

DEVELOPER GUIDE | PUBLIC
Document Version: 7.60 PL2 – 2019-06-28

SAP GUI Scripting API

© 2019 SAP SE or an SAP affiliate company. All rights reserved.

THE BEST RUN SAP

Content

1 SAP GUI Scripting API. 5

1.1 Requirements and Remarks. 7

1.2 Objects. 8

GuiAbapEditor Object. 16

GuiApoGrid Object. 29

GuiApplication Object 37

GuiBarChart Object 47

GuiBox Object. 51

GuiButton Object. 53

GuiCalendar Object 55

GuiChart Object. 61

GuiCheckBox Object. 65

GuiCollection Collection. 68

GuiColorSelector Object. 69

GuiComboBox Object 72

GuiComboBoxControl Object. 75

GuiComboBoxEntry Object. 78

GuiComponent Object. 78

GuiComponentCollection Collection. 79

GuiConnection Object. 81

GuiContainer Object 83

GuiContainerShell Object. 84

GuiContextMenu Object. 87

GuiCTextField Object. 88

GuiCustomControl Object 91

GuiDialogShell Object. 93

GuiEAIViewer2D Object. 96

GuiEAIViewer3D Object. 98

GuiEnum Object. 101

GuiFrameWindow Object. 101

GuiGOSShell Object. 108

GuiGraphAdapt Object. 110

GuiGridView Object. 113

GuiHTMLViewer Object. 128

GuiInputFieldControl Object. 132

GuiLabel Object. 134

GuiMainWindow Object. 141
GuiMap Object. 147
GuiMenu Object. 150
GuiMenubar Object. 151
GuiMessageWindow. 153
GuiModalWindow Object. 155
GuiNetChart Object. 158
GuiOfficeIntegration Object. 161
GuiOkCodeField Object. 164
GuiPasswordField Object. 166
GuiPicture Object. 169
GuiRadioButton Object. 173
GuiSapChart Object. 177
GuiScrollbar Object. 179
GuiScrollContainer Object. 179
GuiSession Object. 181
GuiSessionInfo Object. 192
GuiShell Object. 195
GuiSimpleContainer Object. 198
GuiSplit Object. 201
GuiSplitterContainer Object 204
GuiStage Object. 206
GuiStatusbar Object. 209
GuiStatusPane Object. 213
GuiTab Object. 214
GuiTableColumn Collection. 216
GuiTableControl Object. 218
GuiTableRow Collection. 222
GuiTabStrip Object. 224
GuiTextedit Object. 227
GuiTextField. 232
GuiTitlebar Object 236
GuiToolbar Object. 238
GuiToolbarControl. 240
GuiTree Object. 244
GuiUserArea Object. 257
GuiUtils Object 260
GuiVComponent Object. 263
GuiVContainer Object 268
GuiVHViewSwitch Object. 270
1.3 Events. 273

Change Event - Additional Remarks. . . . 276
1.4 Enumerations. . . . 279
GuiComponentType. . . . 279
GuiErrorType. . . . 280
GuiEventType. . . . 282
GuiImageType. . . . 283
GuiMagicDispIDs. . . . 283
GuiMessageBoxOption. . . . 293
GuiMessageBoxResult. . . . 293
GuiMessageBoxType. . . . 294
GuiScrollbarType. . . . 294
GuiTableSelectionType. . . . 295

2 SAP GUI Scripting ROT Entry Helper. . . . 296
2.1 SapGuiAuto Object. . . . 296

3 SAP GUI Scripting ROT Access Helper. . . . 298
3.1 CSapROTWrapper Object. . . . 298

1 SAP GUI Scripting API

Purpose

In SAP R/3 4.6C active elements ("controls") where introduced in SAP GUI and consumed by many applications. Since existing automation approaches at that point of time where not able to handle controls and therefore could no longer be used to automate user interaction with modern applications, SAP GUI Scripting was created.

Examples of affected use cases are:

- Automatic testing of SAP functionality
- Customized front end applications replacing the SAP GUI
- Tools to customize applications on the SAP GUI level -> GuiXT
- E-Learning applications that guide a user through SAP transactions

Integration

Many of the available SAP GUI controls were designed exclusively with user interaction in mind. As their business functionality is closely coupled with the user interface they cannot be instantiated outside the SAP GUI in a batch-like fashion.

We therefore decided not to add the business functionality of the SAP GUI controls to a low-level integration component. Instead the controls run within the SAP GUI, which itself exposes a new interface allowing the automation of tasks.

Features

We developed an object model representing the SAP GUI at runtime as a hierarchy of objects. Most of these expose an interface to an element of the user interface. These interfaces can be used to perform all the actions a user could do with the given element. In addition we offer outgoing interfaces through which an external application can receive notifications about events occurring within the SAP GUI.

The SAP GUI Scripting API is available in SAP GUI for Windows and SAP GUI for Java. However, SAP GUI for Java does not support the complete set of objects / methods / properties available in SAP GUI for Windows. An example is the GuiOfficeIntegration object, which is only available in SAP GUI for Windows ("SAP Desktop Office Integration").

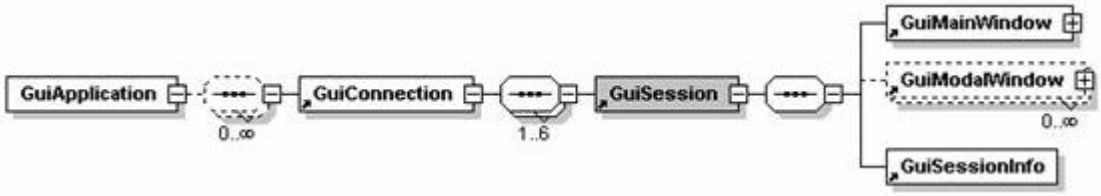
Available uses for the scripting component include

- Listening to the actions a user performs in the SAP GUI and record them as a script
- Running a script that emulates user interaction
- Logging the SAP system information, such as response time

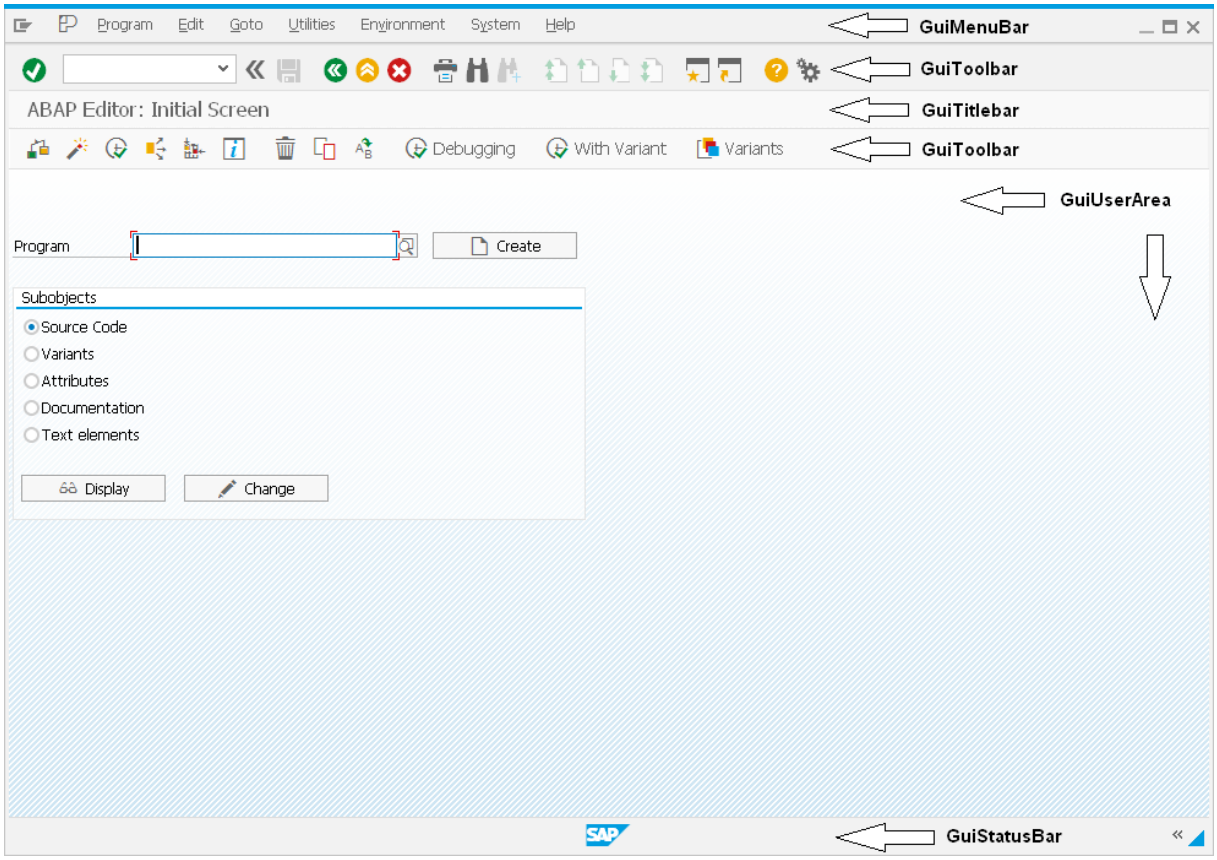
Runtime hierarchy overview

Top level administrative objects

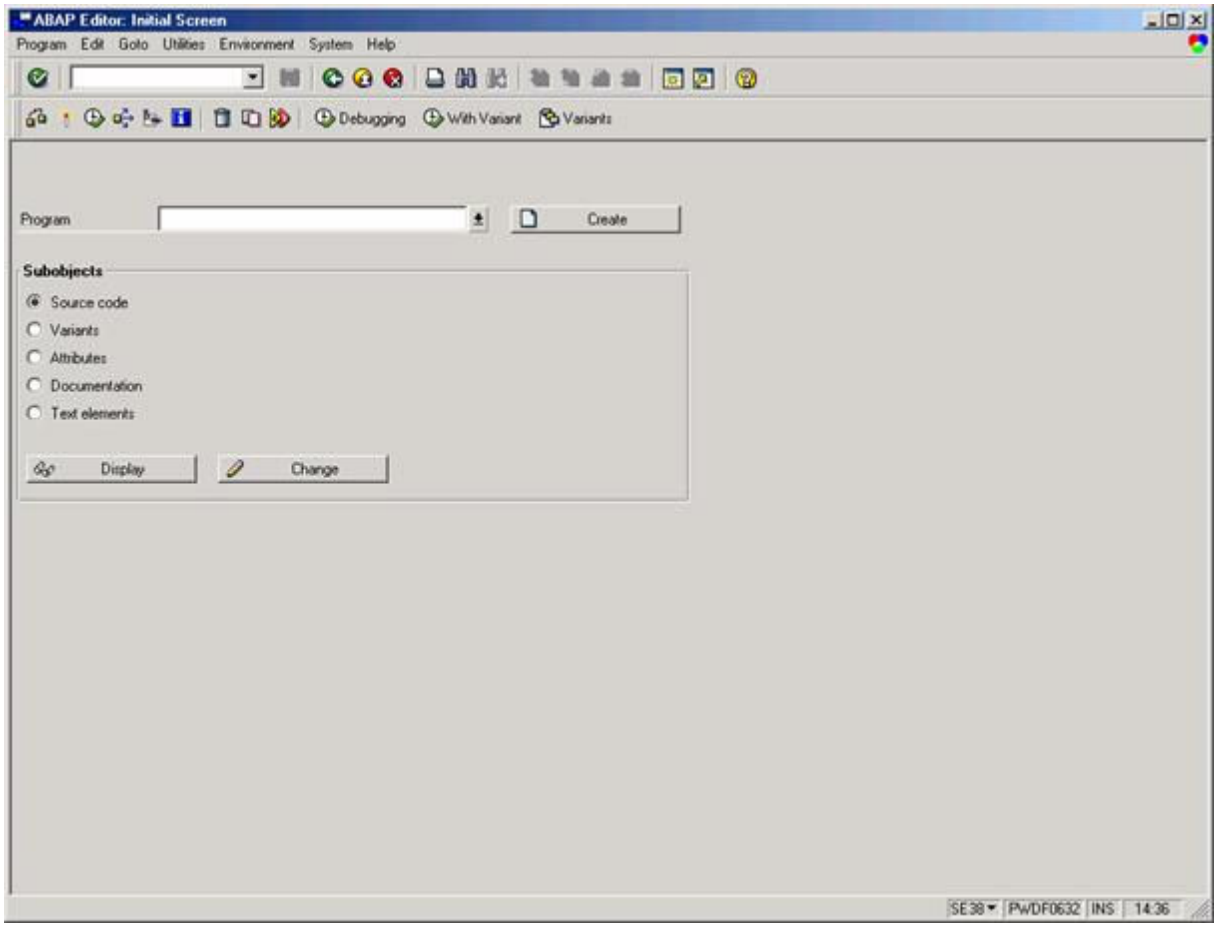
All objects defined in the scripting component's object model are available at runtime as members of a hierarchical tree with the root object being GuiApplication.



The children of the GuiMainWindow are easily identified straightforward.



Please note that the availability of some of these objects depends on the design mode used. The titlebar, for example, is available only in New Visual Design, not in classic mode, as can be seen in the following screenshot.



1.1 Requirements and Remarks

SAP System

Scripting support is available for SAP R/3 3.1I, 4.0B, 4.5B, 4.6B, 4.6C, 4.6D, and for all products based on later versions of SAP_BASIS.

For the releases 3.1I to 6.10 ABAP support packages and SAP kernel patches are available to add the support, while they are already part of 6.20 and later releases. SAP Note 480149 lists the required patch levels.

SAP GUI

The scripting interface can be installed with the SAP GUI release 6.20 and later releases. This document is based on SAP GUI for Windows 7.60. Some of the functionality described here may be missing from older versions of SAP GUI.

Remarks

SAP Notes on SAP GUI Scripting

480149	Describes the ABAP and kernel patch level requirements
587202	Limitations of SAP GUI Scripting
619459	SAP GUI Scripting support of SAP applications
692245	Additional server based security options for Scripting
839115	Load test with SAP GUI scripting
983990	SAP GUI Scripting security: Enable Scripting per system user
1441550	SAP GUI scripting: Sample applications

SAP Support Component for SAP GUI Scripting

- BC-FES-GUI

1.2 Objects

Classes

Class Module	Description
GuiAbapEditor Object [page 16]	The GuiAbapEditor object represents the new ABAP editor control available as of SAP_BASIS release 6.20 (see also SAP Note 930742). GuiAbapEditor extends GuiShell.
GuiApoGrid Object [page 29]	The GuiApoGrid object is component, which is similar to GuiGridView, but which contains additional SCM specific functions (used for example in transaction /sapapo/sdp94). GuiApoGrid extends GuiShell.

Class Module	Description
GuiApplication Object [page 37]	The GuiApplication represents the process in which all SAP GUI activity takes place. If the scripting component is accessed by attaching to an SAPlogon process, then GuiApplication will represent SAPlogon. GuiApplication is a creatable class. However, there must be only one component of this type in any process. GuiApplication extends GuiContainer.
GuiBarChart Object [page 47]	The GuiBarChart is a powerful tool to display and modify time scale diagrams. The object is of a very technical nature. It should only be used for recording and playback, as most of the parameters cannot be determined in any other way. GuiBarChart extends GuiShell.
GuiBox Object [page 51]	A GuiBox is a simple frame with a name. The items inside the frame are not children of the box. GuiBox extends GuiVComponent. The type prefix is box. The name property is the ABAP fieldname.
GuiButton Object [page 53]	GuiButton represents all push buttons that are on dynpros, the toolbar or in table controls. GuiButton extends GuiVComponent. The type prefix is btn, the name property is the fieldname taken from the SAP data dictionary There is one exception: for tabstrip buttons, it is the button id set in screen painter that is taken from the SAP data dictionary.
GuiCalendar Object [page 55]	The calendar control can be used to select single dates or periods of time. GuiCalendar extends GuiShell.
GuiChart Object [page 61]	The GuiChart object is of a very technical nature. It should only be used for recording and playback, as most of the parameters cannot be determined in any other way.
GuiCheckBox Object [page 65]	GuiCheckBox extends GuiVComponent. The type prefix is chk, the name is the fieldname taken from the SAP data dictionary.
GuiCollection Collection [page 68]	GuiCollection is similar to GuiComponentCollection, but its members are not necessarily extensions of GuiComponent. It can be used to pass a collection as a parameter to functions of scriptable objects. An object of this class is created by calling the CreateGuiCollection function of the GuiApplication.
GuiColorSelector Object [page 69]	GuiColorSelector displays a set of colors for selection. It extends GuiShell.
GuiComboBox Object [page 72]	The GuiComboBox looks somewhat similar to GuiCTextField, but has a completely different implementation. While pressing the combo box button of a GuiCTextField will open a new dynpro or control in which a selection can be made, GuiComboBox retrieves all possible choices on initialization from the server, so the selection is done solely on the client. GuiComboBox extends GuiVComponent. The type prefix is cmb, the name is the fieldname taken from the SAP data dictionary.

Class Module	Description
GuiComboBoxControl Object [page 75]	GuiComboboxControl offers a combo box that can be used inside control containers (unlike the Dynpro element represented by GuiComboBox). GuiComboBoxControl extends GuiShell.
GuiComboBoxEntry Object [page 78]	Members of the Entries collection of a GuiComboBox are of type GuiComBoxEntry.
GuiComponent Object [page 78]	GuiComponent is the base class for most classes in the Scripting API. It was designed to allow generic programming, meaning you can work with objects without knowing their exact type.
GuiComponentCollection Collection [page 79]	The GuiComponentCollection is used for collections elements such as the `children` property of containers. Each element of the collection is an extension of GuiComponent.
GuiConnection Object [page 81]	A GuiConnection represents the connection between SAP GUI and an application server. Connections can be opened from SAPlogon or from GuiApplication's openConnection and openConnectionByConnectionString methods. GuiConnection extends GuiContainer. The type prefix for GuiConnection is con, the name is con plus the connection number in square brackets.
GuiContainer Object [page 83]	This interface resembles GuiVContainer. The only difference is that it is not intended for visual objects but rather administrative objects such as connections or sessions. Objects exposing this interface will therefore support GuiComponent but not GuiVComponent. GuiContainer extends GuiComponent.
GuiContainerShell Object [page 84]	A GuiContainerShell is a wrapper for a set of GuiShell objects. GuiContainerShell extends GuiVContainer. The type prefix is shellcont, the name is the last part of the id, shellcont[n].
GuiContextMenu Object [page 87]	A GuiContextMenu may have other GuiContextMenu objects as children. GuiContextMenu extends GuiMenu. The type is mnu, the name is the function code that is sent to the system when the menu item is selected.
GuiCTextField Object [page 88]	If the cursor is set into a text field of type GuiCTextField a combo box button is displayed to the right of the text field. Pressing this button is equivalent to pressing the F4 key. The button is not represented in the scripting object model as a separate object; it is considered to be part of the text field. There are no other differences between GuiTextField and GuiCTextField. GuiCTextField extends GuiTextField. The type prefix is ctxt, the name is the Fieldname taken from the SAP data dictionary.

Class Module	Description
GuiCustomControl Object [page 91]	The GuiCustomControl is a wrapper object that is used to place ActiveX controls onto dynpro screens. While GuiCustomControl is a dynpro element itself, its children are of GuiContainerShell type, which is a container for controls. GuiCustomControl extends GuiVContainer. The type prefix is cntl, the name is the fieldname taken from the SAP data dictionary.
GuiDialogShell Object [page 93]	The GuiDialogShell is an external window that is used as a container for other shells, for example a toolbar. GuiDialogShell extends GuiVContainer. The type prefix is shellcont, the name is the last part of the id, shellcont[n].
GuiEAIViewer2D Object [page 96]	The GuiEAIViewer2D control is used to view 2-dimensional graphic images in the SAP system. The user can carry out redlining over the loaded image. The scripting wrapper for this control records all user actions during the redlining process and reproduces the same actions when the recorded script is replayed. GuiEAIViewer2D extends GuiShell.
GuiEAIViewer3D Object [page 98]	The GuiEAIViewer3D control is used to view 3-dimensional graphic images in the SAP system. GuiEAIViewer3D extends GuiShell.
GuiEnum Object [page 101]	GuiEnum is the base class for some enumerators used in SAP GUI Scripting.
GuiFrameWindow Object [page 101]	A GuiFrameWindow is a high level visual object in the runtime hierarchy. It can be either the main window or a modal popup window. See the GuiMainWindow and GuiModalWindow sections for examples. GuiFrameWindow itself is an abstract interface. GuiFrameWindow extends GuiVContainer. The type prefix is wnd, the name is wnd plus the window number in square brackets.
GuiGOSShell Object [page 108]	The GuiGosShell is only available in New Visual Design mode. GuiGOSShell extends GuiVContainer. The type prefix is shellcont, the name is the last part of the id, shellcont[n].
GuiGraphAdapt Object [page 110]	For the graphic adapter control only basic members from GuiShell are available. Recording and playback is not possible.
GuiGridView Object [page 113]	The grid view is similar to the dynpro table control, but significantly more powerful. GuiGridView extends GuiShell.
GuiHTMLViewer Object [page 128]	The GuiHTMLViewer is used to display an HTML document inside SAP GUI. GuiHTMLViewer extends GuiShell.

Class Module	Description
GuiInputFieldControl Object [page 132]	GuiInputFieldControl offers an input field that can be used inside control containers (unlike the Dynpro element represented by GuiTextField). GuiInputFieldControl extends GuiShell.
GuiLabel Object [page 134]	GuiLabel extends GuiVComponent. The type prefix is lbl, the name is the fieldname taken from the SAP data dictionary.
GuiMainWindow Object [page 141]	This window represents the main window of an SAP GUI session. GuiMainWindow extends GuiFrameWindow.
GuiMap Object [page 147]	For the map control only basic members from GuiShell are available. Recording and playback is not possible.
GuiMenu Object [page 150]	A GuiMenu may have other GuiMenu objects as children. GuiMenu extends GuiVContainer. The type prefix is menu, the name is the text of the menu item. If the item does not have a text, which is the case for separators, then the name is the last part of the id, menu[n].
GuiMenubar Object [page 151]	Only the main window has a menubar. The children of the menubar are menus. GuiMenubar extends GuiVContainer. The type prefix and name are mbar.
GuiMessageWindow [page 153]	GuiMessageWindow is a message box displayed by message showMessageBox of GuiUtils.
GuiModalWindow Object [page 155]	A GuiModalWindow is a dialog pop-up. GuiModalWindow extends GuiFrameWindow.
GuiNetChart Object [page 158]	The GuiNetChart is a powerful tool to display and modify entity relationship diagrams. It is of a very technical nature and should only be used for recording and playback, as most of the parameters cannot be determined in any other way.
GuiOfficeIntegration Object [page 161]	The GuiOfficeIntegration object (Desktop Office Integration) offers a container for hosting different kinds of Office applications (Microsoft Word, Microsoft Excel, Microsoft Powerpoint). GuiOfficeIntegration extends GuiShell.
GuiOkCodeField Object [page 164]	The GuiOkCodeField is placed on the upper toolbar of the main window. It is a combo box into which commands can be entered. Setting the text of GuiOkCodeField will not execute the command until server communication is started, for example by emulating the Enter key (VKey 0). GuiOkCodeField extends GuiVComponent. The type prefix is okcd, the name is empty.

Class Module	Description
GuiPasswordField Object [page 166]	The only difference between GuiTextField and GuiPasswordField is that the Text property cannot be read for a password field. The returned text is always empty. GuiPasswordField extends GuiTextField. The type prefix is pwd, the name is the fieldname taken from the SAP data dictionary.
GuiPicture Object [page 169]	The picture control displays a picture on an SAP GUI screen. GuiPicture extends GuiShell.
GuiRadioButton Object [page 173]	GuiRadioButton extends GuiVComponent. The type prefix is rad, the name is the fieldname taken from the SAP data dictionary.
GuiSapChart Object [page 177]	For the SAP chart control only basic members from GuiShell are available. Recording and playback is not possible.
GuiScrollbar Object [page 179]	The GuiScrollbar class is a utility class used for example in GuiScrollContainer or GuiTableControl.
GuiScrollContainer Object [page 179]	This container represents scrollable subscreens. A subscreen may be scrollable without actually having a scrollbar, because the existence of a scrollbar depends on the amount of data displayed and the size of the GuiUserArea. GuiScrollContainer extends GuiVContainer. The type prefix is ssub, the name is generated from the data dictionary settings.
GuiSession Object [page 181]	The GuiSession provides the context in which a user performs a certain task such as working with a transaction. It is therefore the access point for applications, which record a user's actions regarding a specific task or play back those actions. GuiSession extends GuiContainer. The type prefix is ses, the name is ses plus the session number in square brackets.
GuiSessionInfo Object [page 192]	GuiSessionInfo is a member of all GuiSession objects. It makes available technical information about the session. Some of its properties are displayed in the right corner of the SAP GUI status line.
GuiShell Object [page 195]	GuiShell is an abstract object whose interface is supported by all the controls. GuiShell extends GuiVContainer. The type prefix is shell, the name is the last part of the id, shell[n].
GuiSimpleContainer Object [page 198]	This container represents non-scrollable subscreens. It does not have any functionality apart from to the inherited interfaces. GuiSimpleContainer extends GuiVContainer. The type prefix is sub, the name is generated from the data dictionary settings.
GuiSplit Object [page 201]	GuiSplit extends GuiShell.
GuiSplitterContainer Object [page 204]	The GuiSplitterContainer represents the dynpro splitter element, which was introduced in the Web Application Server ABAP in NetWeaver 7.1. The dynpro splitter element is similar to the activeX based splitter control, but it is a plain dynpro element.

Class Module	Description
GuiStage Object [page 206]	For the stage control only basic members from GuiShell are available. Recording and playback is not possible.
GuiStatusbar Object [page 209]	GuiStatusbar represents the message displaying part of the status bar on the bottom of the SAP GUI window. It does not include the system and login information displayed in the rightmost area of the status bar as these are available from the GuiSessionInfo object. GuiStatusbar extends GuiVComponent. The type prefix is sbar.
GuiStatusPane Object [page 213]	The parent of the GuiStatusPane objects is the status bar (see also GuiStatusbar Object). The GuiStatusPane objects reflect the individual areas of the status bar, for example "pane[0]" refers to the section of the status bar where the messages are displayed.
GuiTab Object [page 214]	The GuiTab objects are the children of a GuiTabStrip object. GuiTab extends GuiVContainer. The type prefix is tabp, the name is the id of the tab's button taken from SAP data dictionary.
GuiTableColumn Collection [page 216]	GuiTableColumn represents a column in a table control. GuiTableColumn extends GuiComponentCollection.
GuiTableControl Object [page 218]	The table control is a standard dynpro element, in contrast to the GuiCtrlGridView, which looks similar. GuiTableControl extends GuiVContainer. The type prefix is tbl, the name is the fieldname taken from the SAP data dictionary.
GuiTableRow Collection [page 222]	GuiTableRow represents a row in a table control. GuiTableRow extends GuiComponentCollection.
GuiTabStrip Object [page 224]	A tab strip is a container whose children are of type GuiTab. GuiTabStrip extends GuiVContainer. The type prefix is tabs, the name is the fieldname taken from the SAP data dictionary.
GuiTextedit Object [page 227]	GuiTextField extends GuiVComponent. The type prefix is txt, the name is the fieldname taken from the SAP data dictionary.
GuiTextField [page 232]	GuiTextField extends GuiVComponent. The type prefix is txt, the name is the fieldname taken from the SAP data dictionary.
GuiTitlebar Object [page 236]	The titlebar is only displayed and exposed as a separate object in New Visual Design mode. GuiTitlebar extends GuiVContainer. The type prefix and name of GuiTitlebar are titl.

Class Module	Description
GuiToolbar Object [page 238]	Every GuiFrameWindow has a GuiToolbar. The GuiMainWindow has two toolbars unless the second has been turned off by the ABAP application. The upper toolbar is the system toolbar, while the second toolbar is the application toolbar. The children of a GuiToolbar are buttons. The indexes for toolbar buttons are determined by the virtual key values defined for the button. GuiToolbar extends GuiVContainer. The type prefix and name are tbar.
GuiToolbarControl [page 240]	GuiToolbarControl represents a button bar control that can host different types of buttons. GuiToolbarControl extends GuiShell.
GuiTree Object [page 244]	This object represents a tree control that can be displayed in multiple different ways (simple tree, column tree,...). GuiTree extends GuiShell and
GuiUserArea Object [page 257]	The GuiUserArea comprises the area between the toolbar and statusbar for windows of GuiMainWindow type and the area between the titlebar and toolbar for modal windows, and may also be limited by docker controls. The standard dynpro elements can be found only in this area, with the exception of buttons, which are also found in the toolbars. GuiUserArea extends GuiVContainer. The type prefix and name are usr.
GuiUtils Object [page 260]	GuiUtils is a utility class that offers some methods for accessing files or showing message boxes.
GuiVComponent Object [page 263]	The GuiVComponent interface is exposed by all visual objects, such as windows, buttons or text fields. Like GuiComponent, it is an abstract interface. Any object supporting the GuiVComponent interface also exposes the GuiComponent interface. GuiVComponent extends GuiComponent.
GuiVContainer Object [page 268]	An object exposes the GuiVContainer interface if it is both visible and can have children. It will then also expose GuiComponent and GuiVComponent. Examples of this interface are windows and subscreens, toolbars or controls having children, such as the splitter control. GuiVContainer extends GuiContainer and GuiVComponent.
GuiVHViewSwitch Object [page 270]	GuiVHViewSwitch represents the “View Switch” object that was introduced with the Belize theme in SAP GUI. The View Switch is placed in the header area of the SAP GUI main window and can be used to select different views within an application.

1.2.1 GuiAbapEditor Object

The GuiAbapEditor object represents the new ABAP editor control available as of SAP_BASIS release 6.20 (see also SAP Note 930742). GuiAbapEditor extends GuiShell.

Methods

Method Syntax	Description
All methods of the GuiVComponent Object [page 263]: • DumpState • SetFocus • Visualize	
All methods of the GuiContainer Object [page 83]: • FindById	
All methods of the GuiVContainer Object [page 268]: • FindAllByName • FindAllByNameEx • FindByName • FindByNameEx	
All methods of the GuiShell Object [page 195]: • SelectContextMenuItem • SelectContextMenuItemByPosition • SelectContextMenuItemByText	
AutoBraceEnabled `Public Function AutoBraceEnabled() As Byte`	Returns True if the auto brace facility is currently switched on.
AutoComplete `Public Sub AutoComplete()`	Invokes the auto complete list box.
AutoCorrectEnabled `Public Function AutoCorrectEnabled() As Byte`	Returns True if the auto correct facility is currently switched on.

Method	
Syntax	**Description**
AutoExpand `Public Sub AutoExpand()`	Invokes the auto expand code template mechanism.
AutoIndentEnabled `Public Function AutoIndentEnabled() As Byte`	Returns True if the auto indent facility is currently switched on.
Capitalize `Public Sub Capitalize()`	Makes the first alphabetic character of each word in the selected text uppercase. All other characters are made lower case.
ClipboardCopy `Public Sub ClipboardCopy()`	Performs a clipboard copy operation on the currently selected text.
ClipboardCut `Public Sub ClipboardCut()`	Performs a clipboard cut operation on the currently selected text.
ClipboardPaste `Public Sub ClipboardPaste()`	Pastes the current contents of the clipboard beginning from the current cursor position.

ClipboardRingPaste
`Public Sub ClipboardRingPaste( _`
`ByVal Index As Long _`
`)`
	Pastes an entry from the editor's clipboard ring to the editor. Index : One-based index of the clipboard entry as it appears in the ABAP editor context menu.

CodeHintsEnabled `Public Function CodeHintsEnabled() As Byte`	Returns True if code hints are currently enabled.
CommentSelectedLines `Public Sub CommentSelectedLines()`	Encloses the selected lines in comments.
CorrectCapsEnabled `Public Function CorrectCapsEnabled() As Byte`	Returns True if the correct caps function is currently switched on.
Delete `Public Sub Delete()`	Deletes the character, which proceeds the current cursor position. Equivalent to pressing the <DEL> key.

Method	
Syntax	**Description**
DeleteBack `Public Sub DeleteBack()`	Moves the cursor to the previous column, deleting the character currently residing there. Equivalent to pressing the backspace key.

DeleteRange
`Public Sub DeleteRange( _`
`  ByVal LineStart As Long, _`
`  ByVal ColumnStart As Long, _`
`  ByVal LineEnd As Long, _`
`  ByVal ColumnEnd As Long _`
`)`
	Defines a region of text for deletion.
	• LineStart specifies the line number from where deletion is to begin.
	• ColumnStart (p2) specifies the number of the column from where deletion is to begin.
	• LineEnd (p3) specifies the number of the line where deletion will end.
	• ColumnEnd (p4) specifies the number of the column where deletion will end.

DeleteSelection `Public Sub DeleteSelection()`	Deletes the currently selected text.
DeleteWord `Public Sub DeleteWord()`	Deletes the word, which proceeds the current character position.
DeleteWordBack `Public Sub DeleteWordBack()`	Deletes the word, which precedes the current cursor position.
DuplicateLine `Public Sub DuplicateLine()`	Takes the contents of the line upon which the cursor currently resides and inserts a copy of the line contents on the line below the cursor.
FormatSelectedLines `Public Sub FormatSelectedLines()`	Formats the selected lines according to "Pretty Printer" and "Formatting" settings e.g. Auto Indent, Smart Tab.
GetAutoCompleteEntryCount `Public Function GetAutoCompleteEntryCount() As Long`	Returns the number of available entries displayed in the auto completion list box.

GetAutoCompleteEntryText
`Public Function GetAutoCompleteEntryText( _`
`  ByVal Index As Long _`
`) As String`
	Returns a string representing the auto completion list box entry corresponding to the index supplied as a parameter.

Method	
Syntax	**Description**
GetAutoCompleteIconType `Public Function GetAutoCompleteIconType( _ ByVal Index As Long _ ) As String`	Returns the index of the image associated with the auto complete entry specified in Index. Returns -1 if no image is associated.
GetAutoCompleteSubIconType `Public Function GetAutoCompleteSubIconType( _ ByVal Index As Long _ ) As String`	Returns the index of the sub image associated with the auto complete entry specified in Index. Returns -1 if no sub image is associated.
GetAutoCompleteToolbarButtonToolTip `Public Function GetAutoCompleteToolbarButtonToolTip( _ ByVal Index As Long _ ) As String`	Returns the tooltip text, which is displayed by the autocomplete toolbar button specified in Index.
GetAutoCompleteToolTipDelay `Public Function GetAutoCompleteToolTipDelay() As Long`	Returns the number of milliseconds, which elapse between highlighting an entry in the autocomplete listbox and the appearance of the corresponding tooltip window.
GetCurrentToolTipText `Public Function GetCurrentToolTipText() As String`	Retrieves the text in the currently displayed code hint or autocomplete listbox tooltip window. Multiple lines are separated with \n characters.
GetCursorColumnPosition `Public Function GetCursorColumnPosition() As Long`	Returns the column number in which the cursor currently resides.
GetCursorLinePosition `Public Function GetCursorLinePosition() As Long`	Returns the number of the line upon which the cursor currently resides.
GetFirstVisibleLine `Public Function GetFirstVisibleLine() As Long`	Returns the line number of the top-most visible line in the current editor session.
GetHTMLClipboardContents `Public Function GetHTMLClipboardContents() As String`	Returns a string containing the current contents of the clipboard in HTML format. Returns an empty string if the clipboard does not contain anything in HTML format.

Method	
Syntax	**Description**
GetLastVisibleLine `Public Function GetLastVisibleLine() As Long`	Returns the line number of the bottom-most visible line in the current editor session.
GetLineCount `Public Function GetLineCount() As Long`	Returns the total number of lines contained in the document in the current session.

GetLineText
`Public Function GetLineText( _`
`  ByVal Line As Long _`
`) As String`
	Returns a string containing the contents of the line number specified as a parameter.

GetNumberedBookmarks
`Public Function GetNumberedBookmarks( _`
`  ByVal Line As Long _`
`) As Object`
	Returns a collection of bookmark numbers assigned to the line number passed as a parameter. The number of the bookmark can range from 0 to 9. If no numbered bookmark is assigned then the collection is empty.

GetRTFClipboardContents `Public Function GetRTFClipboardContents() As String`	Returns a string containing the current contents of the clipboard in Rich Text format. Returns an empty string if the clipboard does not contain anything in Rich Text format.
GetSelectedAutoComplete `Public Function GetSelectedAutoComplete() As Long`	Returns the zero based index of the currently selected entry in the auto completion list box. The method will return -1 if no entry is selected.
GetSelectedText `Public Function GetSelectedText() As String`	Returns a string containing the text currently highlighted or selected in the editor session. If the selected text spans more than one line then any line terminator characters will be included in the string returned by this method.

GetStructureBlockEndLine
`Public Function GetStructureBlockEndLine( _`
`  ByVal Line As Long _`
`) As Long`
	Returns the end line of the structure block relevant to the line number passed to the method. If the line does not reside within a structure block at all then the method returns -1.

GetStructureBlockStartLine
`Public Function GetStructureBlockStartLine( _`
`  ByVal Line As Long _`
`) As Long`
	Returns the start line of the structure block relevant to the line number passed to the method. If the line resides within a nested block then the start line of the innermost block will be returned. If the line does not reside within a structure block at all then the method returns -1.

Method	
Syntax	**Description**
GetUndoPosition `Public Function GetUndoPosition() As Long`	Returns the current position of the document in the undo/redo buffer.
GetWordWrapMode `Public Function GetWordWrapMode() As Long`	Returns an integer corresponding to the currently set Word wrap mode: • 0 - Word wrap disabled. • 1 - Wrap at edge of window. • 2 - Wrap by page width. • 3 - Wrap by page width inserting hard break.
GetWordWrapPosition `Public Function GetWordWrapPosition() As Long`	Returns the current page width assigned to word wrap. The number returned is the number of columns after which word wrap will be applied.
GoNextBookMark `Public Sub GoNextBookMark()`	Navigates to the line where the next none numbered bookmark is set.

GoNumberedBookmark
`Public Sub GoNumberedBookmark( _`
`  ByVal Mark As Long _`
`)`
	Navigates to the line where the bookmark numbered Mark resides.

GoPreviousBookMark `Public Sub GoPreviousBookMark()`	Navigates to the line where the previous none numbered bookmark is set.
InsertTab `Public Sub InsertTab()`	Inserts a TAB at the current cursor position. Equivalent to pressing the TAB key.

InsertText
`Public Sub InsertText( _`
`  ByVal Text As String, _`
`  ByVal Line As Long, _`
`  ByVal Column As Long _`
`)`
	Places the text specified in Text at the position specified in Line and Column as if the text had been typed into the editor from the keyboard.

IsAutoCompleteEntryBold
`Public Function IsAutoCompleteEntryBold( _`
`  ByVal Index As Long _`
`) As Byte`
	Returns True if the auto complete entry specified in Index is bold.

Method	
Syntax	**Description**
IsAutoCompleteOpen `Public Function IsAutoCompleteOpen() As Byte`	Returns True if the auto completion list box is currently open.
IsAutoCompleteToolbarButtonPressed `Public Function IsAutoCompleteToolbarButtonPressed( _ ByVal Index As Long _ ) As Byte`	Returns True if the autocomplete toolbar button specified in Index is currently pressed. Otherwise False is returned.
IsAutoCompleteToolTipVisible `Public Function IsAutoCompleteToolTipVisible() As Byte`	Returns True if the tooltip corresponding to an entry in the auto complete listbox is currently visible.
IsBookmark `Public Function IsBookmark( _ ByVal Line As Long _ ) As Byte`	Returns True if the line is bookmarked with a standard bookmark which is not numbered. The method does not provide information about whether the line is marked using a numbered bookmark.
IsBreakpointSet `Public Function IsBreakpointSet( _ ByVal Line As Long _ ) As Byte`	Returns True if a breakpoint is set on the line number passed in as a parameter.
IsLineCollapsed `Public Function IsLineCollapsed( _ ByVal Line As Long _ ) As Byte`	Returns True if the line number passed to it corresponds to a line, which signifies the beginning of a collapsible block, which is currently in the collapsed state.
IsLineComment `Public Function IsLineComment( _ ByVal Line As Long _ ) As Byte`	Returns True if the line number specified in Line contains comments. Otherwise False is returned.
IsLineModified `Public Function IsLineModified( _ ByVal Line As Long _ ) As Byte`	Returns True if the line has been modified during the course of the current editor session.
IsModified `Public Function IsModified() As Byte`	Returns True if any part of the current document has been modified during \ the course of the current editor session.

Method

Syntax	Description
JoinSelectedLines `Public Sub JoinSelectedLines()`	Merges currently selected lines of text into a single line of text.
LowerCase `Public Sub LowerCase()`	Forces selected text into lower case.
MoveCursorDocumentEnd `Public Sub MoveCursorDocumentEnd()`	Positions the cursor in the last column of the very last line of the document.
MoveCursorLineDown `Public Sub MoveCursorLineDown()`	Moves the cursor down one line from its current position.
MoveCursorLineEnd `Public Sub MoveCursorLineEnd()`	Positions the cursor in the last column of the current line.
MoveCursorLineHome `Public Sub MoveCursorLineHome()`	Positions the cursor in the first column of the current line.
MoveCursorLineUp `Public Sub MoveCursorLineUp()`	Moves the cursor up one line from its current position.
MoveLineDown `Public Sub MoveLineDown()`	Moves the contents of the line where the cursor resides to the line below and moves the contents of the line below the cursor up one line.
MoveLineUp `Public Sub MoveLineUp()`	Moves the contents of the line where the cursor resides to the line above and moves the contents of the line above the cursor down one line.
MoveWordLeft `Public Sub MoveWordLeft()`	Moves the cursor to the column preceding the next word found to the left of the cursor's current position.
MoveWordRight `Public Sub MoveWordRight()`	Moves the cursor to the column preceding the next word found to the right of the cursor's current position.
OverwriteModeEnabled `Public Function OverwriteModeEnabled() As Byte`	Returns True if Overwrite mode is enabled, False if in Insert mode.

Method	
Syntax	**Description**
RemoveAllBookmarks `Public Sub RemoveAllBookmarks()`	Removes all types of bookmarks from the document. Both numbered and none numbered bookmarks are removed.
RemoveAllBreakpoints `Public Sub RemoveAllBreakpoints()`	Removes all breakpoints from the current document.

RemoveBookmarks
`Public Sub RemoveBookmarks( _`
`  ByVal Bookmarks As String _`
`)`
	Removes all bookmarks specified in the supplied string.

RemoveBreakpoint
`Public Sub RemoveBreakpoint( _`
`  ByVal Line As Long _`
`)`
	Removes the breakpoint on line number Line.

ReplaceSelection
`Public Sub ReplaceSelection( _`
`  ByVal Text As String _`
`)`
	Replaces the currently selected text with the text contained in the Text parameter.

SaveToFile
`Public Sub SaveToFile( _`
`  ByVal p1 As String _`
`)`	

ScrollToLine
`Public Sub ScrollToLine( _`
`  ByVal Line As Long _`
`)`
	Scrolls to the line number specified in Line if not already visible on the screen.

