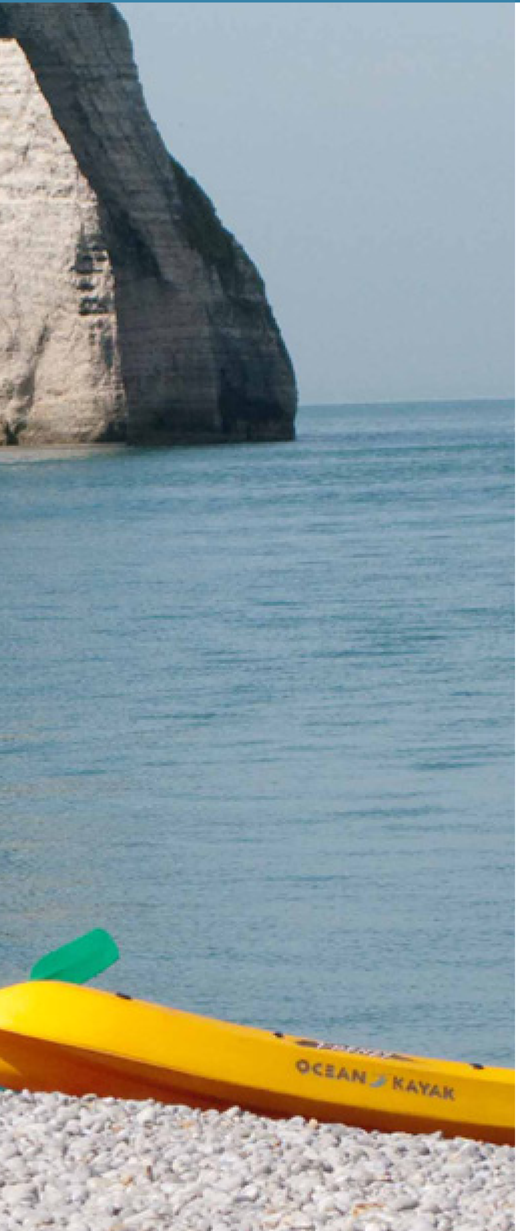
© Indiscripts.com | June 2012

What's new in InDesign CS6 Scripting DOM



1 | New Object Members

The data presented below are extracted from the Scripting DOM reference. They only include the objects and object members introduced in InDesign 8 (CS6). A full hypertext reference is already available here: http://www.jongware.com/idjshelp.html

This document has been automatically generated by parsing the Type Library files provided by Adobe from InDesign 3.0 (CS) to InDesign 8 (CS6). It may contain typos or adventious case mistakes due to the automation process.

- To easily browse the contents, use the PDF bookmarks: **View > Navigation Panels > Bookmarks**.
- A separate document focuses on new **Enumeration** values.

LEGEND:

Type	Primitive argument/value type (string, boolean, etc.)
{Type}	Any DOM object type.
#Type	Enumeration type.
•	Indicates a read-only property.

Application

Object available in ID 3, 4, 5, 6, 7, 7.5, 8

APPLICATION CS6 VALUES	PROTOTYPE	DESCRIPTION
cellStyleMappings	{CellStyleMappings} •cellStyleMappings	*A collection of cell style mappings.*
charStyleMappings	{CharStyleMappings} •charStyleMappings	*A collection of char style mappings.*
contentPlacer	{ContentPlacerObject} •contentPlacer	*The Content Placer. Used to get a reference to the content placer object.*
fontLockingPreferences	{FontLockingPreference} •fontLockingPreferences	*Font locking preference settings.*
linkedPageItemOptions	{LinkedPageItemOption} •linkedPageItemOptions	*Linked Page Item options.*
paraStyleMappings	{ParaStyleMappings} •paraStyleMappings	*A collection of para style mappings.*
pngExportPreferences	{PNGExportPreference} •pngExportPreferences	*PNG export preferences.*
selectionKeyObject	*variant* selectionKeyObject	*The key object of the selection. Type: PageItem or idNothingEnum enumerator.*
tableStyleMappings	{TableStyleMappings} •tableStyleMappings	*A collection of table style mappings.*

Button

Object available in ID 3, 4, 5, 6, 7, 7.5, 8

BUTTON CS6 VALUES	PROTOTYPE	DESCRIPTION
contentPlace	*variant* contentPlace(Objects PageItems, *boolean* LinkPageItems, *boolean* LinkStories, *boolean* MapStyles, *boolean* ShowingOptions)	*Duplicate an object and place it into the target page item. / Return value: The placed content. Type: Array of PageItems, Graphics, Stories or Texts. / PageItems: One or more page items to place or load. / LinkPageItems: Whether to link pageItems in content placer (if true it will override link stories value). / LinkStories: Whether to link stories in content placer (only applicable for single story, pageItem links will also be created in case of more than one item). / MapStyles: Whether to map styles in content placer. / ShowingOptions: Whether to display the link options dialog.*
clearFormBehaviors	{ClearFormBehaviors} •clearFormBehaviors	*A collection of clear form behavior objects.*
hiddenUntilTriggered	*boolean* hiddenUntilTriggered	*If true, the form field/push button is hidden until triggered in the exported PDF.*
horizontalLayoutConstraints	*variant* horizontalLayoutConstraints	*The left margin, width, and right margin constraints this item is subject to when using the object-based layout rule. Type: Array of 3 idDimensionsConstraints enumerators.*
linkedPageItemOptions	{LinkedPageItemOption} •linkedPageItemOptions	*Linked Page Item options.*
printableInPdf	*boolean* printableInPdf	*If true, the form field/push button is printable in the exported PDF.*
printFormBehaviors	{PrintFormBehaviors} •printFormBehaviors	*A collection of print form behavior objects.*
submitFormBehaviors	{SubmitFormBehaviors} •submitFormBehaviors	*A collection of submit form behavior objects.*
verticalLayoutConstraints	*variant* verticalLayoutConstraints	*The top margin, height, and bottom margin constraints this item is subject to when using the object-based layout rule. Type: Array of 3 idDimensionsConstraints enumerators.*

Cell

Object available in ID 3, 4, 5, 6, 7, 7.5, 8

CELL CS6 VALUES	PROTOTYPE	DESCRIPTION
checkBoxes	{CheckBoxes} •checkBoxes	*A collection of checkboxes.*
comboBoxes	{ComboBoxes} •comboBoxes	*A collection of comboboxes.*
listBoxes	{ListBoxes} •listBoxes	*A collection of listboxes.*
radioButtons	{RadioButtons} •radioButtons	*A collection of radio buttons.*
signatureFields	{SignatureFields} •signatureFields	*A collection of signature fields.*
textBoxes	{TextBoxes} •textBoxes	*A collection of text boxes.*

Character

Object available in ID 3, 4, 5, 6, 7, 7.5, 8

CHARACTER CS6 VALUES	PROTOTYPE	DESCRIPTION
checkBoxes	{CheckBoxes} •checkBoxes	*A collection of checkboxes.*
comboBoxes	{ComboBoxes} •comboBoxes	*A collection of comboboxes.*
listBoxes	{ListBoxes} •listBoxes	*A collection of listboxes.*
paragraphKashidaWidth	*double* paragraphKashidaWidth	*Paragraph kashida width. 0 is none, 1 is short, 2 is medium, 3 is long.*
radioButtons	{RadioButtons} •radioButtons	*A collection of radio buttons.*
signatureFields	{SignatureFields} •signatureFields	*A collection of signature fields.*
textBoxes	{TextBoxes} •textBoxes	*A collection of text boxes.*

Document

Object available in ID 3, 4, 5, 6, 7, 7.5, 8

DOCUMENT CS6 VALUES	PROTOTYPE	DESCRIPTION
createAlternateLayout	*void* createAlternateLayout(Objects SpreadItems, *string* Name, *variant* Width, *variant* Height, *boolean* CreateTextStyles, *boolean* LinkTextStories, #LayoutRuleOptions LayoutRule)	*Creates an alternate layout for a given list of spreads. / SpreadItems: The spreads to create an alternate layout for. / Name: The name of the alternate layout. Note: This is used for the named layout for the new section for the duplicated spreads. / Width: The width of the pages created. Type: Unit (Double or String). / Height: The height of the pages created. Type: Unit (Double or String). / CreateTextStyles: Whether to create new text styles. / LinkTextStories: Whether to create linked text for duplicated text stories. / LayoutRule: What layout rule to set on the pages.*
deleteAlternateLayout	*void* deleteAlternateLayout(*string* Name)	*Deletes an alternate layout. / Name: The name of the alternate layout to delete.*
cellStyleMappings	{CellStyleMappings} •cellStyleMappings	*A collection of cell style mappings.*
charStyleMappings	{CharStyleMappings} •charStyleMappings	*A collection of char style mappings.*

DOCUMENT CS6 VALUES	PROTOTYPE	DESCRIPTION
checkBoxes	{CheckBoxes} •checkBoxes	*A collection of checkboxes.*
comboBoxes	{ComboBoxes} •comboBoxes	*A collection of comboboxes.*
linkedPageItemOptions	{LinkedPageItemOption} •linkedPageItemOptions	*Linked Page Item options.*
listBoxes	{ListBoxes} •listBoxes	*A collection of listboxes.*
paraStyleMappings	{ParaStyleMappings} •paraStyleMappings	*A collection of para style mappings.*
radioButtons	{RadioButtons} •radioButtons	*A collection of radio buttons.*
selectionKeyObject	*variant* selectionKeyObject	*The key object of the selection. Type: PageItem or idNothingEnum enumerator.*
signatureFields	{SignatureFields} •signatureFields	*A collection of signature fields.*
tableStyleMappings	{TableStyleMappings} •tableStyleMappings	*A collection of table style mappings.*
textBoxes	{TextBoxes} •textBoxes	*A collection of text boxes.*

DocumentPreference

Object available in ID 3, 4, 5, 6, 7, 7.5, 8

DOCUMENTPREFERENCE CS6 VALUES	PROTOTYPE	DESCRIPTION
createPrimaryTextFrame	*boolean* createPrimaryTextFrame	*If true, the document A-master has primary text frames when a new document is created.*

DocumentPreset

Object available in ID 3, 4, 5, 6, 7, 7.5, 8

DOCUMENTPRESET CS6 VALUES	PROTOTYPE	DESCRIPTION
createPrimaryTextFrame	*boolean* createPrimaryTextFrame	*If true, the document A-master has primary text frames when a new document is created.*

EPS

Object available in ID 3, 4, 5, 6, 7, 7.5, 8

EPS CS6 VALUES	PROTOTYPE	DESCRIPTION
contentPlace	*variant* contentPlace(Objects PageItems, *boolean* LinkPageItems, *boolean* LinkStories, *boolean* MapStyles, *boolean* ShowingOptions)	*Duplicate an object and place it into the target page item. / Return value: The placed content. Type: Array of PageItems, Graphics, Stories or Texts. / PageItems: One or more page items to place or load. / LinkPageItems: Whether to link pageItems in content placer (if true it will override link stories value). / LinkStories: Whether to link stories in content placer (only applicable for single story, pageItem links will also be created in case of more than one item). / MapStyles: Whether to map styles in content placer. / ShowingOptions: Whether to display the link options dialog.*

