients should know that what they have is not the original version, so that the original author's reputation will not be affected by problems that might be introduced by others.

Finally, software patents pose a constant threat to the existence of any free program. We wish to make sure that a company cannot effectively restrict the users of a free program by obtaining a restrictive license from a patent holder. Therefore, we insist that any patent license obtained for a version of the library must be consistent with the full freedom of use specified in this license.

Most GNU software, including some libraries, is covered by the ordinary GNU General Public License. This license, the GNU Lesser General Public License, applies to certain designated libraries, and is quite different from the ordinary General Public License. We use this license for certain libraries in order to permit linking those libraries into non-free programs.

When a program is linked with a library, whether statically or using a shared library, the combination of the two is legally speaking a combined work, a derivative of the original

library. The ordinary General Public License therefore permits such linking only if the entire combination fits its criteria of freedom. The Lesser General Public License permits more lax criteria for linking other code with the library.

We call this license the "Lesser" General Public License because it does Less to protect the user's freedom than the ordinary General Public License. It also provides other free software developers Less of an advantage over competing non-free programs. These disadvantages are the reason we use the ordinary General Public License for many libraries. However, the Lesser license provides advantages in certain special circumstances.

For example, on rare occasions, there may be a special need to encourage the widest possible use of a certain library, so that it becomes a de-facto standard. To achieve this, non-free programs must be allowed to use the library. A more frequent case is that a free library does the same job as widely used non-free libraries. In this case, there is little to gain by limiting the free library to free software only, so we use the Lesser General Public License.

In other cases, permission to use a particular library in non-free programs enables a greater number of people to use a large body of free software. For example, permission to use the GNU C Library in non-free programs enables many more people to use the whole GNU operating system, as well as its variant, the GNU/Linux operating system.

Although the Lesser General Public License is Less protective of the users' freedom, it does ensure that the user of a program that is linked with the Library has the freedom and the wherewithal to run that program using a modified version of the Library.

The precise terms and conditions for copying, distribution and modification follow. Pay close attention to the difference between a "work based on the library" and a "work that uses the library". The former contains code derived from the library, whereas the latter must be combined with the library in order to run.

TERMS AND CONDITIONS FOR COPYING, DISTRIBUTION AND MODIFICATION

0. This License Agreement applies to any software library or other program which contains a notice placed by the copyright holder or other authorized party saying it may be distributed under the terms of this Lesser General Public License (also called "this License"). Each licensee is addressed as "you".

A "library" means a collection of software functions and/or data prepared so as to be conveniently linked with application programs (which use some of those functions and data) to form executables.

The "Library", below, refers to any such software library or work which has been distributed under these terms. A "work based on the Library" means either the Library or any derivative work under copyright law: that is to say, a work containing the Library or a portion of it, either verbatim or with modifications and/or translated straightforwardly into another language. (Hereinafter, translation is included without limitation in the term "modification".)

"Source code" for a work means the preferred form of the work for making modifications to it. For a library, complete source code means all the source code for all modules it contains, plus any associated interface definition files, plus the scripts used to control compilation and installation of the library.

Activities other than copying, distribution and modification are not covered by this License; they are outside its scope. The act of running a program using the Library is not restricted,

and output from such a program is covered only if its contents constitute a work based on the Library (independent of the use of the Library in a tool for writing it). Whether that is true depends on what the Library does and what the program that uses the Library does.

1. You may copy and distribute verbatim copies of the Library's complete source code as you receive it, in any medium, provided that you conspicuously and appropriately publish on each copy an appropriate copyright notice and disclaimer of warranty; keep intact all the notices that refer to this License and to the absence of any warranty; and distribute a copy of this License along with the Library.

You may charge a fee for the physical act of transferring a copy, and you may at your option offer warranty protection in exchange for a fee.

2. You may modify your copy or copies of the Library or any portion of it, thus forming a work based on the Library, and copy and distribute such modifications or work under the terms of Section 1 above, provided that you also meet all of these conditions:

a) The modified work must itself be a software library.

b) You must cause the files modified to carry prominent notices stating that you changed the files and the date of any change.

c) You must cause the whole of the work to be licensed at no charge to all third parties under the terms of this License.

d) If a facility in the modified Library refers to a function or a table of data to be supplied by an application program that uses the facility, other than as an argument passed when the facility is invoked, then you must make a good faith effort to ensure that, in the event an application does not supply such function or table, the facility still operates, and performs whatever part of its purpose remains meaningful.