SelectAll `Public Sub SelectAll()`	Highlights all text in the current document for selection.

Method	
Syntax	**Description**

SelectBlockRange
`Public Sub SelectBlockRange( _`
`  ByVal LineStart As Long, _`
`  ByVal ColumnStart As Long, _`
`  ByVal LineEnd As Long, _`
`  ByVal ColumnEnd As`
`  Long _`
`)`
	Highlights a region of text in block mode for selection. Equivalent to holding down the ALT key while dragging the mouse over the text.
	• LineStart specifies the line number from where selection is to begin.
	• ColumnStart specifies the number of the column from where selection is to begin.
	• LineEnd specifies the number of the line where selection will end.
	• ColumnEnd specifies the number of the column where selection will end.

SelectRange
`Public Sub SelectRange( _`
`  ByVal LineStart As Long, _`
`  ByVal ColumnStart As Long, _`
`  ByVal LineEnd As Long, _`
`  ByVal ColumnEnd As`
`  Long _`
`)`
	Highlights a region of text for selection.
	• LineStart specifies the line number from where selection is to begin.
	• ColumnStart specifies the number of the column from where selection is to begin.
	• LineEnd specifies the number of the line where selection will end.
	• ColumnEnd specifies the number of the column where selection will end.

SelectWordLeft `Public Sub SelectWordLeft()`	Selects the word to the left of the current cursor position.
SelectWordRight `Public Sub SelectWordRight()`	Selects the word to the right of the current cursor position.
Sentencize `Public Sub Sentencize()`	Makes the first character of each sentence upper case. Sentences are delimited by "." characters. All other characters are made lower case.

SetAutoBrace
`Public Sub SetAutoBrace( _`
`  ByVal Status As Byte _`
`)`
	Switches the auto brace facility on or off.

SetAutoCorrect
`Public Sub SetAutoCorrect( _`
`  ByVal Status As Byte _`
`)`
	Switches the auto complete facility on or off.

Method	
Syntax	**Description**

SetAutoIndent
`Public Sub SetAutoIndent( _`
`  ByVal Status As Byte _`
`)`
	Switches the auto indent facility on or off.

SetBookmarks
`Public Sub SetBookmarks( _`
`  ByVal Bookmarks As String _`
`)`
	Set bookmarks.
	Takes a string of the following format:
	<line>[(<no>)][,<line>] e.g. "10(1),22(2),33,42",
	<line>={1,...,n}, <no>={1,...

SetBreakpoint
`Public Sub SetBreakpoint( _`
`  ByVal Line As Long _`
`)`
	Sets a breakpoint on line number Line.

SetCodeHints
`Public Sub SetCodeHints( _`
`  ByVal Status As Byte _ )`
	Switches code hints on or off.

SetCorrectCaps
`Public Sub SetCorrectCaps( _`
`  ByVal Status As Byte _`
`)`
	Switches the caps correction feature on or off.

SetLineFeedStyle
`Public Sub SetLineFeedStyle( _`
`  ByVal p1 As Long _`
`)`	

SetOverwriteMode
`Public Sub SetOverwriteMode( _`
`  ByVal Status As Byte _`
`)`
	Switches between Insert and Overwrite modes. If called with True then Overwrite mode is enabled. Otherwise the editor is in Insert mode.

SetSelectionPosInLine
`Public Sub SetSelectionPosInLine( _`
`  ByVal Line As Long, _`
`  ByVal Column As`
`  Long _`
`)`
	Places the cursor on line <Line> and column <Column>.

Method	
Syntax	**Description**

SetSmartTab
`Public Sub SetSmartTab( _`
`  ByVal Status As Byte _`
`)`
	Switches the smart tab facility on or off.

SetWordWrapMode
`Public Sub SetWordWrapMode( _`
`  ByVal Mode As Long _`
`)`
	Sets the word wrap mode according to the number supplied in Mode:
	• 0 - Word wrap disabled.
	• 1 - Wrap at edge of window.
	• 2 - Wrap by page width.
	• 3 - Wrap by page width inserting hard break.

SetWordWrapPosition
`Public Sub SetWordWrapPosition( _`
`  ByVal Pos As Long _`
`)`
	Pos specifies the number of columns to be displayed before word wrap is applied.

SmartTabEnabled `Public Function SmartTabEnabled() As Byte`	Switches the smart tab facility on or off.
SortSelectedLines `Public Sub SortSelectedLines()`	Rearranges the selected lines in alphanumeric order.
SwapCase `Public Sub SwapCase()`	Inverts the case setting for the selected text. Upper case characters are switched to lower case and vice versa.
ToggleCapsLock `Public Sub ToggleCapsLock()`	Switches caps lock on or off.

ToggleNumberedBookmark
`Public Sub ToggleNumberedBookmark( _`
`  ByVal Mark As Long, _`
`  ByVal Line As Long _`
`)`
	Toggles the state of the numbered bookmark Mark on line Line. If a book mark with the number Mark already exists on Line then it will be removed. Otherwise it will be added.

ToggleStructureBlock
`Public Sub ToggleStructureBlock( _`
`  ByVal Line As Long _`
`)`
	If the line number specified in Line is the first line of a collapsible block of code then this method will toggle the expanded/collapsed status of the block.

Method

Syntax	Description
TransposeLine `Public Sub TransposeLine()`	Swaps the contents of the line where the cursor currently resides with the contents of the line above the current cursor position.
UncommentSelectedLines `Public Sub UncommentSelectedLines()`	Removes comments from the selected lines.

Undo
`Public Sub Undo( _`
`  ByVal UndoPosition As Long _`
`)`
	Performs either an undo or a redo depending on UndoPosition. UndoPosition specifies a zero based position in the undo/redo buffer. If -1 is passed then a single step undo is preformed.

UnTab `Public Sub UnTab()`	Removes a TAB at the current cursor position. Equivalent to pressing <SHIFT> + <TAB>.
UpperCase `Public Sub UpperCase()`	Forces selected text into upper case.

Properties

Property

Syntax	Description
All properties of the GuiComponent Object [page 78]: • ContainerType • Id • Name • Parent • Type • TypeAsNumber	

Property	
Syntax	**Description**
All properties of the GuiVComponent Object [page 263]: • AccLabelCollection • AccText • AccTextOnRequest • AccTooltip • Changeable • DefaultTooltip • Height • IconName • IsSymbolFont • Left • Modified • ParentFrame • ScreenLeft • ScreenTop • Text • Tooltip • Top • Width	
All properties of the GuiContainer Object [page 83]: • Children	
All properties of the GuiShell Object [page 195]: • AccDescription • CurrentContextMenu • DragDropSupported • Handle • OcxEvents • SubType	

1.2.2 GuiApoGrid Object

The GuiApoGrid is an object specifically created for SAP SCM applications. It implements a planning board, which is similar to a GuiGridView control. GuiApoGrid extends GuiShell.

Remarks

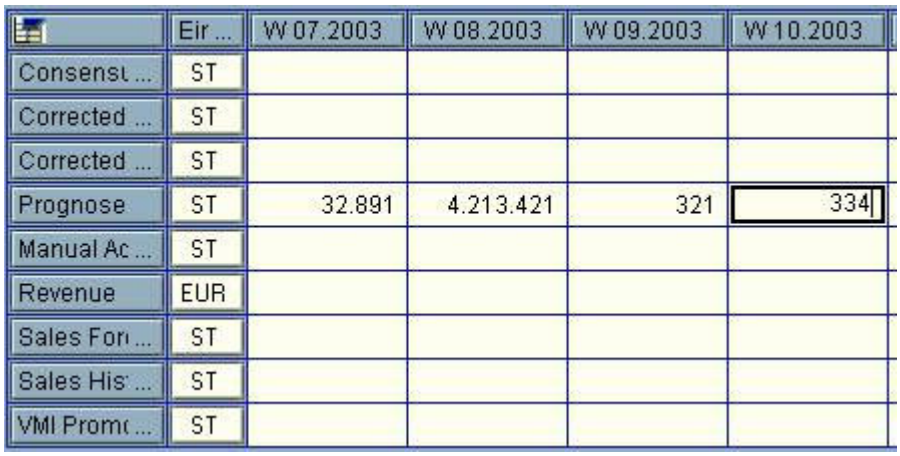
The columns and rows are identified by their position starting with zero:

0 <= row < RowCount

0 <= column < ColumnCount

After a drill-down the rows are re-numbered so that the row number of any given row may change. Scrolling horizontally does not affect the number of a column.

Example



	Eir ...	W 07.2003	W 08.2003	W 09.2003	W 10.2003
Consensı ...	ST				
Corrected ...	ST				
Corrected ...	ST				
Prognose	ST	32.891	4.213.421	321	334
Manual Ac ...	ST				
Revenue	EUR				
Sales For ...	ST				
Sales His ...	ST				
VMI Promı ...	ST				

Methods

Method	
Syntax	**Description**
All methods of the GuiVComponent Object [page 263]: • DumpState • SetFocus • Visualize	
All methods of the GuiContainer Object [page 83]: • FindById	

Method Syntax	Description
All methods of the GuiVContainer Object [page 268]: • FindAllByName • FindAllByNameEx • FindByName • FindByNameEx	
All methods of the GuiShell Object [page 195]: • SelectContextMenuItem • SelectContextMenuItemByPosition • SelectContextMenuItemByText	
CancelCut `Public Sub CancelCut()`	Abort the cut operation.
ClearSelection `Public Sub ClearSelection()`	Calling clearSelection removes all row, column and cell selections.

ContextMenu
`Public Sub ContextMenu( _`
`  ByVal Column As Long, _`
`  ByVal Row As`
`  Long _`
`)`
	Calling contextMenu emulates the context menu request.

Cut `Public Sub Cut()`	Cut the selected cells.

DeselectCell
`Public Sub DeselectCell( _`
`  ByVal Column As Long, _`
`  ByVal Row As Long _`
`)`
	Deselect the specified cells. This function removes the specified cells from the collection of selected cells.

DeselectColumn
`Public Sub DeselectColumn( _`
`  ByVal Column As Long _`
`)`
	This function removes the specified column from the collection of the selected columns.

Method	
Syntax	**Description**

DeselectRow
`Public Sub DeselectRow( _`
`  ByVal Row As Long _`
`)`
	This function removes the specified row from the collection of the selected rows.

DoubleClickCell
`Public Sub DoubleClickCell( _`
`  ByVal Column As Long, _`
`  ByVal Row As`
`  Long _`
`)`
	This function emulates a mouse double-click on a given cell if the parameters are valid and raises an exception otherwise.

GetBgdColorInfo
`Public Function GetBgdColorInfo( _`
`  ByVal Row As Long, _`
`  ByVal Column As Long _`
`) As String`
	This function returns the background color of the specified cell.

GetCellChangeable
`Public Function GetCellChangeable( _`
`  ByVal Column As Long, _`
`  ByVal Row As Long _`
`) As Byte`
	This function returns True if the specified cell is editable.

GetCellFormat
`Public Function GetCellFormat( _`
`  ByVal Column As Long, _`
`  ByVal Row As Long _`
`) As String`	

GetCellTooltip
`Public Function GetCellTooltip( _`
`  ByVal Column As Long, _`
`  ByVal Row As Long _`
`) As String`
	This function returns the tooltip of the specified cell.

GetCellValue
`Public Function GetCellValue( _`
`  ByVal Column As Long, _`
`  ByVal Row As Long _`
`) As String`
	This function returns the value of the specified cell as a string.

Method	
Syntax	**Description**

GetFgdColorInfo
`Public Function GetFgdColorInfo( _`
`  ByVal Row As Long, _`
`  ByVal Column As Long _`
`) As String`
	This function returns the font color of the specified cell.

GetIconInfo
`Public Function GetIconInfo( _`
`  ByVal Row As Long, _`
`  ByVal Column As Long _`
`) As String`	

IsCellSelected
`Public Function IsCellSelected( _`
`  ByVal Column As Long, _`
`  ByVal Row As Long _`
`) As Byte`
	Returns True if the specified cell is selected.

IsColSelected
`Public Function IsColSelected( _`
`  ByVal col As Long _`
`) As Byte`
	Returns True if the specified column is selected.

IsRowSelected
`Public Function IsRowSelected( _`
`  ByVal Row As Long _`
`) As Byte`
	Returns True if the specified row is selected.

Paste
`Public Function Paste( _`
`  ByVal CellValues As Object, _`
`  ByVal ColumnCount As Long _`
`) As Long`
	Triggers a paste operation.

PressEnter `Public Sub PressEnter()`	This emulates pressing the Enter key.
SelectAll `Public Sub SelectAll()`	This function selects the whole grid content (i.e. all rows and all columns).

Method

Syntax	Description

SelectCell
`Public Sub SelectCell( _`
`ByVal Column As Long, _`
`ByVal Row As`
`Long _ )`
	Select the specified cell.

SelectColumn
`Public Sub SelectColumn( _`
`ByVal Column As Long _`
`)`
	Select the specified column.

SelectRow
`Public Sub SelectRow( _`
`ByVal Row As Long _`
`)`
	Select the specified row.

SetCellValue
`Public Function SetCellValue( _`
`ByVal Column As Long, _`
`ByVal Row As Long, _`
`ByVal Value As String _`
`) As String`
	This function enters the specified value in the specified cell.

Properties

Property

Syntax	Description
All properties of the GuiComponent Object [page 78]: • ContainerType • Id • Name • Parent • Type • TypeAsNumber	

Property

Syntax	Description
All properties of the GuiVComponent Object [page 263]: • AccLabelCollection • AccText • AccTextOnRequest • AccTooltip • Changeable • DefaultTooltip • Height • IconName • IsSymbolFont • Left • Modified • ParentFrame • ScreenLeft • ScreenTop • Text • Tooltip • Top • Width	
All properties of the GuiShell Object [page 195]: • AccDescription • CurrentContextMenu • DragDropSupported • Handle • OcxEvents • SubType	
All properties of the GuiContainer Object [page 83]: • Children	
ColumnCount (Read-only) `Public Property ColumnCount As Long`	This property represents the number of columns in the control.
CurrentCellColumn (Read-only) `Public Property CurrentCellColumn As Long`	The index of the column that contains the current cell.

Property

Syntax	Description
CurrentCellRow (Read-only) `Public Property CurrentCellRow As Long`	The row index of the current cell ranges from 0 to the number of rows less 1, with -1 being the index of the title row.
FirstVisibleColumn (Read-only) `Public Property FirstVisibleColumn As Long`	This property represents the first visible column of the scrollable area of the APOGrid control.
FirstVisibleRow (Read-only) `Public Property FirstVisibleRow As Long`	This is the index of the first visible row in the grid. Setting this property to an invalid row index will raise an exception.
FixedColumnsLeft (Read-only) `Public Property FixedColumnsLeft As Long`	The number of fixed columns at the left side of the grid.
FixedColumnsRight (Read-only) `Public Property FixedColumnsRight As Long`	The number of fixed columns at the right side of the grid.
FixedRowsBottom (Read-only) `Public Property FixedRowsBottom As Long`	The number of fixed rows at the bottom of the grid.
FixedRowsTop (Read-only) `Public Property FixedRowsTop As Long`	The number of fixed rows at the top of the grid.
RowCount (Read-only) `Public Property RowCount As Long`	This property represents the number of rows in the control.
SelectedCells (Read-only) `Public Property SelectedCells As Object`	The collection of selected cells. Trying to set this property to an invalid value will raise an exception.
SelectedColumns (Read-only) `Public Property SelectedColumns As String`	The selected columns are available as a collection. Setting this property can raise an exception, if the new collection contains an invalid column.

Property	
Syntax	**Description**
SelectedColumnsObject (Read-only) `Public Property SelectedColumnsObject As Object`	
SelectedRows (Read-only) `Public Property SelectedRows As String`	The selected rows are available as a collection. Setting this property can raise an exception, if the new collection contains an invalid row.
SelectedRowsObject (Read-only) `Public Property SelectedRowsObject As Object`	
VisibleColumnCount (Read-only) `Public Property VisibleColumnCount As Long`	Retrieves the number of visible columns of the grid.
VisibleRowCount (Read-only) `Public Property VisibleRowCount As Long`	Retrieves the number of visible rows of the grid.

1.2.3 GuiApplication Object

Description

The GuiApplication represents the process in which all SAP GUI activity takes place. If the scripting component is accessed by attaching to an SAP Logon process, then GuiApplication will represent SAP Logon. GuiApplication is a creatable class. However, there must be only one component of this type in any process. GuiApplication extends the GuiContainer Object [page 83].

Remarks

When running a recorded script in Excel you may receive an error message 'Invalid use of property'. This may be caused by a name collision for the 'application' object. In Excel this name is predefined, and it will collide with the code generated by the SAP GUI Scripting recorder. To fix the problem, rename the SAP GUI application object.

Example

Example

You can instantiate an SAP GUI application object using CreateObject.

```
Rem Create the GuiApplication object
Set Application =  CreateObject( "Sapgui.ScriptingCtrl.1")
Rem Open a connection in synchronous mode
Set Connection  = Application.OpenConnection( "U9C [PUBLIC]",  True)
Set Session     = Connection.Children(0)
Rem Do something: Either fill out the login screen
Rem or in case of Single-Sign-On start a transaction.
Session.SendCommand( "/nbibs")
MsgBox "Waiting..."
Rem Shutdown the connection
Set Session     =  Nothing
Connection.CloseSession( "ses[0]")
Set Connection  =  Nothing
Rem Wait a bit for the connection to be closed completely
Wscript.Sleep 1000
Set Application =  Nothing
MsgBox "Done"
```

Example

You can attach to a running instance of SAP GUI through the Running Object Table.

```
Rem Get the application object from the Running Object Table
Set rotEntry  =  GetObject( "SAPGUI")
Set application = rotEntry.GetScriptingEngine
Rem Get the first session of the first connection
Rem You may have to adjust this to access a different session
Set connection = application.Children(0)
Set session    = connection.Children(0)
Rem Start a transaction
session.findById( "wnd[0]/tbar[0]/okcd").text =  "/nbibs"
session.findById( "wnd[0]").sendVKey 0
```

Methods

Method	
Syntax	**Description**
All methods of the GuiContainer Object [page 83]: • FindById	

Method Syntax	Description

AddHistoryEntry
`Public Function AddHistoryEntry( _`
`  ByVal Fieldname As String, _`
`  ByVal Value As String _`
`) As Byte`
	SAP GUI for Windows has an input history functionality, which displays for text fields the entries made in the past as a suggestion. With this function, an entry can be added to the history database so that it will be available the next time the end user accesses the text field with the given field name.

CreateGuiCollection
`Public Function CreateGuiCollection()`
`As Object`
	Some functions accept collections as parameters. This function creates a collection object that is independent of the scripting language used, such as VBScript or JavaScript.

DropHistory `Public Function DropHistory() As Byte`	Calling this function will delete all entries from the input history. The function returns True if the history data have been deleted successfully. Attention: After dropping the history database, it cannot be restored. Therefore this function must be used with caution.

Ignore
`Public Sub Ignore( _`
`  ByVal WindowHandle As Integer _`
`)`
	eCATT uses this function to prevent scripts from accessing the session in which eCATT itself runs. Otherwise the token handling would cause a dead lock.

Method Syntax	Description

OpenConnection
`Public Function OpenConnection( _`
`  ByVal Description As String, _`
`  Optional ByVal Sync As Variant, _`
`  Optional ByVal Raise As Variant _`
`) As GuiConnection`
	The parameter Description should contain one of the descriptions displayed in SAP Logon, for example "XYZ [PUBLIC]". If you want to create a new SAP GUI instance and place it within your application you may add the suffix "/INPLACE".
	This function will raise the exception E_ACCESSDENIED if the scripting support has been disabled by the administrator or the user.
	Remark: When opening connections manually SAP GUI executes the request asynchronously, so that the SAP Logon dialog remains responsive after requesting a new connection. This behaviour is also the default for SAP GUI Scripting. In the Scripting context it means that the call to openConnection may return before the new connection has been opened. A side effect of this is that when opening a connection fails SAP GUI displays an error popup that cannot be handled from the script. This problem can be solved by setting the sync parameter to True. Then the call to openConnection will not return until a connection has been established, or an error has been detected. If sync is set to True and an error occurs an exception is raised, unless the parameter raise is set to False.

OpenConnectionByConnectionString
`Public Function`
`OpenConnectionByConnectionString( _`
`  ByVal ConnectString As String, _`
`  Optional ByVal Sync As Variant, _`
`  Optional ByVal Raise As Variant _`
`) As GuiConnection`
	The parameter ConnectString is the connection string for the SAP server, for example "/R/ALR/G/SPACE". See the description of the **openConnection** method for a discussion of the sync and raise parameters.
	For detailed information refer to Method OpenConnectionByConnectionString [page 44].

RegisterROT `Public Function RegisterROT() As Byte`	Accessing the SAPGUI entry in the Running Object Table from a C++ application may fail unless the interface is registered with a strong reference. This is not required when using Visual Basic or scripting languages. The reference must be released using revokeRot before shutting down the Scripting component. Failing to do so will lead to undefined behaviour. Most applications do not need to use this method, and shouldn't call it.
RevokeROT `Public Sub RevokeROT()`	This method must be called before shutting down the Scripting component if registerROT was used.

Properties

Property Syntax	Description
All properties of the GuiComponent Object [page 78]: • ContainerType • Id • Name • Parent • Type • TypeAsNumber	
All properties of the GuiContainer Object [page 83]: • Children	
ActiveSession (Read-only) `Public Property ActiveSession As Object`	Returns the Session that the user is currently working with, which will be the topmost window.
AllowSystemMessages (Read-write) `Public Property AllowSystemMessages As Byte`	System messages are displayed when an administrator invokes them on the server to send a notification to users currently logged in. This may happen at any time and interfere with the recording or playback of a script. Setting this property to FALSE will prevent system messages from being displayed.
ButtonbarVisible (Read-write) `Public Property ButtonbarVisible As Byte`	Setting this property to FALSE hides the application toolbar of the main window for newly opened connections.
ConnectionErrorText (Read-only) `Public Property ConnectionErrorText As String`	This property contains the text of a connection error message. If **openConnection** fails, you can retrieve information about the cause of the failure from this property.
Connections (Read-only) `Public Property Connections As GuiComponentCollection`	This property is another name for the **Children** property. It has been added for better readability as all the children of GuiApplication are connections.
HistoryEnabled (Read-write) `Public Property HistoryEnabled As Byte`	The local history function can be enabled or disabled using this property. Disabling it will significantly improve the performance of SAP GUI, which may be crucial during load tests, for example.

Property

Syntax	Description
MajorVersion (Read-only) `Public Property MajorVersion As Long`	Version of the SAP GUI release, for example '7.60'.
MinorVersion (Read-only) `Public Property MinorVersion As Long`	Build number of the scripting component.
NewVisualDesign (Read-only) `Public Property NewVisualDesign As Byte`	Returns whether New Visual Design or Classic mode are used for the user interface.
Patchlevel (Read-only) `Public Property Patchlevel As Long`	Patchlevel of SAP GUI.
Revision (Read-only) `Public Property Revision As Long`	Revision of the SAP GUI release. In SAP GUI for Windows this is the compilation number.
StatusbarVisible (Read-write) `Public Property StatusbarVisible As Byte`	Setting this property to FALSE hides the status bar of the main window for newly opened connections.
TitlebarVisible (Read-write) `Public Property TitlebarVisible As Byte`	Setting this property to FALSE hides the titlebar of the main window for newly opened connections.
ToolbarVisible (Read-write) `Public Property ToolbarVisible As Byte`	Setting this property to FALSE hides the system toolbar of the main window for newly opened connections.
Utils (Read-only) `Public Property Utils As GuiUtils`	This property returns a global GuiUtils object.

Events

Event	
Syntax	**Description**

CreateSession
`Public Event CreateSession( _`
`  ByVal Session As GuiSession _`
`)`
	This event is raised whenever a new session is created, irrespective of whether of the session being created manually, from ABAP or by a script. The event is only raised for a session if the scripting support has been enabled for the corresponding backend.
	
	Example
	The following script attaches itself to the SAPlogon process and displays a pop-up whenever a new session is created.
	`Dim objSapGui`
	`Set objSapGui = GetObject("SAPGUI")`
	`Dim objScriptingEngine`
	`Set objScriptingEngine =`
	`objSapGui.GetScriptingEngine`
	`WScript.ConnectObject`
	`objScriptingEngine,  "Engine_"`
	`Dim Waiting`
	`Waiting = 1`
	`Do While (Waiting = 1)`
	`  WScript.Sleep(100)`
	`Loop`
	`Set objScriptingEngine = Nothing`
	`Set objSapGui = Nothing`
	`Sub Engine_CreateSession(ByVal`
	`Session)`
	`  Dim result`
	`  result = MsgBox("Session`
	`created", vbOKCancel)`
	`  If result = vbCancel then`
	`    Waiting = 0`
	`  End If`
	`End Sub`

Event

Syntax	Description

DestroySession
`Public Event DestroySession( _`
`  ByVal Session As GuiSession _`
`)`
	This event is raised before a session is destroyed . This can be done either by closing the main window manually, or by calling the closeSession method of GuiConnection.
	Example
	You can handle this event from VBScript by adding the following procedure to the sample script on previous page:
	`Sub Engine_DestroySession(ByVal Session)`
	`  Dim result`
	`  result = MsgBox("Session destroyed",vbOKCancel)`
	`  If result = vbCancel then`
	`  Waiting = 0`
	`  End If`
	`End Sub`

Error
`Public Event Error( _`
`  ByVal ErrorId As Long, _`
`  ByVal Desc1 As String, _`
`  ByVal Desc2 As String, _`
`  ByVal Desc3 As String, _`
`  ByVal Desc4 As String _`
`)`
	An error event is currently only raised, if the wrapper library required to access a SAP GUI ActiveX control from a script is not available. This event is also available on the GuiSession in which the error occurred.

IgnoreSession
`Public Event IgnoreSession( _`
`  ByVal SessionMainWindowHandle As Integer _`
`)`
	The event is fired when a session is set to 'Ignored' using IgnoreSession function. This event is only fired when using SAP GUI Scripting while running eCATT in parallel.

1.2.3.1 Method OpenConnectionByConnectionString

Connection Strings

Connection String is a technical term used within SAP GUI. A connection string describes a connection address for a destination, e.g. an SAP system's application server, similar to an Internet URL describes a location for a web page.

Simple Connection Strings

In its simplest form, a connection string contains an IP address and a port number. This information is sufficient for SAP GUI to open a direct TCP connection to a destination, e.g. an application server. IP address and port number are marked with the prefixes '/H/' (for host) and '/S/' (for service). Note that the port number for an SAP application server is by convention 3200 plus the two-digit SAP system number.

> ### Example
>
> Example for a simple connection string with an application server's IP address (172.16.64.17) and port number (3200):
>
> `/H/172.16.64.17/S/3200`

If your network environment supports DNS (Domain Name Services), a hostname can be used instead of the IP address in all kinds of connection strings. (This requires a correct DNS configuration on the client, e.g. via the hosts file).

> ### Example
>
> Example with an application server's hostname (iwdf8997.wdf.sap-ag.de) and port number (3200):
>
> `/H/iwdf8997.wdf.sap-ag.de/S/3200`

If your network environment supports symbolic service names for well-known ports, the symbolic service name can be used instead of the port number in all kinds of connection strings. (This requires a correct service configuration on the client, e.g. in the services file). Note that SAP application server ports are by convention named 'sapdp<SID>', where <SID> is the SAP system id

> ### Example
>
> Example with host name (iwdf8997.wdf.sap-ag.de) and symbolic service name (sapdpIWD):
>
> `/H/iwdf8997.wdf.sap-ag.de/S/sapdpIWD`

Simple connection strings need not to be resolved by the SAP GUI application. Resolution of host names and symbolic service names is done by the operating system's network layer.

SAP Routers

In a WAN (Wide Area Network) environment, SAP routers are used to make connections to remote SAP systems that cannot be reached with a direct TCP connection. Passwords may be used for each SAP router to control access. In order to make a connection, the client is responsible for providing the complete route to the destination, possibly including a chain of several SAP routers. Path information is not provided by the routers. (Strictly speaking, an SAP router is actually better described as an application level proxy with password capabilities and strict source routing). The address for each router is specified by a simple connection string (with the router's host name and port number), optionally followed by '/P/' and the router password. The path from the current location to the destination is described by concatenating all router addresses, followed by the address of the destination SAP system. Thus, a connection string with SAP routers generally has the form `<router 1><router 2>...<router n><destination>`.

Example

Example with two routers (gate.acme.com, port 3299, and gate.sap.com, port 3298), the first using a password (secret), for a connection to the application server iwdf8997.sap.com, port 3200):

```
/H/gate.acme.com/S/3299/P/secret/H/gate.sap.com/S/3298/H/iwdf8997.sap.com/S/
3200<-------- 1st router ---------><---- 2nd router ----><------ app_server
------>
```

Connection strings including SAP routers are passed to SAP GUI's communication layer and resolved step by step by the routers on the path. If host names and symbolic service names are used, each router must have access to correct network configuration information to resolve them.

Message Servers and Logon Groups

For load balancing purposes, application servers from one SAP system are usually configured in logon groups, where each group serves a particular kind of user. The application servers in each group are assigned to users by a least-heavily-loaded strategy. This load balancing is done by message servers. Each SAP system has exactly one message server, which can be reached via TCP on a specific message server port.

Note

Care should be taken that the application server's port number is not confused with the message server's port number. Although the message server's host name may in small installations often be identical to the hostname of an application server, the port number is always different. Symbolic service names for message servers by convention have the form 'sapms<SID>', where <SID> is the SAP system id.

Message server and group information can be used to address an SAP system in a connection string. The address of the message server is specified as a combination of message server host name, message server port and group name. This information is marked with the prefixes `'/M/'` (message server host name), `'/S/'` (message server port) and `'/G/'` (logon group).

Example

Example with message server (hostname alrmain, port number 4253) and logon group (SPACE):

```
/M/alrmain.wdf.sap-ag.de/S/4253/G/SPACE
```

Connection strings with message servers are resolved by SAP GUI by contacting the message server and retrieving the (simple) connection string of an application server for the specified group. This requires network access to the message server at the time the address is resolved. SAP router connection strings may be used in combination with message server connection strings simply by specifying the router address before the message server address. The router is then used for contacting the message server as well as for contacting the resolved application server.

Symbolic System Names

The most user-friendly form of connection string addresses an SAP system only by its symbolic name (per convention, the system id) and the logon group name. This information is marked with the prefixes '/R/' (for the symbolic SAP system name) and '/G/' (for the logon group name).

> **Example**
>
> Example with SAP system (ALR) and logon group (SPACE):
>
> **`/R/ALR/G/SPACE`**
>
> Connection strings with symbolic system names are resolved by SAP GUI by looking up the symbolic SAP system name in the Message Server List (a text file containing a mapping between symbolic system names and message server addresses) and replacing the /R/ part of the connection string with the resulting message server address.
>
> The result is a complete message server connection string, which is then further resolved as explained above.

Formal Syntax

For the technically interested reader, the following BNF grammar formally describes the syntax of connection strings:

> **Sample Code**

```
<connection string> := [<router prefix>]<local>
<local> := <simple>|<message server>|<symbolic>
<simple> := "/H/"<host>"/S/"<service>
<host> := <hostname>|<ipaddr>
<hostname> := (any DNS hostname)
<ipaddr> := (any IP address, in dotted decimal form)
<service> := <servicename>|<port number>
<servicename> := (any IP service name)
<port number> := (any decimal number)
<messageserver> := "/M/"<host>"/S/"<service>"/G/"<group>
<group> := (any ASCII string not containing '/')
<symbolic> := "/R/"<system>"/G/"<group>
<system> := (any ASCII string not containing '/')
<router prefix> := <router>*
<router> := "/H/"<host>"/S/"<service>["/P/"<password>]
<password> := (any ASCII string not containing '/')
```

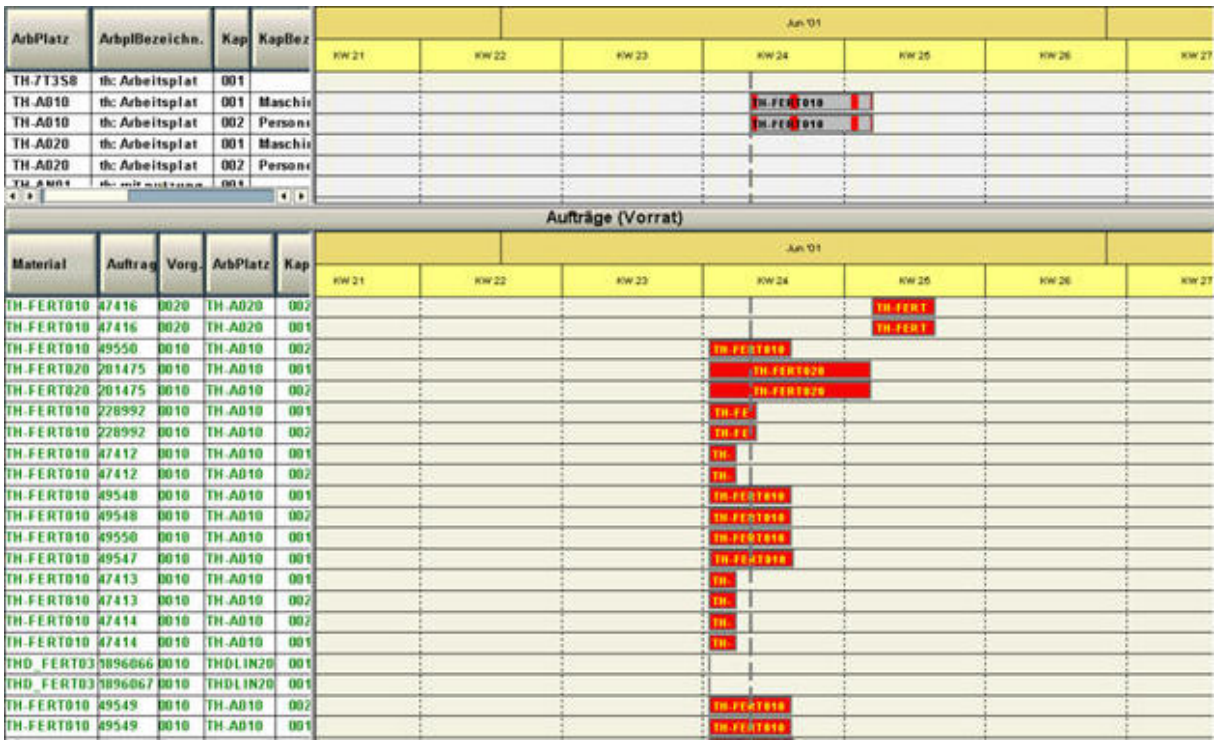
1.2.4 GuiBarChart Object

Description

The GuiBarChart is a powerful tool to display and modify time scale diagrams.

The object is of a very technical nature. It should only be used for recording and playback, as most of the parameters cannot be determined in any other way. GuiBarChart extends the GuiShell Object [page 195].

Example



Methods

Method	
Syntax	**Description**
All methods of the GuiVComponent Object [page 263]: • DumpState • SetFocus • Visualize	

Method Syntax	Description
All methods of the GuiContainer Object [page 83]: • FindById	
All methods of the GuiVContainer Object [page 268]: • FindAllByName • FindAllByNameEx • FindByName • FindByNameEx	
All methods of the GuiShell Object [page 195]: • SelectContextMenuItem • SelectContextMenuItemByPosition • SelectContextMenuItemByText	

BarCount
`Public Function BarCount( _`
`  ByVal chartId As Long _`
`) As Long`
	Returns the number of bars in the given chart.

GetBarContent
`Public Function GetBarContent( _`
`  ByVal chartId As Long, _`
`  ByVal barId As Long, _`
`  ByVal textId As Long _`
`) As String`
	Returns the content of the bar.

GetGridLineContent
`Public Function GetGridLineContent( _`
`  ByVal chartId As Long, _`
`  ByVal gridlineId As Long, _`
`  ByVal textId As Long _`
`) As String`
	Returns the content of the grid line.

GridCount
`Public Function GridCount( _`
`  ByVal chartId As Long _`
`) As Long`
	Returns the number of grids within the chart.

LinkCount
`Public Function LinkCount( _`
`  ByVal chartId As Long _`
`) As Long`
	Returns the number of links within the given chart.

Method

Syntax	Description

SendData
`Public Sub SendData( _`
`  ByVal Data As String _`
`)`
	Send data to the server.

Properties

Property

Syntax	Description
All properties of the GuiComponent Object [page 78]: • ContainerType • Id • Name • Parent • Type • TypeAsNumber	
All properties of the GuiVComponent Object [page 263]: • AccLabelCollection • AccText • AccTextOnRequest • AccTooltip • Changeable • DefaultTooltip • Height • IconName • IsSymbolFont • Left • Modified • ParentFrame • ScreenLeft • ScreenTop • Text • Tooltip • Top • Width	

Property

Syntax	Description
All properties of the GuiContainer Object [page 83]: • Children	
All properties of the GuiShell Object [page 195]: • AccDescription • CurrentContextMenu • DragDropSupported • Handle • OcxEvents • SubType	
ChartCount (Read-only) `Public Property ChartCount As Long`	Number of charts.

1.2.5 GuiBox Object

Description

A GuiBox is a simple frame with a name (also called a "Group Box"). The items inside the frame are not children of the box. The type prefix is "box".

GuiBox extends the GuiVComponent Object [page 263].

Methods

Method

Syntax	Description
All methods of the GuiVComponent Object [page 263]: • DumpState • SetFocus • Visualize	

Properties

Property Syntax	Description
All properties of the GuiComponent Object [page 78]: • ContainerType • Id • Name • Parent • Type • TypeAsNumber	
All properties of the GuiVComponent Object [page 263]: • AccLabelCollection • AccText • AccTextOnRequest • AccTooltip • Changeable • DefaultTooltip • Height • IconName • IsSymbolFont • Left • Modified • ParentFrame • ScreenLeft • ScreenTop • Text • Tooltip • Top • Width	
CharHeight (Read-only) `Public Property CharHeight As Long`	Height of the GuiBox in character metric.
CharLeft (Read-only) `Public Property CharLeft As Long`	Left coordinate of the GuiBox in character metric.
CharTop (Read-only) `Public Property CharTop As Long`	Top coordinate of the GuiBox in character metric.

Property

Syntax	Description
CharWidth (Read-only) `Public Property CharWidth As Long`	Width of the GuiBox in character metric.

1.2.6 GuiButton Object

Description

GuiButton represents all push buttons that are on dynpros, the toolbar or in table controls. GuiButton extends the GuiVComponent Object [page 263]. The type prefix is btn, the name property is the fieldname taken from the SAP data dictionary There is one exception: For tabstrip buttons, it is the button id set in screen painter that is taken from the SAP data dictionary.

Methods

Method

Syntax	Description
All methods of the GuiVComponent Object [page 263]: • DumpState • SetFocus • Visualize	

Press	This emulates manually pressing a button. Pressing a button will always cause server communication to occur, rendering all references to elements below the window level invalid. The following code will therefore fail:
	`Set TextField = session.findById(".../txtF1")`
	`session.findById(".../btnPB5").press`
	`TextField.text = "Hello"`

Properties

Property	
Syntax	**Description**
All properties of the GuiComponent Object [page 78]: • ContainerType • Id • Name • Parent • Type • TypeAsNumber	
All properties of the GuiVComponent Object [page 263]: • AccLabelCollection • AccText • AccTextOnRequest • AccTooltip • Changeable • DefaultTooltip • Height • IconName • IsSymbolFont • Left • Modified • ParentFrame • ScreenLeft • ScreenTop • Text • Tooltip • Top • Width	
All properties of the GuiBox Object [page 51]: • CharHeight • CharLeft • CharTop • CharWidth	

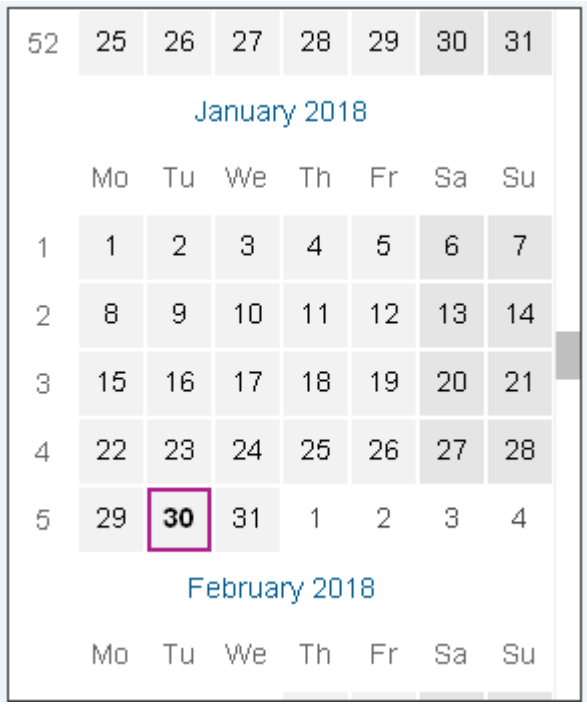
Property Syntax	Description
Emphasized (Read-only) `Public Property Emphasized As Byte`	This property is True if the button is displayed emphasized (in Fiori Visual Themes: The leftmost button in the footer and buttons configured as "Fiori Usage D Display<->Change"). **i Note** • If SAP GUI is running without a Fiori Visual Theme (like Belize) this property is always False. • This property is available as of SAP GUI for Windows 7.60.
LeftLabel `Public Property LeftLabel As GuiVComponent`	Left label of the GuiButton. The label is assigned in the Screen Painter, using the flag 'assign left'.
RightLabel `Public Property RightLabel As GuiVComponent`	Right label of the GuiButton. This property is set in Screen Painter using the 'assign right' flag.

1.2.7 GuiCalendar Object

Description

The calendar control can be used to select single dates or periods of time. GuiCalendar extends the GuiShell Object [page 195].

Example



Methods

Method	
Syntax	**Description**
All methods of the GuiVComponent Object [page 263]: • DumpState • SetFocus • Visualize	
All methods of the GuiContainer Object [page 83]: • FindById	
All methods of the GuiVContainer Object [page 268]: • FindAllByName • FindAllByNameEx • FindByName • FindByNameEx	

Method Syntax	Description
All methods of the GuiShell Object [page 195]: • SelectContextMenuItem • SelectContextMenuItemByPosition • SelectContextMenuItemByText	

ContextMenu
`Public Sub ContextMenu( _`
`  ByVal CtxMenuId As Long, _`
`  ByVal CtxMenuCellRow As Long, _`
`  ByVal CtxMenuCellCol As Long, _`
`  ByVal DateBegin As String, _`
`  ByVal DateEnd As String _`
`)`
	Calling this function opens a context menu.
	Parameter **CtxMenuId** indicates the cell type of the cell in which the context menu was opened:
	Value / **Cell Type** / **Description**
	0 / Date / Invocation on a cell with a single date
	1 / Weekday / Weekday Invocation on a button for a certain day of the week
	2 / Week / Invocation on a button for a specific week

CreateDate
`Public Function CreateDate( _`
`  ByVal day As Long, _`
`  ByVal month As Long, _`
`  ByVal year As Long _`
`) As String`	

GetColor
`Public Function GetColor( _`
`  ByVal from As String _`
`) As Long`	

GetColorInfo
`Public Function GetColorInfo( _`
`  ByVal Color As Long _`
`) As String`
	Returns the explanation defined by the application for semantic colors used in the GuiCalendar (starting with index 0).