EPS CS6 VALUES	PROTOTYPE	DESCRIPTION
horizontalLayoutConstraints	*variant* horizontalLayoutConstraints	*The left margin, width, and right margin constraints this item is subject to when using the object-based layout rule. Type: Array of 3 idDimensionsConstraints enumerators.*
linkedPageItemOptions	{LinkedPageItemOption} •linkedPageItemOptions	*Linked Page Item options.*
verticalLayoutConstraints	*variant* verticalLayoutConstraints	*The top margin, height, and bottom margin constraints this item is subject to when using the object-based layout rule. Type: Array of 3 idDimensionsConstraints enumerators.*

FormField

Object available in ID 3, 4, 5, 6, 7, 7.5, 8

FORMFIELD CS6 VALUES	PROTOTYPE	DESCRIPTION
contentPlace	*variant* contentPlace(Objects PageItems, *boolean* LinkPageItems, *boolean* LinkStories, *boolean* MapStyles, *boolean* ShowingOptions)	*Duplicate an object and place it into the target page item. / Return value: The placed content. Type: Array of PageItems, Graphics, Stories or Texts. / PageItems: One or more page items to place or load. / LinkPageItems: Whether to link pageItems in content placer (if true it will override link stories value). / LinkStories: Whether to link stories in content placer (only applicable for single story, pageItem links will also be created in case of more than one item). / MapStyles: Whether to map styles in content placer. / ShowingOptions: Whether to display the link options dialog.*
horizontalLayoutConstraints	*variant* horizontalLayoutConstraints	*The left margin, width, and right margin constraints this item is subject to when using the object-based layout rule. Type: Array of 3 idDimensionsConstraints enumerators.*
linkedPageItemOptions	{LinkedPageItemOption} •linkedPageItemOptions	*Linked Page Item options.*
verticalLayoutConstraints	*variant* verticalLayoutConstraints	*The top margin, height, and bottom margin constraints this item is subject to when using the object-based layout rule. Type: Array of 3 idDimensionsConstraints enumerators.*

GeneralPreference

Object available in ID 3, 4, 5, 6, 7, 7.5, 8

GENERALPREFERENCE CS6 VALUES	PROTOTYPE	DESCRIPTION
createLinksOnContentPlace	*boolean* createLinksOnContentPlace	*Enable the creation of links on content place.*
customMonitorPpi	*double* customMonitorPpi	*When using a custom monitor resolution, what is the value of that resolution in pixels per inch.*
greekVectorGraphicsOnDrag	*boolean* greekVectorGraphicsOnDrag	*Controls whether or not to greek vector graphics when dragging at high quality.*
mainMonitorPpi	*double* •mainMonitorPpi	*The value of the system reported main monitor resolution.*
mapStylesOnContentPlace	*boolean* mapStylesOnContentPlace	*Enable the mapping of styles on content place.*
previewPages	#PreviewPagesOptions previewPages	*The pages to create preview images for. Note: Valid when include preview is true.*
showConveyor	*boolean* showConveyor	*Show the conveyor on content collector or content placer tool activation.*

GENERALPREFERENCE CS6 VALUES	PROTOTYPE	DESCRIPTION
useCustomMonitorResolution	*boolean* useCustomMonitorResolution	*Enable the use of a custom monitor resolution in pixels per inch as opposed to querying the system settings.*

Graphic

Object available in ID 3, 4, 5, 6, 7, 7.5, 8

GRAPHIC CS6 VALUES	PROTOTYPE	DESCRIPTION
contentPlace	*variant* contentPlace(Objects PageItems, *boolean* LinkPageItems, *boolean* LinkStories, *boolean* MapStyles, *boolean* ShowingOptions)	*Duplicate an object and place it into the target page item. / Return value: The placed content. Type: Array of PageItems, Graphics, Stories or Texts. / PageItems: One or more page items to place or load. / LinkPageItems: Whether to link pageItems in content placer (if true it will override link stories value). / LinkStories: Whether to link stories in content placer (only applicable for single story, pageItem links will also be created in case of more than one item). / MapStyles: Whether to map styles in content placer. / ShowingOptions: Whether to display the link options dialog.*
horizontalLayoutConstraints	*variant* horizontalLayoutConstraints	*The left margin, width, and right margin constraints this item is subject to when using the object-based layout rule. Type: Array of 3 idDimensionsConstraints enumerators.*
linkedPageItemOptions	{LinkedPageItemOption} •linkedPageItemOptions	*Linked Page Item options.*
verticalLayoutConstraints	*variant* verticalLayoutConstraints	*The top margin, height, and bottom margin constraints this item is subject to when using the object-based layout rule. Type: Array of 3 idDimensionsConstraints enumerators.*

GraphicLine

Object available in ID 3, 4, 5, 6, 7, 7.5, 8

GRAPHICLINE CS6 VALUES	PROTOTYPE	DESCRIPTION
contentPlace	*variant* contentPlace(Objects PageItems, *boolean* LinkPageItems, *boolean* LinkStories, *boolean* MapStyles, *boolean* ShowingOptions)	*Duplicate an object and place it into the target page item. / Return value: The placed content. Type: Array of PageItems, Graphics, Stories or Texts. / PageItems: One or more page items to place or load. / LinkPageItems: Whether to link pageItems in content placer (if true it will override link stories value). / LinkStories: Whether to link stories in content placer (only applicable for single story, pageItem links will also be created in case of more than one item). / MapStyles: Whether to map styles in content placer. / ShowingOptions: Whether to display the link options dialog.*
checkBoxes	{CheckBoxes} •checkBoxes	*A collection of checkboxes.*
comboBoxes	{ComboBoxes} •comboBoxes	*A collection of comboboxes.*
horizontalLayoutConstraints	*variant* horizontalLayoutConstraints	*The left margin, width, and right margin constraints this item is subject to when using the object-based layout rule. Type: Array of 3 idDimensionsConstraints enumerators.*
htmlItems	{HtmlItems} •htmlItems	*A collection of embedded HTML page items.*
linkedPageItemOptions	{LinkedPageItemOption} •linkedPageItemOptions	*Linked Page Item options.*
listBoxes	{ListBoxes} •listBoxes	*A collection of listboxes.*

GRAPHICLINE CS6 VALUES	PROTOTYPE	DESCRIPTION
radioButtons	{RadioButtons} •radioButtons	*A collection of radio buttons.*
signatureFields	{SignatureFields} •signatureFields	*A collection of signature fields.*
textBoxes	{TextBoxes} •textBoxes	*A collection of text boxes.*
verticalLayoutConstraints	*variant* verticalLayoutConstraints	*The top margin, height, and bottom margin constraints this item is subject to when using the object-based layout rule. Type: Array of 3 idDimensionsConstraints enumerators.*

Group

Object available in ID 3, 4, 5, 6, 7, 7.5, 8

GROUP CS6 VALUES	PROTOTYPE	DESCRIPTION
contentPlace	*variant* contentPlace(Objects PageItems, *boolean* LinkPageItems, *boolean* LinkStories, *boolean* MapStyles, *boolean* ShowingOptions)	*Duplicate an object and place it into the target page item. / Return value: The placed content. Type: Array of PageItems, Graphics, Stories or Texts. / PageItems: One or more page items to place or load. / LinkPageItems: Whether to link pageItems in content placer (if true it will override link stories value). / LinkStories: Whether to link stories in content placer (only applicable for single story, pageItem links will also be created in case of more than one item). / MapStyles: Whether to map styles in content placer. / ShowingOptions: Whether to display the link options dialog.*
checkBoxes	{CheckBoxes} •checkBoxes	*A collection of checkboxes.*
comboBoxes	{ComboBoxes} •comboBoxes	*A collection of comboboxes.*
horizontalLayoutConstraints	*variant* horizontalLayoutConstraints	*The left margin, width, and right margin constraints this item is subject to when using the object-based layout rule. Type: Array of 3 idDimensionsConstraints enumerators.*
linkedPageItemOptions	{LinkedPageItemOption} •linkedPageItemOptions	*Linked Page Item options.*
listBoxes	{ListBoxes} •listBoxes	*A collection of listboxes.*
radioButtons	{RadioButtons} •radioButtons	*A collection of radio buttons.*
signatureFields	{SignatureFields} •signatureFields	*A collection of signature fields.*
textBoxes	{TextBoxes} •textBoxes	*A collection of text boxes.*
verticalLayoutConstraints	*variant* verticalLayoutConstraints	*The top margin, height, and bottom margin constraints this item is subject to when using the object-based layout rule. Type: Array of 3 idDimensionsConstraints enumerators.*

Guide

Object available in ID 3, 4, 5, 6, 7, 7.5, 8

GUIDE CS6 VALUES	PROTOTYPE	DESCRIPTION
resolve	*variant* resolve(*variant* Location, #CoordinateSpaces In, *boolean* ConsideringRulerUnits)	*Get the coordinates of the given location in the specified coordinate system. / Return value: The resolved location for each of the targeted page items. Type: Array of 2 Doubles or Array of Arrays of 2 Doubles. / Location: The location requested. Type: Array of 2 Doubles, idAnchorPoint enumerator or Array of Arrays of 2 Doubles, idCoordinateSpaces enumerators, idAnchorPoint enumerators, idBoundingBoxLimits enumerators or Long Integers. / In: The coordinate space to use. / ConsideringRulerUnits: If true then a ruler location is interpreted using ruler units rather than points. The default value is false. This parameter has no effect unless the reference point is specified relative to a page.*
transformValuesOf	{Objects} transformValuesOf(#CoordinateSpaces In)	*Get the transformation values of the page item. / Return value: A list of the transformation matrices. / In: The coordinate space to use.*
guideType	#GuideTypeOptions guideType	*The type of the guide.*
guideZone	*variant* guideZone	*The zone of the guide. Type: Unit (Double or String).*

Image

Object available in ID 3, 4, 5, 6, 7, 7.5, 8

IMAGE CS6 VALUES	PROTOTYPE	DESCRIPTION
contentPlace	*variant* contentPlace(Objects PageItems, *boolean* LinkPageItems, *boolean* LinkStories, *boolean* MapStyles, *boolean* ShowingOptions)	*Duplicate an object and place it into the target page item. / Return value: The placed content. Type: Array of PageItems, Graphics, Stories or Texts. / PageItems: One or more page items to place or load. / LinkPageItems: Whether to link pageItems in content placer (if true it will override link stories value). / LinkStories: Whether to link stories in content placer (only applicable for single story, pageItem links will also be created in case of more than one item). / MapStyles: Whether to map styles in content placer. / ShowingOptions: Whether to display the link options dialog.*
horizontalLayoutConstraints	*variant* horizontalLayoutConstraints	*The left margin, width, and right margin constraints this item is subject to when using the object-based layout rule. Type: Array of 3 idDimensionsConstraints enumerators.*
linkedPageItemOptions	{LinkedPageItemOption} •linkedPageItemOptions	*Linked Page Item options.*
verticalLayoutConstraints	*variant* verticalLayoutConstraints	*The top margin, height, and bottom margin constraints this item is subject to when using the object-based layout rule. Type: Array of 3 idDimensionsConstraints enumerators.*