(For example, a function in a library to compute square roots has a purpose that is entirely well-defined independent of the application. Therefore, Subsection 2d requires that any application-supplied function or table used by this function must be optional: if the application does not supply it, the square root function must still compute square roots.)

These requirements apply to the modified work as a whole. If identifiable sections of that work are not derived from the Library, and can be reasonably considered independent and separate works in themselves, then this License, and its terms, do not apply to those sections when you distribute them as separate works. But when you distribute the same sections as part of a whole which is a work based on the Library, the distribution of the whole must be on the terms of this License, whose permissions for other licensees extend to the entire whole, and thus to each and every part regardless of who wrote it.

Thus, it is not the intent of this section to claim rights or contest your rights to work written entirely by you; rather, the intent is to exercise the right to control the distribution of derivative or collective works based on the Library.

In addition, mere aggregation of another work not based on the Library with the Library (or with a work based on the Library) on a volume of a storage or distribution medium does not bring the other work under the scope of this License.

3. You may opt to apply the terms of the ordinary GNU General Public License instead of this License to a given copy of the Library. To do this, you must alter all the notices that refer to this License, so that they refer to the ordinary GNU General Public License, version 2, instead of to this License. (If a newer version than version 2 of the ordinary GNU General

Public License has appeared, then you can specify that version instead if you wish.) Do not make any other change in these notices.

Once this change is made in a given copy, it is irreversible for that copy, so the ordinary GNU General Public License applies to all subsequent copies and derivative works made from that copy.

This option is useful when you wish to copy part of the code of the Library into a program that is not a library.

4. You may copy and distribute the Library (or a portion or derivative of it, under Section 2) in object code or executable form under the terms of Sections 1 and 2 above provided that you accompany it with the complete corresponding machine-readable source code, which must be distributed under the terms of Sections 1 and 2 above on a medium customarily used for software interchange.

If distribution of object code is made by offering access to copy from a designated place, then offering equivalent access to copy the source code from the same place satisfies the requirement to distribute the source code, even though third parties are not compelled to copy the source along with the object code.

5. A program that contains no derivative of any portion of the Library, but is designed to work with the Library by being compiled or linked with it, is called a "work that uses the Library". Such a work, in isolation, is not a derivative work of the Library, and therefore falls outside the scope of this License.

However, linking a "work that uses the Library" with the Library creates an executable that is a derivative of the Library (because it contains portions of the Library), rather than a "work that uses the library". The executable is therefore covered by this License. Section 6 states terms for distribution of such executables.

When a "work that uses the Library" uses material from a header file that is part of the Library, the object code for the work may be a derivative work of the Library even though the source code is not. Whether this is true is especially significant if the work can be linked without the Library, or if the work is itself a library. The threshold for this to be true is not precisely defined by law.

If such an object file uses only numerical parameters, data structure layouts and accessors, and small macros and small inline functions (ten lines or less in length), then the use of the object file is unrestricted, regardless of whether it is legally a derivative work. (Executables containing this object code plus portions of the Library will still fall under Section 6.)

Otherwise, if the work is a derivative of the Library, you may distribute the object code for the work under the terms of Section 6. Any executables containing that work also fall under Section 6, whether or not they are linked directly with the Library itself.

6. As an exception to the Sections above, you may also combine or link a "work that uses the Library" with the Library to produce a work containing portions of the Library, and distribute that work under terms of your choice, provided that the terms permit modification of the work for the customer's own use and reverse engineering for debugging such modifications.

You must give prominent notice with each copy of the work that the Library is used in it and that the Library and its use are covered by this License. You must supply a copy of this License. If the work during execution displays copyright notices, you must include the

copyright notice for the Library among them, as well as a reference directing the user to the copy of this License. Also, you must do one of these things:

- a) Accompany the work with the complete corresponding machine-readable source code for the Library including whatever changes were used in the work (which must be distributed under Sections 1 and 2 above); and, if the work is an executable linked with the Library, with the complete machine-readable "work that uses the Library", as object code and/or source code, so that the user can modify the Library and then relink to produce a modified executable containing the modified Library. (It is understood that the user who changes the contents of definitions files in the Library will not necessarily be able to recompile the application to use the modified definitions.)
- b) Use a suitable shared library mechanism for linking with the Library. A suitable mechanism is one that (1) uses at run time a copy of the library already present on the user's computer system, rather than copying library functions into the executable, and (2) will operate properly with a modified version of the library, if the user installs one, as long as the modified version is interface-compatible with the version that the work was made with.
- c) Accompany the work with a written offer, valid for at least three years, to give the same user the materials specified in Subsection 6a, above, for a charge no more than the cost of performing this distribution.
- d) If distribution of the work is made by offering access to copy from a designated place, offer equivalent access to copy the above specified materials from the same place.
- e) Verify that the user has already received a copy of these materials or that you have already sent this user a copy.

For an executable, the required form of the "work that uses the Library" must include any data and utility programs needed for reproducing the executable from it. However, as a special exception, the materials to be distributed need not include anything that is normally distributed (in either source or binary form) with the major components (compiler, kernel, and so on) of the operating system on which the executable runs, unless that component itself accompanies the executable.

It may happen that this requirement contradicts the license restrictions of other proprietary libraries that do not normally accompany the operating system. Such a contradiction means you cannot use both them and the Library together in an executable that you distribute.

7. You may place library facilities that are a work based on the Library side-by-side in a single library together with other library facilities not covered by this License, and distribute such a combined library, provided that the separate distribution of the work based on the Library and of the other library facilities is otherwise permitted, and provided that you do these two things:

- a) Accompany the combined library with a copy of the same work based on the Library, uncombined with any other library facilities. This must be distributed under the terms of the Sections above.
- b) Give prominent notice with the combined library of the fact that part of it is a work based on the Library, and explaining where to find the accompanying uncombined form of the same work.

8. You may not copy, modify, sublicense, link with, or distribute the Library except as expressly provided under this License. Any attempt otherwise to copy, modify, sublicense,

link with, or distribute the Library is void, and will automatically terminate your rights under this License. However, parties who have received copies, or rights, from you under this License will not have their licenses terminated so long as such parties remain in full compliance.

9. You are not required to accept this License, since you have not signed it. However, nothing else grants you permission to modify or distribute the Library or its derivative works. These actions are prohibited by law if you do not accept this License. Therefore, by modifying or distributing the Library (or any work based on the Library), you indicate your acceptance of this License to do so, and all its terms and conditions for copying, distributing or modifying the Library or works based on it.

10. Each time you redistribute the Library (or any work based on the Library), the recipient automatically receives a license from the original licensor to copy, distribute, link with or modify the Library subject to these terms and conditions. You may not impose any further restrictions on the recipients' exercise of the rights granted herein. You are not responsible for enforcing compliance by third parties with this License.

11. If, as a consequence of a court judgment or allegation of patent infringement or for any other reason (not limited to patent issues), conditions are imposed on you (whether by court order, agreement or otherwise) that contradict the conditions of this License, they do not excuse you from the conditions of this License. If you cannot distribute so as to satisfy simultaneously your obligations under this License and any other pertinent obligations, then as a consequence you may not distribute the Library at all. For example, if a patent license would not permit royalty-free redistribution of the Library by all those who receive copies directly or indirectly through you, then the only way you could satisfy both it and this License would be to refrain entirely from distribution of the Library.

If any portion of this section is held invalid or unenforceable under any particular circumstance, the balance of the section is intended to apply, and the section as a whole is intended to apply in other circumstances.

It is not the purpose of this section to induce you to infringe any patents or other property right claims or to contest validity of any such claims; this section has the sole purpose of protecting the integrity of the free software distribution system which is implemented by public license practices. Many people have made generous contributions to the wide range of software distributed through that system in reliance on consistent application of that system; it is up to the author/donor to decide if he or she is willing to distribute software through any other system and a licensee cannot impose that choice.

This section is intended to make thoroughly clear what is believed to be a consequence of the rest of this License.

12. If the distribution and/or use of the Library is restricted in certain countries either by patents or by copyrighted interfaces, the original copyright holder who places the Library under this License may add an explicit geographical distribution limitation excluding those countries, so that distribution is permitted only in or among countries not thus excluded. In such case, this License incorporates the limitation as if written in the body of this License.

13. The Free Software Foundation may publish revised and/or new versions of the Lesser General Public License from time to time. Such new versions will be similar in spirit to the present version, but may differ in detail to address new problems or concerns.