GetDateTooltip
`Public Function GetDateTooltip( _`
`  ByVal date As String _`
`) As String`
	Returns the tooltip text of the date specified as parameter (in format "YYYYMMDD").

Method Syntax	Description

GetDay
`Public Function GetDay( _`
`  ByVal date As String _`
`) As Long`
	Returns the day of the date specified as parameter (in format “YYYYMMDD”).

GetMonth
`Public Function GetMonth( _`
`  ByVal date As String _`
`) As Long`
	Returns the month of the date specified as parameter (in format “YYYYMMDD”).

GetWeekday
`Public Function GetWeekday( _`
`  ByVal date As String _`
`) As String`
	Returns the week day of the date specified as parameter (in format “YYYYMMDD”).

GetWeekNumber
`Public Function GetWeekNumber( _`
`  ByVal date As String _`
`) As Long`
	Returns the week number of the date specified as parameter (in format “YYYYMMDD”).

GetYear
`Public Function GetYear( _`
`  ByVal date As String _`
`) As Long`
	Returns the year of the date specified as parameter (in format “YYYYMMDD”).

IsWeekend
`Public Function IsWeekend( _`
`  ByVal date As String _`
`) As Long`
	Returns True if the date specified by the parameter is at a weekend.

SelectMonth
`Public Sub SelectMonth( _`
`  ByVal month As Long, _`
`  ByVal year`
`  As Long _ )`
	Selects the month specified by the parameters (starting with index 1).

SelectRange
`Public Sub SelectRange( _`
`  ByVal from As String, _`
`  ByVal to As`
`  String _`
`)`
	Selects the range specified by the parameters (in format “YYYYMMDD”).

Method

Syntax	Description

SelectWeek
`Public Sub SelectWeek( _`
`  ByVal week As Long, _`
`  ByVal year`
`  As Long _`
`)`
	Selects the week specified by the parameters (starting with index 0).

Properties

Property

Syntax	Description
All properties of the GuiComponent Object [page 78]: • ContainerType • Id • Name • Parent • Type • TypeAsNumber	
All properties of the GuiVComponent Object [page 263]: • AccLabelCollection • AccText • AccTextOnRequest • AccTooltip • Changeable • DefaultTooltip • Height • IconName • IsSymbolFont • Left • Modified • ParentFrame • ScreenLeft • ScreenTop • Text • Tooltip • Top • Width	

Property	
Syntax	**Description**
All properties of the GuiContainer Object [page 83]: • Children	
All properties of the GuiShell Object [page 195]: • AccDescription • CurrentContextMenu • DragDropSupported • Handle • OcxEvents • SubType	
endSelection (Read-only) `Public Property endSelection As String`	The last day of the selected date range (in format "YYYYMMDD").
FirstVisibleDate (Read-write) `Public Property FirstVisibleDate As String`	This is the earliest date visible in the calendar control. In the example above the value would be "20171225". Property type
FocusDate (Read-write) `Public Property FocusDate As String`	The currently focused date (identified by the focus border; see picture above) in the calendar control is available in the format "YYYYMMDD". In this example it is "20180130".
FocusedElement (Read-only) `Public Property FocusedElement As Long`	This property indicates which part of a composite GuiCalendar control currently has focus. The following values are possible: • 0 - "InputField": The input field (picker) to manually enter a date currently has focus • 1 - "Button": The push button to open the navigator pane currently has focus • 2 - "Navigator": The popup navigator pane is open and currently has focus **i Note** This property is available as of SAP GUI for Windows 7.50 patchlevel 8 and SAP GUI for Windows 7.60.
horizontal (Read-only) `Public Property horizontal As Long`	This property contains True if the GuiCalendar has a horizontal orientation, else it contains False.

Property

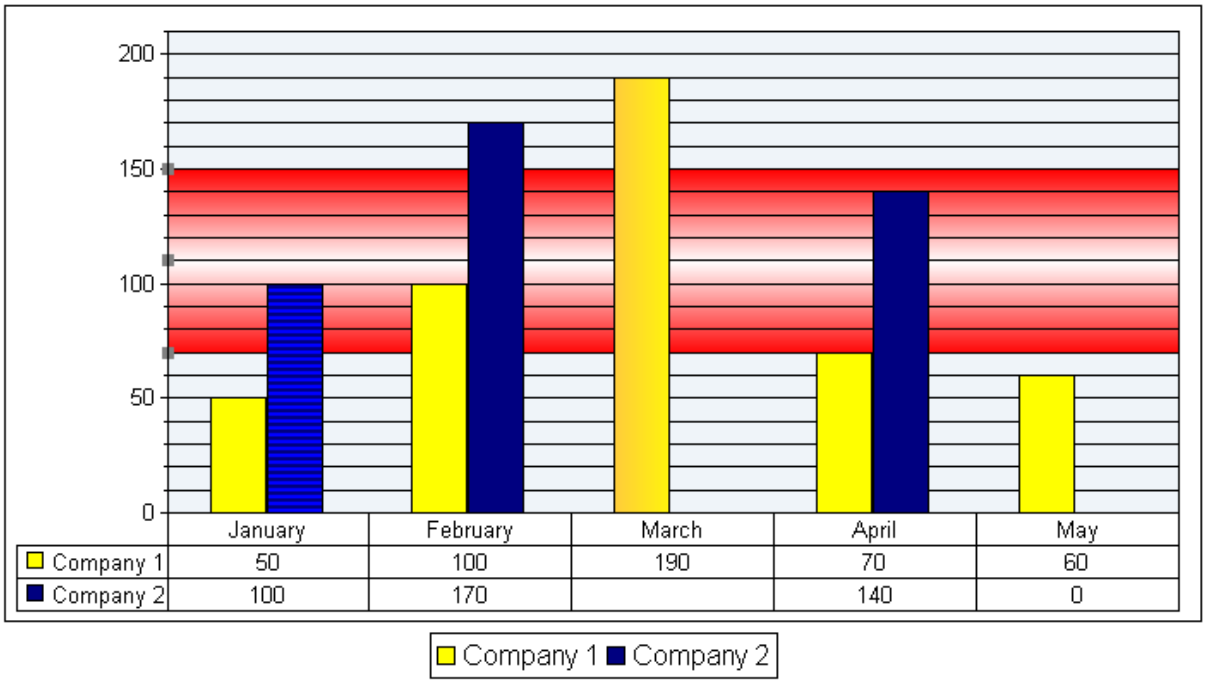
Syntax	Description
LastVisibleDate (Read-write) `Public Property LastVisibleDate As String`	The last date that is currently displayed by the GuiCalendar (in format “YYYYMMDD”).
SelectionInterval (Read-write) `Public Property SelectionInterval As String`	The interval is represented by two concatenated date strings separated by a comma.
startSelection (Read-only) `Public Property startSelection As String`	The starting day of the selected date range (in format “YYYYMMDD”).
Today (Read-only) `Public Property Today As String`	The current day (in format “YYYYMMDD”).

1.2.8 GuiChart Object

Description

The GuiChart object is of a very technical nature. It should only be used for recording and playback, as most of the parameters cannot be determined in any other way.

Example



Methods

Method	
Syntax	**Description**
All methods of the GuiVComponent Object [page 263]: • DumpState • SetFocus • Visualize	
All methods of the GuiContainer Object [page 83]: • FindById	
All methods of the GuiVContainer Object [page 268]: • FindAllByName • FindAllByNameEx • FindByName • FindByNameEx	

Method	
Syntax	**Description**
All methods of the GuiShell Object [page 195]: • SelectContextMenuItem • SelectContextMenuItemByPosition • SelectContextMenuItemByText	

ValueChange
`Public Sub ValueChange( _`
`  ByVal Series As Long, _`
`  ByVal Point As Long, _`
`  ByVal XValue As String, _`
`  ByVal YValue As String, _`
`  ByVal DataChange As Byte, _`
`  ByVal Id As String, _`
`  ByVal ZValue As String, _`
`  ByVal ChangeFlag As Long _`
`)`
	Series: Number of the data set within the row that should be changed.
	Point: Number of the data point within the row that should be changed.
	XValue: New x value.
	YValue: New y value.
	DataChange: Setting this parameter to True means the value was not changed interactively within the graphic but rather by entering the new value on the DataPoint property page.
	Id: GFW data container id of the changed point. May be used instead of the pair series/point.
	ZValue: New z value.
	ChangeFlag: Notify which value was changed or if it was a time value. The value is set as a bit array, using the lower 5 bits.
	1 x
	2 y
	4 x is time value
	8 y is time value
	16 z
	If the new value is a point in time, it should be set using a string of the format **mm/dd/yyyy hh:mm:ss**.

Properties

Property	
Syntax	**Description**
All properties of the GuiComponent Object [page 78]: • ContainerType • Id • Name • Parent • Type • TypeAsNumber	
All properties of the GuiVComponent Object [page 263]: • AccLabelCollection • AccText • AccTextOnRequest • AccTooltip • Changeable • DefaultTooltip • Height • IconName • IsSymbolFont • Left • Modified • ParentFrame • ScreenLeft • ScreenTop • Text • Tooltip • Top • Width	
All additional properties of the GuiContainer Object [page 83]: • Children	
All additional properties of the GuiShell Object [page 195]: • AccDescription • CurrentContextMenu • DragDropSupported • Handle • OcxEvents • SubType	

1.2.9 GuiCheckBox Object

Description

GuiCheckBox extends the GuiVComponent Object [page 263]. The type prefix is chk, the name is the fieldname taken from the SAP data dictionary.

Methods

Method Syntax	Description
All methods of the GuiVComponent Object [page 263]: • DumpState • SetFocus • Visualize	

GetListProperty
`Public Function GetListProperty( _`
`  ByVal Property As String _`
`) As String`
	For more information refer to the documentation about method **GetListProperty** within GuiLabel Object [page 134].

GetListPropertyNonRec
`Public Function`
`GetListPropertyNonRec( _`
`  ByVal Property As String _`
`) As String`
	This method returns information that is compiled on the server to enhance the ABAP lists with accessibility information. See GuiLabel::GetListProperty for a description of available attributes. In contrast to the method GetListProperty, GetListPropertyNonRec will only return information that is set for the specific element, and ignore list properties set for parent elements.

Properties

Property Syntax	Description
All properties of the GuiComponent Object [page 78]: • ContainerType • Id • Name • Parent • Type • TypeAsNumber	
All properties of the GuiVComponent Object [page 263]: • AccLabelCollection • AccText • AccTextOnRequest • AccTooltip • Changeable • DefaultTooltip • Height • IconName • IsSymbolFont • Left • Modified • ParentFrame • ScreenLeft • ScreenTop • Text • Tooltip • Top • Width	
All properties of the GuiBox Object [page 51]: • CharHeight • CharLeft • CharTop • CharWidth	
ColorIndex (Read-only) `Public Property ColorIndex As Long`	This number defines the index of the list color of this element.

Property

Syntax	Description
ColorIntensified (Read-only) `Public Property ColorIntensified As Byte`	This property is True if the Intensified flag is set in Screen Painter for this dynpro element.
ColorInverse (Read-only) `Public Property ColorInverse As Byte`	This property is True if the inverse color style is set in Screen Painter for the element.
Flushing (Read-only) `Public Property Flushing As Byte`	Some components such as radio buttons or checkboxes may cause a round trip when their value is changed. If this is the case, the **Flushing** property is True.
IsLeftLabel (Read-only) `Public Property IsLeftLabel As Byte`	This property is True if the component has the 'assign left' flag.
IsListElement (Read-only) `Public Property IsListElement As Byte`	This property is True if the element is on an ABAP list, not a dynpro screen.
IsRightLabel (Read-only) `Public Property IsRightLabel As Byte`	This property is True if the component has the 'assign right' flag.
LeftLabel (Read-only) `Public Property LeftLabel As GuiVComponent`	Left label of the component. The label is assigned in the Screen Painter, using the flag 'assign left'.
RightLabel (Read-only) `Public Property RightLabel As GuiVComponent`	Right label of the component. This property is set in Screen Painter using the 'assign right' flag.
RowText (Read-only) `Public Property RowText As String`	This property is only available in ABAP list screens. It returns the text of the while line containing the current component.
Selected (Read-write) `Public Property Selected As Byte`	Like radio buttons, checking a checkbox can cause server communication, depending on the ABAP Screen Painter definition.

1.2.10 GuiCollection Collection

Description

GuiCollection is similar to the GuiComponentCollection Collection [page 79], but its members are not necessarily extensions of the GuiComponent Object [page 78]. It can be used to pass a collection as a parameter to functions of scriptable objects. An object of this class is created by calling the **CreateGuiCollection** function of the GuiApplication Object [page 37].

Methods

Method Syntax	Description

Add
`Public Sub Add( _`
`  ByVal Item As Variant _`
`)`
	After a GuiCollection has been created, items can be added by calling the add function.

ElementAt
`Public Function ElementAt( _`
`  ByVal Index As Long _`
`) As Variant`
	This function returns the member in the collection at position index, where index may range from 0 to count-1. If no member can be found for the given index, an exception is raised.

Item
`Public Function Item( _`
`  ByVal Index As Variant _`
`) As Variant`
	This function returns the member in the collection at position index, where index may range from 0 to count-1. It has been added for compatibility with Microsoft Visual Basic collections. If no member can be found for the given index, an exception is raised.

Properties

Property Syntax	Description
Count (Read-only) `Public Property Count As Long`	The number of elements in the collection. This property has been added for compatibility with Microsoft Visual Basic collections.
Length (Read-only) `Public Property Length As Long`	The number of elements in the collection.
NewEnum (Read-only) `Public Property NewEnum As Unknown`	This property has been added for compatibility with Microsoft Visual Basic collections.
Type (Read-only) `Public Property Type As String`	The type information can be used to determine which properties and methods an object supports. The value is the name of the type taken from this documentation. The value for GuiCollection is 'GuiCollection'.
TypeAsNumber (Read-only) `Public Property TypeAsNumber As Long`	While the Type property is a string value, the **TypeAsNumber** property is a long value that can alternatively be used to identify an object's type . It was added for better performance in methods such as **FindByIdEx**. Possible values for this property are taken from the GuiComponentType [page 279]enumeration.

1.2.11 GuiColorSelector Object

Description

GuiColorSelector displays a set of colors for selection. It extends the GuiShell Object [page 195].

Example



Methods

Method Syntax	Description
All methods of the GuiVComponent Object [page 263]: • DumpState • SetFocus • Visualize	
All methods of the GuiContainer Object [page 83]: • FindById	
All methods of the GuiVContainer Object [page 268]: • FindAllByName • FindAllByNameEx • FindByName • FindByNameEx	
All methods of the GuiShell Object [page 195]: • SelectContextMenuItem • SelectContextMenuItemByPosition • SelectContextMenuItemByText	

ChangeSelection
`Public Sub ChangeSelection( _`
`  ByVal i As Integer _`
`)`
	This function emulates the user's selection of the color at the given index position.

Properties

Property Syntax	Description
All properties of the GuiComponent Object [page 78]: • ContainerType • Id • Name • Parent • Type • TypeAsNumber	
All properties of the GuiVComponent Object [page 263]: • AccLabelCollection • AccText • AccTextOnRequest • AccTooltip • Changeable • DefaultTooltip • Height • IconName • IsSymbolFont • Left • Modified • ParentFrame • ScreenLeft • ScreenTop • Text • Tooltip • Top • Width	
All properties of the GuiContainer Object [page 83]: • Children	
All additional properties of the GuiShell Object [page 195]: • AccDescription • CurrentContextMenu • DragDropSupported • Handle • OcxEvents • SubType	

1.2.12 GuiComboBox Object

Description

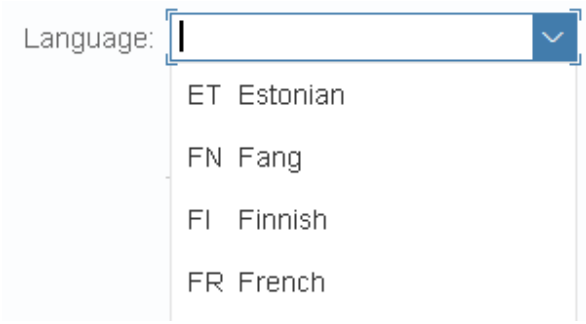
The GuiComboBox looks somewhat similar to GuiCTextField, but has a completely different implementation. While pressing the combo box button of a GuiCTextField will open a new dynpro or control in which a selection can be made, GuiComboBox retrieves all possible choices on initialization from the server, so the selection is done solely on the client. GuiComboBox extends the GuiVComponent Object [page 263]. The type prefix is cmb, the name is the fieldname taken from the SAP data dictionary. GuiComboBox inherits from the GuiVComponent Object [page 263].

Methods

Method Syntax	Description
All methods of the GuiVComponent Object [page 263]: • DumpState • SetFocus • Visualize	
SetKeySpace `Public Sub SetKeySpace()`	This function sets the key property of the combo box to the space character. It was introduced for eCATT.

Properties

Property Syntax	Description
All properties of the GuiComponent Object [page 78]: • ContainerType • Id • Name • Parent • Type • TypeAsNumber	
All properties of the GuiVComponent Object [page 263]: • AccLabelCollection • AccText • AccTextOnRequest • AccTooltip • Changeable • DefaultTooltip • Height • IconName • IsSymbolFont • Left • Modified • ParentFrame • ScreenLeft • ScreenTop • Text • Tooltip • Top • Width	
CharHeight (Read-only) `Public Property CharHeight As Long`	Height of the GuiComboBox in character metric.
CharLeft (Read-only) `Public Property CharLeft As Long`	Left coordinate of the GuiComboBox in character metric.
CharTop (Read-only) `Public Property CharTop As Long`	Top coordinate of the GuiComboBox in character metric.

Property	
Syntax	**Description**
CharWidth (Read-only) `Public Property CharWidth As Long`	Width of the GuiComboBox in character metric.
CurListBoxEntry (Read-only) `Public Property CurListBoxEntry As GuiComboBoxEntry`	The currently focused entry of the dropdown list.
Entries (Read-only) `Public Property Entries() As GuiCollection`	All members of this collection are of GuiComboBoxEntry type and have just two properties, key and value, both of type String. The key data can be displayed in SAP GUI by setting the 'Show keys...' options in SAP GUI options dialog.  In this example the first column contains the key property and the second column contains the value property.
Flushing (Read-only) `Public Property Flushing As Byte`	Some components such as radio buttons, checkboxes or combo boxes may cause a round trip when their value is changed. If this is the case, the **Flushing** property is True.
Highlighted (Read-only) `Public Property Highlighted As Byte`	This property is True if the Highlighted flag is set in the Screen Painter for the combo box.
IsLeftLabel (Read-only) `Public Property IsLeftLabel As Byte`	This property is True if the combo box has the 'assign left' flag.
IsListBoxActive (Read-only) `Public Property IsListBoxActive As Byte`	This property is True if the list box of the combo box is currently open.
IsRightLabel (Read-only) `Public Property IsRightLabel As Byte`	This property is True if the combo box has the 'assign right' flag.

Property

Syntax	Description
Key (Read-write) `Public Property Key As String`	This is the key of the currently selected item. You can change this item by setting the **Key** property to a new value.
LeftLabel (Read-only) `Public Property Modified As Byte`	This label has been defined in ABAP Screen Painter to be the left label of the combo box.
Required (Read-only) `Public Property Required As Byte`	If the required flag is set for a combo box then the empty entry is not selectable from the list.
RightLabel (Read-only) `Public Property RightLabel As GuiVComponent`	This label has been defined in ABAP Screen Painter to be the right label of the combo box.
ShowKey (Read-only) `Public Property ShowKey As Byte`	This property is True if the combo box shows both keys and values (this can be configured by setting the 'Show keys...' options in SAP GUI options dialog).
Value (Read-write) `Public Property Value As String`	This is the value of the currently selected item. You can change this item by setting the value property to a new value.

1.2.13 GuiComboBoxControl Object

Methods

Method

Syntax	Description
All methods of the GuiVComponent Object [page 263]: • DumpState • SetFocus • Visualize	
All methods of the GuiContainer Object [page 83]: • FindById	

Method

Syntax	Description
All methods of the GuiVContainer Object [page 268]: • FindAllByName • FindAllByNameEx • FindByName • FindByNameEx	
All methods of the GuiShell Object [page 195]: • SelectContextMenuItem • SelectContextMenuItemByPosition • SelectContextMenuItemByText	
FireSelected `Public Sub FireSelected()`	Sends "selected" event.

Properties

Property

Syntax	Description
All properties of the GuiComponent Object [page 78]: • ContainerType • Id • Name • Parent • Type • TypeAsNumber	

Property

Syntax	Description
All properties of the GuiVComponent Object [page 263]: • AccLabelCollection • AccText • AccTextOnRequest • AccTooltip • Changeable • DefaultTooltip • Height • IconName • IsSymbolFont • Left • Modified • ParentFrame • ScreenLeft • ScreenTop • Text • Tooltip • Top • Width	
All properties of the GuiContainer Object [page 83]: • Children	
All properties of the GuiShell Object [page 195]: • AccDescription • CurrentContextMenu • DragDropSupported • Handle • OcxEvents • SubType	
Entries (Read-only) `Public Property Entries As GuiCollection`	Entries are again a GuiCollection with: key(index=0), text(index=1) the text for every entry you can get via this collection.
LabelText (Read-only) `Public Property LabelText As String`	Text of the label.
Selected (Read-write) `Public Property Selected As String`	The key of the currently selected entry of the combo box.

Property Syntax	Description
Text (Read-only) `Public Property Text As String`	Current text of the combo box.

1.2.14 GuiComboBoxEntry Object

Description

Members of the Entries collection of a GuiComboBox are of type GuiComBoxEntry.

Properties

Property Syntax	Description
Key (Read-only) `Public Property Key As String`	Key value of the combo box entry.
Pos (Read-only) `Public Property Pos As Long`	Position of the combo box entry. The range is from 1 to the number of entries in the combo box.
Value (Read-only) `Public Property Value As String`	Value of the combo box entry.

1.2.15 GuiComponent Object

Description

GuiComponent is the base class for most classes in the Scripting API. It was designed to allow generic programming, meaning you can work with objects without knowing their exact type.

Properties

Property Syntax	Description
ContainerType (Read-only) `Public Property ContainerType As Byte`	This property is TRUE, if the object is a container and therefore has the **Children** property.
Id (Read-only) `Public Property Id As String`	An object id is a unique textual identifier for the object. It is built in a URLlike formatting, starting at the GuiApplication object and drilling down to the respective object.
Name (Read-only) `Public Property Name As String`	The name property is especially useful when working with simple scripts that only access dynpro fields. In that case a field can be found using its name and type information, which is easier to read than a possibly very long id. However, there is no guarantee that there are no two objects with the same name and type in a given dynpro.
Parent (Read-only) `Public Property Parent As Object`	The parent of an object is one level higher in the runtime hierarchy. An object is always in the children collection of its parent.
Type (Read-only) `Public Property Type As String`	The type information of GuiComponent can be used to determine which properties and methods an object supports. The value of the type string is the name of the type taken from this documentation.
TypeAsNumber (Read-only) `Public Property TypeAsNumber As Long`	While the **Type** property is a string value, the **TypeAsNumber** property is a long value that can alternatively be used to identify an object's type . It was added for better performance in methods such as **FindByIdEx**. Possible values for this property are taken from the GuiComponentType [page 279]enumeration.

1.2.16 GuiComponentCollection Collection

Description

The GuiComponentCollection is used for collections elements such as the **Children** property of containers. Each element of the collection is an extension of GuiComponent.

Methods

Method	
Syntax	**Description**

ElementAt
`Public Function ElementAt( _`
`  ByVal Index As Long _`
`) As GuiComponent`
	This function returns the member in the collection at position index, where index may range from 0 to count-1. If no member can be found for the given index, the exception Gui_Err_Enumerator_Index (614) is raised.

Item
`Public Function Item( _`
`  ByVal Index As Variant _`
`) As GuiComponent`
	This function returns the member in the collection at position index, where index may range from 0 to count-1. It has been added for compatibility with Microsoft Visual Basic collections. If no member can be found for the given index the exception Gui_Err_Enumerator_Index (614) is raised.

Properties

Property	
Syntax	**Description**
Count (Read-only) `Public Property Count As Long`	The number of elements in the collection. This property is used implicitly from Visual Basic applications.
Length (Read-only) `Public Property Length As Long`	The number of elements in the collection.
NewEnum (Read-only) `Public Property NewEnum As Unknown`	This property is used implicitly from Visual Basic applications.

Property

Syntax	Description
Type (Read-only) `Public Property Type As String`	The type information can be used to determine which properties and methods an object supports. The value of the type string is the name of the type taken from this documentation. The value is 'GuiComponentCollection'.
TypeAsNumber (Read-only) `Public Property TypeAsNumber As Long`	While the **Type** property is a string value, the **TypeAsNumber** property is a long value that can alternatively be used to identify an object's type . It was added for better performance in methods such as **FindByIdEx**. Possible values for this property are taken from the GuiComponentType [page 279]enumeration.

1.2.17 GuiConnection Object

Description

A GuiConnection represents the connection between SAP GUI and an application server. Connections can be opened from SAP Logon or from GuiApplication's openConnection and openConnectionByConnectionString methods. GuiConnection extends the GuiContainer Object [page 83]. The type prefix for GuiConnection is con, the name is con plus the connection number in square brackets.

Remarks

It is possible to connect to an application server from ABAP using the following command:

```
CALL FUNCTION func DESTINATION dest.
```

However, this connection is implemented as a re-direction between the two application servers involved. There will therefore be no new GuiConnection object available and the existing object will not reflect the server switch.

Methods

Method Syntax	Description
All methods of the GuiContainer Object [page 83]: • FindById	
CloseConnection `Public Sub CloseConnection()`	This method closes a connection along with all its sessions.

CloseSession
`Public Sub CloseSession( _`
`  ByVal Id As String _`
`)`
	A session can be closed by calling this method of the connection. Closing the last session of a connection will close the connection, too.
	The parameter "Id" must contain the id of the session to close (like "/app/con[0]/ses[0]").

Properties

Property Syntax	Description
All properties of the GuiComponent Object [page 78]: • ContainerType • Id • Name • Parent • Type • TypeAsNumber	
Children (Read-only) `Public Property Children As GuiComponentCollection`	This collection contains all direct children of the object.
ConnectionString (Read-only) `Public Property ConnectionString As String`	This property contains the connection string defining the backend connection. It is more difficult to read, but it doesn't rely on the SAP Logon entries. More information on connection strings can be found in chapter Method OpenConnectionByConnectionString [page 44].

Property

Syntax	Description
Description (Read-only) `Public Property Description As String`	This description is only available if the connection was started either from SAP Logon or using GuiApplication.OpenConnection. In both cases the description can then be used when calling the **OpenConnection** method to play back a script on the same system.
DisabledByServer (Read-only) `Public Property DisabledByServer As Byte`	This property is set to True if the scripting support has not been enabled for the application server.
Sessions (Read-only) `Public Property Sessions As GuiComponentCollection`	This property is another name for the **Children** property. It was added for better readability as all the children of GuiConnection are sessions. Accessing either the children property or the **Sessions** property can cause the exception Gui_Err_Scripting_Disabled_Srv (624) to be raised if the respective application server has not enabled the scripting support.

1.2.18 GuiContainer Object

Description

This interface resembles GuiVContainer. The only difference is that it is not intended for visual objects but rather administrative objects such as connections or sessions. Objects exposing this interface will therefore support GuiComponent but not GuiVComponent. GuiContainer extends the GuiComponent Object [page 78].

Methods

Method	
Syntax	**Description**

FindById
`Public Function FindById( _`
`  ByVal Id As String, _`
`  Optional ByVal Raise As Variant _`
`) As GuiComponent`
	Search through the object's descendants for a given id. If the parameter is a fully qualified id, the function will first check if the container object's id is a prefix of the id parameter. If that is the case, this prefix is truncated. If no descendant with the given id can be found the function raises an exception unless the optional parameter raise is set to False.

Properties

Property	
Syntax	**Description**
All properties of the GuiComponent Object [page 78]: • ContainerType • Id • Name • Parent • Type • TypeAsNumber	
Children (Read-only) `Public Property Children As GuiComponentCollection`	This collection contains all direct children of the object.

1.2.19 GuiContainerShell Object

Description

A GuiContainerShell is a wrapper for a set of the GuiShell Object [page 195]. GuiContainerShell extends the GuiVContainer Object [page 268]. The type prefix is shellcont, the name is the last part of the id, shellcont[n].

Methods

Method	
Syntax	**Description**
All methods of the GuiVComponent Object [page 263]: • DumpState • SetFocus • Visualize	
All methods of the GuiContainer Object [page 83]: • FindById	
All methods of the GuiVContainer Object [page 268]: • FindAllByName • FindAllByNameEx • FindByName • FindByNameEx	

Properties

Property	
Syntax	**Description**
All properties of the GuiComponent Object [page 78]: • ContainerType • Id • Name • Parent • Type • TypeAsNumber	

Property

Syntax	Description
All properties of the GuiVComponent Object [page 263]: • AccLabelCollection • AccText • AccTextOnRequest • AccTooltip • Changeable • DefaultTooltip • Height • IconName • IsSymbolFont • Left • Modified • ParentFrame • ScreenLeft • ScreenTop • Text • Tooltip • Top • Width	
All properties of the GuiContainer Object [page 83]: • Children	
DockerIsVertical (Read-only) `Public Property DockerIsVertical As Byte`	Is TRUE if the container is a vertical docker control.
DockerPixelSize (Read-write) `Public Property DockerPixelSize As Long`	Returns the size of the docker control in pixels.

1.2.20 GuiContextMenu Object

Description

A GuiContextMenu may have other GuiContextMenu objects as children. GuiContextMenu extends the GuiMenu Object [page 150]. The type is mnu, the name is the function code that is sent to the system when the menu item is selected.

Methods

Method Syntax	Description
All methods of the GuiVComponent Object [page 263]: • DumpState • SetFocus • Visualize	
All methods of the GuiContainer Object [page 83]: • FindById	
All additional methods of the GuiVContainer Object [page 268]: • FindAllByName • FindAllByNameEx • FindByName • FindByNameEx	
Select `Public Sub Select()`	Select the context menu item.

Properties

Property Syntax	Description
All properties of the GuiComponent Object [page 78]: • ContainerType • Id • Name • Parent • Type • TypeAsNumber	
All properties of the GuiVComponent Object [page 263]: • AccLabelCollection • AccText • AccTextOnRequest • AccTooltip • Changeable • DefaultTooltip • Height • IconName • IsSymbolFont • Left • Modified • ParentFrame • ScreenLeft • ScreenTop • Text • Tooltip • Top • Width	
All properties of the GuiContainer Object [page 83]: • Children	

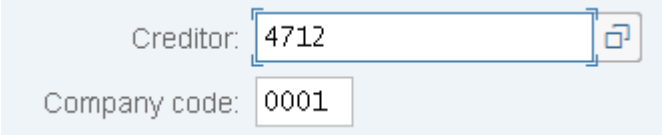
1.2.21 GuiCTextField Object

Description

If the cursor is set into a text field of type GuiCTextField a combo box button is displayed to the right of the text field. Pressing this button is equivalent to pressing the F4 key. The button is not represented in the scripting object model as a separate object; it is considered to be part of the text field.

There are no other differences between GuiTextField and GuiCTextField. GuiCTextField extends the GuiTextField [page 232]. The type prefix is ctxt, the name is the Fieldname taken from the SAP data dictionary.

Example



This is an example of GuiCTextField type text field, where the upper field has the focus. Please note that the button is only displayed when the corresponding input field has the focus unless the ABAP application has defined the button to be shown permanently.

Methods

Method	
Syntax	**Description**
All methods of the GuiVComponent Object [page 263]: • DumpState • SetFocus • Visualize	

Properties

Property	
Syntax	**Description**
All properties of the GuiComponent Object [page 78]: • ContainerType • Id • Name • Parent • Type • TypeAsNumber	
All properties of the GuiVComponent Object [page 263]: • AccLabelCollection • AccText • AccTextOnRequest • AccTooltip • Changeable • DefaultTooltip • Height • IconName • IsSymbolFont • Left • Modified • ParentFrame • ScreenLeft • ScreenTop • Text • Tooltip • Top • Width	
All properties of the GuiBox Object [page 51]: • CharHeight • CharLeft • CharTop • CharWidth	

Property	
Syntax	**Description**
All properties of the GuiTextField [page 232] with one exception: Property `IsListElement` in not available for this object since F4 help is not available for input fields within ABAP lists! • CaretPosition • DisplayedText • Highlighted • HistoryCurEntry • HistoryCurIndex • HistoryIsActive • HistoryList • IsHotspot • IsLeftLabel • IsOField • IsRightLabel • LeftLabel • MaxLength • Numerical • Required • RightLabel	

1.2.22 GuiCustomControl Object

Description

The GuiCustomControl is a wrapper object that is used to place ActiveX controls onto dynpro screens. While GuiCustomControl is a dynpro element itself, its children are of GuiContainerShell type, which is a container for controls. GuiCustomControl extends the GuiVContainer Object [page 268]. The type prefix is cntl, the name is the fieldname taken from the SAP data dictionary.

Methods

Method	
Syntax	**Description**
All methods of the GuiVComponent Object [page 263]: • DumpState • SetFocus • Visualize	
All methods of the GuiContainer Object [page 83]: • FindById	
All additional methods of the GuiVContainer Object [page 268]: • FindAllByName • FindAllByNameEx • FindByName • FindByNameEx	

Properties

Property	
Syntax	**Description**
All properties of the GuiComponent Object [page 78]: • ContainerType • Id • Name • Parent • Type • TypeAsNumber	

Property

Syntax	Description
All properties of the GuiVComponent Object [page 263]: • AccLabelCollection • AccText • AccTextOnRequest • AccTooltip • Changeable • DefaultTooltip • Height • IconName • IsSymbolFont • Left • Modified • ParentFrame • ScreenLeft • ScreenTop • Text • Tooltip • Top • Width	
All properties of the GuiContainer Object [page 83]: • Children	
CharHeight (Read-only) `Public Property CharHeight As Long`	Height of the GuiCustomControl in character metric.
CharLeft (Read-only) `Public Property CharLeft As Long`	Left coordinate of the GuiCustomControl in character metric.
CharTop (Read-only) `Public Property CharTop As Long`	Top coordinate of the GuiCustomControl in character metric.
CharWidth (Read-only) `Public Property CharWidth As Long`	Width of the GuiCustomControl in character metric.

1.2.23 GuiDialogShell Object

Description

The GuiDialogShell is an external window that is used as a container for other shells, for example a toolbar. GuiDialogShell extends the GuiVContainer Object [page 268]. The type prefix is shellcont, the name is the last part of the id, shellcont[n].

Example



Methods

Method Syntax	Description
All methods of the GuiVComponent Object [page 263]: • DumpState • SetFocus • Visualize	
All methods of the GuiContainer Object [page 83]: • FindById	
All additional methods of the GuiVContainer Object [page 268]: • FindAllByName • FindAllByNameEx • FindByName • FindByNameEx	
Close `Public Sub Close()`	This method closes the external window.

Properties

Property Syntax	Description
All properties of the GuiComponent Object [page 78]: • ContainerType • Id • Name • Parent • Type • TypeAsNumber	
All properties of the GuiVComponent Object [page 263]: • AccLabelCollection • AccText • AccTextOnRequest • AccTooltip • Changeable • DefaultTooltip • Height • IconName • IsSymbolFont • Left • Modified • ParentFrame • ScreenLeft • ScreenTop • Text • Tooltip • Top • Width	
All properties of the GuiContainer Object [page 83]: • Children	
Title (Read-only) `Public Property Title As String`	Title of the dialog.

1.2.24 GuiEAIViewer2D Object

Description

The GuiEAIViewer2D control is used to view 2-dimensional graphic images in the SAP system. The user can carry out redlining over the loaded image. The scripting wrapper for this control records all user actions during the redlining process and reproduces the same actions when the recorded script is replayed.

GuiEAIViewer2D extends the GuiShell Object [page 195].

Methods

Method	
Syntax	**Description**
All methods of the GuiVComponent Object [page 263]: • DumpState • SetFocus • Visualize	
All methods of the GuiContainer Object [page 83]: • FindById	
All methods of the GuiVContainer Object [page 268]: • FindAllByName • FindAllByNameEx • FindByName • FindByNameEx	
All methods of the GuiShell Object [page 195]: • SelectContextMenuItem • SelectContextMenuItemByPosition • SelectContextMenuItemByText	

annotationTextRequest
`Public Sub annotationTextRequest( _`
`  ByVal strText As String _`
`)`	

Properties

Property Syntax	Description
All properties of the GuiComponent Object [page 78]: • ContainerType • Id • Name • Parent • Type • TypeAsNumber	
All properties of the GuiVComponent Object [page 263]: • AccLabelCollection • AccText • AccTextOnRequest • AccTooltip • Changeable • DefaultTooltip • Height • IconName • IsSymbolFont • Left • Modified • ParentFrame • ScreenLeft • ScreenTop • Text • Tooltip • Top • Width	
All properties of the GuiContainer Object [page 83]: • Children	
All additional properties of the GuiShell Object [page 195]: • AccDescription • CurrentContextMenu • DragDropSupported • Handle • OcxEvents • SubType	

Property

Syntax	Description
AnnotationEnabled (Read-write) `Public Property AnnotationEnabled As Long`	The value of this property is set to 1 when redlining is started. The wrapper control starts recording user actions as soon as this property is set to value 1.
AnnotationMode (Read-write) `Public Property AnnotationMode As Integer`	During redlining, the selected redlining mode is stored in this property.
RedliningStream (Read-write) `Public Property RedliningStream As String`	This property stores the redlining layer as BLOB (Binary large data object). During recording, the whole BLOB is copied into the generated script.

1.2.25 GuiEAIViewer3D Object

The GuiEAIViewer3D control is used to view 3-dimensional graphic images in the SAP system.

GuiEAIViewer3D extends the GuiShell Object [page 195].

Methods

Method

Syntax	Description
All methods of the GuiVComponent Object [page 263]: • DumpState • SetFocus • Visualize	
All methods of the GuiContainer Object [page 83]: • FindById	

Method

Syntax	Description
All methods of the GuiVContainer Object [page 268]: • FindAllByName • FindAllByNameEx • FindByName • FindByNameEx	
All methods of the GuiShell Object [page 195]: • SelectContextMenuItem • SelectContextMenuItemByPosition • SelectContextMenuItemByText	

Properties

Property

Syntax	Description
All properties of the GuiComponent Object [page 78]: • ContainerType • Id • Name • Parent • Type • TypeAsNumber	

Property

Syntax	Description
All properties of the GuiVComponent Object [page 263]: • AccLabelCollection • AccText • AccTextOnRequest • AccTooltip • Changeable • DefaultTooltip • Height • IconName • IsSymbolFont • Left • Modified • ParentFrame • ScreenLeft • ScreenTop • Text • Tooltip • Top • Width	
All properties of the GuiContainer Object [page 83]: • Children	
All additional properties of the GuiShell Object [page 195]: • AccDescription • CurrentContextMenu • DragDropSupported • Handle • OcxEvents • SubType	

1.2.26 GuiEnum Object

Methods

Method	
Syntax	Description

Clone
`Public Function Clone( _`
`  ByRef ppenum As GuiEnum _`
`) As HResult`	

Next
`Public Function Next( _`
`  ByVal celt As ULong, _`
`  ByRef rgvar As Variant, _`
`  ByRef pceltFetched As ULong _`
`) As HResult`	

Reset `Public Function Reset() As HResult`	

Skip
`Public Function Skip( _`
`  ByVal celt As ULong _`
`) As HResult`	

1.2.27 GuiFrameWindow Object

Description

A GuiFrameWindow is a high level visual object in the runtime hierarchy. It can be either the main window or a modal popup window. See the GuiMainWindow and GuiModalWindow sections for examples. GuiFrameWindow itself is an abstract interface. GuiFrameWindow extends the GuiVContainer Object [page 268]. The type prefix is wnd, the name is wnd plus the window number in square brackets.

Methods

Method Syntax	Description
All methods of the GuiVComponent Object [page 263]: • DumpState • SetFocus • Visualize	
All methods of the GuiContainer Object [page 83]: • FindById	
All methods of the GuiVContainer Object [page 268]: • FindAllByName • FindAllByNameEx • FindByName • FindByNameEx	
Close `Public Sub Close()`	The function attempts to close the window. Trying to close the last main window of a session will not succeed immediately; the dialog 'Do you really want to log off?' will be displayed first.

CompBitmap
`Public Function CompBitmap( _`
`  ByVal Filename1 As String, _`
`  ByVal Filename2 As String _`
`) As Long`
	This method compares two bitmap files pixel by pixel.
	Return Type
	The method returns one of the following values:
	• 0: The files do not differ
	• 1: The files differ in size
	• 2: The files have different content
	• 3: There was an error

Method	
Syntax	**Description**

HardCopy
`Public Function HardCopy( _`
`  ByVal Filename As String, _`
`  Optional ByVal ImageType As Variant, _`
`  Optional ByVal xPos As Variant, _`
`  Optional ByVal yPos As Variant, _`
`  Optional ByVal nWidth As Variant, _`
`  Optional ByVal nHeight As Variant _`
`) As String`
	This function dumps a hardcopy of the window as a bitmap file to disk. The parameter is the name of the file. If the function succeeds, then the return value will be the fully qualified path of the file. If no path information is given, then the file will be written to the SAP GUI Documents Folder.

Filename	
ImageType	The following values are valid: • 0: BMP • 1: JPG • 2: PNG • 3: GIF • 4: TIFF BMP is the default format.
xPos	If the optional parameters xPos, yPos, nWidth and nHeight are set, only the specified rectangle of the main window will be captured.
yPos	If the optional parameters xPos, yPos, nWidth and nHeight are set, only the specified rectangle of the main window will be captured.
nWidth	If the optional parameters xPos, yPos, nWidth and nHeight are set, only the specified rectangle of the main window will be captured.
nHeight	If the optional parameters xPos, yPos, nWidth and nHeight are set, only the specified rectangle of the main window will be captured.