IMEPreference

Object available in ID 3, 4, 5, 6, 7, 7.5, 8

IMEPREFERENCE CS6 VALUES	PROTOTYPE	DESCRIPTION
useNativeDigits	*boolean* useNativeDigits	*If true, use native digits for Arabic languages.*

InsertionPoint

Object available in ID 3, 4, 5, 6, 7, 7.5, 8

INSERTIONPOINT CS6 VALUES	PROTOTYPE	DESCRIPTION
checkBoxes	{CheckBoxes} •checkBoxes	*A collection of checkboxes.*
comboBoxes	{ComboBoxes} •comboBoxes	*A collection of comboboxes.*
listBoxes	{ListBoxes} •listBoxes	*A collection of listboxes.*
paragraphKashidaWidth	*double* paragraphKashidaWidth	*Paragraph kashida width. 0 is none, 1 is short, 2 is medium, 3 is long.*
radioButtons	{RadioButtons} •radioButtons	*A collection of radio buttons.*
signatureFields	{SignatureFields} •signatureFields	*A collection of signature fields.*
textBoxes	{TextBoxes} •textBoxes	*A collection of text boxes.*

Layer

Object available in ID 3, 4, 5, 6, 7, 7.5, 8

LAYER CS6 VALUES	PROTOTYPE	DESCRIPTION
checkBoxes	{CheckBoxes} •checkBoxes	*A collection of checkboxes.*
comboBoxes	{ComboBoxes} •comboBoxes	*A collection of comboboxes.*
listBoxes	{ListBoxes} •listBoxes	*A collection of listboxes.*
radioButtons	{RadioButtons} •radioButtons	*A collection of radio buttons.*
signatureFields	{SignatureFields} •signatureFields	*A collection of signature fields.*
textBoxes	{TextBoxes} •textBoxes	*A collection of text boxes.*

LayoutWindow

Object available in ID 3, 4, 5, 6, 7, 7.5, 8

LAYOUTWINDOW CS6 VALUES	PROTOTYPE	DESCRIPTION
selectionKeyObject	*variant* selectionKeyObject	*The key object of the selection. Type: PageItem or idNothingEnum enumerator.*

Line

Object available in ID 3, 4, 5, 6, 7, 7.5, 8

LINE CS6 VALUES	PROTOTYPE	DESCRIPTION
checkBoxes	{CheckBoxes} •checkBoxes	*A collection of checkboxes.*
comboBoxes	{ComboBoxes} •comboBoxes	*A collection of comboboxes.*

LINE CS6 VALUES	PROTOTYPE	DESCRIPTION
listBoxes	{ListBoxes} •listBoxes	*A collection of listboxes.*
paragraphKashidaWidth	*double* paragraphKashidaWidth	*Paragraph kashida width. 0 is none, 1 is short, 2 is medium, 3 is long.*
radioButtons	{RadioButtons} •radioButtons	*A collection of radio buttons.*
signatureFields	{SignatureFields} •signatureFields	*A collection of signature fields.*
textBoxes	{TextBoxes} •textBoxes	*A collection of text boxes.*

Link

Object available in ID 3, 4, 5, 6, 7, 7.5, 8

LINK CS6 VALUES	PROTOTYPE	DESCRIPTION
goToSource	*void* goToSource()	*Opens the source file of the link in InDesign for SharedContent links.*

MasterSpread

Object available in ID 3, 4, 5, 6, 7, 7.5, 8

MASTERSPREAD CS6 VALUES	PROTOTYPE	DESCRIPTION
contentPlace	*variant* contentPlace(Objects PageItems, *boolean* LinkPageItems, *boolean* LinkStories, *boolean* MapStyles, *variant* PlacePoint, *variant* DestinationLayer, *boolean* ShowingOptions)	*Duplicate an object and place it into the target. / Return value: The placed content. Type: Array of PageItems, Graphics, Stories or Texts. / PageItems: One or more page items to place or load. / LinkPageItems: Whether to link pageItems in content placer (if true it will override link stories value). / LinkStories: Whether to link stories in content placer (only applicable for single story, pageItem links will also be created in case of more than one item). / MapStyles: Whether to map styles in content placer. / PlacePoint: The point at which to place. Type: Array of 2 Units (Doubles or Strings). / DestinationLayer: The layer on which to place. Type: Layer. / ShowingOptions: Whether to display the link options dialog.*
checkBoxes	{CheckBoxes} •checkBoxes	*A collection of checkboxes.*
comboBoxes	{ComboBoxes} •comboBoxes	*A collection of comboboxes.*
listBoxes	{ListBoxes} •listBoxes	*A collection of listboxes.*
primaryTextFrame	*variant* primaryTextFrame	*The primary text frame through which text flows on the MasterSpread. Must be a text frame or a type on a path spline. Type: PageItem or idNothingEnum enumerator.*
radioButtons	{RadioButtons} •radioButtons	*A collection of radio buttons.*
signatureFields	{SignatureFields} •signatureFields	*A collection of signature fields.*
textBoxes	{TextBoxes} •textBoxes	*A collection of text boxes.*

Movie

Object available in ID 3, 4, 5, 6, 7, 7.5, 8

MOVIE CS6 VALUES	PROTOTYPE	DESCRIPTION
contentPlace	*variant* contentPlace(Objects PageItems, *boolean* LinkPageItems, *boolean* LinkStories, *boolean* MapStyles, *boolean* ShowingOptions)	*Duplicate an object and place it into the target page item. / Return value: The placed content. Type: Array of PageItems, Graphics, Stories or Texts. / PageItems: One or more page items to place or load. / LinkPageItems: Whether to link pageItems in content placer (if true it will override link stories value). / LinkStories: Whether to link stories in content placer (only applicable for single story, pageItem links will also be created in case of more than one item). / MapStyles: Whether to map styles in content placer. / ShowingOptions: Whether to display the link options dialog.*
horizontalLayoutConstraints	*variant* horizontalLayoutConstraints	*The left margin, width, and right margin constraints this item is subject to when using the object-based layout rule. Type: Array of 3 idDimensionsConstraints enumerators.*
linkedPageItemOptions	{LinkedPageItemOption} •linkedPageItemOptions	*Linked Page Item options.*
verticalLayoutConstraints	*variant* verticalLayoutConstraints	*The top margin, height, and bottom margin constraints this item is subject to when using the object-based layout rule. Type: Array of 3 idDimensionsConstraints enumerators.*

Oval

Object available in ID 3, 4, 5, 6, 7, 7.5, 8

OVAL CS6 VALUES	PROTOTYPE	DESCRIPTION
contentPlace	*variant* contentPlace(Objects PageItems, *boolean* LinkPageItems, *boolean* LinkStories, *boolean* MapStyles, *boolean* ShowingOptions)	*Duplicate an object and place it into the target page item. / Return value: The placed content. Type: Array of PageItems, Graphics, Stories or Texts. / PageItems: One or more page items to place or load. / LinkPageItems: Whether to link pageItems in content placer (if true it will override link stories value). / LinkStories: Whether to link stories in content placer (only applicable for single story, pageItem links will also be created in case of more than one item). / MapStyles: Whether to map styles in content placer. / ShowingOptions: Whether to display the link options dialog.*
checkBoxes	{CheckBoxes} •checkBoxes	*A collection of checkboxes.*
comboBoxes	{ComboBoxes} •comboBoxes	*A collection of comboboxes.*
horizontalLayoutConstraints	*variant* horizontalLayoutConstraints	*The left margin, width, and right margin constraints this item is subject to when using the object-based layout rule. Type: Array of 3 idDimensionsConstraints enumerators.*
htmlItems	{HtmlItems} •htmlItems	*A collection of embedded HTML page items.*
linkedPageItemOptions	{LinkedPageItemOption} •linkedPageItemOptions	*Linked Page Item options.*
listBoxes	{ListBoxes} •listBoxes	*A collection of listboxes.*
radioButtons	{RadioButtons} •radioButtons	*A collection of radio buttons.*
signatureFields	{SignatureFields} •signatureFields	*A collection of signature fields.*
textBoxes	{TextBoxes} •textBoxes	*A collection of text boxes.*

OVAL CS6 VALUES	PROTOTYPE	DESCRIPTION
verticalLayoutConstraints	*variant* verticalLayoutConstraints	*The top margin, height, and bottom margin constraints this item is subject to when using the object-based layout rule. Type: Array of 3 idDimensionsConstraints enumerators.*

Page

Object available in ID 3, 4, 5, 6, 7, 7.5, 8

PAGE CS6 VALUES	PROTOTYPE	DESCRIPTION
contentPlace	*variant* contentPlace(Objects PageItems, *boolean* LinkPageItems, *boolean* LinkStories, *boolean* MapStyles, *variant* PlacePoint, *variant* DestinationLayer, *boolean* ShowingOptions)	*Duplicate an object and place it into the target. / Return value: The placed content. Type: Array of PageItems, Graphics, Stories or Texts. / PageItems: One or more page items to place or load. / LinkPageItems: Whether to link pageItems in content placer (if true it will override link stories value). / LinkStories: Whether to link stories in content placer (only applicable for single story, pageItem links will also be created in case of more than one item). / MapStyles: Whether to map styles in content placer. / PlacePoint: The point at which to place. Type: Array of 2 Units (Doubles or Strings). / DestinationLayer: The layer on which to place. Type: Layer. / ShowingOptions: Whether to display the link options dialog.*
deleteAllLayoutSnapshots	*void* deleteAllLayoutSnapshots()	*Delete all layout snapshots for this Page.*
deleteLayoutSnapshot	*void* deleteLayoutSnapshot()	*Delete the snapshot of the layout for the current Page size and shape.*
snapshotCurrentLayout	*void* snapshotCurrentLayout()	*Create a snapshot of the layout for the current Page size and shape.*
appliedAlternateLayout	{Section} •appliedAlternateLayout	*The alternate layout section to which the page belongs.*
checkBoxes	{CheckBoxes} •checkBoxes	*A collection of checkboxes.*
comboBoxes	{ComboBoxes} •comboBoxes	*A collection of comboboxes.*
layoutRule	#LayoutRuleOptions layoutRule	*layout rule.*
listBoxes	{ListBoxes} •listBoxes	*A collection of listboxes.*
optionalPage	*boolean* optionalPage	*optional page for HTML5 pagination.*
radioButtons	{RadioButtons} •radioButtons	*A collection of radio buttons.*
signatureFields	{SignatureFields} •signatureFields	*A collection of signature fields.*
snapshotBlendingMode	#SnapshotBlendingModes snapshotBlendingMode	*snapshot blending mode.*
textBoxes	{TextBoxes} •textBoxes	*A collection of text boxes.*