Each version is given a distinguishing version number. If the Library specifies a version number of this License which applies to it and "any later version", you have the option of following the terms and conditions either of that version or of any later version published by the Free Software Foundation. If the Library does not specify a license version number, you may choose any version ever published by the Free Software Foundation.

14. If you wish to incorporate parts of the Library into other free programs whose distribution conditions are incompatible with these, write to the author to ask for permission. For software which is copyrighted by the Free Software Foundation, write to the Free Software Foundation; we sometimes make exceptions for this. Our decision will be guided by the two goals of preserving the free status of all derivatives of our free software and of promoting the sharing and reuse of software generally.

NO WARRANTY

15. BECAUSE THE LIBRARY IS LICENSED FREE OF CHARGE, THERE IS NO WARRANTY FOR THE LIBRARY, TO THE EXTENT PERMITTED BY APPLICABLE LAW. EXCEPT WHEN OTHERWISE STATED IN WRITING THE COPYRIGHT HOLDERS AND/OR OTHER PARTIES PROVIDE THE LIBRARY "AS IS" WITHOUT WARRANTY OF ANY KIND, EITHER EXPRESSED OR IMPLIED, INCLUDING, BUT NOT LIMITED TO, THE IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS FOR A PARTICULAR PURPOSE. THE ENTIRE RISK AS TO THE QUALITY AND PERFORMANCE OF THE LIBRARY IS WITH YOU. SHOULD THE LIBRARY PROVE DEFECTIVE, YOU ASSUME THE COST OF ALL NECESSARY SERVICING, REPAIR OR CORRECTION.

16. IN NO EVENT UNLESS REQUIRED BY APPLICABLE LAW OR AGREED TO IN WRITING WILL ANY COPYRIGHT HOLDER, OR ANY OTHER PARTY WHO MAY MODIFY AND/OR REDISTRIBUTE THE LIBRARY AS PERMITTED ABOVE, BE LIABLE TO YOU FOR DAMAGES, INCLUDING ANY GENERAL, SPECIAL, INCIDENTAL OR CONSEQUENTIAL DAMAGES ARISING OUT OF THE USE OR INABILITY TO USE THE LIBRARY (INCLUDING BUT NOT LIMITED TO LOSS OF DATA OR DATA BEING RENDERED INACCURATE OR LOSSES SUSTAINED BY YOU OR THIRD PARTIES OR A FAILURE OF THE LIBRARY TO OPERATE WITH ANY OTHER SOFTWARE), EVEN IF SUCH HOLDER OR OTHER PARTY HAS BEEN ADVISED OF THE POSSIBILITY OF SUCH DAMAGES.

END OF TERMS AND CONDITIONS

Index

A

Application.AboutMUI 47
 Application.AboutMUIRoyale..... 47
 Application.AddWindow 47
 Application.Base 48
 Application.ConfigChange..... 48
 Application.DoubleStart..... 49
 Application.DropObject 49
 Application.HelpFile 49
 Application.Icon 50
 Application.Iconified 50
 Application.Load 51
 Application.Menustrip 52
 Application.OpenConfigWindow..... 52
 Application.RemoveWindow..... 53
 Application.Save 53
 Application.SingleTask..... 54
 Application.Sleep 55
 Application.WindowList 55
 Area.Background 57
 Area.BackgroundBrush 58
 Area.BackgroundImage 59
 Area.BackgroundRGB 59
 Area.BottomEdge 60
 Area.ContextMenu 60
 Area.ContextMenuTrigger..... 61
 Area.ControlChar 61
 Area.CycleChain 61
 Area.Disabled 62
 Area.FixHeight 62
 Area.FixHeightTxt 63
 Area.FixWidth 63
 Area.FixWidthTxt 63
 Area.Font 64
 Area.Frame 64
 Area.FramePhantomHoriz 65
 Area.FrameTitle 66
 Area.Height 66
 Area.Hide 66
 Area.HorizWeight 67
 Area.InnerBottom 67
 Area.InnerLeft 67
 Area.InnerRight 68
 Area.InnerTop 68
 Area.InputMode 68
 Area.LeftEdge 69
 Area.MaxHeight 69
 Area.MaxWidth 70
 Area.Pressed 70
 Area.RightEdge 71
 Area.Selected 71
 Area.ShortHelp 72
 Area.ShowSelState 72
 Area.TopEdge 72