Method	
Syntax	**Description**
HardCopyToMemory `Public Function HardCopyToMemory( _ Optional ByVal ImageType As Variant _ ) As Variant`	This function returns a hardcopy of the window as a safe array of bytes. The following values are valid: • 0: BMP • 1: JPG • 2: PNG • 3: GIF BMP is the default format. **Sample Code** The following example shows the hardcopy of an SAP GUI main window (**"wnd[0]"**). `If Not IsObject(application) Then Set SapGuiAuto = GetObject("SAPGUI") Set application = SapGuiAuto.GetScriptingEngine End If If Not IsObject(connection) Then Set connection = application.Children(0) End If If Not IsObject(session) Then Set session = connection.Children(0) End If Image = session.findById("wnd[0]").HardCopyToMemory() Const adTypeBinary = 1 Const adSaveCreateOverWrite = 2 Dim BinaryStream Set BinaryStream = CreateObject("ADODB.Stream") BinaryStream.Type = adTypeBinary BinaryStream.Open BinaryStream.Write Image BinaryStream.SaveToFile "C:\screenshot.bmp", adSaveCreateOverWrite MsgBox "Done"`
Iconify `Public Sub Iconify()`	This will set a window to the iconified state. It is not possible to iconify a specific window of a session; both the main window and all existing modals will be iconfied.

Method	
Syntax	**Description**

IsVKeyAllowed
`Public Function IsVKeyAllowed( _`
`  ByVal VKey As Integer _`
`) As Byte`
	This function returns True if the virtual key VKey is currently available. The VKeys are defined in the menu painter.

JumpBackward `Public Sub JumpBackward()`	Execute the Ctrl+Shift+Tab key on the window to jump backward one block.
JumpForward `Public Sub JumpForward()`	Execute the Ctrl+Tab key on the window to jump forward one block.
Maximize `Public Sub Maximize()`	This will maximize a window. It is not possible to maximize a modal window; it is always the main window which will be maximized.
Restore `Public Sub Restore()`	This will restore a window from its iconified state. It is not possible to restore a specific window of a session; both the main window and all existing modals will be restored.

SendVKey
`Public Sub SendVKey( _`
`  ByVal VKey As Integer _`
`)`
	The virtual key VKey is executed on the window. The VKeys are defined in the menu painter.

ShowMessageBox
`Public Function ShowMessageBox( _`
`  ByVal Title As String, _`
`  ByVal Text As String, _`
`  ByVal MsgIcon As Long, _`
`  ByVal MsgType As Long _`
`) As Long`
	This method shows the message box modal to the GuiFrameWindow. The title and text parameters set the title and text of the message box. The return value will be one of the MESSAGE_RESULT_* values.
	Title
	Text
	MsgIcon — The msgIcon parameter sets the icon to be used for the message box and should be set to one of the MESSAGE_TYPE_* constants.
	MsgType — msgType sets the buttons available on the message box and should be set to one of the MESSAGE_OPTION* constants.

Method

Syntax	Description
TabBackward `Public Sub TabBackward()`	Execute the Shift+Tab key on the window to jump backward one element.
TabForward `Public Sub TabForward()`	Execute the Tab key on the window to jump forward one element.

Properties

Property

Syntax	Description
All properties of the GuiComponent Object [page 78]: • ContainerType • Id • Name • Parent • Type • TypeAsNumber	

Property

Syntax	Description
All properties of the GuiVComponent Object [page 263]: • AccLabelCollection • AccText • AccTextOnRequest • AccTooltip • Changeable • DefaultTooltip • Height • IconName • IsSymbolFont • Left • Modified • ParentFrame • ScreenLeft • ScreenTop • Text • Tooltip • Top • Width	
All properties of the GuiContainer Object [page 83]: • Children	
ElementVisualizationMode (Read-write) `Public Property ElementVisualizationMode() As Boolean`	When elementVisualizationMode is enabled, a hit test can be performed on SAP GUI by moving the cursor over the window. The hit event of the session is fired when a component was found at the mouse position.
GuiFocus (Read-only) `Public Property GuiFocus() As GuiVComponent`	The SystemFocus only supports dynpro elements. To receive information about the currently focused ActiveX control you can access the GuiFocus property.
Handle (Read-only) `Public Property Handle() As Long`	The window handle of the control that is connected to the GuiShell. This is the handle of the underlying window in Microsoft Windows.
Iconic (Read-only) `Public Property Iconic() As Boolean`	This property is True if the window is iconified. It is possible to execute script commands on an iconified window , but there may be undefined results, especially when controls are involved, as these may have invalid size settings.

Property

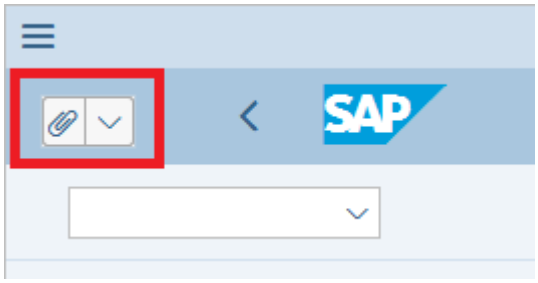
Syntax	Description
SystemFocus (Read-only) `Public Property SystemFocus() As GuiVComponent`	The systemFocus specifies the component that the SAP system is currently seeing as being focused. This value is only valid for dynpro elements and might therefore differ from the focus as seen on the frontend.
WorkingPaneHeight (Read-only) `Public Property WorkingPaneHeight() As Long`	This is the height of the working pane in character metric.
WorkingPaneWidth (Read-only) `Public Property WorkingPaneWidth() As Long`	This is the width of the working pane in character metric. The working pane is the area between the toolbars in the upper area of the window and the status bar at the bottom of the window.

1.2.28 GuiGOSShell Object

Description

The GuiGosShell is only available in New Visual Design mode. GuiGOSShell extends the GuiVContainer Object [page 268]. The type prefix is shellcont, the name is the last part of the id, shellcont[n].

Example



Methods

Method	
Syntax	**Description**
All methods of the GuiVComponent Object [page 263]: • DumpState • SetFocus • Visualize	
All methods of the GuiContainer Object [page 83]: • FindById	
All methods of the GuiVContainer Object [page 268]: • FindAllByName • FindAllByNameEx • FindByName • FindByNameEx	

Properties

Property	
Syntax	**Description**
All properties of the GuiComponent Object [page 78]: • ContainerType • Id • Name • Parent • Type • TypeAsNumber	

Property	
Syntax	**Description**
All properties of the GuiVComponent Object [page 263]: • AccLabelCollection • AccText • AccTextOnRequest • AccTooltip • Changeable • DefaultTooltip • Height • IconName • IsSymbolFont • Left • Modified • ParentFrame • ScreenLeft • ScreenTop • Text • Tooltip • Top • Width	
All properties of the GuiContainer Object [page 83]: • Children	

1.2.29 GuiGraphAdapt Object

Description

For the graphic adapter control only basic members from GuiShell are available. Recording and playback is not possible.

Remarks

In addition to the new, activeX based controls SAP GUI also comes with a set of external graphics executables, for example to display a GANTT chart. These executables are not supported within the API. If during the execution of a script one of these executables is launched, then the script will be blocked.

If you need to automate a process during which a graphics executable is displayed, then you need an automation tool then allows you to both manipulate SAP GUI using the Scripting API, and other Windows applications using native methods.

Methods

Method Syntax	Description
All methods of the GuiVComponent Object [page 263]: • DumpState • SetFocus • Visualize	
All methods of the GuiContainer Object [page 83]: • FindById	
All methods of the GuiVContainer Object [page 268]: • FindAllByName • FindAllByNameEx • FindByName • FindByNameEx	
All methods of the GuiShell Object [page 195]: • SelectContextMenuItem • SelectContextMenuItemByPosition • SelectContextMenuItemByText	

Properties

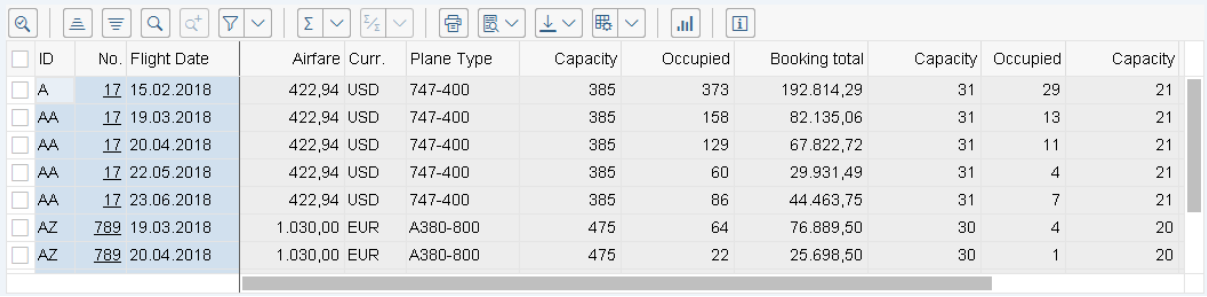
Property Syntax	Description
All properties of the GuiComponent Object [page 78]: • ContainerType • Id • Name • Parent • Type • TypeAsNumber	
All properties of the GuiVComponent Object [page 263]: • AccLabelCollection • AccText • AccTextOnRequest • AccTooltip • Changeable • DefaultTooltip • Height • IconName • IsSymbolFont • Left • Modified • ParentFrame • ScreenLeft • ScreenTop • Text • Tooltip • Top • Width	
All properties of the GuiContainer Object [page 83]: • Children	
All additional properties of the GuiShell Object [page 195]: • AccDescription • CurrentContextMenu • DragDropSupported • Handle • OcxEvents • SubType	

1.2.30 GuiGridView Object

Description

The grid view is similar to the dynpro table control, but significantly more powerful. GuiGridView extends the GuiShell Object [page 195].

Example



ID	No.	Flight Date	Airfare	Curr.	Plane Type	Capacity	Occupied	Booking total	Capacity	Occupied	Capacity
A	17	15.02.2018	422,94	USD	747-400	385	373	192.814,29	31	29	21
AA	17	19.03.2018	422,94	USD	747-400	385	158	82.135,06	31	13	21
AA	17	20.04.2018	422,94	USD	747-400	385	129	67.822,72	31	11	21
AA	17	22.05.2018	422,94	USD	747-400	385	60	29.931,49	31	4	21
AA	17	23.06.2018	422,94	USD	747-400	385	86	44.463,75	31	7	21
AZ	789	19.03.2018	1.030,00	EUR	A380-800	475	64	76.889,50	30	4	20
AZ	789	20.04.2018	1.030,00	EUR	A380-800	475	22	25.698,50	30	1	20

Methods

Method Syntax	Description
All methods of the GuiVComponent Object [page 263]: • DumpState • SetFocus • Visualize	
All methods of the GuiContainer Object [page 83]: • FindById	
All methods of the GuiVContainer Object [page 268]: • FindAllByName • FindAllByNameEx • FindByName • FindByNameEx	

Method Syntax	Description
All methods of the GuiShell Object [page 195]: • SelectContextMenuItem • SelectContextMenuItemByPosition • SelectContextMenuItemByText	
ClearSelection `Public Sub ClearSelection()`	Calling clearSelection removes all row, column and cell selections.

Click
`Public Sub Click( _`
`  ByVal Row As Long, _`
`  ByVal Column As String _`
`)`
	This function emulates a mouse click on a given cell if the parameters are valid and raises an exception otherwise.

ClickCurrentCell `Public Sub ClickCurrentCell()`	This function emulates a mouse click on the current cell.
ContextMenu `Public Sub ContextMenu()`	Calling contextMenu emulates the context menu request.
CurrentCellMoved `Public Sub CurrentCellMoved()`	This function notifies the server that a different cell has been made the current cell. It must be called whenever the current cell is changed.

DeleteRows
`Public Sub DeleteRows( _`
`  ByVal Rows As String _`
`)`
	The parameter rows is a comma separated string of indices or index ranges, for example "3,5-8,14,15". The entries must be ordered and not overlap, otherwise an exception is raised.

DeselectColumn
`Public Sub DeselectColumn( _`
`  ByVal Column As String _`
`)`
	This function removes the specified column from the collection of the selected columns.

DoubleClick
`Public Sub DoubleClick( _`
`  ByVal Row As Long, _`
`  ByVal Column As String _`
`)`
	This function emulates a mouse double click on a given cell if the parameters are valid and raises an exception otherwise.

Method	
Syntax	**Description**
DoubleClickCurrentCell `Public Sub DoubleClickCurrentCell()`	This function emulates a mouse double click on the current cell.

DuplicateRows
`Public Sub DuplicateRows( _`
`  ByVal Rows As String _`
`)`
	The parameter rows is a comma separated string of indices or index ranges, for example "3,5-8,14,15". For any single index a copy of the row will be inserted at the given index. If a range of indexes is duplicated then all the new lines are inserted as one block, before the old lines. The entries must be ordered and not overlap, otherwise an exception is raised.
	Example
	0 Value A
	1 Value B
	If rows is "0,1" then the resulting table would be:
	0 Value A
	1 Value A
	2 Value B
	3 Value B
	If on the other hand rows is "0-1" then the resulting table is:
	0 Value A
	1 Value B
	2 Value A
	3 Value B

GetCellChangeable
`Public Function GetCellChangeable( _`
`  ByVal Row As Long, _`
`  ByVal Column As String _`
`) As Byte`
	This function returns True if the specified cell is changeable.

GetCellCheckBoxChecked
`Public Function GetCellCheckBoxChecked( _`
`  ByVal Row As Long, _`
`  ByVal Column As String _`
`) As Byte`
	Returns True if the checkbox at the specified position is checked. Throws an exception if there is no checkbox in the specified cell.

Method	
Syntax	**Description**

GetCellColor
`Public Function GetCellColor( _`
`  ByVal Row As Long, _`
`  ByVal Column As String _`
`) As Long`
	Returns an identifier for the color of the cell. This can be used to retrieve the color information using GetColorInfo.

GetCellHeight
`Public Function GetCellHeight( _`
`  ByVal Row As Long, _`
`  ByVal Column As String _`
`) As Long`
	Returns the height of the cell in pixels.

GetCellIcon
`Public Function GetCellIcon( _`
`  ByVal Row As Long, _`
`  ByVal Column As String _`
`) As String`
	Return the icon string of the cell, if the cell contains an icon. The string has the ABAP icon format '@xy@', where xy is a number or character.

GetCellLeft
`Public Function GetCellLeft( _`
`  ByVal Row As Long, _`
`  ByVal Column As String _`
`) As Long`
	Returns the left position of the cell in client coordinates.

GetCellMaxLength
`Public Function GetCellMaxLength( _`
`  ByVal Row As Long, _`
`  ByVal Column As String _`
`) As Long`
	Returns the maximum length of the cell in number of bytes.

GetCellState
`Public Function GetCellState( _`
`  ByVal Row As Long, _`
`  ByVal Column As String _`
`) As String`
	Returns the state of the cell. Possible values are:
	• Normal
	• Error
	• Warning
	• Info

GetCellTooltip
`Public Function GetCellTooltip( _`
`  ByVal Row As Long, _`
`  ByVal Column As String _`
`) As String`
	Returns the tooltip of the cell.

Method	
Syntax	**Description**

GetCellTop
`Public Function GetCellTop( _`
`  ByVal Row As Long, _`
`  ByVal Column As String _`
`) As Long`
	Returns the top position of the cell in client coordinates.

GetCellType
`Public Function GetCellType( _`
`  ByVal Row As Long, _`
`  ByVal Column As String _`
`) As String`
	This function returns the type of the specified cell. Possible values are:
	• Normal
	• Button
	• Checkbox
	• ValueList
	• RadioButton

GetCellValue
`Public Function GetCellValue( _`
`  ByVal Row As Long, _`
`  ByVal Column As String _`
`) As String`
	Returns the value of the cell as a string.

GetCellWidth
`Public Function GetCellWidth( _`
`  ByVal Row As Long, _`
`  ByVal Column As String _`
`) As Long`
	Returns the width of the cell in pixels.

GetColorInfo
`Public Function GetColorInfo( _`
`  ByVal Color As Long _`
`) As String`
	Returns the description for the color of the cell.

GetColumnDataType
`Public Function GetColumnDataType( _`
`  ByVal Column As String _`
`) As String`
	Returns the data type of the column according to the 'built-in datatypes' of the XML schema standard.

GetColumnPosition
`Public Function GetColumnPosition( _`
`  ByVal Column As String _`
`) As Long`
	Returns the position of the column as shown on the screen, starting from 1.

Method	
Syntax	**Description**

GetColumnSortType
`Public Function GetColumnSortType( _`
`  ByVal Column As String _`
`) As String`
	Returns the sort type of the column. Possible values are:
	• None
	• Ascending
	• Descending

GetColumnTitles
`Public Function GetColumnTitles( _`
`  ByVal Column As String _`
`) As Object`
	This function returns a collection of strings that are used to display the title of a column. The control chooses the appropriate title according to the width of the column.

GetColumnTooltip
`Public Function GetColumnTooltip( _`
`  ByVal Column As String _`
`) As String`
	The tooltip of a column contains a text which is designed to help the user understands the meaning of the column.

GetColumnTotalType
`Public Function GetColumnTotalType( _`
`  ByVal Column As String _`
`) As String`
	Returns the total type of the column. Possible values are:
	• None
	• Total
	• Subtotal

GetDisplayedColumnTitle
`Public Function`
`GetDisplayedColumnTitle( _`
`  ByVal Column As String _`
`) As String`
	This function returns the title of the column that is currently displayed. This text is one of the values of the collection returned from the function “getColumnTitles”.

GetRowTotalLevel
`Public Function GetRowTotalLevel( _`
`  ByVal Row As Long _`
`) As Long`
	Returns the level of the row.

GetSymbolInfo
`Public Function GetSymbolInfo( _`
`  ByVal Symbol As String _`
`) As String`
	Returns the description for the symbol in the cell.

GetToolbarButtonChecked
`Public Function`
`GetToolbarButtonChecked( _`
`  ByVal ButtonPos As Long _`
`) As Byte`
	Returns True if the button is currently checked (pressed).

Method	
Syntax	**Description**
GetToolbarButtonEnabled `Public Function GetToolbarButtonEnabled( _ ByVal ButtonPos As Long _ ) As Byte`	Indicates if the button can be pressed.
GetToolbarButtonIcon `Public Function GetToolbarButtonIcon( _ ByVal ButtonPos As Long _ ) As String`	Returns the name of the icon of the specified toolbar button.
GetToolbarButtonId `Public Function GetToolbarButtonId( _ ByVal ButtonPos As Long _ ) As String`	Returns the ID of the specified toolbar button, as defined in the ABAP data dictionary.
GetToolbarButtonText `Public Function GetToolbarButtonText( _ ByVal ButtonPos As Long _ ) As String`	Returns the text of the specified toolbar button.
GetToolbarButtonTooltip `Public Function GetToolbarButtonTooltip( _ ByVal ButtonPos As Long _ ) As String`	Returns the tooltip of the specified toolbar button.
GetToolbarButtonType `Public Function GetToolbarButtonType( _ ByVal ButtonPos As Long _ ) As String`	Returns the type of the specified toolbar button. Possible values are • Button • ButtonAndMenu • Menu • Separator • Group • CheckBox
GetToolbarFocusButton `Public Function GetToolbarFocusButton() As Long`	Returns the position of the toolbar button that has the focus. If no button in the toolbar has the focus, the method returns -1.

Method	
Syntax	**Description**

HasCellF4Help
`Public Function HasCellF4Help( _`
`  ByVal Row As Long, _`
`  ByVal Column As String _`
`) As Byte`
	Returns True if the cell has a value help.

HistoryCurEntry
`Public Function HistoryCurEntry( _`
`  ByVal Row As Long, _`
`  ByVal Column As String _`
`) As String`
	Returns the text of the presently selected entry of the history list in the specified cell.
	i Note
	• You can only use this method from an external program (like Freedom Scientific JAWS), because the history list is collapsed when a script accesses SAP GUI
	• If an invalid row index or column name is specified, the method raises an exception (**RowIndexOutOfRange** / **WrongColumnName**)
	• This method is available as of SAP GUI for Windows 7.60

HistoryCurIndex
`Public Function HistoryCurIndex( _`
`  ByVal Row As Long, _`
`  ByVal Column As String _`
`) As Long`
	Returns the index (0-based) of the presently selected entry of the history list in the specified cell.
	i Note
	• You can only use this method from an external program (like Freedom Scientific JAWS), because the history list is collapsed when a script accesses SAP GUI
	• If an invalid row index or column name is specified, the method raises an exception (**RowIndexOutOfRange** / **WrongColumnName**)
	• This method is available as of SAP GUI for Windows 7.60

Method	
Syntax	**Description**

HistoryIsActive
`Public Function HistoryIsActive( _`
`  ByVal Row As Long, _`
`  ByVal Column As String _`
`) As Byte`
	This method returns **true** if the input history list is open for the specified cell
	Note
	• You can only use this method from an external program (like Freedom Scientific JAWS), because the history list is collapsed when a script accesses SAP GUI
	• If an invalid row index or column name is specified, the method raises an exception (**RowIndexOutOfRange** / **WrongColumnName**)
	• This method is available as of SAP GUI for Windows 7.60

HistoryList
`Public Function HistoryList( _`
`  ByVal Row As Long, _`
`  ByVal Column As String _`
`) As GuiCollection`
	This method retrieves the list of input history entries of the specified GuiGridView cell as a GuiCollection.
	Note
	• The values of the history list depend on the current value contained in the cell
	• If an invalid row index or column name is specified, the method raises an exception (**RowIndexOutOfRange** / **WrongColumnName**)
	• This method is available as of SAP GUI for Windows 7.60

Method	
Syntax	**Description**

InsertRows
`Public Sub InsertRows( _`
`  ByVal Rows As String _`
`)`
	The parameter rows is a comma separated text of indices or index ranges, for example "3,5-8,14,15". For any single index, a new row will be added at the given index, moving the old row one line down. If a range of indexes is inserted then all the new lines are inserted as one block, before any of the old lines. The entries must be ordered and not overlap, otherwise, an exception is raised.
	Example
	0 Value A
	1 Value B
	If rows is "0,1", then the resulting table would be:
	0
	1 Value A
	2
	3 Value B
	If, on the other hand, rows is "0-1", then the resulting table is:
	0
	1
	2 Value A
	3 Value B

IsCellHotspot
`Public Function IsCellHotspot( _`
`  ByVal Row As Long, _`
`  ByVal Column As String _`
`) As Byte`
	Returns True if the cell is a link.

IsCellSymbol
`Public Function IsCellSymbol( _`
`  ByVal Row As Long, _`
`  ByVal Column As String _`
`) As Byte`
	Returns True if the text in the cell is displayed in the SAP symbol font.

Method	
Syntax	**Description**

IsCellTotalExpander
`Public Function IsCellTotalExpander( _`
`  ByVal Row As Long, _`
`  ByVal Column As String _`
`) As Byte`
	Returns True if the cell contains a total expander button.

IsColumnFiltered
`Public Function IsColumnFiltered( _`
`  ByVal Column As String _`
`) As Byte`
	Returns True if a filter was applied to the column.

IsColumnKey
`Public Function IsColumnKey( _`
`  ByVal Column As String _`
`) As Byte`
	Returns True if the column is marked as a key column.

IsTotalRowExpanded
`Public Function IsTotalRowExpanded( _`
`  ByVal Row As Long _`
`) As Byte`
	Returns true if the row containing an expander is currently expanded.

ModifyCell
`Public Sub ModifyCell( _`
`  ByVal Row As Long, _`
`  ByVal Column As String, _`
`  ByVal Value As String _`
`)`
	If row and column identify a valid editable cell and value has a valid type for this cell, then the value of the cell is changed. Otherwise, an exception is raised.

ModifyCheckBox
`Public Sub ModifyCheckBox( _`
`  ByVal Row As Long, _`
`  ByVal Column As String, _`
`  ByVal Checked As Boolean _`
`)`
	If row and column identify a valid editable cell containing a checkbox, then the value of the cell is changed. Otherwise, an exception is raised.

MoveRows
`Public Sub MoveRows( _`
`  ByVal FromRow As Long, _`
`  ByVal ToRow As Long, _`
`  ByVal DestRow As Long _`
`)`
	The rows with an index greater than or equal to fromRow up to an index less than or equal to toRow are moved to the position of destRow.
	Passing invalid index values as parameters raises an exception.

Method

Syntax	Description

PressButton
`Public Sub PressButton( _`
`  ByVal Row As Long, _`
`  ByVal Column As String _`
`)`
	This function emulates pressing a button placed in a given cell. It will raise an exception if the cell does not contain a button, or does not even exist.

PressButtonCurrentCel `Public Sub PressButtonCurrentCell()`	This function emulates pressing a button placed in the current cell. It will raise an exception if the cell does not contain a button.

PressColumnHeader
`Public Sub PressColumnHeader( _`
`  ByVal Column As String _`
`)`
	This function emulates a mouse click on the header of the column if the parameter identifies a valid column and raises an exception otherwise.

PressEnter `Public Sub PressEnter()`	This emulates pressing the Enter key.
PressF1 `Public Sub PressF1()`	This emulates pressing the F1 key while the focus is on the grid view.
PressF4 `Public Sub PressF4()`	This emulates pressing the F4 key.

PressToolbarButton
`Public Sub PressToolbarButton( _`
`  ByVal Id As String _`
`)`
	This function emulates clicking a button in the grid view's toolbar.

PressToolbarContextButton
`Public Sub`
`PressToolbarContextButton( _`
`  ByVal Id As String _`
`)`
	This emulates opening the context menu of the grid view's toolbar.

PressTotalRow
`Public Sub PressTotalRow( _`
`  ByVal Row As Long, _`
`  ByVal Column As String _`
`)`
	Pressing the total row button expands or condenses the grouped rows. If the selected cell is not a total row cell an exception is raised.

Method	
Syntax	**Description**
PressTotalRowCurrentCell `Public Sub PressTotalRowCurrentCell()`	This function differs from pressTotalRow only in that it tries to press the expansion button on the current cell.
SelectAll `Public Sub SelectAll()`	This function selects the whole grid content (i.e. all rows and all columns).

SelectColumn
`Public Sub SelectColumn( _`
`  ByVal Column As String _`
`)`
	This function adds the specified column to the collection of the selected columns.

SelectionChanged `Public Sub SelectionChanged()`	This function notifies the server that the selection has changed.

SelectToolbarMenuItem
`Public Sub SelectToolbarMenuItem( _`
`  ByVal Id As String _`
`)`
	This function emulates the selection of an item from the context menu of the grid view's toolbar. The parameter should be the function code of the item.

SetColumnWidth
`Public Sub SetColumnWidth( _`
`  ByVal Column As String, _`
`  ByVal Width As Long _`
`)`
	The width of a column can be set using this function. The width is given in characters. For proportional fonts this refers to the width of an average character. Depending on the contents of the cell more or less characters may fit in the column. If the parameter is invalid an exception is raised.

SetCurrentCell
`Public Sub SetCurrentCell( _`
`  ByVal Row As Long, _`
`  ByVal Column As String _`
`)`
	If row and column identify a valid cell, this cell becomes the current cell. Otherwise, an exception is raised.

TriggerModified `Public Sub TriggerModified()`	Notifies the server of multiple changes in cells. Typically this method should be called after multiple calls to ModifyCell.

Properties

Property Syntax	Description
All properties of the GuiComponent Object [page 78]: • ContainerType • Id • Name • Parent • Type • TypeAsNumber	
All properties of the GuiVComponent Object [page 263]: • AccLabelCollection • AccText • AccTextOnRequest • AccTooltip • Changeable • DefaultTooltip • Height • IconName • IsSymbolFont • Left • Modified • ParentFrame • ScreenLeft • ScreenTop • Text • Tooltip • Top • Width	
All properties of the GuiContainer Object [page 83]: • Children	
All additional properties of the GuiShell Object [page 195]: • AccDescription • CurrentContextMenu • DragDropSupported • Handle • OcxEvents • SubType	

Property

Syntax	Description
ColumnCount (Read-only) `Public Property ColumnCount As Long`	This property represents the number of columns in the control.
ColumnOrder (Read-write) `Public Property ColumnOrder As Object`	This collection contains all the column identifiers in the order in which they are currently displayed. Passing an invalid column identifier to this property will raise an exception.
CurrentCellColumn (Read-write) `Public Property CurrentCellColumn As String`	The string identifying a column is the field name defined in the SAP data dictionary. In the example above the identifiers are named CARRID, CONNID, FLDATE, PRICE etc.
CurrentCellRow (Read-write) `Public Property CurrentCellRow As Long`	The row index of the current cell ranges from 0 to the number of rows less 1, with -1 being the index of the title row.
FirstVisibleColumn (Read-write) `Public Property FirstVisibleColumn As String`	This property represents the first visible column of the scrollable area of the grid view. Fixed columns are ignored. Setting the property to an invalid column identifier will raise an exception.
FirstVisibleRow (Read-write) `Public Property FirstVisibleRow As Long`	This is the index of the first visible row in the grid. Setting this property to an invalid row index will raise an exception.
FrozenColumnCount (Read-only) `Public Property FrozenColumnCount As Long`	This property represents the number of columns that are excluded from horizontal scrolling.
RowCount (Read-only) `Public Property RowCount As Long`	This property represents the number of rows in the control.
SelectedCells (Read-write) `Public Property SelectedCells As Object`	The collection of selected cells contains strings, each of which has the format "<index of the row>,<column identifier>", such as "0,CARRID". Trying to set this property to an invalid value will raise an exception.
SelectedColumns (Read-write) `Public Property SelectedColumns As Object`	The selected columns are available as a collection of strings like the currentCellColumn string. Setting this property can raise an exception, if the new collection contains an invalid column identifier.

Property

Syntax	Description
SelectedRows (Read-write) `Public Property SelectedRows As String`	The string is a comma separated list of row index numbers or index ranges, such as "1,2,4-8,10".Setting this property to an invalid string or a string containing invalid row indices will raise an exception.
SelectionMode (Read-only) `Public Property SelectionMode As String`	Possible values are • RowsAndColumns: Only rows and columns can be selected. Individual rectangular areas of cells are not allowed. • ListboxSingle: Only one single row can be selected. • ListboxMultiple: One or more rows can be selected. • Free: Any kind of selection can be made.
Title (Read-only) `Public Property Title As String`	This property represents title of the grid control.
ToolbarButtonCount (Read-only) `Public Property ToolbarButtonCount As Long`	The number of toolbar buttons including separators.
VisibleRowCount (Read-only) `Public Property VisibleRowCount As Long`	Retrieves the number of visible rows of the grid.

1.2.31 GuiHTMLViewer Object

Description

The GuiHTMLViewer is used to display an HTML document inside SAP GUI. GuiHTMLViewer extends the GuiShell Object [page 195].

Methods

Method	
Syntax	**Description**
All methods of the GuiVComponent Object [page 263]: • DumpState • SetFocus • Visualize	
All methods of the GuiContainer Object [page 83]: • FindById	
All methods of the GuiVContainer Object [page 268]: • FindAllByName • FindAllByNameEx • FindByName • FindByNameEx	
All methods of the GuiShell Object [page 195]: • SelectContextMenuItem • SelectContextMenuItemByPosition • SelectContextMenuItemByText	
ContextMenu `Public Sub ContextMenu()`	Calling contextMenu emulates the context menu request. Note that this function applies only to context menus provided by the backend, not to the local context menu, which is generated by the HTML control.

Method

Syntax	Description

SapEvent
`Public Sub SapEvent( _`
`  ByVal FrameName As String, _`
`  ByVal PostData As String, _`
`  ByVal Url As String _`
`)`
	This function submits an HTML form to the backend.
	Remarks
	If the form is to be submitted using the GET method, the data is appended to the event name in the usual http URL fashion, for example:
	Sample Code
	`sapEvent("Frame1","","sapevent:SUBMIT_FORM_AS_GET_METHOD?FirstName=John&LastName=Smith");`
	In this case, postData is always an empty string.
	If the form is to be submitted using the POST method, the data is transported in the postData parameter:
	Sample Code
	`sapEvent("Frame1","FirstName=John&LastName=Smith","sapevent:SUBMIT_FORM_AS_POST_METHOD");`

FrameName	This is the name of the frame in which the HTML form that has been submitted lives.
PostData	Contains the form data when a submit is made using the POST method.
Url	This is the URL, which is submitted to the backend. The protocol name for the URL string is “sapevent:”. This is followed by the name of the event as defined in the Action Property of the HTML form, which is called.

Properties

Property Syntax	Description
All properties of the GuiComponent Object [page 78]: • ContainerType • Id • Name • Parent • Type • TypeAsNumber	
All properties of the GuiVComponent Object [page 263]: • AccLabelCollection • AccText • AccTextOnRequest • AccTooltip • Changeable • DefaultTooltip • Height • IconName • IsSymbolFont • Left • Modified • ParentFrame • ScreenLeft • ScreenTop • Text • Tooltip • Top • Width	
All properties of the GuiContainer Object [page 83]: • Children	
All additional properties of the GuiShell Object [page 195]: • AccDescription • CurrentContextMenu • DragDropSupported • Handle • OcxEvents • SubType	

Property

Syntax	Description
BrowserHandle (Read-only) `Public Property BrowserHandle As Object`	
DocumentComplete (Read-only) `Public Property DocumentComplete As Long`	

1.2.32 GuiInputFieldControl Object

Methods

Method

Syntax	Description
All methods of the GuiVComponent Object [page 263]: • DumpState • SetFocus • Visualize	
All methods of the GuiContainer Object [page 83]: • FindById	
All methods of the GuiVContainer Object [page 268]: • FindAllByName • FindAllByNameEx • FindByName • FindByNameEx	
All methods of the GuiShell Object [page 195]: • SelectContextMenuItem • SelectContextMenuItemByPosition • SelectContextMenuItemByText	

Method

Syntax	Description
Submit `Public Sub Submit()`	Submits the input to the application.

Properties

Property

Syntax	Description
All properties of the GuiComponent Object [page 78]: • ContainerType • Id • Name • Parent • Type • TypeAsNumber	
All properties of the GuiVComponent Object [page 263]: • AccLabelCollection • AccText • AccTextOnRequest • AccTooltip • Changeable • DefaultTooltip • Height • IconName • IsSymbolFont • Left • Modified • ParentFrame • ScreenLeft • ScreenTop • Text • Tooltip • Top • Width	

Property Syntax	Description
All properties of the GuiContainer Object [page 83]: • Children	
All properties of the GuiShell Object [page 195]: • AccDescription • CurrentContextMenu • DragDropSupported • Handle • OcxEvents • SubType	
ButtonTooltip (Read-only) `Public Property ButtonTooltip As String`	Tooltip of the *submit / find* button.
FindButtonActivated (Read-only) `Public Property FindButtonActivated As Boolean`	This property is True when the *Find* button is active
HistoryOpened (Read-only) `Public Property HistoryOpened As Byte`	This property is True when the input history is opened.
LabelText (Read-only) `Public Property LabelText As String`	The text of the label belonging to the input field.
Text (Read-write) `Public Property Text As String`	Text content of the input field itself.

1.2.33 GuiLabel Object

Description

GuiLabel extends the GuiVComponent Object [page 263]. The type prefix is lbl, the name is the fieldname taken from the SAP data dictionary.

Methods

Method	
Syntax	**Description**
All methods of the GuiVComponent Object [page 263]: • DumpState • SetFocus • Visualize	

Method	
Syntax	**Description**

GetListProperty
`Public Function GetListProperty( _`
`  ByVal Property As String _`
`) As String`
	Remarks
	Attributes of containers in general
	• ContainerType
	  ○ L: Entire list
	  ○ T: A table
	  ○ G: A group inside a table
	  ○ S: A subgroup (inside a group)
	  ○ R: A line in the body of a table
	  ○ B: A text box
	  ○ E: A tree
	  ○ F: Simple "free" text outside any box
	• ContainerTitle: Title of a table (if provided) or of a text box
	• ContainerInputFields: Number of input fields, used if a table or a tree has input fields (incl. checkboxes)
	Attributes of containers of type L (Entire list)
	• ListTablesTotal: Number of tables on the list
	• ListTextBoxesTotal: Number of text boxes on the list
	• ListTreesTotal: Number of trees on the list
	• ListErrorMessage: Used iff the structure recognition detected a (severe) error.
	• ListInputType
	  ○ N: list contains no input fields
	  ○ C: list contains check boxes
	  ○ E: list contains edit fields
	  ○ A: list contains edit fields and check boxes
	Attributes of containers of type T (Table), G (Group) and S (Subgroup)
	• RowsTotal: Number of logical rows in the table body. If this is an attribute of a (sub-)group, number of logical rows until the next (sub-)group starts. The numbers do NOT include summation lines and inserted lines.
	• RowsSummation: Number of rows with color COL_SUMMING INTENSIFIED ON (if there are any).
	• RowsSubSummation: Number of rows with color COL_SUMMING INTENSIFIED OFF (if there are any).
	• RowsInserted: Number of inserted rows (if there are any).
	Attributes of containers of type T (Table)
	• TableNo: Number of the table if there is more than one table on the list
	• ColumnsTotal: Number of logical columns

Method Syntax	Description

	• SuperColumnsTotal: Used iff the table has a hierarchical header
	• TableHierarchical: Used iff the table is hierarchical-sequential
	  ○ A: ALV-like 2-level hierarchical-seq.
	  ○ 2: 2-level hierarchical-seq.
	  ○ 3: 3-level hierarchical-seq.
	• TableGroupsTotal: Used iff the table is hierarchical-seq.: Number of groups (not counting subgroups)
	• Columns2LevelALV: Used iff TableHierarchical is "A": Number of columns in the group header
	• HeaderRows2LevelALV: Used iff TableHierarchical is "A": Number of lines in the group header
	• TableHierarchicalHeader: Used iff the table has a hierarchical header
	• TableMultipleRows: Used iff the table is a multiple-line table: Number of physical lines per logical line
	
	Attributes of containers of type G (Group) and S (Subgroup)
	
	• GroupNo: Number of current group if container is of type G.
	• SubGroupNo: Number of current subgroup if container is of type S
	• SubgroupsTotal: Number of subgroups if table is 3-level hierarchical-sequential and container is of type G
	• GroupHeaderRows: Number of physical lines in the group header
	• GroupHeaderValues: Number of label-value pairs in the group header if the table is 2- or 3- level hierarchical-sequential
	
	Attributes of containers of type R (Row)
	
	• RowType: Used iff the row has a special type
	  ○ S: Color COL_SUMMING INTENSIFIED ON
	  ○ U: Color COL_SUMMING INTENSIFIED OFF
	  ○ I: Inserted line
	• RowNo: Number of current (logical) row, relative to the beginning of the (sub-)group if the table is 3- or 2-level hierarchical-seq.
	• RowMultipleRows: Number of physical lines for current logical line; used iff > 1 (multiple-line tables). Lines with totals may or may not be multiple lines
	• RowInputFields: Number of input fields in the current line (if any)
	
	Attributes of fields in tables
	
	• FieldHeader: The text of the column header (unavailable if the field itself is in the header, or the field is the label

Method	
Syntax	**Description**

	of a label-value pair in a hierarchical-sequential table, or the field is in an inserted line and does not belong to any column).
	• FieldSuperHeader: Text of the supercolumn if the field is in the lower line of a hierarchical header or in the table body (and belongs to a column).
	• ColumnNo: The number of the logical column (if the field belongs to a column).
	• LabelType: Used iff the field is in the header or is a label of a label-value pair in a hierarchical-seq. table.
	  ○ N: normal header field (lowest level in hierar.-seq. tables)
	  ○ H: header field in a supercolumn (upper line of a hierarchical header)
	  ○ A: group header field (COL_GROUPING INTENSIFIED ON) in table whereTableHierarchical is A
	  ○ G: group header field (COL_GROUPING INTENSIFIED ON) in 2-level hierarchical-sequential table
	  ○ S: subgroup header field (COL_HEADING INTENSIFIED ON) in 3-level hierarchical-sequential table
	  ○ T: title-field COL_NORMAL INTENSIFIED ON
	• ColumnType: Used if the field is a column header of a special column.
	  ○ C: column contains checkboxes
	  ○ S: column contains symbols and/or icons
	• SubordinateColumns: Number of subordinate columns if the field is in the upper line of a hierarchical header.
	• FieldMultipleRows: Used iff the field is in a table header and word wrapping was done: Number of physical lines of the "logical field".
	• FieldWithEllipsis: The field is directly followed by SYM_ELLIPSIS, i.e. "...".
	
	Attributes of fields in tables, trees or title lines of text boxes
	• FieldPhysRowNo: If we are in the body of a multiple-line table or in a multiple-line node of a tree or word wrapping is used in the table header: Current physical line number within the logical line.
	
	Attributes of text boxes
	• TextBoxNo: Number of the text box if there is more than one text box on the list
	
	Attributes of containers in an SEUT tree
	• TreeNo: Number of the tree if there is more than one tree on the list and the current container is the root node
	• NodeName: Text of the first field of the node (STREENODE-NAME)

Method

Syntax	Description
	• NodeLevelNo: The current level number; the root has level 0 • NodeNo: The current node number; the "oldest brother" has number 1. • NodeExpandable: Used iff the current node can be expanded (folder with "+"). • NodeMarked: Node has been marked (yellow in SEUT). • NodeChildrenTotal: Used iff the current node is expanded (folder with "-"): Number of children. (Grand children are not counted.) • NodeMultipleRows: Used iff the current node has more than one physical line: Number of physical lines
GetListPropertyNonRec `Public Function GetListPropertyNonRec( _ ByVal Property As String _ ) As String`	This method returns information that is compiled on the server to enhance the ABAP lists with accessibility information. See GuiLabel::GetListProperty for a description of available attributes. In contrast to the method GetListProperty, GetListPropertyNonRec will only return information that is set for the specific element, and ignore list properties set for parent elements.

Properties

Property

Syntax	Description
All properties of the GuiComponent Object [page 78]: • ContainerType • Id • Name • Parent • Type • TypeAsNumber	

Property

Syntax	Description
All properties of the GuiVComponent Object [page 263]: • AccLabelCollection • AccText • AccTextOnRequest • AccTooltip • Changeable • DefaultTooltip • Height • IconName • IsSymbolFont • Left • Modified • ParentFrame • ScreenLeft • ScreenTop • Text • Tooltip • Top • Width	
All properties of the GuiBox Object [page 51]: • CharHeight • CharLeft • CharTop • CharWidth	
CaretPosition (Read-write) `Public Property CaretPosition As Long`	Setting the caret position within a label is possible even though it is not visualized as a caret by SAP GUI. However, the position is transmitted to the server, so ABAP application logic may depend on this position.
ColorIndex (Read-only) `Public Property ColorIndex As Long`	This number defines the index of the list color of this element.
ColorIntensified (Read-only) `Public Property ColorIntensified As Byte`	This property is True if the Intensified flag is set in screen painter for this dynpro element.
ColorInverse (Read-only) `Public Property ColorInverse As Byte`	This property is True if the inverse color style is set in screen painter for the element.

Property

Syntax	Description
DisplayedText (Read-only) `Public Property DisplayedText As String`	This property contains the text as it is displayed on the screen, including preceding or trailing blanks. These blanks are stripped from the text property.
Highlighted (Read-only) `Public Property Highlighted As Byte`	This property is True if the Highlighted flag is set in the screen painter for the dynpro element.
IsHotspot (Read-only) `Public Property IsHotspot As Byte`	Dynpro elements such as labels may be configured to cause a round trip when they are clicked. In that case the mouse cursor changes to the hand shape. This is called a hot spot.
IsLeftLabel (Read-only) `Public Property IsLeftLabel As Byte`	This property is set if the label has been assigned as the left label of another control.
IsListElement (Read-only) `Public Property IsListElement As Byte`	This property is True if the element is on an ABAP list, not a dynpro screen.
IsRightLabel (Read-only) `Public Property IsRightLabel As Byte`	This property is set if the label has been assigned as the right label of another control.
MaxLength (Read-only) `Public Property MaxLength As Long`	The maximum text length of a label is counted in code units. On non-Unicode clients these are equivalent to bytes.
Numerical (Read-only) `Public Property Numerical As Byte`	This flag is True if the label may only contain numbers.
RowText (Read-only) `Public Property RowText As String`	This property is only available in ABAP list screens. It returns the text of the while line containing the current component.