PageItem

Object available in ID 3, 4, 5, 6, 7, 7.5, 8

PAGEITEM CS6 VALUES	PROTOTYPE	DESCRIPTION
contentPlace	*variant* contentPlace(Objects PageItems, *boolean* LinkPageItems, *boolean* LinkStories, *boolean* MapStyles, *boolean* ShowingOptions)	*Duplicate an object and place it into the target page item. / Return value: The placed content. Type: Array of PageItems, Graphics, Stories or Texts. / PageItems: One or more page items to place or load. / LinkPageItems: Whether to link pageItems in content placer (if true it will override link stories value). / LinkStories: Whether to link stories in content placer (only applicable for single story, pageItem links will also be created in case of more than one item). / MapStyles: Whether to map styles in content placer. / ShowingOptions: Whether to display the link options dialog.*
horizontalLayoutConstraints	*variant* horizontalLayoutConstraints	*The left margin, width, and right margin constraints this item is subject to when using the object-based layout rule. Type: Array of 3 idDimensionsConstraints enumerators.*
linkedPageItemOptions	{LinkedPageItemOption} •linkedPageItemOptions	*Linked Page Item options.*
verticalLayoutConstraints	*variant* verticalLayoutConstraints	*The top margin, height, and bottom margin constraints this item is subject to when using the object-based layout rule. Type: Array of 3 idDimensionsConstraints enumerators.*

Paragraph

Object available in ID 3, 4, 5, 6, 7, 7.5, 8

PARAGRAPH CS6 VALUES	PROTOTYPE	DESCRIPTION
checkBoxes	{CheckBoxes} •checkBoxes	*A collection of checkboxes.*
comboBoxes	{ComboBoxes} •comboBoxes	*A collection of comboboxes.*
listBoxes	{ListBoxes} •listBoxes	*A collection of listboxes.*
paragraphKashidaWidth	*double* paragraphKashidaWidth	*Paragraph kashida width. 0 is none, 1 is short, 2 is medium, 3 is long.*
radioButtons	{RadioButtons} •radioButtons	*A collection of radio buttons.*
signatureFields	{SignatureFields} •signatureFields	*A collection of signature fields.*
textBoxes	{TextBoxes} •textBoxes	*A collection of text boxes.*

ParagraphStyle

Object available in ID 3, 4, 5, 6, 7, 7.5, 8

PARAGRAPHSTYLE CS6 VALUES	PROTOTYPE	DESCRIPTION
paragraphKashidaWidth	*double* paragraphKashidaWidth	*Paragraph kashida width. 0 is none, 1 is short, 2 is medium, 3 is long.*

PDF

Object available in ID 3, 4, 5, 6, 7, 7.5, 8

PDF CS6 VALUES	PROTOTYPE	DESCRIPTION
contentPlace	*variant* contentPlace(Objects PageItems, *boolean* LinkPageItems, *boolean* LinkStories, *boolean* MapStyles, *boolean* ShowingOptions)	*Duplicate an object and place it into the target page item. / Return value: The placed content. Type: Array of PageItems, Graphics, Stories or Texts. / PageItems: One or more page items to place or load. / LinkPageItems: Whether to link pageItems in content placer (if true it will override link stories value). / LinkStories: Whether to link stories in content placer (only applicable for single story, pageItem links will also be created in case of more than one item). / MapStyles: Whether to map styles in content placer. / ShowingOptions: Whether to display the link options dialog.*
horizontalLayoutConstraints	*variant* horizontalLayoutConstraints	*The left margin, width, and right margin constraints this item is subject to when using the object-based layout rule. Type: Array of 3 idDimensionsConstraints enumerators.*
linkedPageItemOptions	{LinkedPageItemOption} •linkedPageItemOptions	*Linked Page Item options.*
verticalLayoutConstraints	*variant* verticalLayoutConstraints	*The top margin, height, and bottom margin constraints this item is subject to when using the object-based layout rule. Type: Array of 3 idDimensionsConstraints enumerators.*

PICT

Object available in ID 3, 4, 5, 6, 7, 7.5, 8

PICT CS6 VALUES	PROTOTYPE	DESCRIPTION
contentPlace	*variant* contentPlace(Objects PageItems, *boolean* LinkPageItems, *boolean* LinkStories, *boolean* MapStyles, *boolean* ShowingOptions)	*Duplicate an object and place it into the target page item. / Return value: The placed content. Type: Array of PageItems, Graphics, Stories or Texts. / PageItems: One or more page items to place or load. / LinkPageItems: Whether to link pageItems in content placer (if true it will override link stories value). / LinkStories: Whether to link stories in content placer (only applicable for single story, pageItem links will also be created in case of more than one item). / MapStyles: Whether to map styles in content placer. / ShowingOptions: Whether to display the link options dialog.*
horizontalLayoutConstraints	*variant* horizontalLayoutConstraints	*The left margin, width, and right margin constraints this item is subject to when using the object-based layout rule. Type: Array of 3 idDimensionsConstraints enumerators.*
linkedPageItemOptions	{LinkedPageItemOption} •linkedPageItemOptions	*Linked Page Item options.*
verticalLayoutConstraints	*variant* verticalLayoutConstraints	*The top margin, height, and bottom margin constraints this item is subject to when using the object-based layout rule. Type: Array of 3 idDimensionsConstraints enumerators.*

Polygon

Object available in ID 3, 4, 5, 6, 7, 7.5, 8

POLYGON CS6 VALUES	PROTOTYPE	DESCRIPTION
contentPlace	*variant* contentPlace(Objects PageItems, *boolean* LinkPageItems, *boolean* LinkStories, *boolean* MapStyles, *boolean* ShowingOptions)	*Duplicate an object and place it into the target page item. / Return value: The placed content. Type: Array of PageItems, Graphics, Stories or Texts. / PageItems: One or more page items to place or load. / LinkPageItems: Whether to link pageItems in content placer (if true it will override link stories value). / LinkStories: Whether to link stories in content placer (only applicable for single story, pageItem links will also be created in case of more than one item). / MapStyles: Whether to map styles in content placer. / ShowingOptions: Whether to display the link options dialog.*
checkBoxes	{CheckBoxes} •checkBoxes	*A collection of checkboxes.*
comboBoxes	{ComboBoxes} •comboBoxes	*A collection of comboboxes.*
horizontalLayoutConstraints	*variant* horizontalLayoutConstraints	*The left margin, width, and right margin constraints this item is subject to when using the object-based layout rule. Type: Array of 3 idDimensionsConstraints enumerators.*
htmlItems	{HtmlItems} •htmlItems	*A collection of embedded HTML page items.*
linkedPageItemOptions	{LinkedPageItemOption} •linkedPageItemOptions	*Linked Page Item options.*
listBoxes	{ListBoxes} •listBoxes	*A collection of listboxes.*
radioButtons	{RadioButtons} •radioButtons	*A collection of radio buttons.*
signatureFields	{SignatureFields} •signatureFields	*A collection of signature fields.*
textBoxes	{TextBoxes} •textBoxes	*A collection of text boxes.*
verticalLayoutConstraints	*variant* verticalLayoutConstraints	*The top margin, height, and bottom margin constraints this item is subject to when using the object-based layout rule. Type: Array of 3 idDimensionsConstraints enumerators.*

Rectangle

Object available in ID 3, 4, 5, 6, 7, 7.5, 8

RECTANGLE CS6 VALUES	PROTOTYPE	DESCRIPTION
contentPlace	*variant* contentPlace(Objects PageItems, *boolean* LinkPageItems, *boolean* LinkStories, *boolean* MapStyles, *boolean* ShowingOptions)	*Duplicate an object and place it into the target page item. / Return value: The placed content. Type: Array of PageItems, Graphics, Stories or Texts. / PageItems: One or more page items to place or load. / LinkPageItems: Whether to link pageItems in content placer (if true it will override link stories value). / LinkStories: Whether to link stories in content placer (only applicable for single story, pageItem links will also be created in case of more than one item). / MapStyles: Whether to map styles in content placer. / ShowingOptions: Whether to display the link options dialog.*
checkBoxes	{CheckBoxes} •checkBoxes	*A collection of checkboxes.*
comboBoxes	{ComboBoxes} •comboBoxes	*A collection of comboboxes.*
horizontalLayoutConstraints	*variant* horizontalLayoutConstraints	*The left margin, width, and right margin constraints this item is subject to when using the object-based layout rule. Type: Array of 3 idDimensionsConstraints enumerators.*

RECTANGLE CS6 VALUES	PROTOTYPE	DESCRIPTION
htmlItems	{HtmlItems} •htmlItems	*A collection of embedded HTML page items.*
linkedPageItemOptions	{LinkedPageItemOption} •linkedPageItemOptions	*Linked Page Item options.*
listBoxes	{ListBoxes} •listBoxes	*A collection of listboxes.*
radioButtons	{RadioButtons} •radioButtons	*A collection of radio buttons.*
signatureFields	{SignatureFields} •signatureFields	*A collection of signature fields.*
textBoxes	{TextBoxes} •textBoxes	*A collection of text boxes.*
verticalLayoutConstraints	*variant* verticalLayoutConstraints	*The top margin, height, and bottom margin constraints this item is subject to when using the object-based layout rule. Type: Array of 3 idDimensionsConstraints enumerators.*

Section

Object available in ID 3, 4, 5, 6, 7, 7.5, 8

SECTION CS6 VALUES	PROTOTYPE	DESCRIPTION
alternateLayout	*string* alternateLayout	*The alternate layout name for a set of pages.*
alternateLayoutLength	*integer* •alternateLayoutLength	*The number of pages in the alternate layout section.*
pagination	#PaginationOption pagination	*The pagination option for this section for adding and removing pages in HTML5.*
paginationMaster	*variant* paginationMaster	*The master to apply when pages are added in HTML5. Type: MasterSpread or idNothingEnum enumerator.*

Sound

Object available in ID 3, 4, 5, 6, 7, 7.5, 8

SOUND CS6 VALUES	PROTOTYPE	DESCRIPTION
contentPlace	*variant* contentPlace(Objects PageItems, *boolean* LinkPageItems, *boolean* LinkStories, *boolean* MapStyles, *boolean* ShowingOptions)	*Duplicate an object and place it into the target page item. / Return value: The placed content. Type: Array of PageItems, Graphics, Stories or Texts. / PageItems: One or more page items to place or load. / LinkPageItems: Whether to link pageItems in content placer (if true it will override link stories value). / LinkStories: Whether to link stories in content placer (only applicable for single story, pageItem links will also be created in case of more than one item). / MapStyles: Whether to map styles in content placer. / ShowingOptions: Whether to display the link options dialog.*
horizontalLayoutConstraints	*variant* horizontalLayoutConstraints	*The left margin, width, and right margin constraints this item is subject to when using the object-based layout rule. Type: Array of 3 idDimensionsConstraints enumerators.*
linkedPageItemOptions	{LinkedPageItemOption} •linkedPageItemOptions	*Linked Page Item options.*
verticalLayoutConstraints	*variant* verticalLayoutConstraints	*The top margin, height, and bottom margin constraints this item is subject to when using the object-based layout rule. Type: Array of 3 idDimensionsConstraints enumerators.*