Area.VertWeight 73
 Area.Weight 73
 Area.Width 74

B

Busy.Move 75
 Busy.Speed 75
 Button.HiChar 77
 Button.Label 77
 Button.NoAutoKey 77
 Button.Pressed 78
 Button.Selected 78
 Button.Toggle 78

C

Checkmark.Selected 81
 Coloradjust.RGB 83
 Colorfield.RGB 85
 Cycle.Active 87

D

Dirlist.AcceptPattern 90
 Dirlist.Directory 90
 Dirlist.DrawersOnly 91
 Dirlist.FilesOnly 91
 Dirlist.FilterDrawers 91
 Dirlist.MultiSelDirs 92
 Dirlist.NumBytes 92
 Dirlist.NumDrawers 92
 Dirlist.NumFiles 92
 Dirlist.Path 93
 Dirlist.RejectIcons 93
 Dirlist.RejectPattern 93
 Dirlist.ReRead 94
 Dirlist.SortDirs 94
 Dirlist.SortHighLow 94
 Dirlist.SortType 95
 Dirlist.Status 95
 Dirlist.Title 96

F

Floattext.Justify 97
 Floattext.TabSize 97
 Floattext.Text 97

G

Gauge.Current	99
Gauge.Divide	99
Gauge.Horiz	99
Gauge.InfoText	100
Gauge.Max	100
Group.ActivePage	104
Group.AddHead	104
Group.AddTail	105
Group.ChildList	106
Group.Columns	106
Group.ExitChange	107
Group.HorizSpacing	108
Group.InitChange	108
Group.Insert	108
Group.PageMode	109
Group.Remove	110
Group.SameHeight	111
Group.SameSize	111
Group.SameWidth	111
Group.Spacing	112
Group.Title	112
Group.VertSpacing	113

H

Hollywood.Display	117
Hollywood.MaxHeight	117
Hollywood.MaxWidth	118
Hollywood.MinHeight	118
Hollywood.MinWidth	118
HSpace.Width	115

I

Image.Brush	121
Image.FreeHoriz	121
Image.FreeVert	121
Image.Source	122
Image.State	123
Imagebutton.Pressed	125

L

Label.Centered	127
Label.DoubleFrame	127
Label.FreeVert	127
Label.Key	128
Label.LeftAligned	128
Label.SingleFrame	128
Listtree.Active	132
Listtree.AutoLineHeight	132
Listtree.Close	133
Listtree.DoubleClick	133
Listtree.DragDropSort	134
Listtree.EmptyNodes	134
Listtree.Exchange	134

Listtree.FindName	135
Listtree.GetEntry	137
Listtree.Insert	138
Listtree.MinLineHeight	140
Listtree.Move	140
Listtree.Open	141
Listtree.Quiet	142
Listtree.Remove	142
Listtree.Rename	143
Listtree.Sort	143
Listtree.SortHook	144
Listtree.Title	144
Listtreenode.Frozen	147
Listtreenode.Name	147
Listtreenode.NoSign	147
Listtreenode.Open	148
Listview.Active	150
Listview.AdjustHeight	150
Listview.AdjustWidth	150
Listview.AutoLineHeight	151
Listview.AutoVisible	151
Listview.Clear	151
Listview.ClickColumn	152
Listview.DefClickColumn	152
Listview.DoubleClick	152
Listview.Entries	153
Listview.Exchange	153
Listview.First	154
Listview.GetEntry	154
Listview.GetSelection	154
Listview.Input	155
Listview.Insert	155
Listview.InsertPosition	156
Listview.Jump	156
Listview.MinLineHeight	157
Listview.Move	157
Listview.MultiSelect	158
Listview.Quiet	158
Listview.Remove	159
Listview.Rename	159
Listview.ScrollerPos	160
Listview.Select	160
Listview.SelectChange	161
Listview.Sort	161
Listview.SortFunc	162
Listview.TitleClick	163
Listview.Visible	163
Listviewcolumn.Bar	165
Listviewcolumn.Col	165
Listviewcolumn.Delta	165
Listviewcolumn.Hidden	166
Listviewcolumn.MaxWidth	167
Listviewcolumn.MinWidth	166
Listviewcolumn.PreParse	167
Listviewcolumn.Title	167
Listviewcolumn.Weight	168