1.2.34 GuiMainWindow Object

Description

This window represents the main window of an SAP GUI session.

GuiMainWindow extends the GuiFrameWindow Object [page 101].

Methods

Method	
Syntax	**Description**
All methods of the GuiVComponent Object [page 263]: • DumpState • SetFocus • Visualize	
All methods of the GuiContainer Object [page 83]: • FindById	
All additional methods of the GuiVContainer Object [page 268]: • FindAllByName • FindAllByNameEx • FindByName • FindByNameEx	
All additional methods of the GuiFrameWindow Object [page 101]: • Close • CompBitmap • HardCopy • HardCopyToMemory • Iconify • IsVKeyAllowed • JumpBackward • JumpForward • Maximize • Restore • SendVKey • ShowMessageBox • TabBackward • TabForward	

Method

Syntax	Description

ResizeWorkingPane
`Public Sub ResizeWorkingPane( _`
`  ByVal Width As Long, _`
`  ByVal Height As Long, _`
`  ByVal ThrowOnFail As Boolean _`
`)`
	The ResizeWorkingPane function will resize the window so that the available working area has the given width and height in character metric.
	ThrowOnFail: The throwOnFail parameter has been added for use in the SAP GUI for Java because some window managers may not support a program driven resize of a window.

ResizeWorkingPaneEx
`Public Sub ResizeWorkingPaneEx( _`
`  ByVal Width As Long, _`
`  ByVal Height As Long, _`
`  ByVal ThrowOnFail As Boolean _`
`)`
	The ResizeWorkingPaneEx function will resize the window so that the available working area has the given width and height in pixels.
	Remarks
	This method is only used during recording if the DWORD registry key ResizeWorkingPaneEx in patch HKCU\Software\SAP\SAPGUI Front\SAP Frontend Server\Scripting exists and has the value 1.

Table GUI_FKEY

Refer to Table GUI_FKEY [page 145]

Properties

Property

Syntax	Description
All properties of the GuiComponent Object [page 78]: • ContainerType • Id • Name • Parent • Type • TypeAsNumber	

Property

Syntax	Description
All properties of the GuiVComponent Object [page 263]: • AccLabelCollection • AccText • AccTextOnRequest • AccTooltip • Changeable • DefaultTooltip • Height • IconName • IsSymbolFont • Left • Modified • ParentFrame • ScreenLeft • ScreenTop • Text • Tooltip • Top • Width	
All properties of the GuiContainer Object [page 83]: • Children	
All properties of the GuiFrameWindow Object [page 101]: • ElementVisualizationMode • GuiFocus • Handle • Iconic • SystemFocus • WorkingPaneHeight • WorkingPaneWidth	
ButtonbarVisible (Read-write) `Public Property ButtonbarVisible As Byte`	This property it True if the application toolbar, the lower toolbar within SAP GUI, is visible. Setting this property to False will hide the application toolbar.
StatusbarVisible (Read-write) `Public Property StatusbarVisible As Byte`	This property it True if the status bar at the bottom of the SAP GUI window is visible. Setting this property to False will hide the status bar. When the status bar is hidden, messages will be displayed in a popup instead.

Property

Syntax	Description
TitlebarVisible (Read-write) `Public Property TitlebarVisible As Byte`	This property it True if the title bar is visible. Setting this property to False will hide the title bar. **Remarks** The title bar is only available in New Visual Design, not in Classic Design.
ToolbarVisible (Read-write) `Public Property ToolbarVisible As Byte`	This property it True if the system toolbar, the upper toolbar within SAP GUI, is visible. Setting this property to False will hide the system toolbar.

1.2.34.1 Table GUI_FKEY

VKey	Keyboard Combination
00	Enter
01	F1
02	F2
03	F3
04	F4
05	F5
06	F6
07	F7
08	F8
09	F9
10	F10
11	Ctrl+S
12	F12
13	Shift+F1
14	Shift+F2
15	Shift+F3
16	Shift+F4
17	Shift+F5
18	Shift+F6
19	Shift+F7

VKey	Keyboard Combination
20	Shift+F8
21	Shift+F9
22	Shift+Ctrl+0
23	Shift+F11
24	Shift+F12
25	Ctrl+F1
26	Ctrl+F2
27	Ctrl+F3
28	Ctrl+F4
29	Ctrl+F5
30	Ctrl+F6
31	Ctrl+F7
32	Ctrl+F8
33	Ctrl+F9
34	Ctrl+F10
35	Ctrl+F11
36	Ctrl+F12
37	Ctrl+Shift+F1
38	Ctrl+Shift+F2
39	Ctrl+Shift+F3
40	Ctrl+Shift+F4
41	Ctrl+Shift+F5
42	Ctrl+Shift+F6
43	Ctrl+Shift+F7
44	Ctrl+Shift+F8
45	Ctrl+Shift+F9
46	Ctrl+Shift+F10
47	Ctrl+Shift+F11
48	Ctrl+Shift+F12
70	Ctrl+E
71	Ctrl+F
72	Ctrl+/
73	Ctrl+\
74	Ctrl+N

VKey	Keyboard Combination
75	Ctrl+O
76	Ctrl+X
77	Ctrl+C
78	Ctrl+V
79	Ctrl+Z
80	Ctrl+PageUp
81	PageUp
82	PageDown
83	Ctrl+PageDown
84	Ctrl+G
85	Ctrl+R
86	Ctrl+P

1.2.35 GuiMap Object

Description

For the map control only basic members from GuiShell are available. Recording and playback is not possible.

Methods

Method Syntax	Description
All methods of the GuiVComponent Object [page 263]: • DumpState • SetFocus • Visualize	
All methods of the GuiContainer Object [page 83]: • FindById	

Method

Syntax	Description
All methods of the GuiVContainer Object [page 268]: • FindAllByName • FindAllByNameEx • FindByName • FindByNameEx	
All additional methods of the GuiShell Object [page 195]: • SelectContextMenuItem • SelectContextMenuItemByPosition • SelectContextMenuItemByText	

Properties

Property

Syntax	Description
All properties of the GuiComponent Object [page 78]: • ContainerType • Id • Name • Parent • Type • TypeAsNumber	

Property

Syntax	Description
All properties of the GuiVComponent Object [page 263]: • AccLabelCollection • AccText • AccTextOnRequest • AccTooltip • Changeable • DefaultTooltip • Height • IconName • IsSymbolFont • Left • Modified • ParentFrame • ScreenLeft • ScreenTop • Text • Tooltip • Top • Width	
All properties of the GuiContainer Object [page 83]: • Children	
All additional properties of the GuiShell Object [page 195]: • AccDescription • CurrentContextMenu • DragDropSupported • Handle • OcxEvents • SubType	

1.2.36 GuiMenu Object

Description

A GuiMenu may have other GuiMenu objects as children. GuiMenu extends the GuiVContainer Object [page 268]. The type prefix is menu, the name is the text of the menu item. If the item does not have a text, which is the case for separators, then the name is the last part of the id, menu[n].

Methods

Method Syntax	Description
All methods of the GuiVComponent Object [page 263]: • DumpState • SetFocus • Visualize	
All methods of the GuiContainer Object [page 83]: • FindById	
All methods of the GuiVContainer Object [page 268]: • FindAllByName • FindAllByNameEx • FindByName • FindByNameEx	
Select `Public Sub Select()`	Select the menu.

Properties

Property Syntax	Description
All properties of the GuiComponent Object [page 78]: • ContainerType • Id • Name • Parent • Type • TypeAsNumber	
All properties of the GuiVComponent Object [page 263]: • AccLabelCollection • AccText • AccTextOnRequest • AccTooltip • Changeable • DefaultTooltip • Height • IconName • IsSymbolFont • Left • Modified • ParentFrame • ScreenLeft • ScreenTop • Text • Tooltip • Top • Width	
All properties of the GuiContainer Object [page 83]: • Children	

1.2.37 GuiMenubar Object

Description

Only the main window has a menubar. The children of the menubar are menus. GuiMenubar extends the GuiVContainer Object [page 268]. The type prefix and name are mbar.

Methods

Method	
Syntax	**Description**
All methods of the GuiVComponent Object [page 263]: • DumpState • SetFocus • Visualize	
All methods of the GuiContainer Object [page 83]: • FindById	
All methods of the GuiVContainer Object [page 268]: • FindAllByName • FindAllByNameEx • FindByName • FindByNameEx	

Properties

Property	
Syntax	**Description**
All properties of the GuiComponent Object [page 78]: • ContainerType • Id • Name • Parent • Type • TypeAsNumber	

Property

Syntax	Description
All properties of the GuiVComponent Object [page 263]: • AccLabelCollection • AccText • AccTextOnRequest • AccTooltip • Changeable • DefaultTooltip • Height • IconName • IsSymbolFont • Left • Modified • ParentFrame • ScreenLeft • ScreenTop • Text • Tooltip • Top • Width	
All properties of the GuiContainer Object [page 83]: • Children	

1.2.38 GuiMessageWindow

Methods

Method

Syntax	Description
All methods of the GuiVComponent Object [page 263]: • DumpState • SetFocus • Visualize	

Properties

Property Syntax	Description
All properties of the GuiComponent Object [page 78]: • ContainerType • Id • Name • Parent • Type • TypeAsNumber	
All properties of the GuiVComponent Object [page 263]: • AccLabelCollection • AccText • AccTextOnRequest • AccTooltip • Changeable • DefaultTooltip • Height • IconName • IsSymbolFont • Left • Modified • ParentFrame • ScreenLeft • ScreenTop • Text • Tooltip • Top • Width	
FocusedButton (Read-only) `Public Property FocusedButton As Long`	
HelpButtonHelpText (Read-write) `Public Property HelpButtonHelpText As String`	
HelpButtonText (Read-write) `Public Property HelpButtonText As String`	

Property

Syntax	Description
MessageText (Read-write) `Public Property MessageText As String`	
MessageType (Read-write) `Public Property MessageType As Long`	
OKButtonHelpText (Read-write) `Public Property OKButtonHelpText As String`	
OKButtonText (Read-write) `Public Property OKButtonText As String`	
Visible (Read-write) `Public Property Visible As Byte`	

1.2.39 GuiModalWindow Object

Description

A GuiModalWindow is a dialog pop-up.

GuiModalWindow extends the GuiFrameWindow Object [page 101].

Methods

Method	
Syntax	**Description**
All methods of the GuiVComponent Object [page 263]: • DumpState • SetFocus • Visualize	
All methods of the GuiContainer Object [page 83]: • FindById	
All methods of the GuiVContainer Object [page 268]: • FindAllByName • FindAllByNameEx • FindByName • FindByNameEx	
All methods of the GuiFrameWindow Object [page 101]: • Close • CompBitmap • HardCopy • HardCopyToMemory • Iconify • IsVKeyAllowed • JumpBackward • JumpForward • Maximize • Restore • SendVKey • ShowMessageBox • TabBackward • TabForward	

Properties

Property	
Syntax	**Description**
All properties of the GuiComponent Object [page 78]: • ContainerType • Id • Name • Parent • Type • TypeAsNumber	
All properties of the GuiVComponent Object [page 263]: • AccLabelCollection • AccText • AccTextOnRequest • AccTooltip • Changeable • DefaultTooltip • Height • IconName • IsSymbolFont • Left • Modified • ParentFrame • ScreenLeft • ScreenTop • Text • Tooltip • Top • Width	
All properties of the GuiContainer Object [page 83]: • Children	
All properties of the GuiFrameWindow Object [page 101]: • ElementVisualizationMode • GuiFocus • Handle • Iconic • SystemFocus • WorkingPaneHeight • WorkingPaneWidth	

Property

Syntax	Description
IsPopupDialog (Read-write) `Public Property IsPopupDialog() As Boolean`	Some modal windows represent popup dialogs. In this case the IsPopupDialog property is True. Popup dialogs are identified by checking the ABAP source name and dynpro number. Currently the following are supported: • SAPLSPO1 / 500 (Function module Popup_To_Confirm)
PopupDialogText (Read-write) `Public Property PopupDialogText As String`	The text of the input fields of the popup dialog in a concatenated form.

1.2.40 GuiNetChart Object

Description

The GuiNetChart is a powerful tool to display and modify entity relationship diagrams. It is of a very technical nature and should only be used for recording and playback, as most of the parameters cannot be determined in any other way.

Methods

Method

Syntax	Description
All methods of the GuiVComponent Object [page 263]: • DumpState • SetFocus • Visualize	
All methods of the GuiContainer Object [page 83]: • FindById	

Method Syntax	Description
All methods of the GuiVContainer Object [page 268]: • FindAllByName • FindAllByNameEx • FindByName • FindByNameEx	
All methods of the GuiShell Object [page 195]: • SelectContextMenuItem • SelectContextMenuItemByPosition • SelectContextMenuItemByText	

GetLinkContent
`Public Function GetLinkContent( _`
`  ByVal linkId As Long, _`
`  ByVal textId As Long _`
`) As String`
	Returns the content of the link.
	linkId: Index of the link
	textId: Internal value, do be determined during recording.

GetNodeContent
`Public Function GetNodeContent( _`
`  ByVal nodeId As Long, _`
`  ByVal textId As Long _`
`) As String`
	Returns the content of the node.
	nodeId: Index of the node.
	textId: Internal value, do be determined during recording.

SendData
`Public Sub SendData( _`
`  ByVal Data As String _`
`)`
	This function emulates the output of each action triggered at the control side. The result of the action is sent to the server.
	It's currently not possible to select – deselect single objects at the client-side and to replay/script these "local" actions.

Properties

Property Syntax	Description
All properties of the GuiComponent Object [page 78]: • ContainerType • Id • Name • Parent • Type • TypeAsNumber	
All properties of the GuiVComponent Object [page 263]: • AccLabelCollection • AccText • AccTextOnRequest • AccTooltip • Changeable • DefaultTooltip • Height • IconName • IsSymbolFont • Left • Modified • ParentFrame • ScreenLeft • ScreenTop • Text • Tooltip • Top • Width	
All properties of the GuiContainer Object [page 83]: • Children	
All additional properties of the GuiShell Object [page 195]: • AccDescription • CurrentContextMenu • DragDropSupported • Handle • OcxEvents • SubType	

Property

Syntax	Description
LinkCount (Read-only) `Public Property LinkCount As Long`	Number of links in the net.
NodeCount (Read-only) `Public Property NodeCount As Long`	Number of Nodes in the net.

1.2.41 GuiOfficeIntegration Object

Methods

Method

Syntax	Description
All methods of the GuiVComponent Object [page 263]: • DumpState • SetFocus • Visualize	
All methods of the GuiContainer Object [page 83]: • FindById	
All additional methods of the GuiVContainer Object [page 268]: • FindAllByName • FindAllByNameEx • FindByName • FindByNameEx	
All additional methods of the GuiShell Object [page 195]: • SelectContextMenuItem • SelectContextMenuItemByPosition • SelectContextMenuItemByText	

Method	
Syntax	**Description**

AppendRow
`Public Sub AppendRow( _`
`  ByVal Name As String, _`
`  ByVal Row As`
`  String _`
`)`
	This function appends a new row to a table specified by the parameter name in the table collection. The parameter row is the base64 representation of the binary row.

CloseDocument
`Public Sub CloseDocument( _`
`  ByVal Cookie As Long, _`
`  ByVal EverChanged As Byte, _`
`  ByVal ChangedAfterSave As`
`  Byte _`
`)`
	This function sends the close event of the document specified by the parameter cookie to the server.

CustomEvent
`Public Sub CustomEvent( _`
`  ByVal Cookie As Long, _`
`  ByVal EventName As String, _`
`  ByVal ParamCount As Long, _`
`  Optional ByVal Par1 As Variant, _`
`  Optional ByVal Par2 As Variant, _`
`  Optional ByVal Par3 As Variant, _`
`  Optional ByVal Par4 As Variant, _`
`  Optional ByVal Par5 As Variant, _`
`  Optional ByVal Par6 As Variant, _`
`  Optional ByVal Par7 As Variant, _`
`  Optional ByVal Par8 As Variant, _`
`  Optional ByVal Par9 As Variant, _`
`  Optional ByVal Par10 As Variant, _`
`  Optional ByVal Par11 As Variant, _`
`  Optional ByVal Par12 As Variant _`
`)`
	This function sends the custom event eventName to the server. The document specified by the parameter cookie is the source.

RemoveContent
`Public Sub RemoveContent( _`
`  ByVal Name As String _`
`)`
	This function removes the content of a table in the table collection. The parameter name is the name of the table.

SaveDocument
`Public Sub SaveDocument( _`
`  ByVal Cookie As Long, _`
`  ByVal Changed As Byte _`
`)`
	This function sends the save event of the document specified by the parameter cookie to the server.

Method

Syntax	Description

SetDocument
`Public Sub SetDocument( _`
`  ByVal Index As Long, _`
`  ByVal Document As String _`
`)`
	This function replaces or adds a new document with the specified index. The parameter document is the base64-representation of the binary document.

Properties

Property

Syntax	Description
All properties of the GuiComponent Object [page 78]: • ContainerType • Id • Name • Parent • Type • TypeAsNumber	
All properties of the GuiVComponent Object [page 263]: • AccLabelCollection • AccText • AccTextOnRequest • AccTooltip • Changeable • DefaultTooltip • Height • IconName • IsSymbolFont • Left • Modified • ParentFrame • ScreenLeft • ScreenTop • Text • Tooltip • Top • Width	

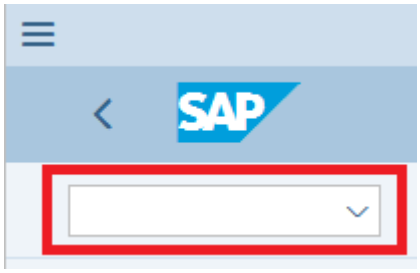
Property Syntax	Description
All properties of the GuiContainer Object [page 83]: • Children	
All additional properties of the GuiShell Object [page 195]: • AccDescription • CurrentContextMenu • DragDropSupported • Handle • OcxEvents • SubType	
Document (Read-only) `Public Property Document As Object`	The document hosted inside the GuiOfficeIntegration object.
HostedApplication (Read-only) `Public Property HostedApplication As Long`	This property contains an index identifying the application hosted in the GuiOfficeIntegartion object. Possible values are: 1. Microsoft Word (value = "1") 2. Microsoft Excel (value = "2") 3. Microsoft Powerpoint (value = "3")

1.2.42 GuiOkCodeField Object

Description

The GuiOkCodeField is placed on the upper toolbar of the main window. It is a combo box into which commands can be entered. Setting the text of GuiOkCodeField will not execute the command until server communication is started, for example by emulating the Enter key (VKey 0). GuiOkCodeField extends the GuiVComponent Object [page 263]. The type prefix is okcd, the name is empty.

Example



Methods

Method	Description
All methods of the GuiVComponent Object [page 263]: • DumpState • SetFocus • Visualize	
PressF1 `Public Sub PressF1()`	Emulate pressing the F1 key while the focus is on the GuiOkCodeField.

Properties

Property Syntax	Description
All properties of the GuiComponent Object [page 78]: • ContainerType • Id • Name • Parent • Type • TypeAsNumber	

Property	
Syntax	**Description**
All properties of the GuiVComponent Object [page 263]: • AccLabelCollection • AccText • AccTextOnRequest • AccTooltip • Changeable • DefaultTooltip • Height • IconName • IsSymbolFont • Left • Modified • ParentFrame • ScreenLeft • ScreenTop • Text • Tooltip • Top • Width	
Opened (Read-only) `Public Property Opened As Byte`	In SAP GUI designs newer than Classic design the GuiOkCodeField can be collapsed using the arrow button to the right of it. In SAP GUI for Windows the GuiOkCodeField may also be collapsed via a setting in the Windows registry. This property contains False is the GuiOkCodeField is collapsed.

1.2.43 GuiPasswordField Object

Description

The only difference between GuiTextField and GuiPasswordField is that the Text property cannot be read for a password field. The returned text is always empty. During recording the password is also not saved in the recorded script. GuiPasswordField extends the GuiTextField [page 232]. The type prefix is pwd, the name is the fieldname taken from the SAP data dictionary.

Methods

Method	
Syntax	**Description**
All methods of the GuiVComponent Object [page 263]: • DumpState • SetFocus • Visualize	

Properties

Property	
Syntax	**Description**
All properties of the GuiComponent Object [page 78]: • ContainerType • Id • Name • Parent • Type • TypeAsNumber	

Property

Syntax	Description
All properties of the GuiVComponent Object [page 263]: • AccLabelCollection • AccText • AccTextOnRequest • AccTooltip • Changeable • DefaultTooltip • Height • IconName • IsSymbolFont • Left • Modified • ParentFrame • ScreenLeft • ScreenTop • Text • Tooltip • Top • Width	
All properties of the GuiBox Object [page 51]: • CharHeight • CharLeft • CharTop • CharWidth	

Property	
Syntax	**Description**
All properties of the GuiTextField [page 232]: • CaretPosition • DisplayedText • Highlighted • HistoryCurEntry • HistoryCurIndex • HistoryIsActive • HistoryList • IsHotspot • sLeftLabel • IsListElement • IsOField • IsRightLabel • LeftLabel • MaxLength • Numerical • Required • RightLabel	

1.2.44 GuiPicture Object

Description

The picture control displays a picture on an SAP GUI screen. GuiPicture extends the GuiShell Object [page 195].

Methods

Method	Description
All methods of the GuiVComponent Object [page 263]: • DumpState • SetFocus • Visualize	
All methods of the GuiContainer Object [page 83]: • FindById	
All methods of the GuiVContainer Object [page 268]: • FindAllByName • FindAllByNameEx • FindByName • FindByNameEx	
Click `Public Sub Click()`	This function emulates a single mouse click on a picture.

ClickControlArea
`Public Sub ClickControlArea( _`
`  ByVal x As Long, _`
`  ByVal y As Long _`
`)`
	The function emulates a click on a given position. The coordinates should be given in pixels with respect to the picture control as it is displayed on the screen.

ClickPictureArea
`Public Sub ClickPictureArea( _`
`  ByVal x As Long, _`
`  ByVal y As Long _`
`)`
	The function emulates a click on a given position. The coordinates should be given in pixels with respect to the original picture file. They may differ from the pixel coordinates of the displayed picture because of scaling.

ContextMenu
`Public Sub ContextMenu( _`
`  ByVal x As Long, _`
`  ByVal y As Long _`
`)`
	The function opens a context menu on the given position. The coordinates should be given in pixels with respect to the picture control as it is displayed on the screen.

DoubleClick `Public Sub DoubleClick()`	This function emulates a double-click on a picture.

Method	Description

DoubleClickControlArea
`Public Sub DoubleClickControlArea( _`
`  ByVal x As Long, _`
`  ByVal y As`
`  Long _`
`)`
	The function emulates a double-click on a given position. The coordinates should be given in pixels with respect to the picture control as it is displayed on the screen.

DoubleClickPictureArea
`Public Sub DoubleClickPictureArea( _`
`  ByVal x As Long, _`
`  ByVal y As`
`  Long _`
`)`
	The function emulates a double-click on a given position. The coordinates should be given in pixels with respect to the original picture file. They may differ from the pixel coordinates of the displayed picture because of scaling.

Properties

Property Syntax	Description
All properties of the GuiComponent Object [page 78] • ContainerType • Id • Name • Parent • Type • TypeAsNumber	

Property

Syntax	Description
All properties of the GuiVComponent Object [page 263]: • AccLabelCollection • AccText • AccTextOnRequest • AccTooltip • Changeable • DefaultTooltip • Height • IconName • IsSymbolFont • Left • Modified • ParentFrame • ScreenLeft • ScreenTop • Text • Tooltip • Top • Width	
All properties of the GuiContainer Object [page 83]: • Children	
All properties of the GuiShell Object [page 195]: • AccDescription • CurrentContextMenu • DragDropSupported • Handle • OcxEvents • SubType	
AltText (Read-only) `Public Property AltText() As String`	This property contains the alternative text that can be assigned to an image (for example used for visually impaired people when a screenreader is used).

Property

Syntax	Description
DisplayMode (Read-only) `Public Property DisplayMode() As String`	Possible values of this property are: • "Normal": This value indicated that the picture is shown in its original size. If the picture's size is larger than the size of the control, the control provides scrollbars. If the picture's size is smaller than the size of the control, the picture is shown in the upper left corner of the control. • "Stretch": The picture is resized in a way that it always occupies the complete area of the control. • "Fit": The picture is resized on way that it fits into the control area without having the need to show scrollbars. In contrast to "Strech" the mode "Fit" preserves the ratio of width and height of the picture. • "NormalCenter": Like "Normal" except that the picture is not shown in the upper left corner but in the center of the control. • "FitCenter": Like "Fit" except that the picture is not shown in the upper left corner but in the center of the control.
Icon (Read-only) `Public Property Icon() As String`	Returns the SAPGUI icon code (e.g. "@01@") of the displayed icon. If no icon is displayed, the property contains an empty string.
Url (Read-only) `Public Property Url() As String`	Returns the URL of the displayed picture. If an icon is displayed (see property "icon"), the property contains an empty string. Depending in the application that used the control the URL may contain temporary URL parts (e.g. UUIDs).

1.2.45 GuiRadioButton Object

Description

GuiRadioButton extends the GuiVComponent Object [page 263]. The type prefix is rad, the name is the fieldname taken from the SAP data dictionary.

Methods

Method	
Syntax	**Description**
All methods of the GuiVComponent Object [page 263]: • DumpState • SetFocus • Visualize	
Select `Public Sub Select()`	Selecting a radio button automatically deselects all the other buttons within that group. This may cause a server round-trip, depending on the definition of the button in the screen painter.

Properties

Property	
Syntax	**Description**
All properties of the GuiComponent Object [page 78]: • ContainerType • Id • Name • Parent • Type • TypeAsNumber	

Property

Syntax	Description
All properties of the GuiVComponent Object [page 263]: • AccLabelCollection • AccText • AccTextOnRequest • AccTooltip • Changeable • DefaultTooltip • Height • IconName • IsSymbolFont • Left • Modified • ParentFrame • ScreenLeft • ScreenTop • Text • Tooltip • Top • Width	
All properties of the GuiButton Object [page 53]: • LeftLabel • RightLabel	
CharHeight (Read-only) `Public Property CharHeight As Long`	Height of the GuiRadioButton in character metric.
CharLeft (Read-only) `Public Property CharLeft As Long`	Left coordinate of the GuiRadioButton in character metric.
CharTop (Read-only) `Public Property CharTop As Long`	Top coordinate of the GuiRadioButton in character metric.
CharWidth (Read-only) `Public Property CharWidth As Long`	Width of the GuiRadioButton in character metric.
Flushing (Read-only) `Public Property Flushing As Byte`	Some components such as radio buttons or checkboxes may cause a round trip when their value is changed. If this is the case, the Flushing property is True.

Property

Syntax	Description
GroupCount (Read-only) `Public Property GroupCount As Long`	The number of radio buttons in the same group the current object belongs to.

GroupMembers (Read-only)
`Public Property GroupMembers As GuiComponentCollection`
	Example
	The collection of GuiRadioButton objects belonging to the same radio button group.
	Example:
	`Set GroupMembers = session.findById("wnd[0]/usr/radRB2").GroupMembers`
	`For  Each GroupMember In GroupMembers`
	`  MsgBox GroupMember.Text`
	`Next`

GroupPos (Read-only) `Public Property GroupPos As Long`	The position of the radio button in the respective radio button group (ranging from 1 to GroupCount).
IsLeftLabel (Read-only) `Public Property IsLeftLabel As Byte`	This property is True if the component has the 'assign left' flag.
IsRightLabel (Read-only) `Public Property IsRightLabel As Byte`	This property is True if the component has the 'assign right' flag.
LeftLabel `Public Property LeftLabel As GuiVComponent`	Left label of the GuiRadioButton. The label is assigned in the Screen Painter, using the flag 'assign left'.
RightLabel `Public Property RightLabel As uiVComponent`	Right label of the GuiRadioButton. This property is set in Screen Painter using the 'assign right' flag.

1.2.46 GuiSapChart Object

Description

For the SAP chart control only basic members from GuiShell are available. Recording and playback is not possible.

Methods

Method	
Syntax	**Description**
All methods of the GuiVComponent Object [page 263]: • DumpState • SetFocus • Visualize	
All methods of the GuiContainer Object [page 83]: • FindById	
All methods of the GuiVContainer Object [page 268]: • FindAllByName • FindAllByNameEx • FindByName • FindByNameEx	
All methods of the GuiShell Object [page 195]: • SelectContextMenuItem • SelectContextMenuItemByPosition • SelectContextMenuItemByText	

Properties

Property Syntax	Description
All properties of the GuiComponent Object [page 78]: • ContainerType • Id • Name • Parent • Type • TypeAsNumber	
All properties of the GuiVComponent Object [page 263]: • AccLabelCollection • AccText • AccTextOnRequest • AccTooltip • Changeable • DefaultTooltip • Height • IconName • IsSymbolFont • Left • Modified • ParentFrame • ScreenLeft • ScreenTop • Text • Tooltip • Top • Width	
All properties of the GuiContainer Object [page 83]: • Children	
All properties of the GuiShell Object [page 195]: • AccDescription • CurrentContextMenu • DragDropSupported • Handle • OcxEvents • SubType	

1.2.47 GuiScrollbar Object

Description

The GuiScrollbar class is a utility class used for example in GuiScrollContainer or GuiTableControl.

Properties

Property Syntax	Description
Maximum (Read-only) `Public Property Maximum As Long`	This is the maximum position of the scrollbar.
Minimum (Read-only) `Public Property Minimum As Long`	This is the minimum position of the scrollbar.
PageSize (Read-only) `Public Property PageSize As Long`	When the user scrolls down a page, position will be increased by the value of pageSize.
Position (Read-write) `Public Property Position As Long`	The position of the thumb of the scrollbar can be set to values from minimum to maximum.
Maximum (Read-only) `Public Property Maximum As Long`	This is the maximum position of the scrollbar.

1.2.48 GuiScrollContainer Object

Description

This container represents scrollable subscreens. A subscreen may be scrollable without actually having a scrollbar, because the existence of a scrollbar depends on the amount of data displayed and the size of the

GuiUserArea. GuiScrollContainer extend sthe GuiVContainer Object [page 268]. The type prefix is ssub, the name is generated from the data dictionary settings.

Methods

Method	
Syntax	**Description**
All methods of the GuiVComponent Object [page 263]: • DumpState • SetFocus • Visualize	
All methods of the GuiContainer Object [page 83]: • FindById	
All methods of the GuiVContainer Object [page 268]: • FindAllByName • FindAllByNameEx • FindByName • FindByNameEx	

Properties

Property	
Syntax	**Description**
All properties of the GuiComponent Object [page 78]: • ContainerType • Id • Name • Parent • Type • TypeAsNumber	

Property	
Syntax	**Description**
All properties of the GuiVComponent Object [page 263]: • AccLabelCollection • AccText • AccTextOnRequest • AccTooltip • Changeable • DefaultTooltip • Height • IconName • IsSymbolFont • Left • Modified • ParentFrame • ScreenLeft • ScreenTop • Text • Tooltip • Top • Width	
All properties of the GuiContainer Object [page 83]: • Children	
All properties of the GuiBox Object [page 51]: • CharHeight • CharLeft • CharTop • CharWidth	
HorizontalScrollbar (Read-write) `Public Property HorizontalScrollbar As GuiScrollbar`	The horizontal scrollbar of the scroll container.
VerticalScrollbar (Read-write) `Public Property VerticalScrollbar As GuiScrollbar`	The vertical scrollbar of the scroll container.

1.2.49 GuiSession Object

Description

The GuiSession provides the context in which a user performs a certain task such as working with a transaction. It is therefore the access point for applications, which record a user's actions regarding a specific task or play back those actions. GuiSession extends GuiContainer. The type prefix is ses, the name is ses plus the session number in square brackets.

Remarks

GuiSession is self-contained in that ids within the context of a session remain valid independently of other connections or sessions being open at the same time. Usually an external application will first determine with which session to interact. Once that is clear, the application will work more or less exclusively on that session. Traversing the object hierarchy from the GuiApplication to the user interface elements, it is the session among whose children the highest level visible objects can be found. In contrast to objects like buttons or text fields, the session remains valid until the corresponding main window has been closed, whereas buttons, for example, are destroyed during each server communication.

Methods

Method Syntax	Description
All methods of the GuiContainer Object [page 83]: • FindById	

AsStdNumberFormat
`Public Function AsStdNumberFormat( _`
`  ByVal Number As String _`
`) As String`
	Depending on the system's number format the minus sign of numbers may be placed to the right of the number. Using this function the minus sign is moved to the left.

ClearErrorList `Public Sub ClearErrorList()`	This method clears the list of errors that may be created when ActiveX controls are found on a screen that do not support SAP GUI Scripting. Otherwise the list is cleared after an error event was raised. This happens at the end of a round trip.
CreateSession `Public Sub CreateSession()`	This function opens a new session, which is then visualized by a new main window. This resembles the "/o" command that can be executed from the command field.

Method	
Syntax	**Description**
EnableJawsEvents `Public Sub EnableJawsEvents()`	Enable the sending of events to the screenreader Freedom Scientific JAWS, which communicates with SAP GUI for Windows via the Scripting API. By default the sending of events is activated.
EndTransaction `Public Sub EndTransaction()`	Calling this function has the same effect as SendCommand("/n").

FindByPosition
`Public Function FindByPosition( _`
`  ByVal x As Long, _`
`  ByVal y As Long, _`
`  Optional ByVal Raise As Variant _`
`) As GuiCollection`
	This method can be used to do a hittest on an SAP GUI session. The parameters x and y should be given in screen coordinates. If no component is found an exception is raised unless raise is set to False. In that case a Null/Nothing object is returned.

GetIconResourceName
`Public Function GetIconResourceName( _`
`  ByVal Text As String _`
`) As String`
	In SAP GUI icons are often described as text in the format @nn@ where nn is a number. The function getIconResourceName translates the @nn@ notation into the name of the resource in sapbtmp.dll.

GetVKeyDescription
`Public Function GetVKeyDescription( _`
`  ByVal VKey As Long _`
`) As String`
	When a script is recorded, it will often contain sendVKey(n) calls, where n is a number. The method getVKeyDescription translates these numbers into a readable text. For example the number 0 is translated into the text “Enter”.

LockSessionUI `Public Sub LockSessionUI()`	This method locks the session so that no user interaction is possible until the session is unlocked using UnlockSessionUI.

SendCommand
`Public Sub SendCommand( _`
`  ByVal Command As String _`
`)`
	Using this function you can execute any command string, which could otherwise be entered in the command field combo box.

StartTransaction
`Public Sub StartTransaction( _`
`  ByVal Transaction As String _`
`)`
	Calling this function with parameter "xyz" has the same effect as SendCommand("/nxyz").

UnlockSessionUI `Public Sub UnlockSessionUI()`	This method unlocks the session after it was locked using LockSessionUI.

Properties

Property Syntax	Description
All properties of the GuiComponent Object [page 78]: • ContainerType • Id • Name • Parent • Type • TypeAsNumber	
All properties of the GuiContainer Object [page 83]: • Children	
AccEnhancedTabChain (Read-write) `Public Property AccEnhancedTabChain As Byte`	This property is True if the respective option "Include read-only and disabled elements in tab chain" has been set in the SAP GUI options dialog.
AccSymbolReplacement (Read-write) `Public Property AccSymbolReplacement As Byte`	This property is True if the respective option "Display symbols in lists as letters" has been set in the SAP GUI options dialog.
ActiveWindow (Read-only) `Public Property ActiveWindow As GuiFrameWindow`	All windows can be found in the children collection of GuiSession. However, most of the time an application will access the currently activated window of the session, as that is the window with which a user will probably interact. This property is intended as a shortcut to this window.
Busy (Read-write) `Public Property Busy As Byte`	While SAP GUI is waiting for data from the server, any Scripting call will not return, which blocks the executing thread. This may not be acceptable for advanced applications. A way to prevent this is to check the busy property of the session. If this property is True, then a subsequent Scripting call will wait for the server communication to be finished.
ErrorList (Read-write) `Public Property ErrorList As GuiCollection`	
Info (Read-only) `Public Property Info As GuiSessionInfo`	Info is of type GuiSessionInfo. It contains technical information about the current connection, the login data, the running SAP application and more.

Property

Syntax	Description
IsActive (Read-write) `Public Property IsActive As Byte`	TRUE if the session window is active. FALSE overwise.
IsListBoxActive (Read-only) `Public Property IsListBoxActive As Byte`	This property is True if a listbox is currently open (for a GuiComboBox).
ListBoxCurrEntry (Read-only) `Public Property ListBoxCurrEntry As Long`	The index of the currently selected listbox entry.
ListBoxCurrEntryHeight (Read-only) `Public Property ListBoxCurrEntryHeight As Long`	The height of the current entry of the listbox in pixels.
ListBoxCurrEntryLeft (Read-only) `Public Property ListBoxCurrEntryLeft As Long`	The left position of the current entry of the listbox in pixels.
ListBoxCurrEntryTop (Read-only) `Public Property ListBoxCurrEntryTop As Long`	The top position of the current entry of the listbox in pixels.
ListBoxCurrEntryWidth (Read-only) `Public Property ListBoxCurrEntryWidth As Long`	The width of the current entry of the listbox in pixels.
ListBoxHeight (Read-only) `Public Property ListBoxHeight As Long`	The height of the open listbox in pixels.
ListBoxLeft (Read-only) `Public Property ListBoxLeft As Long`	The left position of the open listbox in pixels.
ListBoxTop (Read-only) `Public Property ListBoxTop As Long`	The top position of the open listbox in pixels.
ListBoxWidth (Read-only) `Public Property ListBoxWidth As Long`	The width of the open listbox in pixels.

Property

Syntax	Description
PassportPreSystemId (Read-write) `Public Property PassportPreSystemId As String`	The pre-system ID. Part of the passport information.
PassportSystemId (Read-write) `Public Property PassportSystemId As String`	The system ID. Part of the passport information.
PassportTransactionId (Read-write) `Public Property PassportTransactionId As String`	The unique ID of the transaction. Part of the passport information.
ProgressPercent (Read-only) `Public Property ProgressPercent As LongPublic`	The percentage displayed by the SAP GUI progress indicator.
ProgressText (Read-only) `Public Property ProgressText As String`	The text displayed by the progress indicator.
Record (Read-write) `Public Property Record As Byte`	Setting this property to True enables the recording mode of the session. In this mode changes to elements of the user interface are recorded within SAP GUI and sent to a recording application using the Change event described later. **Remarks** Some elements of the user interface may behave differently in record mode than during playback or manual interaction. • The F4 help dialog is always displayed as a modal window. • Drag & Drop is disabled.
RecordFile (Read-write) `Public Property RecordFile As String`	A simple way to record a script it to set the recordFile property to a valid filename and then enable the record property. A Visual Basic Script file of the given name will be created in the SAP GUI Scripts Folder on the client PC. **Remarks** This property only accepts simple filenames without path information.
SaveAsUnicode (Read-write) `Public Property SaveAsUnicode As Byte`	If this property is set to TRUE, the recorded scripts will be saved in UNICODE encoding. Overwise is the current system codepage.

Property

Syntax	Description
ShowDropdownKeys (Read-write) `Public Property ShowDropdownKeys As Byte`	If this property is TRUE, the dropdowns show not only the text of dropdown entries, but also the keys.
SuppressBackendPopups (Read-write) `Public Property SuppressBackendPopups As Byte`	
TestToolMode (Read-write) `Public Property TestToolMode As Long`	During internal tests some aspects of the user interface proved to be difficult to handle with test tools using the Scripting API to automate SAP GUI. For this reason a special mode has been added in which the following changes are administered. • While success (S), warning (W) and error (E) messages are always displayed in the status bar, information (I) and abort (A) messages are displayed as pop-up windows unless testToolMode is set. • The update mode of the application server is changed to immediate mode for the connection. • System messages are ignored so that they do not interrupt the recording or playback of scripts. **Remarks** The test tool mode requires one of the following versions of the SAP kernel: • 6.20 Patch level 29 and all following kernel versions • 4.6D Patch level 1208, see note 511310. Currently only the following values are allowed for this property: • 0: Disable testToolMode • 1: Enable testToolMode

Events

Event

Syntax	Description

AbapScriptingEvent
`Public Event AbapScriptingEvent( _`
`  ByVal param As String _`
`)`	

Activated
`Public Event Activated( _`
`  ByVal Session As GuiSession _`
`)`	

AutomationFCode
`Public Event AutomationFCode( _`
`  ByVal Session As GuiSession, _`
`  ByVal FunctionCode As String _`
`)`
	The event is only fired when using the SAP Workplace. It notifies the listener that SAP GUI executes a function code that was set by the Workplace framework.

Event

Syntax	Description

Change
`Public Event Change( _`
`  ByVal Session As GuiSession, _`
`  ByVal Component As GuiComponent, _`
`  ByVal CommandArray As Variant _`
`)`
	In record mode, the session collects changes to elements of the user interface and sends these changes to a listening application whenever server communication is about to start or if the record mode is turned off. The change events are raised immediately before the startRequest event. There is at least one event for every modified element in the recorded session.
	Remarks
	Only changes made at the SAP GUI level are recorded. Transactions may preset some of the entry fields with values from parameters stored in the SAP system. If these data are not changed in SAP GUI, they will not be recorded. This may cause problems during playback of scripts, if the entry fields are preset with different values.
	If any of the following techniques is used in a transaction, the user should manually modify all the entries he wants to see recorded:
	• Usage of SAP parameters
	• Variants
	• Hold Data, from the menu System -> User Profile
	Playback of the changes will only work, if the order of the calls is the same as during recording.
	Each event represents one line of script code. The Component parameter specifies the object on which to invoke a method or property. Therefore the first thing to record is Component.id for later use with findById. The recorder may however also decide to record other properties of Component. If, for example, a line in a table control or list is selected, it may be prudent not to record the position of the line, but rather the values in it. That way, a script can be generated that is more robust with respect to changes in the number, and therefore in the position, of lines.
	If new function modules have been added, selecting a line from the list might return the wrong function module.

Type	Method/Property name	Parameters
"SP"	"Text"	"Hello World"

This sets the parameter Text to value “Hello World”.

Event	
Syntax	**Description**

Type	Method/Property name	Parameters
"SP"	"RecordMode"	True

This sets the parameter RecordMode to the Boolean value True. It is up to the recorder to generate a script line with a valid textual representation of Boolean values, such as “true”, “True” or “TRUE” for example.

Type	Method/Property name	Parameters
"SP"	"TestToolMode"	0

This sets the parameter TestToolMode to value 0.