Spread

Object available in ID 3, 4, 5, 6, 7, 7.5, 8

SPREAD CS6 VALUES	PROTOTYPE	DESCRIPTION
contentPlace	*variant* contentPlace(Objects PageItems, *boolean* LinkPageItems, *boolean* LinkStories, *boolean* MapStyles, *variant* PlacePoint, *variant* DestinationLayer, *boolean* ShowingOptions)	*Duplicate an object and place it into the target. / Return value: The placed content. Type: Array of PageItems, Graphics, Stories or Texts. / PageItems: One or more page items to place or load. / LinkPageItems: Whether to link pageItems in content placer (if true it will override link stories value). / LinkStories: Whether to link stories in content placer (only applicable for single story, pageItem links will also be created in case of more than one item). / MapStyles: Whether to map styles in content placer. / PlacePoint: The point at which to place. Type: Array of 2 Units (Doubles or Strings). / DestinationLayer: The layer on which to place. Type: Layer. / ShowingOptions: Whether to display the link options dialog.*
checkBoxes	{CheckBoxes} •checkBoxes	*A collection of checkboxes.*
comboBoxes	{ComboBoxes} •comboBoxes	*A collection of comboboxes.*
listBoxes	{ListBoxes} •listBoxes	*A collection of listboxes.*
radioButtons	{RadioButtons} •radioButtons	*A collection of radio buttons.*
signatureFields	{SignatureFields} •signatureFields	*A collection of signature fields.*
textBoxes	{TextBoxes} •textBoxes	*A collection of text boxes.*

StaticText

Object available in ID 3, 4, 5, 6, 7, 7.5, 8

STATICTEXT CS6 VALUES	PROTOTYPE	DESCRIPTION
staticAlignment	#StaticAlignmentOptions staticAlignment	*Text alignment for the StaticText.*

Story

Object available in ID 3, 4, 5, 6, 7, 7.5, 8

STORY CS6 VALUES	PROTOTYPE	DESCRIPTION
cellStyleMappings	{CellStyleMappings} •cellStyleMappings	*A collection of cell style mappings.*
charStyleMappings	{CharStyleMappings} •charStyleMappings	*A collection of char style mappings.*
checkBoxes	{CheckBoxes} •checkBoxes	*A collection of checkboxes.*
comboBoxes	{ComboBoxes} •comboBoxes	*A collection of comboboxes.*
listBoxes	{ListBoxes} •listBoxes	*A collection of listboxes.*
paragraphKashidaWidth	*double* paragraphKashidaWidth	*Paragraph kashida width. 0 is none, 1 is short, 2 is medium, 3 is long.*
paraStyleMappings	{ParaStyleMappings} •paraStyleMappings	*A collection of para style mappings.*
radioButtons	{RadioButtons} •radioButtons	*A collection of radio buttons.*

STORY CS6 VALUES	PROTOTYPE	DESCRIPTION
signatureFields	{SignatureFields} •signatureFields	*A collection of signature fields.*
tableStyleMappings	{TableStyleMappings} •tableStyleMappings	*A collection of table style mappings.*
textBoxes	{TextBoxes} •textBoxes	*A collection of text boxes.*

StoryWindow

Object available in ID 3, 4, 5, 6, 7, 7.5, 8

STORYWINDOW CS6 VALUES	PROTOTYPE	DESCRIPTION
selectionKeyObject	*variant* selectionKeyObject	*The key object of the selection. Type: PageItem or idNothingEnum enumerator.*

Table

Object available in ID 3, 4, 5, 6, 7, 7.5, 8

TABLE CS6 VALUES	PROTOTYPE	DESCRIPTION
checkBoxes	{CheckBoxes} •checkBoxes	*A collection of checkboxes.*
comboBoxes	{ComboBoxes} •comboBoxes	*A collection of comboboxes.*
listBoxes	{ListBoxes} •listBoxes	*A collection of listboxes.*
radioButtons	{RadioButtons} •radioButtons	*A collection of radio buttons.*
signatureFields	{SignatureFields} •signatureFields	*A collection of signature fields.*
textBoxes	{TextBoxes} •textBoxes	*A collection of text boxes.*

Text

Object available in ID 3, 4, 5, 6, 7, 7.5, 8

TEXT CS6 VALUES	PROTOTYPE	DESCRIPTION
checkBoxes	{CheckBoxes} •checkBoxes	*A collection of checkboxes.*
comboBoxes	{ComboBoxes} •comboBoxes	*A collection of comboboxes.*
listBoxes	{ListBoxes} •listBoxes	*A collection of listboxes.*
paragraphKashidaWidth	*double* paragraphKashidaWidth	*Paragraph kashida width. 0 is none, 1 is short, 2 is medium, 3 is long.*
radioButtons	{RadioButtons} •radioButtons	*A collection of radio buttons.*
signatureFields	{SignatureFields} •signatureFields	*A collection of signature fields.*
textBoxes	{TextBoxes} •textBoxes	*A collection of text boxes.*

TextColumn

Object available in ID 3, 4, 5, 6, 7, 7.5, 8

TEXTCOLUMN CS6 VALUES	PROTOTYPE	DESCRIPTION
checkBoxes	{CheckBoxes} •checkBoxes	*A collection of checkboxes.*
comboBoxes	{ComboBoxes} •comboBoxes	*A collection of comboboxes.*
listBoxes	{ListBoxes} •listBoxes	*A collection of listboxes.*
paragraphKashidaWidth	*double* paragraphKashidaWidth	*Paragraph kashida width. 0 is none, 1 is short, 2 is medium, 3 is long.*
radioButtons	{RadioButtons} •radioButtons	*A collection of radio buttons.*
signatureFields	{SignatureFields} •signatureFields	*A collection of signature fields.*
textBoxes	{TextBoxes} •textBoxes	*A collection of text boxes.*

TextDefault

Object available in ID 3, 4, 5, 6, 7, 7.5, 8

TEXTDEFAULT CS6 VALUES	PROTOTYPE	DESCRIPTION
paragraphKashidaWidth	*double* paragraphKashidaWidth	*Paragraph kashida width. 0 is none, 1 is short, 2 is medium, 3 is long.*

TextFrame

Object available in ID 3, 4, 5, 6, 7, 7.5, 8

TEXTFRAME CS6 VALUES	PROTOTYPE	DESCRIPTION
contentPlace	*variant* contentPlace(Objects PageItems, *boolean* LinkPageItems, *boolean* LinkStories, *boolean* MapStyles, *boolean* ShowingOptions)	*Duplicate an object and place it into the target page item. / Return value: The placed content. Type: Array of PageItems, Graphics, Stories or Texts. / PageItems: One or more page items to place or load. / LinkPageItems: Whether to link pageItems in content placer (if true it will override link stories value). / LinkStories: Whether to link stories in content placer (only applicable for single story, pageItem links will also be created in case of more than one item). / MapStyles: Whether to map styles in content placer. / ShowingOptions: Whether to display the link options dialog.*
checkBoxes	{CheckBoxes} •checkBoxes	*A collection of checkboxes.*
comboBoxes	{ComboBoxes} •comboBoxes	*A collection of comboboxes.*
horizontalLayoutConstraints	*variant* horizontalLayoutConstraints	*The left margin, width, and right margin constraints this item is subject to when using the object-based layout rule. Type: Array of 3 idDimensionsConstraints enumerators.*
linkedPageItemOptions	{LinkedPageItemOption} •linkedPageItemOptions	*Linked Page Item options.*
listBoxes	{ListBoxes} •listBoxes	*A collection of listboxes.*
objectExportOptions	{ObjectExportOption} •objectExportOptions	*Export options for the object.*
radioButtons	{RadioButtons} •radioButtons	*A collection of radio buttons.*

TEXTFRAME CS6 VALUES	PROTOTYPE	DESCRIPTION
signatureFields	{SignatureFields} •signatureFields	*A collection of signature fields.*
textBoxes	{TextBoxes} •textBoxes	*A collection of text boxes.*
verticalLayoutConstraints	*variant* verticalLayoutConstraints	*The top margin, height, and bottom margin constraints this item is subject to when using the object-based layout rule. Type: Array of 3 idDimensionsConstraints enumerators.*

TextFramePreference

Object available in ID 3, 4, 5, 6, 7, 7.5, 8

TEXTFRAMEPREFERENCE CS6 VALUES	PROTOTYPE	DESCRIPTION
autoSizingReferencePoint	#AutoSizingReferenceEnum autoSizingReferencePoint	*The reference point for auto sizing of text frame. Reference point is automatically adjusted to the suitable value depending on the auto-sizing type value. As an example, top left reference point becomes top center for height only dimension.*
autoSizingType	#AutoSizingTypeEnum autoSizingType	*Auto-sizing type of text frame. Based on type, reference value is automatically adjusted. For example, for height only type, top-left reference point becomes top-center. Recommended to change auto-sizing type, after setting other auto-sizing attributes.*
minimumHeightForAutoSizing	*variant* minimumHeightForAutoSizing	*The minimum height for auto-sizing of the text frame. Type: Unit (Double or String).*
minimumWidthForAutoSizing	*variant* minimumWidthForAutoSizing	*The minimum width for auto-sizing of the text frame. Type: Unit (Double or String).*
textColumnMaxWidth	*variant* textColumnMaxWidth	*The maximum column width of the columns in the text frame. Use 0 to indicate no upper limit. Type: Unit (Double or String) (0 - 8640 points).*
useFlexibleColumnWidth	*boolean* useFlexibleColumnWidth	*If true, maintains column width between a min and max range when the text frame is resized. If false, causes columns to resize when the text frame is resized. Note: When true, resizing the frame can change the number of columns in the frame.*
useMinimumHeightForAutoSizing	*boolean* useMinimumHeightForAutoSizing	*If true, minimum height value is used during the auto-sizing of text frame.*
useMinimumWidthForAutoSizing	*boolean* useMinimumWidthForAutoSizing	*If true, minimum width value is used during the auto-sizing of text frame.*
useNoLineBreaksForAutoSizing	*boolean* useNoLineBreaksForAutoSizing	*If true, line-breaks are not introduced after auto sizing.*

TextPreference

Object available in ID 3, 4, 5, 6, 7, 7.5, 8

TEXTPREFERENCE CS6 VALUES	PROTOTYPE	DESCRIPTION
quoteCharactersRotatedInVertical	*boolean* quoteCharactersRotatedInVertical	*If true, Japanese composer treats quotes as half width and rotates them in vertical.*