M

Menu.AddHead	169
Menu.AddTail	169
Menu.Disabled	170
Menu.Insert	170
Menu.Remove	170
Menu.Title	171
MenuItem.CommandString	173
MenuItem.Disabled	173
MenuItem.Exclude	174
MenuItem.Selected	174
MenuItem.Shortcut	174
MenuItem.Title	174
MenuItem.Type	175
MenuStrip.AddHead	178
MenuStrip.AddTail	178
MenuStrip.Insert	179
MenuStrip.Remove	179
mui.CreateGUI	37
mui.CreateObject	38
mui.DoMethod	39
mui.FreeGUI	40
mui.FreeImage	40
mui.FreeObject	41
mui.Get	41
mui.HaveObject	42
mui.IsVersion4	42
mui.Notify	43
mui.Request	44
mui.Set	45
Popfile.Contents	194
Popfile.Pattern	195
Popfile.RejectIcons	195
Popfile.RejectPattern	195
Popfile.SaveMode	195
Popfile.ShowPattern	196
Popfile.Title	196
Popfont.Acknowledge	197
Popfont.Active	197
Popfont.AdvanceOnCR	198
Popfont.Contents	198
Popfont.FixedWidthOnly	198
Popfont.MaxHeight	199
Popfont.MinHeight	199
Popfont.Title	199
Poplist.Acknowledge	201
Poplist.AdvanceOnCR	201
Poplist.Contents	202
Poppen.RGB	203
Poppen.Title	203
Prop.Decrease	205
Prop.Entries	205
Prop.First	205
Prop.Horiz	206
Prop.Increase	206
Prop.Slider	206
Prop.UseWinBorder	207
Prop.Visible	207

N

Notify.AppMessage	181
Notify.Class	181
Notify.ExportID	182
Notify.HelpLine	182
Notify.HelpNode	182
Notify.ID	183
Notify.MUIClass	183
Notify.NoNotify	183
Notify.NotifyData	184
Notify.Revision	185
Notify.UserData	185
Notify.Version	185

P

Popdrawer.Acknowledge	189
Popdrawer.Active	189
Popdrawer.AdvanceOnCR	190
Popdrawer.Contents	190
Popdrawer.SaveMode	190
Popdrawer.Title	191
Popfile.AcceptPattern	193
Popfile.Acknowledge	193
Popfile.Active	193
Popfile.AdvanceOnCR	194

R

Radio.Active	209
Rectangle.BarTitle	211
Rectangle.HBar	211
Rectangle.VBar	212
Register.ActivePage	213
Register.AddPage	214
Register.Closable	214
Register.ClosePage	214
Register.CloseRequest	215
Register.GetPageID	215
Register.InsertPage	216
Register.Pages	216
Register.Position	217

S

Scale.Horiz	219
Scrollbar.IncDecSize	221
Scrollbar.Type	221
Scrollgroup.FreeHoriz	223
Scrollgroup.FreeVert	223
Scrollgroup.UseWinBorder	224
Slider.Format	225
Slider.Horiz	225
Slider.Level	226
Slider.Max	226
Slider.Min	226
Slider.Pressed	227
Slider.Quiet	227
Slider.Reverse	227
Slider.Stringify	227
String.Accept	229
String.Acknowledge	229
String.AdvanceOnCR	230
String.Contents	230
String.Copy	230
String.CursorPos	231
String.Cut	231
String.Insert	231
String.MarkEnd	232
String.MarkStart	232
String.MaxLen	232
String.Paste	233
String.Redo	233
String.Reject	233
String.Secret	234
String.Undo	234

T

Text.Contents	235
Text.HiChar	235
Text.PreParse	236
Text.SetMax	236
Text.SetMin	236
Text.SetVMax	237
Texteditor.ActiveObjectOnClick	239
Texteditor.Align	239
Texteditor.AreaMarked	240
Texteditor.AutoClip	240
Texteditor.Clear	241
Texteditor.Color	241
Texteditor.ColorMap	242
Texteditor.Columns	242
Texteditor.Contents	242
Texteditor.ConvertTabs	243
Texteditor.Copy	243
Texteditor.CursorX	243
Texteditor.CursorY	244
Texteditor.Cut	244
Texteditor.Erase	244
Texteditor.ExportHook	244
Texteditor.ExportWrap	245