Type	Method/Property name	Parameters
"M"	"Resize"	96
		32
		False

The method Resize is called with three parameters. In this case the third member of the CommandArray is an array with 3 elements.

ContextMenu

```
Public Event ContextMenu( _
   ByVal Session As GuiSession, _
   ByVal Component As GuiVComponent _
)
```

The contextMenu event is fired when SAP GUI is about to display a context menu. There are currently the following limitations:

- Only context menus of controls of type GuiShell are supported.
- The event is not fired for “cached” context menus, which are not retrieved from the server when being opened.

Destroy

```
Public Event Destroy( _
   ByVal Session As GuiSession _
)
```

This event is raised before a session is destroyed.

Event

Syntax	Description

EndRequest
`Public Event EndRequest( _`
`  ByVal Session As GuiSession _`
`)`
	endRequest is called immediately after the session is unlocked after server communication.

Error
`Public Event Error( _`
`  ByVal Session As GuiSession, _`
`  ByVal ErrorId As Long, _`
`  ByVal Desc1 As String, _`
`  ByVal Desc2 As String, _`
`  ByVal Desc3 As String, _`
`  ByVal Desc4 As String _`
`)`
	An error event is currently only raised, if the wrapper library required to access an SAP GUI ActiveX control from a script is not available. error events from all sessions are also available at the GuiApplication.

FocusChanged
`Public Event FocusChanged( _`
`  ByVal Session As GuiSession, _`
`  ByVal NewFocusedControl As GuiVComponent _`
`)`
	This event is triggered when the focus in SAP GUI is moved to a new item. Using the parameters one can identify which item in which session received focus.

HistoryOpened
`Public Event HistoryOpened( _`
`  ByVal Session As GuiSession, _`
`  ByVal NewFocusedControl As GuiVComponent _`
`)`
	This event is triggered when the SAP GUI input history is opened. Using the parameters one can identify the session and the object for which the history was opened.

Hit
`Public Event Hit( _`
`  ByVal Session As GuiSession, _`
`  ByVal Component As GuiComponent, _`
`  ByVal InnerObject As String _`
`)`
	The hit event is only raised when elementVisualizationMode is set to True, which turns on the hit test mode of SAP GUI. If in this mode a SAP GUI component is identified, the hit event is raised. The parameters of this event are
	• The session on which the component was hit
	• The component that was hit
	• A description of an inner object of the component if an inner object was hit

ProgressIndicator
`Public Event ProgressIndicator( _`
`  ByVal percentage As Long, _`
`  ByVal Text As String _`
`)`
	This event is triggered when the SAP GUI progress indicator is displayed. The properties contain the current percentage and the text of the progress indicator.

Event

Syntax	Description

StartRequest
`Public Event StartRequest( _`
`  ByVal Session As GuiSession _`
`)`
	The startRequest event is raised before the session is locked during server communication. At this point user input can be checked before it is sent to the server. It is not possible to prevent server communication from this event.

1.2.50 GuiSessionInfo Object

Description

GuiSessionInfo is a member of all GuiSession objects. It makes available technical information about the session. Some of its properties are displayed in the system information area (either in the status bar or the title area depending on the SAP GUI theme used).

Properties

Property

Syntax	Description
ApplicationServer (Read-only) `Public Property ApplicationServer As String`	The name of the application server is set only if the session belongs to a connection that was started without load balancing, by specifying an application server.
Client (Read-only) `Public Property Client As String`	The client selected on the login screen.
Codepage (Read-only) `Public Property Codepage As Long`	The codepage specified in SAP Logon in the properties of the connection.
Flushes (Read-only) `Public Property Flushes As Long`	The property flushes counts the number of flushes in the automation queue during server communication.

Property

Syntax	Description
Group (Read-only) `Public Property Group As String`	The login group information is available only if the session belongs to a connection which was started using load balancing.
GuiCodepage (Read-only) `Public Property GuiCodepage As Long`	A list of codepages is available in table TCP00A of the SAP system. On a client running Microsoft Windows with codepage 1252 (Latin I) the property guiCodepage is 1160.
I18NMode (Read-only) `Public Property I18NMode As Byte`	The I18N mode of SAP GUI is required for multi-byte character sets.
InterpretationTime (Read-only) `Public Property InterpretationTime As Long`	The interpretation time begins after the data have arrived from the server. It comprises the parsing of the data and distribution to the SAP GUI elements. The unit is milliseconds.
IsLowSpeedConnection (Read-only) `Public Property IsLowSpeedConnection As Byte`	The property is True if the connection to which the session belongs runs with the low speed connection flag. This flag can be set on the advanced connection properties page of the SAPLogon dialog. The SAP GUI Scripting support is very limited for low speed connections, because information required to identify SAP GUI objects is not being sent.
Language (Read-only) `Public Property Language As String`	The language specified on the login screen.
MessageServer (Read-only) `Public Property MessageServer As String`	The message server information is available only if the session belongs to a connection which was started using load balancing.
Program (Read-only) `Public Property Program As String`	The name of the source program that is currently being executed.
ResponseTime (Read-only) `Public Property ResponseTime As Long`	This is the time that is spent on network communication from the moment data are sent to the server to the moment the server response arrives. The unit is milliseconds.
RoundTrips (Read-only) `Public Property RoundTrips As Long`	Before SAP GUI sends data to the server it locks the user interface. In many cases it will not unlock the interface once data arrive from the server, but instead will send a new request to the server immediately. Controls in particular use this technology to load the data they need for visualization. The count of these token switches between SAP GUI and the server is the roundTrips property.

Property

Syntax	Description
ScreenNumber (Read-only) `Public Property ScreenNumber As Long`	The number of the screen currently displayed.
ScriptingModeReadOnly (Read-only) `Public Property ScriptingModeReadOnly As Byte`	The read only mode can be enabled using an application server profile parameter. In this mode the state of SAP applications cannot be changed through the Scripting API, which means: • Properties can only be read, but not set • Functions can only be called if they do not change the control's state. **Remarks** In this mode scripts can be recorded and information about the application can be read from SAP GUI. However a transaction cannot be run from a script. Additional documentation is available in note 692245 and in the SAP GUI Scripting security documentation on the service market place.
ScriptingModeRecordingDisabled (Read-only) `Public Property ScriptingModeRecordingDisabled As Byte`	The recording disabled mode can be enabled using an application server profile parameter. In this mode SAP GUI Scripting does not fire any events. This implies that user interaction cannot be recorded. However data can be read from SAP GUI and scripts can be used to run transactions.
SessionNumber (Read-only) `Public Property SessionNumber As Long`	The number of the session is also displayed in SAP GUI on the status bar.
SystemName (Read-only) `Public Property SystemName As String`	This is the name of the SAP system.
SystemNumber (Read-only) `Public Property SystemNumber As Long`	The system number is set only if the session belongs to a connection that was started without load balancing, by specifying an application server.
SystemSessionId (Read-only) `Public Property SystemSessionId As String`	All SAP GUI sessions of the same connection are represented on the server with the same SystemSessionId. Using SystemSessionId and SessionNumber, it is possible to find a matching SAP GUI session from an ABAP application.
Transaction (Read-only) `Public Property Transaction As String`	The transaction that is currently being executed.

Property

Syntax	Description
UI_GUIDELINE (Read-only) `Public Property UI_GUIDELINE As String`	This property can be used to identify whether the SAP GUI session is running with Fiori Visual Theme (Belize) or not. The return value is • 1 if the session is running without Fiori Visual Theme (Belize) • 2 if the session is running with Fiori Visual Theme (Belize)
User (Read-only) `Public Property User As String`	The SAP name of the user logged into the system.

1.2.51 GuiShell Object

Description

GuiShell is an abstract object whose interface is supported by all the controls. GuiShell extends the GuiVContainer Object [page 268]. The type prefix is shell, the name is the last part of the id, shell[n].

Methods

Method

Syntax	Description
All methods of the GuiVComponent Object [page 263]: • DumpState • SetFocus • Visualize	
All methods of the GuiContainer Object [page 83]: • FindById	

Method Syntax	Description
All methods of the GuiVContainer Object [page 268]: • FindAllByName • FindAllByNameEx • FindByName • FindByNameEx	

SelectContextMenuItem
`Public Sub SelectContextMenuItem( _`
`  ByVal FunctionCode As String _`
`)`
	Select an item from the control's context menu.

SelectContextMenuItemByPosition
`Public Sub`
`SelectContextMenuItemByPosition( _`
`  ByVal PositionDesc As String _`
`)`
	This method allows you to select a context menu item using the position of the item. It is therefore independent of the menu item text.

SelectContextMenuItemByText
`Public Sub`
`SelectContextMenuItemByText( _`
`  ByVal Text As String _`
`)`
	Select a menu item of a context menu using the text of the item and possible higher level menus.

Properties

Property Syntax	Description
All properties of the GuiComponent Object [page 78]: • ContainerType • Id • Name • Parent • Type • TypeAsNumber	

Property

Syntax	Description
All properties of the GuiVComponent Object [page 263]: • AccLabelCollection • AccText • AccTextOnRequest • AccTooltip • Changeable • DefaultTooltip • Height • IconName • IsSymbolFont • Left • Modified • ParentFrame • ScreenLeft • ScreenTop • Text • Tooltip • Top • Width	
All additional properties of the GuiContainer Object [page 83]: • Children	
AccDescription (Read-only) `Public Property AccDescription As String`	Accessibility description of the shell. This description can be used for shells that do not have a title element.
CurrentContextMenu (Read-only) `Public Property CurrentContextMenu As GuiContextMenu`	This property is only set when a context menu is available at the shell object.
DragDropSupported (Read-only) `Public Property DragDropSupported As Byte`	This property is True if the shell allows drag and drop operations.
Handle (Read-only) `Public Property Handle As Long`	The window handle of the control that is connected to the GuiShell.

Property

Syntax	Description
OcxEvents (Read-only) `Public Property OcxEvents As GuiCollection`	Returns a collection containing the event ids of the ActiveX control. These are the events that the control may send to the server.
SubType (Read-only) `Public Property SubType As String`	Additional type information to identify the control represented by the shell, for example Picture, TextEdit, GridView...

1.2.52 GuiSimpleContainer Object

Description

This container represents non-scrollable subscreens. It does not have any functionality apart from to the inherited interfaces. GuiSimpleContainer extends the GuiVContainer Object [page 268]. The type prefix is sub, the name is is generated from the data dictionary settings.

Methods

Method

Syntax	Description
All methods of the GuiVComponent Object [page 263]: • DumpState • SetFocus • Visualize	
All methods of the GuiContainer Object [page 83]: • FindById	

Method

Syntax	Description
All methods of the GuiVContainer Object [page 268]: • FindAllByName • FindAllByNameEx • FindByName • FindByNameEx	

GetListProperty
`Public Function GetListProperty( _`
`  ByVal Property As String _`
`) As String`
	For more information refer to the documentation about method **GetListProperty** within GuiLabel Object [page 134].

GetListPropertyNonRec
`Public Function`
`GetListPropertyNonRec( _`
`  ByVal Property As String _`
`) As String`	

Properties

Property

Syntax	Description
All properties of the GuiComponent Object [page 78]: • ContainerType • Id • Name • Parent • Type • TypeAsNumber	

Property

Syntax	Description
All properties of the GuiVComponent Object [page 263]: • AccLabelCollection • AccText • AccTextOnRequest • AccTooltip • Changeable • DefaultTooltip • Height • IconName • IsSymbolFont • Left • Modified • ParentFrame • ScreenLeft • ScreenTop • Text • Tooltip • Top • Width	
All properties of the GuiContainer Object [page 83]: • Children	
All properties of the GuiBox Object [page 51]: • CharHeight • CharLeft • CharTop • CharWidth	
IsListElement (Read-only) `Public Property IsListElement As Byte`	This property is True if the element is on an ABAP list, not a dynpro screen.
IsStepLoop (Read-only) `Public Property IsStepLoop As Byte`	This property is True if the container is a step loop container.
LoopColCount (Read-only) `Public Property LoopColCount As Long`	If the container is a step loop container, then this property contains the number of columns in the step loop.

Property

Syntax	Description
LoopCurrentCol (Read-only) `Public Property LoopCurrentCol As Long`	If the container is a step loop container, then this property contains the current row number in the step loop.
LoopCurrentColCount (Read-only) `Public Property LoopCurrentColCount As Long`	If the container is a step loop container, then this property contains the number of columns in the current row of the step loop. Please note that depending on the type of steploop the number of columns per row may be different per row. **i Note** This property is available as of SAP GUI for Windows 7.50 patchlevel 9 and SAP GUI for Windows 7.60.
LoopCurrentRow (Read-only) `Public Property LoopCurrentRow As Long`	If the container is a step loop container, then this property contains the current column number in the step loop.
LoopRowCount (Read-only) `Public Property LoopRowCount As Long`	If the container is a step loop container, then this property contains the number of rows in the step loop.

1.2.53 GuiSplit Object

Description

GuiSplit extends the GuiShell Object [page 195].

Methods

Method	Description
All methods of the GuiVComponent Object [page 263]: • DumpState • SetFocus • Visualize	
All methods of the GuiContainer Object [page 83]: • FindById	
All methods of the GuiVContainer Object [page 268]: • FindAllByName • FindAllByNameEx • FindByName • FindByNameEx	
All methods of the GuiShell Object [page 195]: • SelectContextMenuItem • SelectContextMenuItemByPosition • SelectContextMenuItemByText	

GetColSize
`Public Function GetColSize( _`
`  ByVal Id As Long _`
`) As Long`
	This method returns the size of the splitter column specified by the parameter Id (starting with index 1) in percent.

GetRowSize
`Public Function GetRowSize( _`
`  ByVal Id As Long _`
`) As Long`
	This method returns the size of the splitter row specified by the parameter Id (starting with index 1) in percent.

SetColSize
`Public Sub SetColSize( _`
`  ByVal Id As Long, _`
`  ByVal Size`
`  As Long _`
`)`
	This method sets the size of the splitter column specified by the parameter Id (starting with index 1) to the percentage specified by parameter Size.

SetRowSize
`Public Sub SetRowSize( _`
`  ByVal Id As Long, _`
`  ByVal Size`
`  As Long _`
`)`
	This method sets the size of the splitter row specified by the parameter Id (starting with index 1) to the percentage specified by parameter Size.

Properties

Property Syntax	Description
All properties of the GuiComponent Object [page 78]: • ContainerType • Id • Name • Parent • Type • TypeAsNumber	
All properties of the GuiVComponent Object [page 263]: • AccLabelCollection • AccText • AccTextOnRequest • AccTooltip • Changeable • DefaultTooltip • Height • IconName • IsSymbolFont • Left • Modified • ParentFrame • ScreenLeft • ScreenTop • Text • Tooltip • Top • Width	
All properties of the GuiContainer Object [page 83]: • Children	
All properties of the GuiShell Object [page 195]: • AccDescription • CurrentContextMenu • DragDropSupported • Handle • OcxEvents • SubType	

Property	
Syntax	**Description**
IsVertical (Read-only) `Public Property IsVertical As Long`	This property contains True if the splitter cells of the GuiSplit are vertically aligned and False if they are horizontally aligned.

1.2.54 GuiSplitterContainer Object

Description

The GuiSplitterContainer represents the dynpro splitter element, which was introduced in the Web Application Server ABAP in NetWeaver 7.1. The dynpro splitter element is similar to the activeX based splitter control, but it is a plain dynpro element.

Methods

Method	
Syntax	**Description**
All methods of the GuiVComponent Object [page 263]: • DumpState • SetFocus • Visualize	
All methods of the GuiContainer Object [page 83]: • FindById	
All methods of the GuiVContainer Object [page 268]: • FindAllByName • FindAllByNameEx • FindByName • FindByNameEx	

Method

Syntax	Description
All methods of the GuiShell Object [page 195]: • SelectContextMenuItem • SelectContextMenuItemByPosition • SelectContextMenuItemByText	

Properties

Property

Syntax	Description
All properties of the GuiComponent Object [page 78]: • ContainerType • Id • Name • Parent • Type • TypeAsNumber	
All properties of the GuiVComponent Object [page 263]: • AccLabelCollection • AccText • AccTextOnRequest • AccTooltip • Changeable • DefaultTooltip • Height • IconName • IsSymbolFont • Left • Modified • ParentFrame • ScreenLeft • ScreenTop • Text • Tooltip • Top • Width	

Property Syntax	Description
All properties of the GuiContainer Object [page 83]: • Children	
All properties of the GuiShell Object [page 195]: • AccDescription • CurrentContextMenu • DragDropSupported • Handle • OcxEvents • SubType	
IsVertical (Read-only) `Public Property IsVertical As Byte`	This property contains True if the splitter cells of the GuiSplitterContainer are vertically aligned and False if they are horizontally aligned.
SashPosition (Read-write) `Public Property SashPosition As Long`	Contains the position of the splitter sash in characters.

1.2.55 GuiStage Object

Description

For the stage control only basic members from GuiShell are available. Recording and playback is not possible.

Methods

Method Syntax	Description
All methods of the GuiVComponent Object [page 263]: • DumpState • SetFocus • Visualize	

Method	
Syntax	**Description**
All methods of the GuiContainer Object [page 83]: • FindById	
All methods of the GuiVContainer Object [page 268]: • FindAllByName • FindAllByNameEx • FindByName • FindByNameEx	
All methods of the GuiShell Object [page 195]: • SelectContextMenuItem • SelectContextMenuItemByPosition • SelectContextMenuItemByText	

ContextMenu
`Public Sub ContextMenu( _`
`  ByVal strId As String _`
`)`
	Calling this function opens a context menu.

DoubleClick
`Public Sub DoubleClick( _`
`  ByVal strId As String _`
`)`
	This function emulates a mouse double click.

SelectItems
`Public Sub SelectItems( _`
`  ByVal strItems As String _`
`)`
	Select the items specified by the parameter strItems.

Properties

Property Syntax	Description
All properties of the GuiComponent Object [page 78]: • ContainerType • Id • Name • Parent • Type • TypeAsNumber	
All properties of the GuiVComponent Object [page 263]: • AccLabelCollection • AccText • AccTextOnRequest • AccTooltip • Changeable • DefaultTooltip • Height • IconName • IsSymbolFont • Left • Modified • ParentFrame • ScreenLeft • ScreenTop • Text • Tooltip • Top • Width	
All properties of the GuiContainer Object [page 83]: • Children	
All additional properties of the GuiShell Object [page 195]: • AccDescription • CurrentContextMenu • DragDropSupported • Handle • OcxEvents • SubType	

1.2.56 GuiStatusbar Object

Description

GuiStatusbar represents the message displaying part of the status bar on the bottom of the SAP GUI window. It does not include the system and login information displayed in the rightmost area of the status bar as these are available from the GuiSessionInfo object. GuiStatusbar extends the GuiVComponent Object [page 263]. The type prefix is sbar.

Methods

Method	
Syntax	**Description**
All methods of the GuiVComponent Object [page 263]: • DumpState • SetFocus • Visualize	
All methods of the GuiContainer Object [page 83]: • FindById	
All methods of the GuiVContainer Object [page 268]: • FindAllByName • FindAllByNameEx • FindByName • FindByNameEx	
DoubleClick `Public Sub DoubleClick()`	When a message is displayed in the GuiStatusbar, this message can be double clicked. This will usually open the SAP performance assistant.

Properties

Property Syntax	Description
All properties of the GuiComponent Object [page 78]: • ContainerType • Id • Name • Parent • Type • TypeAsNumber	
All properties of the GuiVComponent Object [page 263]: • AccLabelCollection • AccText • AccTextOnRequest • AccTooltip • Changeable • DefaultTooltip • Height • IconName • IsSymbolFont • Left • Modified • ParentFrame • ScreenLeft • ScreenTop • Text • Tooltip • Top • Width	
All properties of the GuiContainer Object [page 83]: • Children	
Handle (Read-only) `Public Property Handle As Long`	The window handle of the control that is connected to the GuiShell.
MessageAsPopup (Read-only) `Public Property MessageAsPopup As Byte`	Some messages may be displayed not only on the status bar but also as a pop-up window. In such cases, this property is set to True so that a script knows it has to close a pop-up to continue.

Property

Syntax	Description
MessageHasLongText (Read-only) `Public Property MessageHasLongText As Long`	This property can be used to determine whether the currently displayed message has a long text or not (in Belize theme or newer themes this means that the *View Details* link is displayed for this message). Possible return values • -1: Presently no message is displayed in the statusbar • 0: The message which is displayed does not have a long text • 1: The message which is displayed has a long text This property is available as of patchlevel 2 of SAP GUI for Windows 7.60.
MessageId (Read-only) `Public Property MessageId As String`	This is the name of the message class used in the ABAP message call.
MessageNumber (Read-only) `Public Property MessageNumber As String`	This is the name of the message number used in the ABAP message call. It will usually be a number, but this is not enforced by the system.

Property

Syntax	Description

MessageParameter (Read-only)
`Public Property MessageParameter As String`
	These are the values of the parameters used to expand the placeholders in the message text definition in the data dictionary. The text property of the GuiStatusbar already contains the expanded text of the message. A maximum of 8 parameter values can be provided in the ABAP coding, so index should be in the range from 0 to 7.
	Example
	The ABAP language line
	Sample Code
	`message e319(01) with 'test1' 'test2' 'test3' 'test4'.`
	will result in the following property values:
	Sample Code
	`Text = E:  test1 test2 test3 test4`
	`Type          = E`
	`Id            = 01`
	`Number        = 319`
	`Parameter 0   = test1`
	`Parameter 1   = test2`
	`Parameter 2   = test3`
	`Parameter 3   = test4`
	`Parameter 4   =`
	`Parameter 5   =`
	`Parameter 6   =`
	`Parameter 7   =`
	`as Popup      = False`
	The message 319 in message class 01 is defined as ' & & & &', with '&' being a placeholder.

MessageType (Read-only) `Public Property MessageType As String`	This property may have any of the following values: **Value** / **Description** S – Success W – Warning E – Error A – Abort I – Information

1.2.57 GuiStatusPane Object

The parent of the GuiStatusPane objects is the status bar (see also GuiStatusbar Object). The GuiStatusPane objects reflect the individual areas of the status bar, for example "pane[0]" refers to the section of the status bar where the messages are displayed. See also GuiStatusbar Object [page 209].

Methods

Method	
Syntax	**Description**
All methods of the GuiVComponent Object [page 263]: • DumpState • SetFocus • Visualize	

Properties

Property	
Syntax	**Description**
All properties of the GuiComponent Object [page 78]: • ContainerType • Id • Name • Parent • Type • TypeAsNumber	

Property	
Syntax	**Description**
All properties of the GuiVComponent Object [page 263]: • AccLabelCollection • AccText • AccTextOnRequest • AccTooltip • Changeable • DefaultTooltip • Height • IconName • IsSymbolFont • Left • Modified • ParentFrame • ScreenLeft • ScreenTop • Text • Tooltip • Top • Width	

1.2.58 GuiTab Object

Description

The GuiTab objects are the children of a GuiTabStrip object. GuiTab extends the GuiVContainer Object [page 268]. The type prefix is tabp, the name is the id of the tab's button taken from SAP data dictionary.

Methods

Method Syntax	Description
All methods of the GuiVComponent Object [page 263]: • DumpState • SetFocus • Visualize	
All methods of the GuiContainer Object [page 83]: • FindById	
All methods of the GuiVContainer Object [page 268]: • FindAllByName • FindAllByNameEx • FindByName • FindByNameEx	
ScrollToLeft `Public Sub ScrollToLeft()`	ScrollToLeft shifts the tabs so that a certain tab becomes the leftTab of the tab strip.
Select `Public Sub Select()`	This function sets the tab to be the tab strip's selected tab. Changing the selected tab of a tab strip may cause server communication.

Properties

Property Syntax	Description
All properties of the GuiComponent Object [page 78]: • ContainerType • Id • Name • Parent • Type • TypeAsNumber	

Property	
Syntax	**Description**
All properties of the GuiVComponent Object [page 263]: • AccLabelCollection • AccText • AccTextOnRequest • AccTooltip • Changeable • DefaultTooltip • Height • IconName • IsSymbolFont • Left • Modified • ParentFrame • ScreenLeft • ScreenTop • Text • Tooltip • Top • Width	
All properties of the GuiContainer Object [page 83]: • Children	

1.2.59 GuiTableColumn Collection

Description

GuiTableColumn extends the GuiComponentCollection Collection [page 79].

Methods

Method	
Syntax	**Description**

ElementAt
`Public Function ElementAt( _`
`  ByVal Index As Long _`
`) As GuiComponent`
	This function returns the member in the collection at position index, where index may range from 0 to count-1. If no member can be found for the given index, an exception is raised.

Item
`Public Function Item( _`
`  ByVal Index As Variant _`
`) As GuiComponent`
	This function returns the member in the collection at position index, where index may range from 0 to count-1. It has been added for compatibility with Microsoft Visual Basic collections. If no member can be found for the given index, an exception is raised.

Properties

Property	
Syntax	**Description**
Count (Read-only) `Public Property Count As Long`	Number of cells in the column.
DefaultTooltip (Read-only) `Public Property DefaultTooltip As String`	Tooltip text generated from the short text defined in the data dictionary for the given screen element type.
Fixed (Read-only) `Public Property Fixed As Byte`	Some columns may be fixed, which means that they will not be scrolled with the rest of the columns.
IconName (Read-only) `Public Property IconName As String`	If the object has been assigned an icon, then this property is the name of the icon, otherwise it is an empty string.
Length (Read-only) `Public Property Length As Long`	Number of cells in the column.
NewEnum (Read-only) `Public Property NewEnum As Unknown`	Property for VB collection handling.

Property

Syntax	Description
Selected (Read-write) `Public Property Selected As Byte`	This property is true if the column is selected.
Title (Read-only) `Public Property Title As String`	This is the caption of the column.
Tooltip (Read-only) `Public Property Tooltip As String`	The tooltip contains a text, which is designed to help a user understand the meaning of a given text field or button.
Type (Read-only) `Public Property Type As String`	The type information of GuiComponent can be used to determine which properties and methods an object supports. The value of the type string is the name of the type taken from this documentation.
TypeAsNumber (Read-only) `Public Property TypeAsNumber As Long`	While the type property is a string value, the typeAsNumber property is a long value that can alternatively be used to identify an object's type . It was added for better performance in methods such as FindByIdEx. Possible values for this property are taken from the GuiComponentType enumeration.

1.2.60 GuiTableControl Object

Description

The table control is a standard dynpro element, in contrast to the GuiCtrlGridView, which looks similar. GuiTableControl extends theGuiVContainer Object [page 268]. The type prefix is tbl, the name is the fieldname taken from the SAP data dictionary.

Methods

Method Syntax	Description
All methods of the GuiVComponent Object [page 263]: • DumpState • SetFocus • Visualize	
All methods of the GuiContainer Object [page 83]: • FindById	
All additional methods of the GuiVContainer Object [page 268]: • FindAllByName • FindAllByNameEx • FindByName • FindByNameEx	
ConfigureLayout `Public Sub ConfigureLayout()`	In the configuration dialog the layout of the table can be changed. This dialog is a GuiModalWindow.
DeselectAllColumns `Public Sub DeselectAllColumns()`	This function can be used for table controls with a button that allows, to deselect all columns in one step.

GetAbsoluteRow
`Public Function GetAbsoluteRow( _`
`  ByVal Index As Long _`
`) As GuiTableRow`
	Unlike the rows collection, the indexing supported by this function does not reset the index after scrolling, but counts the rows starting with the first row with respect to the first scroll position. If the selected row is not currently visible then an exception is raised.

GetCell
`Public Function GetCell( _`
`  ByVal Row As Long, _`
`  ByVal Column As Long _`
`) As GuiVComponent`
	This method returns a given table cell. It is more efficient than accessing a single cell using the rows or columns collections. Syntax
	Row: Zero-based index of the row.
	Column: Zero-based index of the column.

ReorderTable
`Public Sub ReorderTable( _`
`  ByVal Permutation As String _`
`)`
	The parameter permutation describes a new ordering of the columns. For example “1 3 2” will move column 3 to second position.

Method

Syntax	Description
SelectAllColumns `Public Sub SelectAllColumns()`	This function can be used for table controls with a button that allows, to select all columns in one step.

Properties

Property

Syntax	Description
All properties of the GuiComponent Object [page 78]: • ContainerType • Id • Name • Parent • Type • TypeAsNumber	
All properties of the GuiVComponent Object [page 263]: • AccLabelCollection • AccText • AccTextOnRequest • AccTooltip • Changeable • DefaultTooltip • Height • IconName • IsSymbolFont • Left • Modified • ParentFrame • ScreenLeft • ScreenTop • Text • Tooltip • Top • Width	

Property

Syntax	Description
All properties of the GuiContainer Object [page 83]: • Children	
CharHeight (Read-only) `Public Property CharHeight As Long`	Height of the GuiTableControl in character metric.
CharLeft (Read-only) `Public Property CharLeft As Long`	Left coordinate of the GuiTableControl in character metric.
CharTop (Read-only) `Public Property CharTop As Long`	Top coordinate of the GuiTableControl in character metric.
CharWidth (Read-only) `Public Property CharWidth As Long`	Width of the GuiTableControl in character metric.
ColSelectMode (Read-only) `Public Property ColSelectMode As GuiTableSelectionType`	There are three different modes for selecting columns or rows, which are defined in the enumeration type GuiTableSelectionType.
Columns (Read-only) `Public Property Columns As GuiCollection`	The members of this collection are of GuiTableColumn type. Therefore they do not support properties like id or name.
CurrentCol (Read-only) `Public Property CurrentCol As Long`	Zero-based index of the current column.
CurrentRow (Read-only) `Public Property CurrentRow As Long`	Zero-based index of the current row.
HorizontalScrollbar (Read-only) `Public Property HorizontalScrollbar As GuiScrollbar`	The horizontal scrollbar of the table control.
RowCount (Read-only) `Public Property RowCount As Long`	Number of rows in the table. This includes invisible rows. For the number of visible rows the property VisibleRowCount is available.

Property

Syntax	Description
Rows (Read-only) `Public Property Rows As GuiCollection`	The members of this collection are of GuiTableRow type. Indexing starts with 0 for the first visible row, independent of the current position of the horizontal scrollbar. After scrolling, a different row will have the index 0.
RowSelectMode (Read-only) `Public Property RowSelectMode As GuiTableSelectionType`	There are three different modes for selecting columns or rows, which are defined in the enumeration type GuiTableSelectionType.
TableFieldName (Read-only) `Public Property TableFieldName As String`	The name property of the table control contains the ABAP program name in addition to the plain field name. This property contains just the field name.
VerticalScrollbar (Read-only) `Public Property VerticalScrollbar As GuiScrollbar`	The vertical scrollbar of the table control.
VisibleRowCount (Read-only) `Public Property VisibleRowCount As Long`	Number of visible rows in the table. For the number of all rows the property RowCount is available.

1.2.61 GuiTableRow Collection

Description

GuiTableRow extends the GuiComponentCollection Collection [page 79].

Methods

Method	
Syntax	**Description**

ElementAt
`Public Function ElementAt( _`
`  ByVal Index As Long _`
`) As GuiComponent`
	This function returns the member in the collection at position index, where index may range from 0 to count-1. If no member can be found for the given index, an exception is raised.

Item
`Public Function Item( _`
`  ByVal Index As Variant _`
`) As GuiComponent`
	This function returns the member in the collection at position index, where index may range from 0 to count-1. It has been added for compatibility with Microsoft Visual Basic collections. If no member can be found for the given index, an exception is raised.

Properties

Property	
Syntax	**Description**
Count (Read-only) `Public Property Count As Long`	Number of cells in the row.
Length (Read-only) `Public Property Length As Long`	Number of cells in the row.
NewEnum (Read-only) `Public Property NewEnum As Unknown`	Property for VB collection handling.
Selectable (Read-write) `Public Property Selectable As Byte`	This property is True if the row can be selected.
Selected (Read-write) `Public Property Selected As Byte`	This property is true if the row is selected.
Type (Read-only) `Public Property Type As String`	The type information of GuiComponent can be used to determine which properties and methods an object supports. The value of the type string is the name of the type taken from this documentation.

Property

Syntax	Description
TypeAsNumber (Read-only) `Public Property TypeAsNumber As Long`	While the type property is a string value, the typeAsNumber property is a long value that can alternatively be used for this property are taken from the GuiComponentType enumeration.

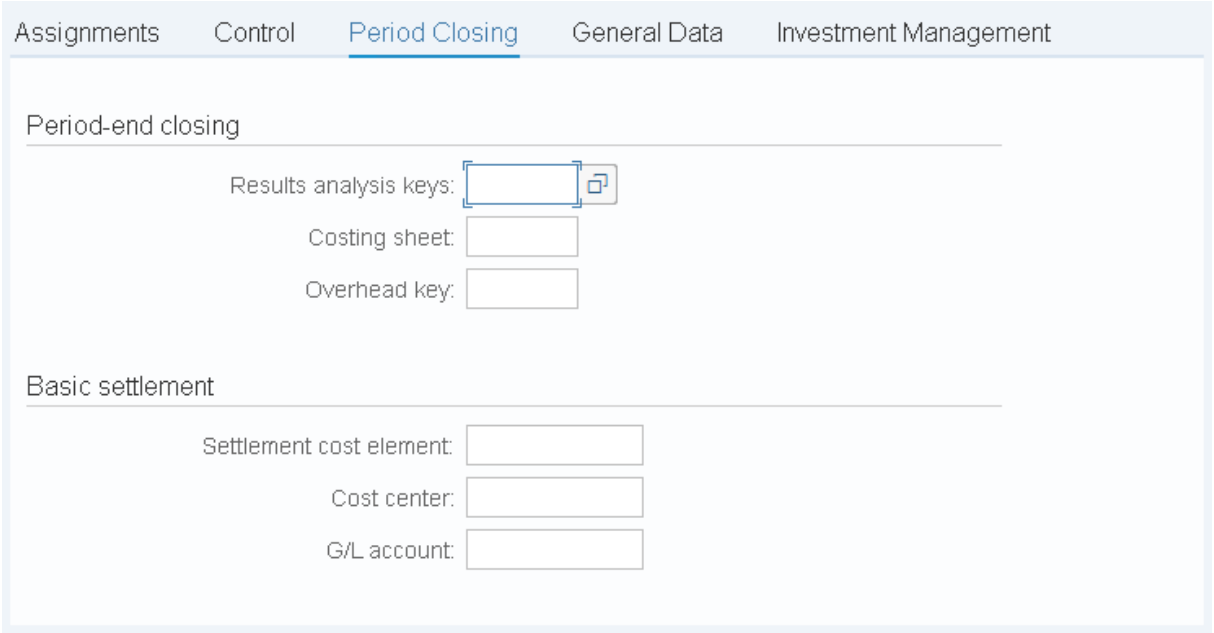
1.2.62 GuiTabStrip Object

Description

A tab strip is a container whose children are of type GuiTab. GuiTabStrip extends the GuiVContainer Object [page 268]. The type prefix is tabs, the name is the fieldname taken from the SAP data dictionary.

Example

The children of the tab strip are the tabs. While all tabs are available at any given time, only the children of the selected tab exist in the object hierarchy for server driven tab strips. So in this example, the text field labeled 'Results analysis keys:' can only be found if the tab labeled 'Period Closing' has been selected.



In some transactions there are local tabs strips where all tabs are available without further server access being required.

Methods

Method	
Syntax	**Description**
All methods of the GuiVComponent Object [page 263]: • DumpState • SetFocus • Visualize	
All methods of the GuiContainer Object [page 83]: • FindById	
All methods of the GuiVContainer Object [page 268]: • FindAllByName • FindAllByNameEx • FindByName • FindByNameEx	

Properties

Property	
Syntax	**Description**
All properties of the GuiComponent Object [page 78]: • ContainerType • Id • Name • Parent • Type • TypeAsNumber	

Property

Syntax	Description
All properties of the GuiVComponent Object [page 263]: • AccLabelCollection • AccText • AccTextOnRequest • AccTooltip • Changeable • DefaultTooltip • Height • IconName • IsSymbolFont • Left • Modified • ParentFrame • ScreenLeft • ScreenTop • Text • Tooltip • Top • Width	
All properties of the GuiContainer Object [page 83]: • Children	
CharHeight (Read-only) `Public Property CharHeight As Long`	Height of the GuiTabStrip in character metric.
CharLeft (Read-only) `Public Property CharLeft As Long`	Left coordinate of the GuiTabStrip in character metric.
CharTop (Read-only) `Public Property CharTop As Long`	Top coordinate of the GuiTabStrip in character metric.
CharWidth (Read-only) `Public Property CharWidth As Long`	Width of the GuiTabStrip in character metric.
LeftTab (Read-only) `Public Property LeftTab As GuiTab`	This is the left most tab whose caption is visible. In the example above it is the one with text 'Period closing'. The leftTab property can be changed by calling the ScrollToLeft method of a different GuiTab, as described below.

Property	
Syntax	**Description**
SelectedTab (Read-only) `Public Property SelectedTab As GuiTab`	The selected tab is the one whose descendants are currently visualized, in the example above it is the 'General data' tab. The selected tab has exactly one child, which is a GuiScroll-Container.

1.2.63 GuiTextedit Object

Description

The TextEdit control is a multiline edit control offering a number of possible benefits. With regard to scripting, the possibility of protecting text parts against editing by the user is especially useful. GuiTextedit extends the GuiShell Object [page 195].

Methods

Method	
Syntax	**Description**
All methods of the GuiVComponent Object [page 263]: • DumpState • SetFocus • Visualize	
All methods of the GuiContainer Object [page 83]: • FindById	
All methods of the GuiVContainer Object [page 268]: • FindAllByName • FindAllByNameEx • FindByName • FindByNameEx	

Method Syntax	Description
All methods of the GuiShell Object [page 195]: • SelectContextMenuItem • SelectContextMenuItemByPosition • SelectContextMenuItemByText	
ContextMenu `Public Sub ContextMenu()`	Calling ContextMenu emulates the context menu request.
DoubleClick `Public Sub DoubleClick()`	This function emulates a mouse double-click. For setting the selection, the function setSelectionIndexes can be called in advance.

GetLineText
`Public Function GetLineText( _`
`  ByVal nLine As Long _`
`) As String`
	Returns the text of the specified line.

GetUnprotectedTextPart
`Public Function`
`GetUnprotectedTextPart( _`
`  ByVal Part As Long _`
`) As String`
	This function retrieves the content of an unprotected text part using the zero based index part.

IsBreakpointLine
`Public Function IsBreakpointLine( _`
`  ByVal nLine As Long _`
`) As Byte`
	Returns TRUE if the specified line contains a breakpoint.

IsCommentLine
`Public Function IsCommentLine( _`
`  ByVal nLine As Long _`
`) As Byte`
	Returns TRUE if the specified line is a comment line.

IsHighlightedLine
`Public Function IsHighlightedLine( _`
`  ByVal nLine As Long _`
`) As Byte`
	Returns TRUE if the specified line is highlighted.

IsProtectedLine
`Public Function IsProtectedLine( _`
`  ByVal nLine As Long _`
`) As Byte`
	Returns TRUE if the specified line is protected.

Method	
Syntax	**Description**

IsSelectedLine
`Public Function IsSelectedLine( _`
`  ByVal nLine As Long _`
`) As Byte`
	Returns TRUE if the specified line is selected.

ModifiedStatusChanged
`Public Sub ModifiedStatusChanged( _`
`  ByVal Status As Boolean _`
`)`
	This function emulates the change of the modified status.

MultipleFilesDropped `Public Sub MultipleFilesDropped()`	Emulate a Drag&Drop operation, in which several files are dropped on the textedit control. The collection contains for each file the fully qualified file name as a string.
PressF1 `Public Sub PressF1()`	This function emulates pressing the F1 key on the keyboard.
PressF4 `Public Sub PressF4()`	This function emulates pressing the F4 key on the keyboard.

SetSelectionIndexes
`Public Sub SetSelectionIndexes( _`
`  ByVal Start As Long, _`
`  ByVal End As`
`  Long _`
`)`
	This function sets the visually selected text range. start and end are absolute, zero based character indexes. start corresponds to the position where the selection begins and end is the position of the first character following the selection. Note that setting start equal to end results in setting the cursor on this position.

SetUnprotectedTextPart
`Public Function`
`SetUnprotectedTextPart( _`
`  ByVal Part As Long, _`
`  ByVal Text As String _`
`) As Byte`
	This function assigns the content of text to the unprotected text part with zero based index part. The function returns True if it was possible to perform the assignment. Otherwise, False is returned.

SingleFileDropped
`Public Sub SingleFileDropped( _`
`  ByVal Filename As String _`
`)`
	This function emulates the drop of a single file with the directory path fileName.

Properties

Property Syntax	Description
All properties of the GuiComponent Object [page 78]: • ContainerType • Id • Name • Parent • Type • TypeAsNumber	
All properties of the GuiVComponent Object [page 263]: • AccLabelCollection • AccText • AccTextOnRequest • AccTooltip • Changeable • DefaultTooltip • Height • IconName • IsSymbolFont • Left • Modified • ParentFrame • ScreenLeft • ScreenTop • Text • Tooltip • Top • Width	
All properties of the GuiContainer Object [page 83]: • Children	
All properties of the GuiShell Object [page 195]: • AccDescription • CurrentContextMenu • DragDropSupported • Handle • OcxEvents • SubType	

Property

Syntax	Description
CurrentColumn (Read-only) `Public Property CurrentColumn As Long`	The number of the column in which the text caret is currently positioned.
CurrentLine (Read-only) `Public Property CurrentLine As Long`	The number of the line in which the text caret is currently positioned.
FirstVisibleLine (Read-write) `Public Property FirstVisibleLine As Long`	The first visible line is visualized at the top border of the control.
LastVisibleLine (Read-only) `Public Property LastVisibleLine As Long`	The number of the last line that is currently visible.
LineCount (Read-only) `Public Property LineCount As Long`	The number of all lines in the current document.
NumberOfUnprotectedTextParts (Read-only) `Public Property NumberOfUnprotectedTextParts As Long`	The number of unprotected text parts, which are contained.
SelectedText (Read-only) `Public Property SelectedText As String`	The currently selected text.
SelectionEndColumn (Read-only) `Public Property SelectionEndColumn As Long`	The number of the column in which the current selection ends.
SelectionEndLine (Read-only) `Public Property SelectionEndLine As Long`	The number of the line in which the current selection ends.
SelectionIndexEnd (Read-only) `Public Property SelectionIndexEnd As Long`	Retrieves the absolute, zero based character index of the ending point from the visually selected text range, i.e. the position where the selection ends. Note that a selection can be degenerated, i.e. selectionIndexStart is equal to selectionIndexEnd.

Property

Syntax	Description
SelectionIndexStart (Read-only) `Public Property SelectionIndexStart As Long`	Retrieves the absolute, zero based character index of the starting point from the visually selected text range, i.e. the position, where the selection begins. Note that a selection can be degenerated, i.e. selectionIndexStart is equal to selectionIndexEnd.
SelectionStartColumn (Read-only) `Public Property SelectionStartColumn As Long`	The number of the column in which the current selection starts.
SelectionStartLine (Read-only) `Public Property SelectionStartLine As Long`	The number of the line in which the current selection starts.