TextStyleRange

Object available in ID 3, 4, 5, 6, 7, 7.5, 8

TEXTSTYLERANGE CS6 VALUES	PROTOTYPE	DESCRIPTION
checkBoxes	{CheckBoxes} •checkBoxes	*A collection of checkboxes.*
comboBoxes	{ComboBoxes} •comboBoxes	*A collection of comboboxes.*
listBoxes	{ListBoxes} •listBoxes	*A collection of listboxes.*
paragraphKashidaWidth	*double* paragraphKashidaWidth	*Paragraph kashida width. 0 is none, 1 is short, 2 is medium, 3 is long.*
radioButtons	{RadioButtons} •radioButtons	*A collection of radio buttons.*
signatureFields	{SignatureFields} •signatureFields	*A collection of signature fields.*
textBoxes	{TextBoxes} •textBoxes	*A collection of text boxes.*

Window

Object available in ID 3, 4, 5, 6, 7, 7.5, 8

WINDOW CS6 VALUES	PROTOTYPE	DESCRIPTION
selectionKeyObject	*variant* selectionKeyObject	*The key object of the selection. Type: PageItem or idNothingEnum enumerator.*

WMF

Object available in ID 3, 4, 5, 6, 7, 7.5, 8

WMF CS6 VALUES	PROTOTYPE	DESCRIPTION
contentPlace	*variant* contentPlace(Objects PageItems, *boolean* LinkPageItems, *boolean* LinkStories, *boolean* MapStyles, *boolean* ShowingOptions)	*Duplicate an object and place it into the target page item. / Return value: The placed content. Type: Array of PageItems, Graphics, Stories or Texts. / PageItems: One or more page items to place or load. / LinkPageItems: Whether to link pageItems in content placer (if true it will override link stories value). / LinkStories: Whether to link stories in content placer (only applicable for single story, pageItem links will also be created in case of more than one item). / MapStyles: Whether to map styles in content placer. / ShowingOptions: Whether to display the link options dialog.*
horizontalLayoutConstraints	*variant* horizontalLayoutConstraints	*The left margin, width, and right margin constraints this item is subject to when using the object-based layout rule. Type: Array of 3 idDimensionsConstraints enumerators.*
linkedPageItemOptions	{LinkedPageItemOption} •linkedPageItemOptions	*Linked Page Item options.*
verticalLayoutConstraints	*variant* verticalLayoutConstraints	*The top margin, height, and bottom margin constraints this item is subject to when using the object-based layout rule. Type: Array of 3 idDimensionsConstraints enumerators.*

Word

Object available in ID 3, 4, 5, 6, 7, 7.5, 8

WORD CS6 VALUES	PROTOTYPE	DESCRIPTION
checkBoxes	{CheckBoxes} •checkBoxes	*A collection of checkboxes.*
comboBoxes	{ComboBoxes} •comboBoxes	*A collection of comboboxes.*
listBoxes	{ListBoxes} •listBoxes	*A collection of listboxes.*
paragraphKashidaWidth	*double* paragraphKashidaWidth	*Paragraph kashida width. 0 is none, 1 is short, 2 is medium, 3 is long.*
radioButtons	{RadioButtons} •radioButtons	*A collection of radio buttons.*
signatureFields	{SignatureFields} •signatureFields	*A collection of signature fields.*
textBoxes	{TextBoxes} •textBoxes	*A collection of text boxes.*

ObjectStyle

Object available in ID 4, 5, 6, 7, 7.5, 8

OBJECTSTYLE CS6 VALUES	PROTOTYPE	DESCRIPTION
enableTextFrameAutoSizingOptions	*boolean* enableTextFrameAutoSizingOptions	*If true, the object style will apply auto-sizing text frame options.*

ChangeGrepPreference

Object available in ID 5, 6, 7, 7.5, 8

CHANGEGREPPREFERENCE CS6 VALUES	PROTOTYPE	DESCRIPTION
paragraphKashidaWidth	*variant* paragraphKashidaWidth	*Paragraph kashida width. 0 is none, 1 is short, 2 is medium, 3 is long. Type: Double or idNothingEnum enumerator.*

ChangeObjectPreference

Object available in ID 5, 6, 7, 7.5, 8

CHANGEOBJECTPREFERENCE CS6 VALUES	PROTOTYPE	DESCRIPTION
autoSizingReferencePoint	*variant* autoSizingReferencePoint	*The reference point for auto sizing of text frame. Reference point is automatically adjusted to the suitable value depending on the auto-sizing type value. As an example, top left reference point becomes top center for height only dimension. Type: idAutoSizingReferenceEnum enumerator or idNothingEnum enumerator.*
autoSizingType	*variant* autoSizingType	*Auto-sizing type of text frame. Based on type, reference value is automatically adjusted. For example, for height only type, top-left reference point becomes top-center. Recommended to change auto-sizing type, after setting other auto-sizing attributes. Type: idAutoSizingTypeEnum enumerator or idNothingEnum enumerator.*

CHANGEOBJECTPREFERENCE CS6 VALUES	PROTOTYPE	DESCRIPTION
minimumHeightForAutoSizing	*variant* minimumHeightForAutoSizing	*The minimum height for auto-sizing of the text frame. Type: Unit (Double or String) or idNothingEnum enumerator.*
minimumWidthForAutoSizing	*variant* minimumWidthForAutoSizing	*The minimum width for auto-sizing of the text frame. Type: Unit (Double or String) or idNothingEnum enumerator.*
useMinimumHeightForAutoSizing	*variant* useMinimumHeightForAutoSizing	*If true, minimum height value is used during the auto-sizing of text frame. . Type: Boolean or idNothingEnum enumerator.*
useMinimumWidthForAutoSizing	*variant* useMinimumWidthForAutoSizing	*If true, minimum width value is used during the auto-sizing of text frame. . Type: Boolean or idNothingEnum enumerator.*
useNoLineBreaksForAutoSizing	*variant* useNoLineBreaksForAutoSizing	*If true, line-breaks are not introduced after auto sizing. . Type: Boolean or idNothingEnum enumerator.*

ChangeTextPreference

Object available in ID 5, 6, 7, 7.5, 8

CHANGETEXTPREFERENCE CS6 VALUES	PROTOTYPE	DESCRIPTION
paragraphKashidaWidth	*variant* paragraphKashidaWidth	*Paragraph kashida width. 0 is none, 1 is short, 2 is medium, 3 is long. Type: Double or idNothingEnum enumerator.*

FindGrepPreference

Object available in ID 5, 6, 7, 7.5, 8

FINDGREPPREFERENCE CS6 VALUES	PROTOTYPE	DESCRIPTION
paragraphKashidaWidth	*variant* paragraphKashidaWidth	*Paragraph kashida width. 0 is none, 1 is short, 2 is medium, 3 is long. Type: Double or idNothingEnum enumerator.*

FindObjectPreference

Object available in ID 5, 6, 7, 7.5, 8

FINDOBJECTPREFERENCE CS6 VALUES	PROTOTYPE	DESCRIPTION
autoSizingReferencePoint	*variant* autoSizingReferencePoint	*The reference point for auto sizing of text frame. Reference point is automatically adjusted to the suitable value depending on the auto-sizing type value. As an example, top left reference point becomes top center for height only dimension. Type: idAutoSizingReferenceEnum enumerator or idNothingEnum enumerator.*
autoSizingType	*variant* autoSizingType	*Auto-sizing type of text frame. Based on type, reference value is automatically adjusted. For example, for height only type, top-left reference point becomes top-center. Recommended to change auto-sizing type, after setting other auto-sizing attributes. Type: idAutoSizingTypeEnum enumerator or idNothingEnum enumerator.*
minimumHeightForAutoSizing	*variant* minimumHeightForAutoSizing	*The minimum height for auto-sizing of the text frame. Type: Unit (Double or String) or idNothingEnum enumerator.*

FINDOBJECTPREFERENCE CS6 VALUES	PROTOTYPE	DESCRIPTION
minimumWidthForAutoSizing	*variant* minimumWidthForAutoSizing	*The minimum width for auto-sizing of the text frame. Type: Unit (Double or String) or idNothingEnum enumerator.*
useMinimumHeightForAutoSizing	*variant* useMinimumHeightForAutoSizing	*If true, minimum height value is used during the auto-sizing of text frame. . Type: Boolean or idNothingEnum enumerator.*
useMinimumWidthForAutoSizing	*variant* useMinimumWidthForAutoSizing	*If true, minimum width value is used during the auto-sizing of text frame. . Type: Boolean or idNothingEnum enumerator.*
useNoLineBreaksForAutoSizing	*variant* useNoLineBreaksForAutoSizing	*If true, line-breaks are not introduced after auto sizing. . Type: Boolean or idNothingEnum enumerator.*

FindTextPreference

Object available in ID 5, 6, 7, 7.5, 8

FINDTEXTPREFERENCE CS6 VALUES	PROTOTYPE	DESCRIPTION
paragraphKashidaWidth	*variant* paragraphKashidaWidth	*Paragraph kashida width. 0 is none, 1 is short, 2 is medium, 3 is long. Type: Double or idNothingEnum enumerator.*

ImportedPage

Object available in ID 5, 6, 7, 7.5, 8

IMPORTEDPAGE CS6 VALUES	PROTOTYPE	DESCRIPTION
contentPlace	*variant* contentPlace(Objects PageItems, *boolean* LinkPageItems, *boolean* LinkStories, *boolean* MapStyles, *boolean* ShowingOptions)	*Duplicate an object and place it into the target page item. / Return value: The placed content. Type: Array of PageItems, Graphics, Stories or Texts. / PageItems: One or more page items to place or load. / LinkPageItems: Whether to link pageItems in content placer (if true it will override link stories value). / LinkStories: Whether to link stories in content placer (only applicable for single story, pageItem links will also be created in case of more than one item). / MapStyles: Whether to map styles in content placer. / ShowingOptions: Whether to display the link options dialog.*
horizontalLayoutConstraints	*variant* horizontalLayoutConstraints	*The left margin, width, and right margin constraints this item is subject to when using the object-based layout rule. Type: Array of 3 idDimensionsConstraints enumerators.*
linkedPageItemOptions	{LinkedPageItemOption} •linkedPageItemOptions	*Linked Page Item options.*
verticalLayoutConstraints	*variant* verticalLayoutConstraints	*The top margin, height, and bottom margin constraints this item is subject to when using the object-based layout rule. Type: Array of 3 idDimensionsConstraints enumerators.*

XmlStory

Object available in ID 5, 6, 7, 7.5, 8

XMLSTORY CS6 VALUES	PROTOTYPE	DESCRIPTION
cellStyleMappings	{CellStyleMappings} •cellStyleMappings	*A collection of cell style mappings.*
charStyleMappings	{CharStyleMappings} •charStyleMappings	*A collection of char style mappings.*
checkBoxes	{CheckBoxes} •checkBoxes	*A collection of checkboxes.*
comboBoxes	{ComboBoxes} •comboBoxes	*A collection of comboboxes.*
listBoxes	{ListBoxes} •listBoxes	*A collection of listboxes.*
paragraphKashidaWidth	*double* paragraphKashidaWidth	*Paragraph kashida width. 0 is none, 1 is short, 2 is medium, 3 is long.*
paraStyleMappings	{ParaStyleMappings} •paraStyleMappings	*A collection of para style mappings.*
radioButtons	{RadioButtons} •radioButtons	*A collection of radio buttons.*
signatureFields	{SignatureFields} •signatureFields	*A collection of signature fields.*
tableStyleMappings	{TableStyleMappings} •tableStyleMappings	*A collection of table style mappings.*
textBoxes	{TextBoxes} •textBoxes	*A collection of text boxes.*