Texteditor.FixedFont	246
Texteditor.GetSelection	246
Texteditor.GetText	246
Texteditor.HasChanged	247
Texteditor.ImportHook	247
Texteditor.ImportWrap	249
Texteditor.Insert	250
Texteditor.Mark	250
Texteditor.MarkAll	251
Texteditor.MarkNone	251
Texteditor.Paste	251
Texteditor.PasteColors	251
Texteditor.PasteStyles	252
Texteditor.ReadOnly	252
Texteditor.Redo	252
Texteditor.RedoAvailable	253
Texteditor.Replace	253
Texteditor.Rows	253
Texteditor.Scrollbar	254
Texteditor.Search	254
Texteditor.SetBold	255
Texteditor.SetColor	255
Texteditor.SetItalic	256
Texteditor.SetUnderline	257
Texteditor.StyleBold	257
Texteditor.StyleItalic	258
Texteditor.StyleUnderline	258
Texteditor.TabSize	258
Texteditor.Undo	259
Texteditor.UndoAvailable	259
Texteditor.UndoLevels	259
Texteditor.WrapBorder	260
Texteditor.WrapMode	260
Texteditor.WrapWords	261
Toolbar.Active	263
Toolbar.BarPos	263
Toolbar.BarSpacer	264
Toolbar.BarSpacerSpacing	264
Toolbar.Borderless	265
Toolbar.BottomBarFrameSpacing	265
Toolbar.BottomInnerSpacing	265
Toolbar.Columns	266
Toolbar.DisMode	266
Toolbar.DontMove	267
Toolbar.EnableKeys	267
Toolbar.Frame	267
Toolbar.Free	268
Toolbar.FreeHoriz	268
Toolbar.FreeVert	268
Toolbar.Horiz	269
Toolbar.HorizInnerSpacing	269
Toolbar.HorizSpacing	270
Toolbar.HorizTextGfxSpacing	270
Toolbar.IgnoreAppearance	270
Toolbar.LabelPos	271
Toolbar.LeftBarFrameSpacing	271
Toolbar.MouseOver	271
Toolbar.NtRaiseActive	272

Toolbar.Raised	272
Toolbar.RightBarFrameSpacing	272
Toolbar.Rows	273
Toolbar.Scale	273
Toolbar.Scaled	273
Toolbar.SpacersSize	274
Toolbar.SpecialSelect	274
Toolbar.Sunny	275
Toolbar.TextOverUseShine	275
Toolbar.TopBarFrameSpacing	275
Toolbar.TopInnerSpacing	276
Toolbar.VertSpacing	276
Toolbar.VertTextGfxSpacing	276
Toolbar.ViewMode	277
Toolbarbutton.Disabled	279
Toolbarbutton.DisImage	279
Toolbarbutton.Exclude	279
Toolbarbutton.Hide	280
Toolbarbutton.Image	280
Toolbarbutton.Immediate	280
Toolbarbutton.NoClick	281
Toolbarbutton.Pressed	281
Toolbarbutton.Selected	281
Toolbarbutton.SelImage	282
Toolbarbutton.ShortHelp	282
Toolbarbutton.Sleep	282
Toolbarbutton.Toggle	283

V

Virtgroup.Height	287
Virtgroup.Horiz	287
Virtgroup.Input	287
Virtgroup.Left	288
Virtgroup.Top	288
Virtgroup.Width	288

Volumelist.Title	291
VSpace.Height	285

W

Window.Activate	293
Window.ActiveObject	293
Window.AppWindow	294
Window.Borderless	294
Window.CloseGadget	294
Window.CloseRequest	295
Window.DefaultObject	295
Window.DepthGadget	295
Window.DragBar	296
Window.Height	296
Window.LeftEdge	297
Window.Menustrip	297
Window.MouseObject	298
Window.MuiID	298
Window.NeedsMouseObject	298
Window.NoMenus	299
Window.Open	299
Window.PubScreen	300
Window.ScreenTitle	300
Window.ScreenToBack	300
Window.ScreenToFront	301
Window.SizeGadget	301
Window.Sleep	301
Window.Snapshot	302
Window.Title	304
Window.ToBack	302
Window.ToFront	302
Window.TopEdge	303
Window.UseBottomBorderScroller	303
Window.UseLeftBorderScroller	304
Window.UseRightBorderScroller	304
Window.Width	305