1.2.64 GuiTextField

Description

GuiTextField extends the GuiVComponent Object [page 263]. The type prefix is txt, the name is the fieldname taken from the SAP data dictionary.

Methods

Method

Syntax	Description
All methods of the GuiVComponent Object [page 263]: • DumpState • SetFocus • Visualize	

Method

Syntax	Description

GetListProperty
`Public Function GetListProperty( _`
`  ByVal Property As String _`
`) As String`
	For more information refer to the documentation about method **GetListProperty** within GuiLabel Object [page 134].

GetListPropertyNonRec
`Public Function`
`GetListPropertyNonRec( _`
`  ByVal Property As String _`
`) As String`
	This method returns information that is compiled on the server to enhance the ABAP lists with accessibility information. See GuiLabel::GetListProperty for a description of available attributes. In contrast to the method GetListProperty, GetListPropertyNonRec will only return information that is set for the specific element, and ignore list properties set for parent elements.

Properties

Property

Syntax	Description
All properties of the GuiComponent Object [page 78]: • ContainerType • Id • Name • Parent • Type • TypeAsNumber	

Property

Syntax	Description
All properties of the GuiVComponent Object [page 263]: • AccLabelCollection • AccText • AccTextOnRequest • AccTooltip • Changeable • DefaultTooltip • Height • IconName • IsSymbolFont • Left • Modified • ParentFrame • ScreenLeft • ScreenTop • Text • Tooltip • Top • Width	
All properties of the GuiBox Object [page 51] • CharHeight • CharLeft • CharTop • CharWidth	
CaretPosition (Read-write) `Public Property CaretPosition As Long`	The position of the caret within a text field may be checked by the ABAP application to determine which word the caret is in. Among other things this is useful for context sensitive help.
DisplayedText (Read-only) `Public Property DisplayedText As String`	This property contains the text as it is displayed on the screen, including preceding or trailing blanks. These blanks are stripped from the text property.
Highlighted (Read-only) `Public Property Highlighted As Byte`	This property is True if the Highlighted flag is set in the screen painter for the dynpro element. See GuiLabel for an example.
HistoryCurEntry (Read-only) `Public Property HistoryCurEntry As String`	Text of the currently focused entry in the history list box.

Property

Syntax	Description
HistoryCurIndex (Read-only) `Public Property HistoryCurIndex As Long`	Currently focused index in the history dropdown list box.
HistoryIsActive (Read-only) `Public Property HistoryIsActive As Byte`	This property is True if the local input field history drop down is currently open.
HistoryList (Read-only) `Public Property HistoryList As GuiCollection`	List of entries in the local history list box.
IsHotspot (Read-only) `Public Property IsHotspot As Byte`	Dynpro elements such as labels may be configured to cause a round trip when they are clicked. In that case the mouse cursor changes to the hand shape. This is called a hot spot.
IsLeftLabel (Read-only) `Public Property IsLeftLabel As Byte`	This property is True if the component has the 'assign left' flag.
IsListElement (Read-only) `Public Property IsListElement As Byte`	This property is True if the element is on an ABAP list, not a dynpro screen.
IsOField (Read-only) `Public Property IsOField As Byte`	OField is a special ABAP dynpro element, the Output Field. These fields can be set programmatically to a value at run-time. In this respect they differ from labels. However they cannot be used to enter data, so they are not input fields.
IsRightLabel (Read-only) `Public Property IsRightLabel As Byte`	This property is True if the component has the 'assign right' flag.
LeftLabel (Read-only) `Public Property LeftLabel As GuiVComponent`	This label has been defined in ABAP Screen Painter to be the left label of the control.
MaxLength (Read-only) `Public Property MaxLength As Long`	The maximum length of text that can be written in a text field is counted in code units. On non-Unicode clients these are equivalent to bytes.
Numerical (Read-only) `Public Property Numerical As Byte`	If this flag is set only numbers and special characters may be written into the text field.

Property

Syntax	Description
Required (Read-only) `Public Property Required As Byte`	This property is True if the component is a required value for the screen.
RightLabel (Read-only) `Public Property RightLabel As GuiVComponent`	This label has been defined in ABAP Screen Painter to be the right label of the control.

1.2.65 GuiTitlebar Object

Description

The titlebar is only displayed and exposed as a separate object in New Visual Design mode. GuiTitlebar extends theGuiVContainer Object [page 268]. The type prefix and name of GuiTitlebar are titl.

Remarks

In some transactions the titlebar may contain objects of GuiGosShell type.

Methods

Method

Syntax	Description
All methods of the GuiVComponent Object [page 263]: • DumpState • SetFocus • Visualize	
All methods of the GuiContainer Object [page 83]: • FindById	

Method	
Syntax	**Description**
All additional methods of the GuiVContainer Object [page 268]: • FindAllByName • FindAllByNameEx • FindByName • FindByNameEx	

Properties

Property	
Syntax	**Description**
All properties of the GuiComponent Object [page 78]: • ContainerType • Id • Name • Parent • Type • TypeAsNumber	

Property	
Syntax	**Description**
All properties of the GuiVComponent Object [page 263]: • AccLabelCollection • AccText • AccTextOnRequest • AccTooltip • Changeable • DefaultTooltip • Height • IconName • IsSymbolFont • Left • Modified • ParentFrame • ScreenLeft • ScreenTop • Text • Tooltip • Top • Width	
All properties of the GuiContainer Object [page 83]: • Children	

1.2.66 GuiToolbar Object

Description

Every GuiFrameWindow has a GuiToolbar. The GuiMainWindow has two toolbars unless the second has been turned off by the ABAP application. The upper toolbar is the system toolbar, while the second toolbar is the application toolbar.

The children of a GuiToolbar are buttons. The indexes for toolbar buttons are determined by the virtual key values defined for the button.

GuiToolbar extends the GuiVContainer Object [page 268]. The type prefix and name are tbar.

Methods

Method	
Syntax	**Description**
All methods of the GuiVComponent Object [page 263]: • DumpState • SetFocus • Visualize	
All methods of the GuiContainer Object [page 83]: • FindById	
All additional methods of the GuiVContainer Object [page 268]: • FindAllByName • FindAllByNameEx • FindByName • FindByNameEx	

Properties

Property	
Syntax	**Description**
All properties of the GuiComponent Object [page 78]: • ContainerType • Id • Name • Parent • Type • TypeAsNumber	

Property	
Syntax	**Description**
All properties of the GuiVComponent Object [page 263]: • AccLabelCollection • AccText • AccTextOnRequest • AccTooltip • Changeable • DefaultTooltip • Height • IconName • IsSymbolFont • Left • Modified • ParentFrame • ScreenLeft • ScreenTop • Text • Tooltip • Top • Width	
All properties of the GuiContainer Object [page 83]: • Children	

1.2.67 GuiToolbarControl

GuiToolbarControl extends the GuiShell Object [page 195].

Description

Methods

Method Syntax	Description
All methods of the GuiVComponent Object [page 263]: • DumpState • SetFocus • Visualize	
All methods of the GuiContainer Object [page 83]: • FindById	
All methods of the GuiVContainer Object [page 268]: • FindAllByName • FindAllByNameEx • FindByName • FindByNameEx	
All methods of the GuiShell Object [page 195]: • SelectContextMenuItem • SelectContextMenuItemByPosition • SelectContextMenuItemByText	

GetButtonChecked
`Public Function GetButtonChecked( _`
`  ByVal ButtonPos As Long _`
`) As Byte`
	Returns if the button is currently checked (pressed).

GetButtonEnabled
`Public Function GetButtonEnabled( _`
`  ByVal ButtonPos As Long _`
`) As Byte`
	Indicates if the button can be pressed.

GetButtonIcon
`Public Function GetButtonIcon( _`
`  ByVal ButtonPos As Long _`
`) As String`
	Returns the name of the icon of the specified toolbar button.

Method	
Syntax	**Description**

GetButtonId
`Public Function GetButtonId( _`
`  ByVal ButtonPos As Long _`
`) As String`
	Returns the ID of the specified toolbar button.

GetButtonText
`Public Function GetButtonText( _`
`  ByVal ButtonPos As Long _`
`) As String`
	Returns the text of the specified toolbar button.

GetButtonTooltip
`Public Function GetButtonTooltip( _`
`  ByVal ButtonPos As Long _`
`) As String`
	Returns the tooltip of the specified toolbar button.

GetButtonType
`Public Function GetButtonType( _`
`  ByVal ButtonPos As Long _`
`) As String`
	Returns the type of the specified toolbar button. Possible values are:"Button", "ButtonAndMenu", "Menu", "Separator", "Group", "CheckBox"

GetMenuItemIdFromPosition
`Public Function`
`GetMenuItemIdFromPosition( _`
`  ByVal Pos As Long _`
`) As String`
	This function returns the identifier of the menu item with index Position.

PressButton
`Public Sub PressButton( _`
`  ByVal Id As String _`
`)`
	This function emulates pressing the button with the given id.

PressContextButton
`Public Sub PressContextButton( _`
`  ByVal Id As String _`
`)`
	This function emulates pressing the context button with the given id.

SelectMenuItem
`Public Sub SelectMenuItem( _`
`  ByVal Id As String _`
`)`
	This function emulates selecting the menu item with the given id.

Method

Syntax	Description

SelectMenuItemByText
`Public Sub SelectMenuItemByText( _`
`    ByVal strText As String _`
`)`
	This function emulates selecting the menu item by menu item text.

Properties

Property

Syntax	Description
All properties of the GuiComponent Object [page 78]: • ContainerType • Id • Name • Parent • Type • TypeAsNumber	
All properties of the GuiVComponent Object [page 263]: • AccLabelCollection • AccText • AccTextOnRequest • AccTooltip • Changeable • DefaultTooltip • Height • IconName • IsSymbolFont • Left • Modified • ParentFrame • ScreenLeft • ScreenTop • Text • Tooltip • Top • Width	

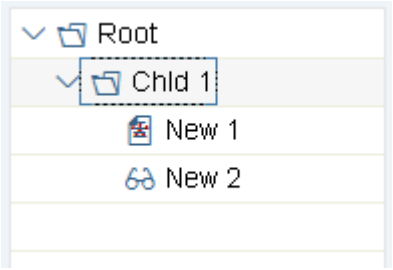
Property Syntax	Description
All properties of the GuiContainer Object [page 83]: • Children	
All additional properties of the GuiShell Object [page 195]: • AccDescription • CurrentContextMenu • DragDropSupported • Handle • OcxEvents • SubType	
ButtonCount `Public Property ButtonCount As Long`	The number of toolbar buttons including separators.
FocusedButton `Public Property FocusedButton As Long`	Zero-based index of the button that currently has the focus.

1.2.68 GuiTree Object

Example

The Tree Control supports three tree types:

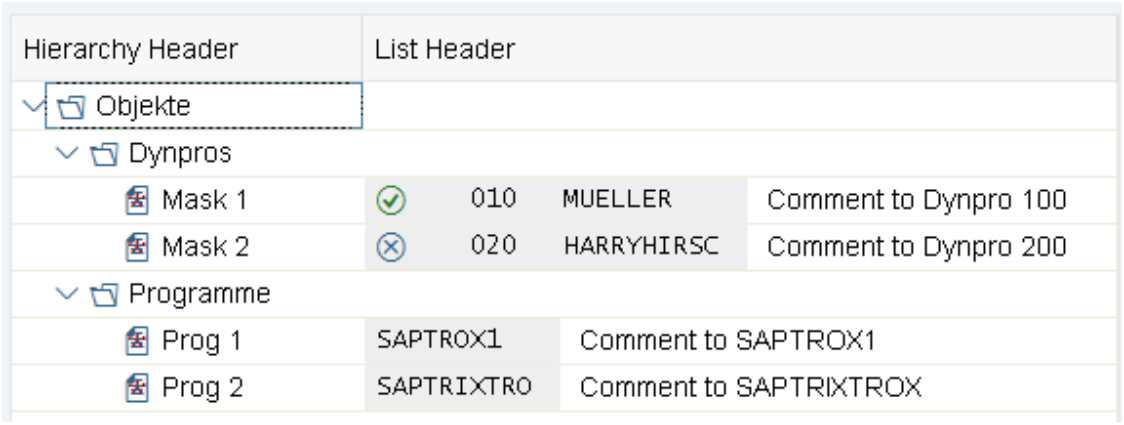
- Simple Tree



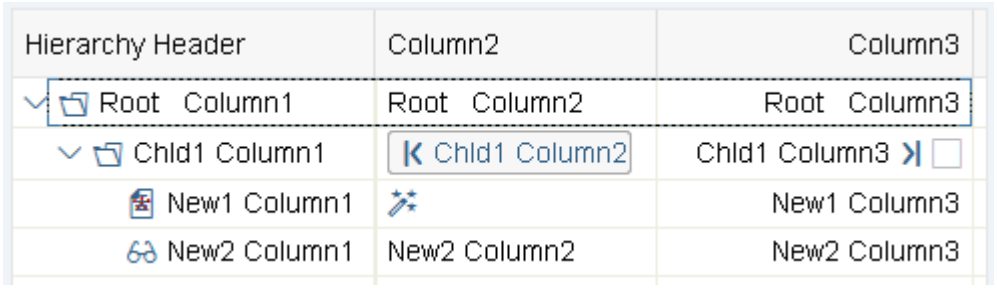
- List Tree
  - without header



- with header



- Column Tree



Methods

Method	
Syntax	**Description**
All methods of the GuiVComponent Object [page 263]: • DumpState • SetFocus • Visualize	

Method	
Syntax	**Description**
All methods of the GuiContainer Object [page 83]: • FindById	
All methods of the GuiVContainer Object [page 268]: • FindAllByName • FindAllByNameEx • FindByName • FindByNameEx	
All methods of the GuiShell Object [page 195]: • SelectContextMenuItem • SelectContextMenuItemByPosition • SelectContextMenuItemByText	

ChangeCheckbox
`Public Sub ChangeCheckbox( _`
`  ByVal NodeKey As String, _`
`  ByVal ItemName As String, _`
`  ByVal Checked As Boolean _`
`)`
	This method emulates changing a checkbox state.

ClickLink
`Public Sub ClickLink( _`
`  ByVal NodeKey As String, _`
`  ByVal ItemName As String _`
`)`
	This function emulates triggering a link.

CollapseNode
`Public Sub CollapseNode( _`
`  ByVal NodeKey As String _`
`)`
	This function closes the node with the key nodeKey.

DefaultContextMenu `Public Sub DefaultContextMenu()`	This method requests a context menu for the whole Tree Control.

DoubleClickItem
`Public Sub DoubleClickItem( _`
`  ByVal NodeKey As String, _`
`  ByVal ItemName As String _`
`)`
	This function emulates double-clicking on a text item.

Method	
Syntax	**Description**

DoubleClickNode
`Public Sub DoubleClickNode( _`
`  ByVal NodeKey As String _ )`
	This function emulates double-clicking a node.

EnsureVisibleHorizontalItem
`Public Sub`
`EnsureVisibleHorizontalItem( _`
`  ByVal NodeKey As String, _`
`  ByVal ItemName As String _`
`)`
	This function scrolls the Tree horizontally until the Item is visible.

ExpandNode
`Public Sub ExpandNode( _`
`  ByVal NodeKey As String _`
`)`
	This function expands the node with the key nodeKey.

FindNodeKeyByPath
`Public Function FindNodeKeyByPath( _`
`  ByVal Path As String _`
`) As String`	

GetAbapImage
`Public Function GetAbapImage( _`
`  ByVal Key As String, _`
`  ByVal Name As String _`
`) As String`	

GetAllNodeKeys `Public Function GetAllNodeKeys() As Object`	

GetCheckBoxState
`Public Function GetCheckBoxState( _`
`  ByVal NodeKey As String, _`
`  ByVal ItemName As String _`
`) As Byte`
	Retrieves the CheckBox state (1 = Checked, 0 = Unchecked).

GetColumnCol
`Public Function GetColumnCol( _`
`  ByVal colName As String _`
`) As Object`
	The keys of all the items in the given column.

Method	
Syntax	**Description**
GetColumnHeaders `Public Function GetColumnHeaders() As Object`	Collection of the titles of the columns.
GetColumnIndexFromName `Public Function GetColumnIndexFromName( _ ByVal Key As String _ ) As Long`	Returns the column index (starting with 1) of the column specified by the parameter.
GetColumnNames `Public Function GetColumnNames() As Object`	Returns a collection of the column names.
GetColumnTitleFromName `Public Function GetColumnTitleFromName( _ ByVal Key As String _ ) As String`	Returns the column title of the column specified by the parameter.
GetColumnTitles `Public Function GetColumnTitles() As Object`	
GetFocusedNodeKey `Public Function GetFocusedNodeKey() As String`	Returns the key of the node that has focus.
GetHierarchyLevel `Public Function GetHierarchyLevel( _ ByVal Key As String _ ) As Long`	Returns the hierarchy level of the specified key starting on level 0.
GetHierarchyTitle `Public Function GetHierarchyTitle() As String`	
GetIsDisabled `Public Function GetIsDisabled( _ ByVal NodeKey As String, _ ByVal ItemName As String _ ) As Byte`	

Method	
Syntax	**Description**

GetIsHighLighted
`Public Function GetIsHighLighted( _`
`  ByVal NodeKey As String, _`
`  ByVal ItemName As String _`
`) As Byte`	

GetItemHeight
`Public Function GetItemHeight( _`
`  ByVal NodeKey As String, _`
`  ByVal ItemName As String _`
`) As Long`
	Retrieves the height of the specified item in pixels.

GetItemLeft
`Public Function GetItemLeft( _`
`  ByVal NodeKey As String, _`
`  ByVal ItemName As String _`
`) As Long`
	Retrieves the left position of the specified item in pixels.

GetItemStyle
`Public Function GetItemStyle( _`
`  ByVal NodeKey As String, _`
`  ByVal ItemName As String _`
`) As Long`	

GetItemText
`Public Function GetItemText( _`
`  ByVal Key As String, _`
`  ByVal Name As String _`
`) As String`
	This function returns the text of the item specified by the key and name parameters.

GetItemTextColor
`Public Function GetItemTextColor( _`
`  ByVal Key As String, _`
`  ByVal Name As String _`
`) As ULong`
	Retrieves the font color of the specified item.

GetItemToolTip
`Public Function GetItemToolTip( _`
`  ByVal Key As String, _`
`  ByVal Name As String _`
`) As String`
	Retrieves the tooltip of the specified item.

Method	
Syntax	**Description**

GetItemTop
`Public Function GetItemTop( _`
`  ByVal NodeKey As String, _`
`  ByVal ItemName As String _`
`) As Long`
	Retrieves the top position of the specified item in pixels.

GetItemType
`Public Function GetItemType( _`
`  ByVal Key As String, _`
`  ByVal Name As String _`
`) As Long`
	Retrieves the column tree item type:
	• trvTreeStructureHierarchy = 0
	• trvTreeStructureImage = 1
	• trvTreeStructureText = 2
	• trvTreeStructureBool = 3
	• trvTreeStructureButton = 4
	• trvTreeStructureLink = 5

GetItemWidth
`Public Function GetItemWidth( _`
`  ByVal NodeKey As String, _`
`  ByVal ItemName As String _`
`) As Long`
	Retrieves the width of the specified item in pixels.

GetListTreeNodeItemCount
`Public Function`
`GetListTreeNodeItemCount( _`
`  ByVal NodeKey As String _`
`) As Long`
	Returns the number of visible items of the specified node for a list tree.

GetNextNodeKey
`Public Function GetNextNodeKey( _`
`  ByVal NodeKey As String _`
`) As String`
	Returns the key of the next node belonging to the same node one level above.

GetNodeAbapImage
`Public Function GetNodeAbapImage( _`
`  ByVal Key As String _`
`) As String`	

GetNodeChildrenCount
`Public Function`
`GetNodeChildrenCount( _`
`  ByVal Key As String _`
`) As Long`
	Returns the number of visible direct children of the specified node.

Method

Syntax	Description
GetNodeChildrenCountByPath `Public Function GetNodeChildrenCountByPath( _ ByVal Path As String _ ) As Long`	This function returns the number of visible children of the node given by the path parameter.
GetNodeHeight `Public Function GetNodeHeight( _ ByVal Key As String _ ) As Long`	Returns the height of the specified node in pixels.
GetNodeIndex `Public Function GetNodeIndex( _ ByVal Key As String _ ) As Long`	Returns the index of the specified key within its node.
GetNodeItemHeaders `Public Function GetNodeItemHeaders( _ ByVal NodeKey As String _ ) As Object`	
GetNodeKeyByPath `Public Function GetNodeKeyByPath( _ ByVal Path As String _ ) As String`	Key of the node specified by the given path description.
GetNodeLeft `Public Function GetNodeLeft( _ ByVal Key As String _ ) As Long`	Returns the left position of the specified node in pixels.
GetNodePathByKey `Public Function GetNodePathByKey( _ ByVal Key As String _ ) As String`	Given a node key, the path is retrieved (e.g. 2\1\2).
GetNodesCol `Public Function GetNodesCol() As Object`	The collection contains the node keys of all the nodes in the tree.

Method	
Syntax	**Description**

GetNodeStyle
`Public Function GetNodeStyle( _`
`  ByVal NodeKey As String _`
`) As Long`	

GetNodeTextByKey
`Public Function GetNodeTextByKey( _`
`  ByVal Path As String _`
`) As String`
	This function returns the text of the node specified by the given key.

GetNodeTextByPath
`Public Function GetNodeTextByPath( _`
`  ByVal Path As String _`
`) As String`
	The text of a node defined by the given path is returned.

GetNodeTextColor
`Public Function GetNodeTextColor( _`
`  ByVal Key As String _`
`) As ULong`
	Returns the font color of the specified node.

GetNodeToolTip
`Public Function GetNodeToolTip( _`
`  ByVal NodeKey As String _`
`) As String`
	Returns the tooltip of the specified node.

GetNodeTop
`Public Function GetNodeTop( _`
`  ByVal Key As String _`
`) As Long`
	Returns the top position of the specified node in pixels.

GetNodeWidth
`Public Function GetNodeWidth( _`
`  ByVal Key As String _`
`) As Long`
	Returns the width of the specified node in pixels.

GetParent
`Public Function GetParent( _`
`  ByVal CKey As String _`
`) As String`
	Key of the parent node of the node specified by the given key.

Method	
Syntax	**Description**

GetPreviousNodeKey
`Public Function GetPreviousNodeKey( _`
`  ByVal NodeKey As String _`
`) As String`
	Returns the key of the previous node belonging to the same node one level above.

GetSelectedNodes `Public Function GetSelectedNodes() As Object`	
GetSelectionMode `Public Function GetSelectionMode() As Integer`	The selection behaviour of a Tree Control instance is set once at the time of creation. **Return Type** • 0: Single Node • 1: Multiple Node • 2: Single Item • 3: Multiple Item

GetStyleDescription
`Public Function GetStyleDescription( _`
`  ByVal nStyle As Long _`
`) As String`	

GetSubNodesCol
`Public Function GetSubNodesCol( _`
`  ByVal Path As String _`
`) As Object`
	Collection of the keys of all subnodes of the node specified by the given key.

GetTreeType `Public Function GetTreeType() As Long`	The returned number has the following meaning: • 0 : Simple tree • 1 : List tree • 2 : Column tree

HeaderContextMenu
`Public Sub HeaderContextMenu( _`
`  ByVal HeaderName As String _ )`
	This method requests a context menu for a header.

IsFolder
`Public Function IsFolder( _`
`  ByVal NodeKey As String _`
`) As Byte`
	Returns True if the specified object is a node and not a leaf.

Method	
Syntax	**Description**

IsFolderExpandable
`Public Function IsFolderExpandable( _`
`  ByVal NodeKey As String _`
`) As Byte`
	Returns True if the folder belonging to the specified node can be expanded.

IsFolderExpanded
`Public Function IsFolderExpanded( _`
`  ByVal NodeKey As String _`
`) As Byte`
	Returns True if the folder belonging to the specified node is expanded.

ItemContextMenu
`Public Sub ItemContextMenu( _`
`  ByVal NodeKey As String, _`
`  ByVal ItemName As String _`
`)`
	This method requests a context menu for an item.

NodeContextMenu
`Public Sub NodeContextMenu( _`
`  ByVal NodeKey As String _`
`)`
	This method requests a context menu for a node.

PressButton
`Public Sub PressButton( _`
`  ByVal NodeKey As String, _`
`  ByVal ItemName As String _`
`)`
	This method emulates pressing a button.

PressHeader
`Public Sub PressHeader( _`
`  ByVal HeaderName As String _`
`)`
	This method emulates clicking a header.

PressKey
`Public Sub PressKey( _`
`  ByVal Key As String _`
`)`
	This method emulates pressing a key.

SelectColumn
`Public Sub SelectColumn( _`
`  ByVal ColumnName As String _`
`)`
	This function adds a column to the column selection. A node or item selection is removed.

Method	
Syntax	**Description**
SelectedItemColumn `Public Function SelectedItemColumn() As String`	The name of the column of the selected item.
SelectedItemNode `Public Function SelectedItemNode() As String`	The node key of the selected item.

SelectItem
`Public Sub SelectItem( _`
`  ByVal NodeKey As String, _`
`  ByVal ItemName As String _`
`)`
	This function emulates the selection of an item. This selection removes all other selections.

SelectNode
`Public Sub SelectNode( _`
`  ByVal NodeKey As String _`
`)`
	The node with the key nodeKey is added to the Node Selection.

SetCheckBoxState
`Public Sub SetCheckBoxState( _`
`  ByVal NodeKey As String, _`
`  ByVal ItemName As String, _`
`  ByVal state As Long _`
`)`
	This method checks or unchecks the checkbox in the specified cell of the tree control (if parameter "state" equals 0 the checkbox is unchecked, if the parameter equals 1 the checkbox is checked.

SetColumnWidth
`Public Sub SetColumnWidth( _`
`  ByVal ColumnName As String, _`
`  ByVal Width As Long _`
`)`
	This function sets the width of a column in pixels.

UnselectAll `Public Sub UnselectAll()`	All selections are removed.

UnselectColumn
`Public Sub UnselectColumn( _`
`  ByVal ColumnName As String _`
`)`
	This function removes a column from the column selection.

Method

Syntax	Description

UnselectNode
`Public Sub UnselectNode( _`
`    ByVal NodeKey As String _`
`)`
	The node with the key nodeKey is removed from the Node Selection.

Properties

Property

Syntax	Description
All properties of the GuiComponent Object [page 78]: • ContainerType • Id • Name • Parent • Type • TypeAsNumber	
All properties of the GuiVComponent Object [page 263]: • AccLabelCollection • AccText • AccTextOnRequest • AccTooltip • Changeable • DefaultTooltip • Height • IconName • IsSymbolFont • Left • Modified • ParentFrame • ScreenLeft • ScreenTop • Text • Tooltip • Top • Width	

Property Syntax	Description
All properties of the GuiContainer Object [page 83]: • Children	
All additional properties of the GuiShell Object [page 195]: • AccDescription • CurrentContextMenu • DragDropSupported • Handle • OcxEvents • SubType	
ColumnOrder (Read-write) `Public Property ColumnOrder As Object`	The property is used for working with a sequence of columns. The name of each column in the Column Tree must occur exactly once. **Remarks** Preconditions: Tree is a Column Tree and the Column Order can be changed.
HierarchyHeaderWidth (Read-write) `Public Property HierarchyHeaderWidth As Long`	The width of the Hierarchy Header in pixels. **Remarks** Precondition: Tree is a Column Tree or a List Tree with Header
SelectedNode (Read-write) `Public Property SelectedNode As String`	This is the key of the currently selected node. Selecting a node removes other selections (that is Column Selection and Item Selection). **Remarks** Precondition: Node Selection Mode is SingleNodeSelection
TopNode (Read-write) `Public Property TopNode As String`	This property influences the vertical scrolling of the Tree Control. TopNode contains the key of the node that is located on the upper edge of the Tree Control. Setting a node x as top node is only possible if there are enough visible nodes below x to fill the display area of the Tree Control.

1.2.69 GuiUserArea Object

Description

The GuiUserArea comprises the area between the toolbar and status bar for windows of GuiMainWindow type and the area between the titlebar and toolbar for modal windows, and may also be limited by docker controls. The standard dynpro elements can be found only in this area, with the exception of buttons, which are also found in the toolbars.

GuiUserArea extends the GuiVContainer Object [page 268]. The type prefix and name are usr.

Methods

Method Syntax	Description
All methods of the GuiVComponent Object [page 263]: • DumpState • SetFocus • Visualize	
All methods of the GuiContainer Object [page 83]: • FindById	
All methods of the GuiVContainer Object [page 268]: • FindAllByName • FindAllByNameEx • FindByName • FindByNameEx	
All methods of the GuiShell Object [page 195]: • SelectContextMenuItem • SelectContextMenuItemByPosition • SelectContextMenuItemByText	

FindByLabel
`Public Function FindByLabel( _`
`  ByVal Text As String, _`
`  ByVal Type As String _`
`) As GuiComponent`
	A very simple method for finding an object is to search by specifying the text of the respective label.

Properties

Property Syntax	Description
All properties of the GuiComponent Object [page 78]: • ContainerType • Id • Name • Parent • Type • TypeAsNumber	
All properties of the GuiVComponent Object [page 263]: • AccLabelCollection • AccText • AccTextOnRequest • AccTooltip • Changeable • DefaultTooltip • Height • IconName • IsSymbolFont • Left • Modified • ParentFrame • ScreenLeft • ScreenTop • Text • Tooltip • Top • Width	
All properties of the GuiContainer Object [page 83]: • Children	
CurrentContextMenu (Read-only) `Public Property CurrentContextMenu As GuiContextMenu`	This property is only set when a context menu is available at the shell object.
HorizontalScrollbar (Read-only) `Public Property HorizontalScrollbar As GuiScrollbar`	The user area is defined to be scrollable even if the scrollbars are not always visible.

Property

Syntax	Description
IsOTFPreview (Read-only) `Public Property IsOTFPreview As Byte`	
VerticalScrollbar (Read-only) `Public Property VerticalScrollbar As GuiScrollbar`	The user area is defined to be scrollable even if the scrollbars are not always visible.

1.2.70 GuiUtils Object

Methods

Method

Syntax	Description

CloseFile
`Public Sub CloseFile( _`
`  ByVal File As Long _`
`)`
	This function closes a file that was opened using OpenFile.

OpenFile
`Public Function OpenFile( _`
`  ByVal Name As String _`
`) As Long`
	The file will be created in the SAP GUI Documents Folder.
	Return Type
	The return value is a handle to the file.
	Name: Name of the text file to be created. For security reasons this name must not contain any path information.

Method

Syntax	Description

ShowMessageBox
`Public Function ShowMessageBox( _`
`  ByVal Title As String, _`
`  ByVal Text As String, _`
`  ByVal MsgIcon As Long, _`
`  ByVal MsgType As Long _`
`) As Long`
	Shows a message box.
	Return Type
	The return value will be one of the GuiMessageBoxResult constants.
	Title: Title of the message box
	Text: Text of the message box.
	MsgIcon: MsgIcon sets the icon to be used for the message box and should be set to one of the GuiMessageBoxType constants.
	MsgType: MsgType sets the buttons available on the message box and should be set to one of the GuiMessageBoxOption constants.

Write
`Public Sub Write( _`
`  ByVal File As Long, _`
`  ByVal Text`
`  As String _`
`)`
	Write text to an open file without a new line at the end.

WriteLine
`Public Sub WriteLine( _`
`  ByVal File As Long, _`
`  ByVal Text`
`  As String _`
`)`
	Write text to an open file with a new line at the end.

Properties

Property

Syntax	Description
MESSAGE_OPTION_OK (Read-only) `Public Property MESSAGE_OPTION_OK As Long`	Belongs to GuiMessageBoxOption: The message box will show an "OK" button.

Property

Syntax	Description
MESSAGE_OPTION_OKCANCEL (Read-only) `Public Property MESSAGE_OPTION_OKCANCEL As Long`	Belongs to GuiMessageBoxOption: The message box will show an "OK" and a "Cancel" button.
MESSAGE_OPTION_YESNO (Read-only) `Public Property MESSAGE_OPTION_YESNO As Long`	Belongs to GuiMessageBoxOption: The message box will show a "Yes" and a "No" button.
MESSAGE_RESULT_CANCEL (Read-only) `Public Property MESSAGE_RESULT_CANCEL As Long`	Belongs to GuiMessageBoxResult: The message box was closed via the "Cancel" button.
MESSAGE_RESULT_NO (Read-only) `Public Property MESSAGE_RESULT_NO As Long`	Belongs to GuiMessageBoxResult: The message box was closed via the "No" button.
MESSAGE_RESULT_OK (Read-only) `Public Property MESSAGE_RESULT_OK As Long`	Belongs to GuiMessageBoxResult: The message box was closed via the "OK" button.
MESSAGE_RESULT_YES (Read-only) `Public Property MESSAGE_RESULT_YES As Long`	Belongs to GuiMessageBoxResult: The message box was closed via the "Yes" button.
MESSAGE_TYPE_ERROR (Read-only) `Public Property MESSAGE_TYPE_ERROR As Long`	Belongs to GuiMessageBoxType; The message box shows the respective icon for an error message.
MESSAGE_TYPE_INFORMATION (Read-only) `Public Property MESSAGE_TYPE_INFORMATION As Long`	Belongs to GuiMessageBoxType; The message box shows the respective icon for an information message.
MESSAGE_TYPE_PLAIN (Read-only) `Public Property MESSAGE_TYPE_PLAIN As Long`	Belongs to GuiMessageBoxType; The message box shows no icon.
MESSAGE_TYPE_QUESTION (Read-only) `Public Property MESSAGE_TYPE_QUESTION As Long`	Belongs to GuiMessageBoxType; The message box shows the respective icon for a question.

Property

Syntax	Description
MESSAGE_TYPE_WARNING (Read-only) `Public Property MESSAGE_TYPE_WARNING As Long`	Belongs to GuiMessageBoxType; The message box shows the respective icon for a warning message.

1.2.71 GuiVComponent Object

The GuiVComponent interface is exposed by all visual objects, such as windows, buttons or text fields. Like GuiComponent, it is an abstract interface. Any object supporting the GuiVComponent interface also exposes the GuiComponent interface. GuiVComponent extends the GuiComponent Object [page 78].

Methods

Method	
Syntax	**Description**

DumpState
`Public Function DumpState( _`
`  ByVal InnerObject As String _`
`) As GuiCollection`
	This function dumps the state of the object. The parameter innerObject may be used to specify for which internal object the data should be dumped. Only the most complex components, such as the GuiCtrlGridView, support this parameter. All other components always dump their full state. All components that support this parameter have in common that they return general information about the control's state if the parameter "innerObject" contains an empty string. The available values for the innerObject parameter are specified as part of the class description for those components that support it.
	
	i Note
	The **DumpState** method returns a hierarchy of collections of type GuiCollection, which is three levels deep.
	• The top (first) level collection contains a second level collection for every property that is to be dumped.
	• The second level collection contains the complete information for one property. There is a third level collection for every sub-expression that might be required to access inner objects.
	• Finally, the third level collection contains the OpCode, the property or method name, the parameter values and depending on the OpCode the return value to be checked.
	The following OpCodes are used:
	• GPR: Get property and compare return value.
	• MR: Execute method and compare return value.
	• GP: Get property and execute the next entry in the second level collection on the result.
	• M: Execute the method and then execute the next entry in the second level collection on the result.
	For example the calls
	`control.ItemCount = 42`
	`control.GetItemValue(3, 2) =`
	`'MyText'`
	`control.GetItem('2','3').Property1.`
	`MethodY('XYZ').Text = 'ABC'`
	result in three entries of the top level collection:

Method Syntax	Description

	• First entry:
	  ○ OpCode Name Parameter1/
	  ○ Property-Value
	  ○ Parameter2 Parameter3
	  ○ GPR ItemCount 42
	• Second entry:
	  ○ OpCode Name Parameter1 Parameter2 Parameter3/
	  ○ Property-Value
	  ○ MR GetItemValue 3 2 MyText
	• Third entry:
	  ○ OpCode Name Parameter1 Parameter2 Parameter3
	  ○ M GetItem 2 3
	  ○ GP Property1
	  ○ M MethodY XYZ
	  ○ GPR Text ABC
	As you can see in this example, for calls that contain return values (MR, GPR) the last value in the third level collection is the return value.

SetFocus `Public Sub SetFocus()`	This function can be used to set the focus onto an object. If a user interacts with SAP GUI, it moves the focus whenever the interaction is with a new object. Interacting with an object through the scripting component does not change the focus. There are some cases in which the SAP application explicitly checks for the focus and behaves differently depending on the focused object.

Visualize
`Public Function Visualize( _`
`  ByVal On As Boolean, _`
`  Optional ByVal InnerObject As Variant _`
`) As Byte`
	Calling this method of a component will display a red frame around the specified component if the parameter on is true. The frame will be removed if on is false. Some components such as GuiCtrlGridView support displaying the frame around inner objects, such as cells. The format of the innerObject string is the same as for the dumpState method.

Properties

Property Syntax	Description
All Properties of GuiComponent Object [page 78]: • ContainerType • Id • Name • Parent • Type • TypeAsNumber	
AccLabelCollection `Public Property AccLabelCollection As GuiComponentCollection`	The collection contains objects of type GuiLabel that were assigned to this control in the ABAP Screen Painter.
AccText (Read-only) `Public Property AccText As String`	An additional text for accessibility support.
AccTextOnRequest (Read-only) `Public Property AccTextOnRequest As String`	An additional text for accessibility support.
AccTooltip (Read-only) `Public Property AccTooltip As String`	An additional tooltip text for accessibility support.
Changeable (Read-only) `Public Property Changeable As Byte`	An object is changeable if it is neither disabled nor read-only.
DefaultTooltip (Read-only) `Public Property DefaultTooltip As String`	Tooltip text generated from the short text defined in the data dictionary for the given screen element type.
Height (Read-only) `Public Property Height As Long`	Height of the component in pixels.
IconName (Read-only) `Public Property IconName As String`	If the object has been assigned an icon, then this property is the name of the icon, otherwise it is an empty string.

Property

Syntax	Description
IsSymbolFont (Read-only) `Public Property IsSymbolFont As Byte`	The property is TRUE if the component's text is visualized in the SAP symbol font.
Left (Read-only) `Public Property Left As Long`	Left position of the element in screen coordinates
Modified `Public Property Modified As Byte`	An object is modified if its state has been changed by the user and this change has not yet been sent to the SAP system.
ParentFrame (Read-only) `Public Property ParentFrame As GuiComponent`	If the control is hosted by the Frame object, the value of the property is this frame. Overwise NULL.
ScreenLeft (Read-only) `Public Property ScreenLeft As Long`	The y position of the component in screen coordinates.
ScreenTop (Read-only) `Public Property ScreenTop As Long`	The x position of the component in screen coordinates.
Text (Read-write) `Public Property Text As String`	The value of this property very much depends on the type of the object on which it is called. This is obvious for text fields or menu items. On the other hand this property is empty for toolbar buttons and is the class id for shells. You can read the text property of a label, but you can't change it, whereas you can only set the text property of a password field, but not read it.
Tooltip (Read-only) `Public Property Tooltip As String`	The tooltip contains a text which is designed to help a user understand the meaning of a given text field or button.
Top `Public Property Top As Long`	Top coordinate of the element in screen coordinates.
Width (Read-only) `Public Property Width As Long`	Width of the component in pixels.

1.2.72 GuiVContainer Object

Description

An object exposes the GuiVContainer interface if it is both visible and can have children. It will then also expose GuiComponent and GuiVComponent. Examples of this interface are windows and subscreens, toolbars or controls having children, such as the splitter control. GuiVContainer extends the GuiContainer Object [page 83] and the GuiVComponent Object [page 263].

Methods

Method Syntax	Description
All methods of the GuiVComponent Object [page 263]: • DumpState • SetFocus • Visualize	
All methods of the GuiContainer Object [page 83]: • FindById	

FindAllByName
`Public Function FindAllByName( _`
`  ByVal Name As String, _`
`  ByVal Type As String _`
`) As GuiComponentCollection`
	The methods **FindByName** and **FindByNameEx** return only the first object with matching name and type. There may however be several matching objects, which will be returned as members of a collection when **FindAllByName** or **FindAllByNameEx** are used.

FindAllByNameEx
`Public Function FindAllByNameEx( _`
`  ByVal Name As String, _`
`  ByVal Type As Long _`
`) As GuiComponentCollection`
	The methods **FindByName** and **FindByNameEx** return only the first object with matching name and type. There may however be several matching objects, which will be returned as members of a collection when **FindAllByName** or **FindAllByNameEx** are used.

Method

Syntax	Description

FindByName
`Public Function FindByName( _`
`  ByVal Name As String, _`
`  ByVal Type As String _`
`) As GuiComponent`
	Unlike **FindById**, this function does not guarantee a unique result. It will simply return the first descendant matching both the name and type parameters.This is a more natural description of the object than the complex id, but it only makes sense on dynpro objects as most other objects do not have a meaningful name. If no descendant with matching name and type can be found, the function raises an exception.

FindByNameEx
`Public Function FindAllByName( _`
`  ByVal Name As String, _`
`  ByVal Type As String _`
`) As GuiComponentCollection`
	The methods **FindByName** and **FindByNameEx** return only the first object with matching name and type. There may however be several matching objects, which will be returned as members of a collection when **FindAllByName** or **FindAllByNameEx** are used.

Properties

Property

Syntax	Description
All properties of the GuiComponent Object [page 78]: • ContainerType • Id • Name • Parent • Type • TypeAsNumber	

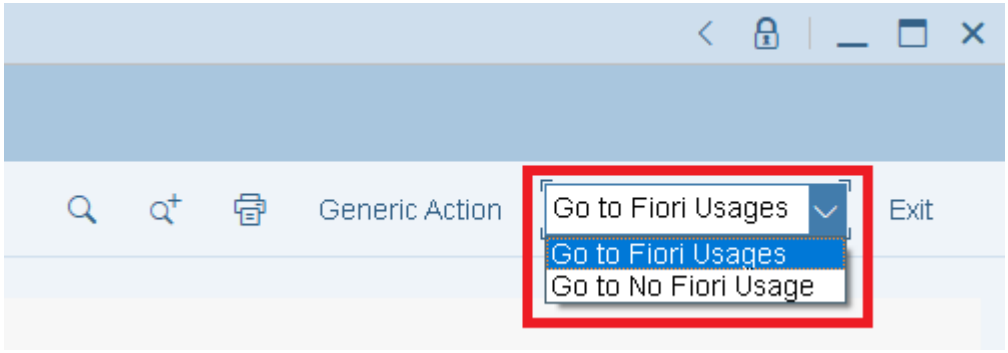
Property	
Syntax	**Description**
All properties of the GuiVComponent Object [page 263]: • AccLabelCollection • AccText • AccTextOnRequest • AccTooltip • Changeable • DefaultTooltip • Height • IconName • IsSymbolFont • Left • Modified • ParentFrame • ScreenLeft • ScreenTop • Text • Tooltip • Top • Width	
All properties of the GuiContainer Object [page 83]: • Children	

1.2.73 GuiVHViewSwitch Object

Description

GuiVHViewSwitch represents the “View Switch” object that was introduced with the Belize theme in SAP GUI. The View Switch is placed in the header area of the SAP GUI main window and can be used to select different

views within an application. Many screens can be displayed in different ways (for example, as a tree or list). To switch from one view to another in a comfortable way, these screens may make use of the View Switch:



GuiVHViewSwitch is very similar to GuiOkCodeField Object [page 164] and extends the GuiVComponent Object [page 263]. The name of the GuiVHViewSwitch object is always **`vhviewswitch`** and only one object of this type can exist at the same time.