EPSText

Object available in ID 7, 7.5, 8

EPSTEXT CS6 VALUES	PROTOTYPE	DESCRIPTION
contentPlace	*variant* contentPlace(Objects PageItems, *boolean* LinkPageItems, *boolean* LinkStories, *boolean* MapStyles, *boolean* ShowingOptions)	*Duplicate an object and place it into the target page item. / Return value: The placed content. Type: Array of PageItems, Graphics, Stories or Texts. / PageItems: One or more page items to place or load. / LinkPageItems: Whether to link pageItems in content placer (if true it will override link stories value). / LinkStories: Whether to link stories in content placer (only applicable for single story, pageItem links will also be created in case of more than one item). / MapStyles: Whether to map styles in content placer. / ShowingOptions: Whether to display the link options dialog.*
horizontalLayoutConstraints	*variant* horizontalLayoutConstraints	*The left margin, width, and right margin constraints this item is subject to when using the object-based layout rule. Type: Array of 3 idDimensionsConstraints enumerators.*
linkedPageItemOptions	{LinkedPageItemOption} •linkedPageItemOptions	*Linked Page Item options.*
verticalLayoutConstraints	*variant* verticalLayoutConstraints	*The top margin, height, and bottom margin constraints this item is subject to when using the object-based layout rule. Type: Array of 3 idDimensionsConstraints enumerators.*

InteractivePDFExportPreference

Object available in ID 7, 7.5, 8

INTERACTIVEPDFEXPORTPREFERENCE CS6 VALUES	PROTOTYPE	DESCRIPTION
usePDFStructureForTabOrder	*boolean* usePDFStructureForTabOrder	*Use tagged PDF structure for interactive elements tab order.*

MediaItem

Object available in ID 7, 7.5, 8

MEDIAITEM CS6 VALUES	PROTOTYPE	DESCRIPTION
contentPlace	*variant* contentPlace(Objects PageItems, *boolean* LinkPageItems, *boolean* LinkStories, *boolean* MapStyles, *boolean* ShowingOptions)	*Duplicate an object and place it into the target page item. / Return value: The placed content. Type: Array of PageItems, Graphics, Stories or Texts. / PageItems: One or more page items to place or load. / LinkPageItems: Whether to link pageItems in content placer (if true it will override link stories value). / LinkStories: Whether to link stories in content placer (only applicable for single story, pageItem links will also be created in case of more than one item). / MapStyles: Whether to map styles in content placer. / ShowingOptions: Whether to display the link options dialog.*
horizontalLayoutConstraints	*variant* horizontalLayoutConstraints	*The left margin, width, and right margin constraints this item is subject to when using the object-based layout rule. Type: Array of 3 idDimensionsConstraints enumerators.*
linkedPageItemOptions	{LinkedPageItemOption} •linkedPageItemOptions	*Linked Page Item options.*
verticalLayoutConstraints	*variant* verticalLayoutConstraints	*The top margin, height, and bottom margin constraints this item is subject to when using the object-based layout rule. Type: Array of 3 idDimensionsConstraints enumerators.*

MultiStateObject

Object available in ID 7, 7.5, 8

MULTISTATEOBJECT CS6 VALUES	PROTOTYPE	DESCRIPTION
contentPlace	*variant* contentPlace(Objects PageItems, *boolean* LinkPageItems, *boolean* LinkStories, *boolean* MapStyles, *boolean* ShowingOptions)	*Duplicate an object and place it into the target page item. / Return value: The placed content. Type: Array of PageItems, Graphics, Stories or Texts. / PageItems: One or more page items to place or load. / LinkPageItems: Whether to link pageItems in content placer (if true it will override link stories value). / LinkStories: Whether to link stories in content placer (only applicable for single story, pageItem links will also be created in case of more than one item). / MapStyles: Whether to map styles in content placer. / ShowingOptions: Whether to display the link options dialog.*
horizontalLayoutConstraints	*variant* horizontalLayoutConstraints	*The left margin, width, and right margin constraints this item is subject to when using the object-based layout rule. Type: Array of 3 idDimensionsConstraints enumerators.*
linkedPageItemOptions	{LinkedPageItemOption} •linkedPageItemOptions	*Linked Page Item options.*
verticalLayoutConstraints	*variant* verticalLayoutConstraints	*The top margin, height, and bottom margin constraints this item is subject to when using the object-based layout rule. Type: Array of 3 idDimensionsConstraints enumerators.*

PlaceGun

Object available in ID 7, 7.5, 8

PLACEGUN CS6 VALUES	PROTOTYPE	DESCRIPTION
checkBoxes	{CheckBoxes} •checkBoxes	*A collection of checkboxes.*
comboBoxes	{ComboBoxes} •comboBoxes	*A collection of comboboxes.*
listBoxes	{ListBoxes} •listBoxes	*A collection of listboxes.*
radioButtons	{RadioButtons} •radioButtons	*A collection of radio buttons.*
signatureFields	{SignatureFields} •signatureFields	*A collection of signature fields.*
textBoxes	{TextBoxes} •textBoxes	*A collection of text boxes.*

Snippet

Object available in ID 7, 7.5, 8

SNIPPET CS6 VALUES	PROTOTYPE	DESCRIPTION
checkBoxes	{CheckBoxes} •checkBoxes	*A collection of checkboxes.*
comboBoxes	{ComboBoxes} •comboBoxes	*A collection of comboboxes.*
listBoxes	{ListBoxes} •listBoxes	*A collection of listboxes.*
radioButtons	{RadioButtons} •radioButtons	*A collection of radio buttons.*
signatureFields	{SignatureFields} •signatureFields	*A collection of signature fields.*
textBoxes	{TextBoxes} •textBoxes	*A collection of text boxes.*

SplineItem

Object available in ID 7, 7.5, 8

SPLINEITEM CS6 VALUES	PROTOTYPE	DESCRIPTION
contentPlace	*variant* contentPlace(Objects PageItems, *boolean* LinkPageItems, *boolean* LinkStories, *boolean* MapStyles, *boolean* ShowingOptions)	*Duplicate an object and place it into the target page item. / Return value: The placed content. Type: Array of PageItems, Graphics, Stories or Texts. / PageItems: One or more page items to place or load. / LinkPageItems: Whether to link pageItems in content placer (if true it will override link stories value). / LinkStories: Whether to link stories in content placer (only applicable for single story, pageItem links will also be created in case of more than one item). / MapStyles: Whether to map styles in content placer. / ShowingOptions: Whether to display the link options dialog.*
checkBoxes	{CheckBoxes} •checkBoxes	*A collection of checkboxes.*
comboBoxes	{ComboBoxes} •comboBoxes	*A collection of comboboxes.*
horizontalLayoutConstraints	*variant* horizontalLayoutConstraints	*The left margin, width, and right margin constraints this item is subject to when using the object-based layout rule. Type: Array of 3 idDimensionsConstraints enumerators.*

SPLINEITEM CS6 VALUES	PROTOTYPE	DESCRIPTION
htmlItems	{HtmlItems} •htmlItems	*A collection of embedded HTML page items.*
linkedPageItemOptions	{LinkedPageItemOption} •linkedPageItemOptions	*Linked Page Item options.*
listBoxes	{ListBoxes} •listBoxes	*A collection of listboxes.*
radioButtons	{RadioButtons} •radioButtons	*A collection of radio buttons.*
signatureFields	{SignatureFields} •signatureFields	*A collection of signature fields.*
textBoxes	{TextBoxes} •textBoxes	*A collection of text boxes.*
verticalLayoutConstraints	*variant* verticalLayoutConstraints	*The top margin, height, and bottom margin constraints this item is subject to when using the object-based layout rule. Type: Array of 3 idDimensionsConstraints enumerators.*

EPubExportPreference

Object available in ID 7.5, 8

EPUBEXPORTPREFERENCE CS6 VALUES	PROTOTYPE	DESCRIPTION
externalStyleSheets	*variant* externalStyleSheets	*The file path of external cascading style sheets. Type: Array of Strings.*
javascripts	*variant* javascripts	*The file path of external javascripts. Type: Array of Strings.*
version	#EpubVersion version	*The version of EPUB.*

HTMLExportPreference

Object available in ID 7.5, 8

HTMLEXPORTPREFERENCE CS6 VALUES	PROTOTYPE	DESCRIPTION
externalStyleSheets	*variant* externalStyleSheets	*The file path of external cascading style sheets. Type: Array of Strings.*
javascripts	*variant* javascripts	*The file path of external javascripts. Type: Array of Strings.*

LinkedStoryOption

Object available in ID 7.5, 8

LINKEDSTORYOPTION CS6 VALUES	PROTOTYPE	DESCRIPTION
applyStyleMappings	*boolean* applyStyleMappings	*If true, style mappings will be applied during linked story creation or update.*

ObjectExportOption

Object available in ID 7.5, 8

OBJECTEXPORTOPTION CS6 VALUES	PROTOTYPE	DESCRIPTION
customLayout	*boolean* customLayout	*If true, custom layout is enabled for object.*
customLayoutType	#CustomLayoutTypeEnum customLayoutType	*Custom Layout settings to be used for object.*

StyleExportTagMap

Object available in ID 7.5, 8

STYLEEXPORTTAGMAP CS6 VALUES	PROTOTYPE	DESCRIPTION
splitDocument	*boolean* splitDocument	*Split Document (EPUB only).*

CellStyleMapping

Object available in ID 8

CELLSTYLEMAPPING CS6 VALUES	PROTOTYPE	DESCRIPTION
delete	*void* delete()	*deletes a style mapping.*
destinationStyleName	*string* destinationStyleName	*The destination style name property.*
index	*integer* •index	*The index of the CellStyleMapping within its containing object.*
mappingRuleType	#MapType mappingRuleType	*The type of the mapping rule.*
parent	*variant* •parent	*The parent of the CellStyleMapping (a Application, Document, Story or XmlStory). Type: Application, Document, Story or XmlStory.*
sourceStyleName	*string* sourceStyleName	*The source style name property.*

CellStyleMappings

Object available in ID 8

CELLSTYLEMAPPINGS CS6 VALUES	PROTOTYPE	DESCRIPTION
add	{CellStyleMapping} add(*string* SourceStyleName, *string* DestinationStyleName, #MapType MappingRuleType)	*Adds a style mapping. / Return value: The added CellStyleMapping. / SourceStyleName: The source style name. / DestinationStyleName: The destination style name. / MappingRuleType: The mapping type.*
anyItem	IDispatch anyItem()	*Returns any object in the collection. / Return value: The object.*
firstItem	IDispatch firstItem()	*Returns the first object in the collection. / Return value: The first object.*
itemByRange	{Objects} itemByRange(*variant* From, *variant* To)	*Returns the objects within the specified range. / Return value: The objects. / From: The object, index, or name at the beginning of the range. Type: Object, Long Integer or String. / To: The object, index, or name at the end of the range. Type: Object, Long Integer or String.*