> **i Note**
>
> - GuiVHViewSwitch exists as of SAP GUI for Windows 7.60 (the UI object itself was introduced in SAP GUI for Windows 7.50, but the extension of the Scripting API is done for SAP GUI for Windows 7.60 and newer SAP GUI versions, only)
> - Objects of type GuiVHViewSwitch can only exist when SAP GUI is running with a Fiori theme like Belize
> - GuiVHViewSwitch does not offer an entry collection. For compatibility reasons the entries of a GuiVHViewSwitch are still GuiButtons which belong to the application toolbar (tbar1).

Methods

Method	Description
All methods of the GuiVComponent Object [page 263]: • DumpState • SetFocus • Visualize	

Properties

Property	
Syntax	**Description**
All properties of the GuiComponent Object [page 78]: • ContainerType • Id • Name • Parent • Type • TypeAsNumber	
All properties of the GuiVComponent Object [page 263]: • AccLabelCollection • AccText • AccTextOnRequest • AccTooltip • Changeable • DefaultTooltip • Height • IconName • IsSymbolFont • Left • Modified • ParentFrame • ScreenLeft • ScreenTop • Text • Tooltip • Top • Width	

1.3 Events

Event Syntax	Description

AbapScriptingEvent
`Public Event AbapScriptingEvent( _`
`  ByVal param As String _`
`)`	

Activated
`Public Event Activated( _`
`  ByVal Session As GuiSession _`
`)`	

AutomationFCode
`Public Event AutomationFCode( _`
`  ByVal Session As GuiSession, _`
`  ByVal FunctionCode As String _`
`)`
	The event is only fired when using the SAP Workplace. It notifies the listener that SAP GUI executes a function code that was set by the Workplace framework.

Change
`Public Event Change( _`
`  ByVal Session As GuiSession, _`
`  ByVal Component As GuiComponent, _`
`  ByVal CommandArray As Variant _`
`)`
	In record mode, the session collects changes to elements of the user interface and sends these changes to a listening application whenever server communication is about to start or if the record mode is turned off. The **Change** events are raised immediately before the **StartRequest** event. There is at least one event for every modified element in the recorded session.
	see also: Change Event - Additional Remarks [page 276]

ContextMenu
`Public Event ContextMenu( _`
`  ByVal Session As GuiSession, _`
`  ByVal Component As GuiVComponent _`
`)`
	The **ContextMenu** event is fired when SAP GUI is about to display a context menu. There are currently the following limitations:
	• Only context menus of controls of type GuiShell are supported.
	• The event is not fired for “cached” context menus, which are not retrieved from the server when being opened.

Event	
Syntax	**Description**

CreateSession
`Public Event CreateSession( _`
`  ByVal Session As GuiSession _`
`)`
	This event is raised whenever a new session is created, irrespective of whether of the session being created manually, from ABAP or by a script. The event is only raised for a session if the scripting support has been enabled for the corresponding backend.
	Example
	Sample Code
	`Dim objSapGui`
	`Set objSapGui = GetObject("SAPGUI")`
	`Dim objScriptingEngine`
	`Set objScriptingEngine =`
	`  objSapGui.GetScriptingEngine`
	`WScript.ConnectObject`
	`objScriptingEngine,  "Engine_"`
	`Dim Waiting`
	`Waiting = 1`
	`Do While (Waiting = 1)`
	`WScript.Sleep(100)`
	`Loop`
	`Set objScriptingEngine = Nothing`
	`Set objSapGui = Nothing`
	`Sub Engine_CreateSession(ByVal`
	`Session)`
	`Dim result`
	`result = MsgBox("Session`
	`  created", vbOKCancel)`
	`If result = vbCancel then`
	`  Waiting = 0`
	`End If`
	`End Sub`

Destroy
`Public Event Destroy( _`
`  ByVal Session As GuiSession _`
`)`
	This event is raised before a session is destroyed.

DestroySession
`Public Event DestroySession( _`
`  ByVal Session As GuiSession _`
`)`
	This event is raised before a session is destroyed . This can be done either by closing the main window manually, or by calling the **closeSession** method of GuiConnection.

EndRequest
`Public Event EndRequest( _`
`  ByVal Session*Session* As GuiSession _`
`)`
	endRequest is called immediately after the session is unlocked after server communication.

Event Syntax	Description

Error (GuiSession Object [page 181])
`Public Event Error( _`
`  ByVal Session As GuiSession, _`
`  ByVal ErrorId As Long, _`
`  ByVal Desc1 As String, _`
`  ByVal Desc2 As String, _`
`  ByVal Desc3 As String, _`
`  ByVal Desc4 As String _`
`)`
	An **Error** event is currently only raised, if the wrapper library required to access a SAP GUI ActiveX control from a script is not available. error events from all sessions are also available at the GuiApplication.

Error (GuiApplication Object [page 37])
`Public Event Error( _`
`  ByVal ErrorId As Long, _`
`  ByVal Desc1 As String, _`
`  ByVal Desc2 As String, _`
`  ByVal Desc3 As String, _`
`  ByVal Desc4 As String _`
`)`
	An **Error** event is currently only raised, if the wrapper library required to access a SAP GUI ActiveX control from a script is not available. This event is also available on the GuiSession in which the error occurred.

FocusChanged
`Public Event FocusChanged( _`
`  ByVal Session As GuiSession, _`
`  ByVal NewFocusedControl As GuiVComponent _`
`)`	

HistoryOpened
`Public Event HistoryOpened( _`
`  ByVal Session As GuiSession, _`
`  ByVal NewFocusedControl As GuiVComponent _`
`)`	

Hit
`Public Event Hit( _`
`  ByVal Session*Session* As GuiSession, _`
`  ByVal Component As GuiComponent, _`
`  ByVal InnerObject As String _`
`)`
	The **Hit** event is only raised when elementVisualizationMode is set to True, which turns on the hit test mode of SAP GUI. If in this mode a SAP GUI component is identified, the **Hit** event is raised. The parameters of this event are
	• The session on which the component was hit
	• The component that was hit
	• A description of an inner object of the component if an inner object was hit

IgnoreSession
`Public Event IgnoreSession( _`
`  ByVal SessionMainWindowHandle As Integer _`
`)`
	The event is fired when a session is set to 'Ignored' using IgnoreSession function. This event is only fired when using SAP GUI Scripting while running eCATT in parallel.

Event	
Syntax	**Description**

ProgressIndicator
`Public Event ProgressIndicator( _`
`  ByVal percentage As Long, _`
`  ByVal Text As String _`
`)`	

StartRequest
`Public Event StartRequest( _`
`  ByVal Session As GuiSession _`
`)`
	The **startRequest** event is raised before the session is locked during server communication. At this point user input can be checked before it is sent to the server. It is not possible to prevent server communication from this event.

1.3.1 Change Event - Additional Remarks

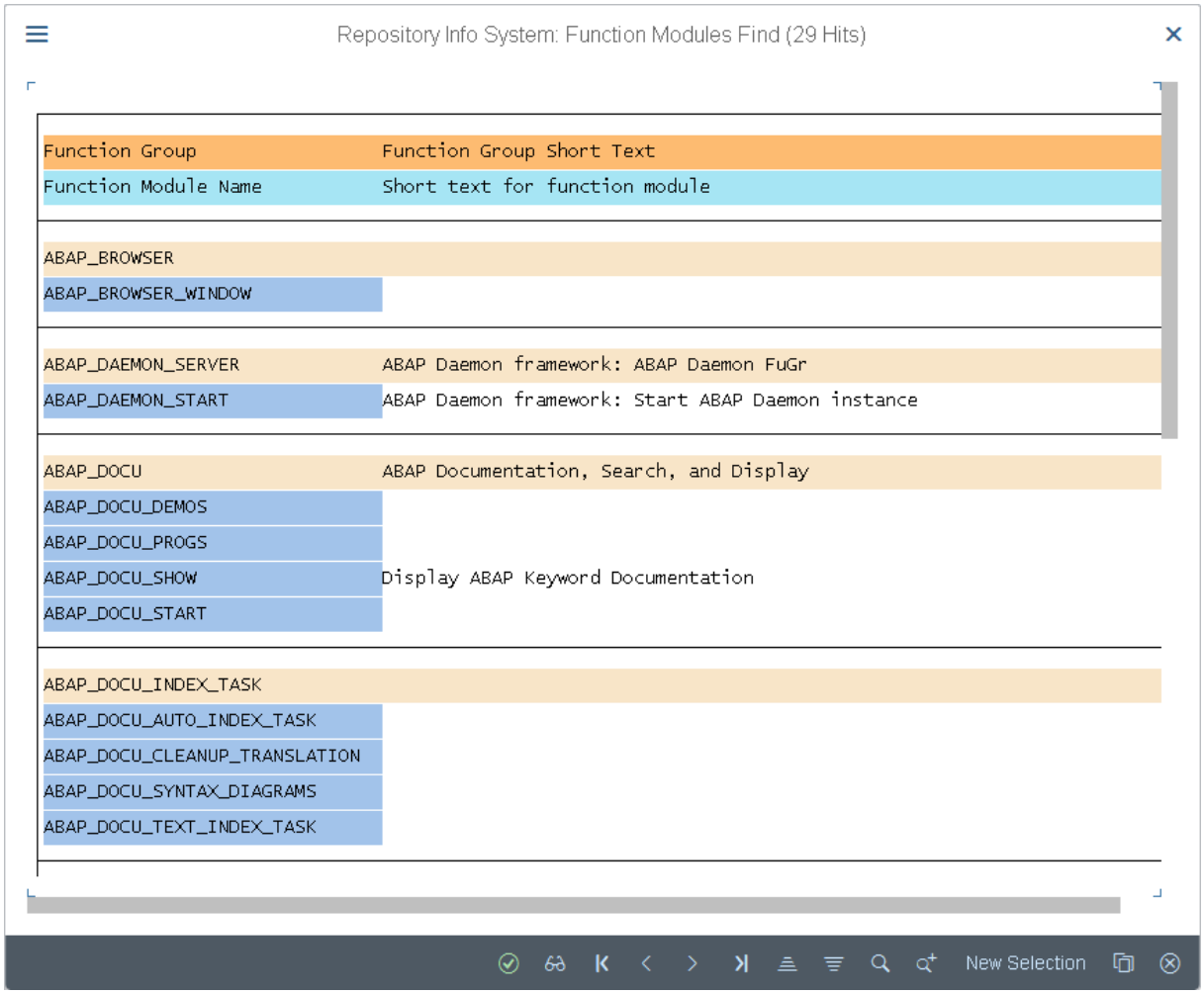
Only changes made at the SAP GUI level are recorded. Transactions may preset some of the entry fields with values from parameters stored in the SAP system. If these data are not changed in SAP GUI, they will not be recorded. This may cause problems during playback of scripts, if the entry fields are preset with different values.

If any of the following techniques is used in a transaction, the user should manually modify all the entries he wants to see recorded:

- Usage of SAP parameters
- Variants
- Hold Data, from the menu System -> User Profile

Playback of the changes will only work, if the order of the calls is the same as during recording.

Each event represents one line of script code. The Component parameter specifies the object on which to invoke a method or property. Therefore the first thing to record is Component.id for later use with **FindById**. The recorder may however also decide to record other properties of Component. If, for example, a line in a table control or list is selected, it may be prudent not to record the position of the line, but rather the values in it. That way, a script can be generated that is more robust with respect to changes in the number, and therefore in the position, of lines.



If new function modules have been added, selecting a line from the list might return the wrong function module.

Type	Method/Property name	Parameters
"SP"	**Text**	"Hello World"

This sets the parameter Text to value “Hello World”.

Type	Method/Property name	Parameters
"SP"	**RecordMode**	True

This sets the parameter RecordMode to the Boolean value True. It is up to the recorder to generate a script line with a valid textual representation of Boolean values, such as “true”, “True” or “TRUE” for example.

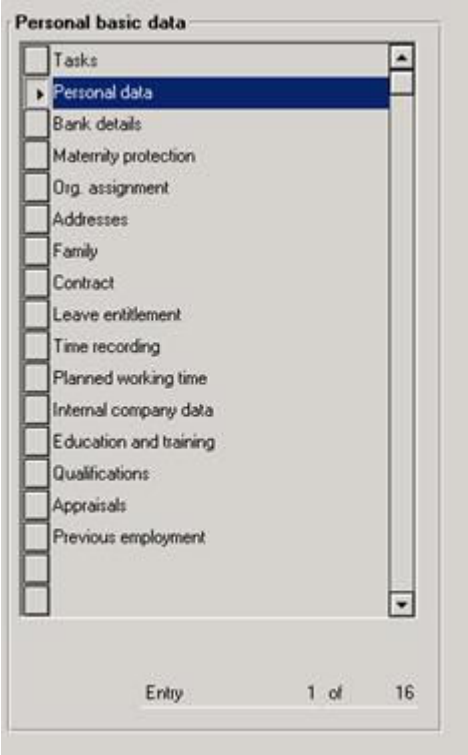
Type	Method/Property name	Parameters
"SP"	**TestToolMode**	0

This sets the parameter TestToolMode to value 0.

Type	Method/Property name	Parameters
"M"	`Resize`	96
		32
		False

The method `Resize` is called with three parameters. In this case the third member of the CommandArray is an array with 3 elements.

There are cases in which the CommandArray contains more than one line.



If a row is selected in this table control, two entries are added to the generated `Change` event's CommandArray parameter.

Type	Method/Property name	Parameters
"M"	`getAbsoluteRow`	1
"SP"	`selected`	True

The script code required to select this line should then look like this:

Sample Code

```
Session.
findById("wnd[0]/usr/tblSAPMBIBSTC537").
getAbsoluteRow("1").
```

```
selected = "True"
```

1.4 Enumerations

1.4.1 GuiComponentType

Members

Member	Value
GuiApplication	10
GuiBox	62
GuiButton	40
GuiCheckBox	42
GuiCollection	120
GuiComboBox	34
GuiComponent	0
GuiComponentCollection	128
GuiConnection	11
GuiContainer	70
GuiContainerShell	51
GuiContextMenu	127
GuiCTextField	32
GuiCustomControl	50
GuiDialogShell	125
GuiDockShell	126
GuiFrameWindow	20
GuiGOSShell	123
GuiLabel	30
GuiListContainer	73
GuiMainWindow	21
GuiMenu	110
GuiMenubar	111

Member	Value
GuiMessageWindow	23
GuiModalWindow	22
GuiOkCodeField	35
GuiPasswordField	33
GuiRadioButton	41
GuiScrollbar	100
GuiScrollContainer	72
GuiSession	12
GuiSessionInfo	121
GuiShell	122
GuiSimpleContainer	71
GuiSplitterContainer	75
GuiSplitterShell	124
GuiStatusbar	103
GuiStatusPane	43
GuiTab	91
GuiTableColumn	81
GuiTableControl	80
GuiTableRow	82
GuiTabStrip	90
GuiTextField	31
GuiTitlebar	102
GuiToolbar	101
GuiUnknown	-1
GuiUserArea	74
GuiVComponent	1
GuiVContainer	2
GuiVHViewSwitch	129

1.4.2 GuiErrorType

Members

Member	Value	Description
Gui_Err_AccessDenied	633	Access denied.
Gui_Err_Bad_Focus	634	Can not set focus to this object.
Gui_Err_Bad_Index_Type	618	Bad index type for collection access.
Gui_Err_Control_Label	615	The control could not be found by label.
Gui_Err_Control_Name	608	The control could not be found by name.
Gui_Err_Control_Position	616	The control could not be found by position.
Gui_Err_Disconnected	621	The object invoked has disconnected from its clients.
Gui_Err_Enumerator_Index	614	The enumerator of the collection cannot find an element with the specified index.
Gui_Err_Enumerator_Reset	612	The enumerator of the collection cannot be reset.
Gui_Err_FindById	619	The control could not be found by id.
Gui_Err_FindByName	620	The control could not be found by name.
Gui_Err_FindByPos	632	The control could not be found by position.
Gui_Err_Front_Module	602	The path of the 'sapfront.dll' could not be determined.
Gui_Err_Init	601	Sapgui engine cannot be initialized.
Gui_Err_Int_Get_Session_Failed	629	Can not get session from TLS
Gui_Err_Int_GetCtrl_Failed	625	Could not get ctrl (Internal Error)
Gui_Err_Int_GetFocusManFromSes_Failed	627	Could not get focus manager from session (Internal Error)
Gui_Err_Int_GetSesFromCtrl_Failed	626	Could not get session from ctrl (Internal Error)
Gui_Err_Int_Invalid_TestToolMode	628	Invalid test tool mode
Gui_Err_Int_View_Not_Set	630	View not set (Internal Error)
Gui_Err_Invalid_Argument	613	The method got an invalid argument.
Gui_Err_Invalid_Context	603	Function called in invalid thread context
Gui_Err_Invalid_Window	611	The required window is invalid.
Gui_Err_Logon_Module	604	The 'Sapgui Logon Component' could not be instantiated.
Gui_Err_Menu_Disabled	623	The menu item is disabled.
Gui_Err_No_Memory	607	The system is out of memory.
Gui_Err_No_Wrapper	622	No wrapper available for this control.
Gui_Err_Not_Implemented	610	The method or property is currently not implemented.
Gui_Err_Permission_Denied	637	Permission denied.
Gui_Err_Property_Readonly	609	The property is readonly.
Gui_Err_Resize_Failed	631	Resize failed.

Member	Value	Description
Gui_Err_Sapgui_Module	605	The 'Sapgui Component' could not be instantiated.
Gui_Err_Save_Image	635	Error saving image.
Gui_Err_Scripting_Disabled_Srv	624	Scripting is disabled by the server.
Gui_Err_Session_Index	606	The session index is out of range.
Gui_Err_Shortcut_Evaluation	636	Shortcut evaluation failed.
Gui_Err_SL_No_Entry	1000	Not a valid SAPLogon entry
Gui_Err_VKey_Disabled	617	The virtual key is not enabled.

1.4.3 GuiEventType

Members

Member	Value	Description
SapApplicationCreateSessionEvent	2002	ApplicationCreateSession
SapApplicationDestroySessionEvent	2003	ApplicationDestroySession
SapApplicationErrorEvent	2004	ApplicationErrorEvent
SapApplicationIgnoreSessionEvent	2005	ApplicationIgnoreSession
SapContextMenuEvent	1282	ContextMenu
SapCustomDataChangedEvent	1280	CustomDataChanged
SapDefaultEvent	0	Default
SapHitSelectEvent	1281	Hit
SapSessionAbapScriptingEvent	1289	AbapScriptingEvent
SapSessionActivatedEvent	1285	SessionActivated
SapSessionAutoFCodeEvent	1284	SessionAutoFCode
SapSessionDestroyEvent	1283	SessionDestroy
SapSessionEndRequestEvent	515	SessionEndRequest
SapSessionErrorEvent	516	SessionError
SapSessionFocusChangedEvent	1286	SessionFocusChanged
SapSessionHistoryOpenedEvent	1287	SessionHistoryOpened
SapSessionProgressIndicatorEvent	1288	ProgressIndicatorOpened
SapSessionStartRequestEvent	514	SessionStartRequest

1.4.4 GuiImageType

Members

Member	Value
BMP	0
GIF	2
JPEG	1
PNG	2

1.4.5 GuiMagicDisplIDs

Members

Member	Value
GuiDisplDBTPress	32200
GuiDisplDCBChecked	32011
GuiDisplDCBCurListBoxEntry	32305
GuiDisplDCBEntries	32302
GuiDisplDCBEntryKey	33800
GuiDisplDCBEntryPos	33802
GuiDisplDCBEntryValue	33801
GuiDisplDCBIsListBoxActive	32304
GuiDisplDCBKey	32300
GuiDisplDCBKeySpace	32303
GuiDisplDCBShowKey	32306
GuiDisplDCBValue	32301
GuiDisplDCollAdd	33103
GuiDisplDCollCount	33100
GuiDisplDCollElAt	33102
GuiDisplDCollLength	33101
GuiDisplDConConnString	33003
GuiDisplDConDescription	33002
GuiDisplDConDisabled	33001
GuiDisplDConnClose	32831

Member	Value
GuiDispIDConSessions	33000
GuiDispIDCTFindAllByName	32035
GuiDispIDCTFindAllByNameEx	32036
GuiDispIDCTFindById	32029
GuiDispIDCTFindByLabel	32027
GuiDispIDCTFindByName	32026
GuiDispIDCTFindByNameEx	32034
GuiDispIDCTFindByPosition	32028
GuiDispIDDockerIsVertical	34301
GuiDispIDDockerPixelSize	34300
GuiDispIDEngAddHist	32913
GuiDispIDEngButtonB	32903
GuiDispIDEngCon	32900
GuiDispIDEngConnErr	32924
GuiDispIDEngCrColl	32911
GuiDispIDEngDropHist	32914
GuiDispIDEngGetEng	1
GuiDispIDEngHistEnabled	32916
GuiDispIDEngIgnore	32908
GuiDispIDEngInplace	32907
GuiDispIDEngMajor	32909
GuiDispIDEngMinor	32910
GuiDispIDEngNoSysMsg	32925
GuiDispIDEngOpenCon	32905
GuiDispIDEngOpenConEx	32918
GuiDispIDEngOpenWDCon	32926
GuiDispIDEngPatchlevel	32919
GuiDispIDEngQuit	32906
GuiDispIDEngRegister	32921
GuiDispIDEngRevision	32920
GuiDispIDEngRevoke	32923
GuiDispIDEngStatusB	32902
GuiDispIDEngTheme	32912
GuiDispIDEngTitleB	32904

Member	Value
GuiDispIDEngToolB	32901
GuiDispIDEngUtils	32917
GuiDispIDEngWDSessions	32927
GuiDispIDErrDesc1	33601
GuiDispIDErrDesc2	33602
GuiDispIDErrDesc3	33603
GuiDispIDErrDesc4	33604
GuiDispIDErrNo	33600
GuiDispIDGActiveSession	32049
GuiDispIDGActiveSession2	32075
GuiDispIDGCAccDescription	33703
GuiDispIDGCAccLabelCol	32043
GuiDispIDGCAccText	32044
GuiDispIDGCAccTextOnReq	32045
GuiDispIDGCAccTooltip	32042
GuiDispIDGCChangeable	32009
GuiDispIDGCCharHeight	32073
GuiDispIDGCCharLeft	32070
GuiDispIDGCCharTop	32071
GuiDispIDGCCharWidth	32072
GuiDispIDGCChildren	32019
GuiDispIDGCClass	32017
GuiDispIDGCColorIndex	32058
GuiDispIDGCColorIntensified	32059
GuiDispIDGCColorInverse	32060
GuiDispIDGCCtxMnu	33701
GuiDispIDGCDefaultTooltip	32069
GuiDispIDGCDisplayedText	32074
GuiDispIDGCDragDrop	33706
GuiDispIDGCDumpState	31194
GuiDispIDGCFlushing	33704
GuiDispIDGCHeight	32006
GuiDispIDGCHwnd	33702
GuiDispIDGCIcon	32037

Member	Value
GuiDispIDGCId	32025
GuiDispIDGCIsContainer	32033
GuiDispIDGCIsHotspot	32051
GuiDispIDGCIsList	32052
GuiDispIDGCIsStepLoop	32062
GuiDispIDGCIsSymbolFont	32061
GuiDispIDGCLeft	32003
GuiDispIDGCLeftLabel	32040
GuiDispIDGCLoopCurrentCol	32065
GuiDispIDGCLoopCurrentRow	32066
GuiDispIDGCLoopHeight	32064
GuiDispIDGCLoopWidth	32063
GuiDispIDGCModified	32030
GuiDispIDGCName	32001
GuiDispIDGCOcxEvents	33705
GuiDispIDGCParent	32038
GuiDispIDGCParentFrame	32050
GuiDispIDGCRightLabel	32041
GuiDispIDGCRowText	32053
GuiDispIDGCScreenLeft	32046
GuiDispIDGCScreenTop	32047
GuiDispIDGCSession	32018
GuiDispIDGCSetFocus	32024
GuiDispIDGCShortId	32031
GuiDispIDGCShowContextMenu	32068
GuiDispIDGCSubType	33700
GuiDispIDGCText	32000
GuiDispIDGCTitle	32048
GuiDispIDGCTooltip	32008
GuiDispIDGCTop	32004
GuiDispIDGCType	32015
GuiDispIDGCTypeAsNum	32032
GuiDispIDGCVisualize	32039
GuiDispIDGCWidth	32005

Member	Value
GuiDispIDGECATTReplay	32076
GuiDispIDGetAbsoluteRow	33407
GuiDispIDGMSelect	33300
GuiDispIDGMWFocusedButton	32433
GuiDispIDGMWHelpButtonHelpText	32440
GuiDispIDGMWHelpButtonText	32435
GuiDispIDGMWMessageText	32437
GuiDispIDGMWMessageType	32436
GuiDispIDGMWOKButtonHelpText	32439
GuiDispIDGMWOKButtonText	32434
GuiDispIDGMWVisible	32438
GuiDispIDGUCharHeight	32603
GuiDispIDGUCharWidth	32602
GuiDispIDGUHorizontalScrollbar	32600
GuiDispIDGUListNav	32605
GuiDispIDGUOTFPreview	32606
GuiDispIDGUResize	32604
GuiDispIDGUVerticalScrollbar	32601
GuiDispIDGWButtonB	32425
GuiDispIDGWClose	32414
GuiDispIDGWCompBitmap	32443
GuiDispIDGWGuiFocus	32422
GuiDispIDGWHandle	32420
GuiDispIDGWHardCopy	32415
GuiDispIDGWHardCopyMem	32441
GuiDispIDGWIconic	32400
GuiDispIDGWIconify	32408
GuiDispIDGWIsPopupDialog	32427
GuiDispIDGWJumpBackward	32432
GuiDispIDGWJumpForward	32431
GuiDispIDGWMaximize	32410
GuiDispIDGWMoveWindow	32407
GuiDispIDGWPopupDialogText	32428
GuiDispIDGWRestore	32409

Member	Value
GuiDispIDGWSpyMode	32413
GuiDispIDGWStatusB	32424
GuiDispIDGWSysFocus	32421
GuiDispIDGWTabBackward	32430
GuiDispIDGWTabForward	32429
GuiDispIDGWTitleB	32426
GuiDispIDGWToolB	32423
GuiDispIDGWVKAllowed	32412
GuiDispIDGWWPHeight	32417
GuiDispIDGWWPMsgBox	32419
GuiDispIDGWWPResize	32418
GuiDispIDGWWPResizeEx	32442
GuiDispIDGWWPWidth	32416
GuiDispIDIsListBoxActive	32840
GuiDispIDLCursor	32022
GuiDispIDLHighlighted	32100
GuiDispIDLIsLeftLabel	32101
GuiDispIDLIsRightLabel	32102
GuiDispIDListBoxCurrEntry	32849
GuiDispIDListBoxCurrEntryHeight	32848
GuiDispIDListBoxCurrEntryLeft	32846
GuiDispIDListBoxCurrEntryTop	32845
GuiDispIDListBoxCurrEntryWidth	32847
GuiDispIDListBoxHeight	32844
GuiDispIDListBoxLeft	32842
GuiDispIDListBoxTop	32841
GuiDispIDListBoxWidth	32843
GuiDispIDLListProperty	32103
GuiDispIDLMaxLength	32012
GuiDispIDLNumerical	32013
GuiDispIDLPassword	32016
GuiDispIDLSimpleListProperty	32104
GuiDispIDMsgAsPopup	34004
GuiDispIDMsgId	34001

Member	Value
GuiDispIDMsgNumber	34002
GuiDispIDMsgPar	34003
GuiDispIDMsgType	34000
GuiDispIDOcxCallbackChange	200889
GuiDispIDOcxCallbackHighlight	200890
GuiDispIDOcxCallbackHit	200891
GuiDispIDOcxControl	271062
GuiDispIDOcxGetRect	31192
GuiDispIDOcxHit	31195
GuiDispIDOcxHitTest	31193
GuiDispIDOcxHover	31196
GuiDispIDOcxIsReadOnlyCall	31191
GuiDispIDOcxNotify	31199
GuiDispIDOcxNotifyContEvSink	31197
GuiDispIDOcxNotifyCtrlEvent	31198
GuiDispIDOKF1	32351
GuiDispIDOKOpened	32350
GuiDispIDRBGroupColl	32504
GuiDispIDRBGroupCount	32502
GuiDispIDRBGroupPos	32503
GuiDispIDRBSelect	32501
GuiDispIDRBSelected	32500
GuiDispIDSBDblClick	32750
GuiDispIDScrollMax	33904
GuiDispIDScrollMin	33905
GuiDispIDScrollPage	33903
GuiDispIDScrollPos	33902
GuiDispIDScrollRange	33900
GuiDispIDSesActivWin	32800
GuiDispIDSesBusy	32803
GuiDispIDSesClearErrorList	32825
GuiDispIDSesClose	32811
GuiDispIDSesCmd	32805
GuiDispIDSesCmdAsync	32806

Member	Value
GuiDispIDSesCreate	32812
GuiDispIDSesEnableAccSymbols	32830
GuiDispIDSesEnableAccTabChain	32829
GuiDispIDSesEnableJaws	32828
GuiDispIDSesEndT	32810
GuiDispIDSesErrorList	32824
GuiDispIDSesFindByPos	32818
GuiDispIDSesIconDesc	33525
GuiDispIDSesInfo	32802
GuiDispIDSesInfoAppSr	33508
GuiDispIDSesInfoCl	33509
GuiDispIDSesInfoCP	33512
GuiDispIDSesInfoDisRec	33521
GuiDispIDSesInfoDynp	33506
GuiDispIDSesInfoFlush	33503
GuiDispIDSesInfoForceNot	33522
GuiDispIDSesInfoGrpN	33515
GuiDispIDSesInfoGuiCP	33523
GuiDispIDSesInfoI18N	33524
GuiDispIDSesInfoITime	33501
GuiDispIDSesInfoLang	33511
GuiDispIDSesInfoModeNo	33517
GuiDispIDSesInfoMsgSrc	33514
GuiDispIDSesInfoMsgSrv	33513
GuiDispIDSesInfoProg	33505
GuiDispIDSesInfoReadOnly	33520
GuiDispIDSesInfoRound	33504
GuiDispIDSesInfoRTime	33500
GuiDispIDSesInfoSesCtx	33518
GuiDispIDSesInfoSysN	33507
GuiDispIDSesInfoSysNo	33516
GuiDispIDSesInfoTrans	33502
GuiDispIDSesInfoUser	33510
GuiDispIDSesInfoWAN	33519

Member	Value
GuiDispIDSesIsActive	32819
GuiDispIDSesLockSessionUI	32826
GuiDispIDSesMenu	32807
GuiDispIDSesPPPSyId	32821
GuiDispIDSesPPSyId	32822
GuiDispIDSesPPTaId	32820
GuiDispIDSesProgressPercent	32832
GuiDispIDSesProgressText	32833
GuiDispIDSesRecFile	32814
GuiDispIDSesRecord	32804
GuiDispIDSesRunScrCtrl	32816
GuiDispIDSesSaveAsUnicode	32823
GuiDispIDSesShowKeys	33527
GuiDispIDSesStartT	32809
GuiDispIDSesStdNumFmt	33526
GuiDispIDSesSuppressBackendPopups	32834
GuiDispIDSesTestTool	32813
GuiDispIDSesUnlockSessionUI	32827
GuiDispIDSesVKey	32808
GuiDispIDSesVKeyDesc	32817
GuiDispIDSHSelCtxtMenIt	34100
GuiDispIDSHSelCtxtMenItPos	34102
GuiDispIDSHSelCtxtMenItTxt	34101
GuiDispIDSplitterIsVertical	34400
GuiDispIDSplitterSashPosition	34401
GuiDispIDTableBase	33400
GuiDispIDTableColBase	33420
GuiDispIDTableColFixed	33421
GuiDispIDTableColSelected	33422
GuiDispIDTableColSelectMode	33401
GuiDispIDTableColTitle	33420
GuiDispIDTableColumns	33402
GuiDispIDTableConfigureLayout	33406
GuiDispIDTableCurrentCol	33410

Member	Value
GuiDispIDTableCurrentRow	33411
GuiDispIDTableDeselAllCols	33414
GuiDispIDTableFieldName	33409
GuiDispIDTabLeftTab	33200
GuiDispIDTableGetCell	33415
GuiDispIDTableReorderTable	33405
GuiDispIDTableRowBase	33430
GuiDispIDTableRowCount	33412
GuiDispIDTableRows	33404
GuiDispIDTableRowSelectable	33431
GuiDispIDTableRowSelected	33430
GuiDispIDTableRowSelectMode	33403
GuiDispIDTableSelAllCols	33408
GuiDispIDTableVisRowCount	33413
GuiDispIDTabSelTab	33201
GuiDispIDTBSelect	32700
GuiDispIDTBToLeft	32701
GuiDispIDTHistoryCurEntry	32057
GuiDispIDTHistoryCurIndex	32056
GuiDispIDTHistoryIsActive	32054
GuiDispIDTHistoryList	32055
GuiDispIDTIsOField	32067
GuiDispIDTRequired	32014
GuiDispIDUtilCloseFile	34202
GuiDispIDUtilMsgBox	34200
GuiDispIDUtilMsgOptOK	34220
GuiDispIDUtilMsgOptOKCan	34222
GuiDispIDUtilMsgOptYesNo	34221
GuiDispIDUtilMsgResCancel	34230
GuiDispIDUtilMsgResNo	34233
GuiDispIDUtilMsgResOK	34231
GuiDispIDUtilMsgResYes	34232
GuiDispIDUtilMsgTypeE	34208
GuiDispIDUtilMsgTypeI	34205

Member	Value
GuiDispIDUtilMsgTypeP	34209
GuiDispIDUtilMsgTypeQ	34206
GuiDispIDUtilMsgTypeW	34207
GuiDispIDUtilOpenFile	34201
GuiDispIDUtilWriteFile	34203
GuiDispIDUtilWriteLnFile	34204

1.4.6 GuiMessageBoxOption

Members

Member	Value	Description
MSG_OPTION_OK	0	Constant value to be used when calling the showMessageBox method. Using this value will display an 'OK' button only. (0)
MSG_OPTION_OKCANCEL	2	Constant value to be used when calling the showMessageBox method. Using this value will display an 'OK' button and a 'Cancel' button. (2)
MSG_OPTION_YESNO	1	Constant value to be used when calling the showMessageBox method. Using this value will display a 'Yes' button and a 'No' button. (1)

1.4.7 GuiMessageBoxResult

Members

Member	Value	Description
MSG_RESULT_CANCEL	0	Constant value to be used as a return value by the showMessageBox method. This value is returned when the 'Cancel' button has been pressed. (0)
MSG_RESULT_NO	3	Constant value to be used as a return value by the showMessageBox method. This value is returned when the 'No' button has been pressed. (3)

Member	Value	Description
MSG_RESULT_OK	1	Constant value to be used as a return value by the showMessageBox method. This value is returned when the 'OK' button has been pressed. (1)
MSG_RESULT_YES	2	Constant value to be used as a return value by the showMessageBox method. This value is returned when the 'Yes' button has been pressed. (2)

1.4.8 GuiMessageBoxType

Members

Member	Value	Description
MSG_TYPE_ERROR	3	Constant value to be used when calling the showMessageBox method. Using this value will display a stop sign as the message box icon. (3)
MSG_TYPE_INFORMATION	0	Constant value to be used when calling the showMessageBox method. Using this value will display the letter 'i' as the message box icon. (0)
MSG_TYPE_PLAIN	4	Constant value to be used when calling the showMessageBox method. Using this value will display no message box icon. (4)
MSG_TYPE_QUESTION	1	Constant value to be used when calling the showMessageBox method. Using this value will display a question mark as the message box icon. (1)
MSG_TYPE_WARNING	2	Constant value to be used when calling the showMessageBox method. Using this value will display an exclamation mark as the message box icon. (2)

1.4.9 GuiScrollbarType

Members

Member	Value
GuiScrollbarTypeHorizontal	2
GuiScrollbarTypeUnknown	0
GuiScrollbarTypeVertical	1

1.4.10 GuiTableSelectionType

Members

Member	Value	Description
MULTIPLE_INTERVAL_SELECTION	2	Several columns/rows can be selected. (2)
NO_SELECTION	0	No selection possible. (0)
SINGLE_SELECTION	1	One column/row can be selected. (1)

2 SAP GUI Scripting ROT Entry Helper

Description

This is a helper library that contains just one object. It is used to allow COM clients to access a running SAP GUI process.

Remarks

Some scripting languages do not allow accessing an entry in the Running Object Table. Even though they may offer a 'GetObject' function, this function will then only support accessing files in the file system or creating COM objects using the COM progid. In this case you need to start by creating another helper object, CSapROTWrapper, which in turn allows you to access the ROT entry for SAP GUI.

2.1 SapGuiAuto Object

Description

This object is registered in the Windows Running Object Table whenever the SAP GUI main process saplogon.exe runs, assuming that SAP GUI Scripting is installed on the client PC.

Methods

Method	Syntax Visual Basic

GetScriptingEngine
This method returns the GuiApplication COM interface for the currently running SAP GUI process.
Example
Visual Basic Script (Visual Basic)
`Set SapGuiAuto = GetObject( "SAPGUI") Set application = SapGuiAuto.GetScriptingEngine`
WinBatch by Wilson WindowWare (Visual Basic)
`SapGuiAuto = ObjectGet("SAPGUI"); Application = SapGuiAuto.GetScriptingEngine;`
	`Public Function GetScriptingEngine() As Object`
	Return Type: **GuiApplication**

3 SAP GUI Scripting ROT Access Helper

Description

This library contains just one object. It is required in cases where the scripting engine does not allow accessing entries in the Running Object Table. This is for example the case for the following engines

- JAWS script, the scripting language of the JAWS Screenreader, by Freedom Scientific
- AutoIt, a freeware Windows automation language

For these languages you need to create a CSapROTWrapper object first, before you can access the ROT entry and get the GuiApplication interface of the running SAP GUI process.

3.1 CSapROTWrapper Object

Methods

Method	Syntax Visual Basic	Parameter

GetROTEntry
This method returns the ROT entry of SAP GUI.
Example
Accessing the GuiApplication interface for the running SAP GUI process from AutoIt
AutoIt (Visual Basic)
`Dim $Wrapper = ObjCreate ( "SapROTWr.SapROTWrapper") Dim $RotSAPGUI = $Wrapper.GetROTEntry ( "SAPGUI") Dim $Application = $RotSAPGUI.GetScriptingEngine()`
Accessing the GuiApplication interface for the running SAP GUI process from JAWS
JAWS Script (Visual Basic)
`Object Wrapper, Object RotSAPGUI, Object SAPGUI Let Wrapper = CreateObject ( "SapROTWr.SapROTWrapper") Let RotSAPGUI = Wrapper.GetROTEntry ( "SAPGUI") Let SAPGUI = RotSAPGUI.GetScriptingEngine`
Accessing the GuiSession interface for the running SAP GUI process from a Yahoo widget
Yahoo Widget (Visual Basic)
`var Wrapper = COM.createObject("SapROTWr.SapROTWrapper"); var SapGuiAuto = Wrapper.GetROTEntry("SAPGUI"); var GuiApplication = SapGuiAuto.GetScriptingEngine(); var GuiConnection = GuiApplication.OpenConnection("My System"); var AllSessions = GuiConnection.Children; var GuiSession = AllSessions.ElementAt(0);`
	`Public Function GetROTEntry( _ ByVal strDisplayName As String _ ) As Object`	*strDisplayName*

Method	Syntax Visual Basic	Parameter
	`alert( "Session: " + GuiSession.Id);`	

Important Disclaimers and Legal Information

Hyperlinks

Some links are classified by an icon and/or a mouseover text. These links provide additional information.
About the icons:

- Links with the icon  : You are entering a Web site that is not hosted by SAP. By using such links, you agree (unless expressly stated otherwise in your agreements with SAP) to this:
  - The content of the linked-to site is not SAP documentation. You may not infer any product claims against SAP based on this information.
  - SAP does not agree or disagree with the content on the linked-to site, nor does SAP warrant the availability and correctness. SAP shall not be liable for any damages caused by the use of such content unless damages have been caused by SAP's gross negligence or willful misconduct.
- Links with the icon : You are leaving the documentation for that particular SAP product or service and are entering a SAP-hosted Web site. By using such links, you agree that (unless expressly stated otherwise in your agreements with SAP) you may not infer any product claims against SAP based on this information.

Beta and Other Experimental Features

Experimental features are not part of the officially delivered scope that SAP guarantees for future releases. This means that experimental features may be changed by SAP at any time for any reason without notice. Experimental features are not for productive use. You may not demonstrate, test, examine, evaluate or otherwise use the experimental features in a live operating environment or with data that has not been sufficiently backed up.
The purpose of experimental features is to get feedback early on, allowing customers and partners to influence the future product accordingly. By providing your feedback (e.g. in the SAP Community), you accept that intellectual property rights of the contributions or derivative works shall remain the exclusive property of SAP.

Example Code

Any software coding and/or code snippets are examples. They are not for productive use. The example code is only intended to better explain and visualize the syntax and phrasing rules. SAP does not warrant the correctness and completeness of the example code. SAP shall not be liable for errors or damages caused by the use of example code unless damages have been caused by SAP's gross negligence or willful misconduct.

Gender-Related Language

We try not to use gender-specific word forms and formulations. As appropriate for context and readability, SAP may use masculine word forms to refer to all genders.

www.sap.com/contactsap

© 2019 SAP SE or an SAP affiliate company. All rights reserved.

No part of this publication may be reproduced or transmitted in any form or for any purpose without the express permission of SAP SE or an SAP affiliate company. The information contained herein may be changed without prior notice.

Some software products marketed by SAP SE and its distributors contain proprietary software components of other software vendors. National product specifications may vary.

These materials are provided by SAP SE or an SAP affiliate company for informational purposes only, without representation or warranty of any kind, and SAP or its affiliated companies shall not be liable for errors or omissions with respect to the materials. The only warranties for SAP or SAP affiliate company products and services are those that are set forth in the express warranty statements accompanying such products and services, if any. Nothing herein should be construed as constituting an additional warranty.

SAP and other SAP products and services mentioned herein as well as their respective logos are trademarks or registered trademarks of SAP SE (or an SAP affiliate company) in Germany and other countries. All other product and service names mentioned are the trademarks of their respective companies.

Please see https://www.sap.com/about/legal/trademark.html for additional trademark information and notices.

THE BEST RUN SAP