CELLSTYLEMAPPINGS CS6 VALUES	PROTOTYPE	DESCRIPTION
lastItem	IDispatch lastItem()	*Returns the last object in the collection. / Return value: The last object.*
middleItem	IDispatch middleItem()	*Returns the middle object in the collection. / Return value: The middle object.*
nextItem	IDispatch nextItem(IDispatch Obj)	*Returns the object whose index follows the specified object in the collection. / Return value: The next object. / Obj: The object whose index comes before the desired object.*
previousItem	IDispatch previousItem(IDispatch Obj)	*Returns the object with the index previous to the specified index. / Return value: The previous object. / Obj: The index of the object that follows the desired object.*
count	*integer* •count	*Displays the number of elements in the CellStyleMapping.*
item	{CellStyleMapping} •item(*variant* Index)	*Returns the object with the specified index or name. Type: Long Integer or String.*

CharStyleMapping

Object available in ID 8

CHARSTYLEMAPPING CS6 VALUES	PROTOTYPE	DESCRIPTION
delete	*void* delete()	*deletes a style mapping.*
destinationStyleName	*string* destinationStyleName	*The destination style name property.*
index	*integer* •index	*The index of the CharStyleMapping within its containing object.*
mappingRuleType	#MapType mappingRuleType	*The type of the mapping rule.*
parent	*variant* •parent	*The parent of the CharStyleMapping (a Application, Document, Story or XmlStory). Type: Application, Document, Story or XmlStory.*
sourceStyleName	*string* sourceStyleName	*The source style name property.*

CharStyleMappings

Object available in ID 8

CHARSTYLEMAPPINGS CS6 VALUES	PROTOTYPE	DESCRIPTION
add	{CharStyleMapping} add(*string* SourceStyleName, *string* DestinationStyleName, #MapType MappingRuleType)	*Adds a style mapping. / Return value: The added CharStyleMapping. / SourceStyleName: The source style name. / DestinationStyleName: The destination style name. / MappingRuleType: The mapping type.*
anyItem	IDispatch anyItem()	*Returns any object in the collection. / Return value: The object.*
firstItem	IDispatch firstItem()	*Returns the first object in the collection. / Return value: The first object.*
itemByRange	{Objects} itemByRange(*variant* From, *variant* To)	*Returns the objects within the specified range. / Return value: The objects. / From: The object, index, or name at the beginning of the range. Type: Object, Long Integer or String. / To: The object, index, or name at the end of the range. Type: Object, Long Integer or String.*
lastItem	IDispatch lastItem()	*Returns the last object in the collection. / Return value: The last object.*

CHARSTYLEMAPPINGS CS6 VALUES	PROTOTYPE	DESCRIPTION
middleItem	IDispatch middleItem()	*Returns the middle object in the collection. / Return value: The middle object.*
nextItem	IDispatch nextItem(IDispatch Obj)	*Returns the object whose index follows the specified object in the collection. / Return value: The next object. / Obj: The object whose index comes before the desired object.*
previousItem	IDispatch previousItem(IDispatch Obj)	*Returns the object with the index previous to the specified index. / Return value: The previous object. / Obj: The index of the object that follows the desired object.*
count	*integer* •count	*Displays the number of elements in the CharStyleMapping.*
item	{CharStyleMapping} •item(*variant* Index)	*Returns the object with the specified index or name. Type: Long Integer or String.*

CheckBox

Object available in ID 8

CHECKBOX CS6 VALUES	PROTOTYPE	DESCRIPTION
applyObjectStyle	*void* applyObjectStyle(ObjectStyle Using, *boolean* ClearingOverrides, *boolean* ClearingOverridesThroughRootObjectStyle)	*Applies the specified object style. / Using: The object style to apply. / ClearingOverrides: If true, clears the CheckBox's existing attributes before applying the style. / ClearingOverridesThroughRootObjectStyle: If true, clears attributes and formatting applied to the CheckBox that are not defined in the object style.*
asynchronousExportFile	{BackgroundTask} asynchronousExportFile(*variant* Format, *string* To, *boolean* ShowingOptions, *variant* Using, *variant* VersionComments, *boolean* ForceSave)	*asynchronously exports the object(s) to a file. / Return value: object representing the asynchronous export. / Format: The export format, specified as an enumeration value or as an extension that appears in the Save as type or Format menu in the Export dialog. Type: idExportFormat enumerator or String. / To: The path to the export file. / ShowingOptions: If true, displays the export options dialog. / Using: The export style. Type: PDFExportPreset. / VersionComments: The comment for this version. Type: String. / ForceSave: If true, forcibly saves a version.*
autoTag	*void* autoTag()	*Tag the object or the parent story using default tags defined in XML preference.*
bringForward	*void* bringForward()	*Brings the CheckBox forward one level in its layer.*
bringToFront	*void* bringToFront(*variant* Reference)	*Brings the CheckBox to the front of its layer or in front of a particular item. / Reference: The reference object to bring the object in front of (must have same parent). Type: PageItem.*
changeObject	{Objects} changeObject(*variant* ReverseOrder)	*Finds objects that match the find what value and replace the objects with the change to value. / Return value: The replaced object(s). / ReverseOrder: If true, returns the results in reverse order. Type: Boolean.*
clearObjectStyleOverrides	*void* clearObjectStyleOverrides()	*Clear overrides for object style.*
clearTransformations	*void* clearTransformations()	*Clears transformations from the CheckBox. Transformations include rotation, scaling, flipping, fitting, and shearing.*

CHECKBOX CS6 VALUES	PROTOTYPE	DESCRIPTION
contentPlace	*variant* contentPlace(Objects PageItems, *boolean* LinkPageItems, *boolean* LinkStories, *boolean* MapStyles, *boolean* ShowingOptions)	*Duplicate an object and place it into the target page item. / Return value: The placed content. Type: Array of PageItems, Graphics, Stories or Texts. / PageItems: One or more page items to place or load. / LinkPageItems: Whether to link pageItems in content placer (if true it will override link stories value). / LinkStories: Whether to link stories in content placer (only applicable for single story, pageItem links will also be created in case of more than one item). / MapStyles: Whether to map styles in content placer. / ShowingOptions: Whether to display the link options dialog.*
convertShape	*void* convertShape(#ConvertShapeOptions Given, *variant* NumberOfSides, *variant* InsetPercentage, *variant* CornerRadius)	*Converts the CheckBox to a different shape. / Given: The CheckBox's new shape. / NumberOfSides: The number of sides for the resulting polygon. (Range: 3 to 100). Type: Long Integer (3 - 100). / InsetPercentage: The star inset percentage for the resulting polygon. (Range: 0.0 to 100.0) . Type: Long Integer (0 - 100). / CornerRadius: The corner radius of the resulting rectangle. Type: Unit (Double or String) (0 - 800 points).*
convertToObject	*void* convertToObject()	*Converts the button object to the page item currently in the active state. Page items from other states will be lost.*
delete	*void* delete()	*Deletes the CheckBox.*
detach	*void* detach()	*Detaches an overridden master page item from the master page.*
duplicate	{PageItem} duplicate(*variant* To, *variant* By)	*Duplicates the CheckBox at the specified location or offset. / Return value: The duplicated CheckBox. / To: The location of the new CheckBox, specified in coordinates in the format [x, y]. Type: Array of 2 Units (Doubles or Strings), Spread, Page or Layer. / By: Amount by which to offset the new CheckBox from the original CheckBox's position. Type: Array of 2 Units (Doubles or Strings).*
export	*void* export(*variant* Format, *string* To, *boolean* ShowingOptions, *variant* Using, *variant* VersionComments, *boolean* ForceSave)	*Exports the object(s) to a file. / Format: The export format, specified as an enumeration value or as an extension that appears in the Save as type or Format menu in the Export dialog. Type: idExportFormat enumerator or String. / To: The path to the export file. / ShowingOptions: If true, displays the export options dialog. / Using: The export style. Type: PDFExportPreset. / VersionComments: The comment for this version. Type: String. / ForceSave: If true, forcibly saves a version.*
extractLabel	*string* extractLabel(*string* Key)	*Gets the label value associated with the specified key. / Return value: The label. / Key: The key.*
findObject	{Objects} findObject(*variant* ReverseOrder)	*Finds objects that match the find what value. / Return value: The found object(s). / ReverseOrder: If true, returns the results in reverse order. Type: Boolean.*
fit	*void* fit(#FitOptions Given)	*Applies the specified fit option to content in a frame. / Given: The fit option to use.*
flipItem	*void* flipItem(#Flip Given, *variant* Around)	*Flips the CheckBox. / Given: The axis around which to flip the CheckBox. / Around: The point around which to flip the CheckBox. Type: Array of 2 Units (Doubles or Strings) or idAnchorPoint enumerator.*
insertLabel	*void* insertLabel(*string* Key, *string* Value)	*Sets the label to the value associated with the specified key. / Key: The key. / Value: The value.*
markup	*void* markup(XMLElement Using)	*Associates the page item with the specified XML element while preserving existing content. / Using: The XML element.*

CHECKBOX CS6 VALUES	PROTOTYPE	DESCRIPTION
move	*void* move(*variant* To, *variant* By)	*Moves the CheckBox to a new location. Note: Either the 'to' or 'by' parameter is required; if both parameters are defined, only the to value is used. / To: The new location of the CheckBox,in the format (x, y). Type: Array of 2 Units (Doubles or Strings), Spread, Page or Layer. / By: The amount (in measurement units) to move the CheckBox relative to its current position, in the format (x, y). Type: Array of 2 Units (Doubles or Strings).*
override	*variant* override(Page DestinationPage)	*Overrides a master page item and places the item on the document page as a new object. / Return value: The new object. Type: PageItem, Guide, Graphic, Movie or Sound. / DestinationPage: The document page that contains the master page item to override.*
placeXML	*void* placeXML(XMLElement Using)	*Places XML content into the specified object. Note: Replaces any existing content. / Using: The XML element whose content you want to place.*
redefineScaling	*void* redefineScaling(*variant* To)	*Apply an item's scaling to its content if possible. / To: The scale factors to be left on the item. The default is {1.0, 1.0}. Type: Array of 2 Doubles.*
reframe	*void* reframe(*variant* In, *variant* OpposingCorners)	*Move the bounding box of the page item. / In: The bounding box to resize. Type: idCoordinateSpaces enumerator or Ordered array containing CoordinateSpace:idCoordinateSpaces enumerator, BoundsKind:idBoundingBoxLimits enumerator. / OpposingCorners: Opposing corners of new bounding box in the given coordinate space. Type: Array of 2 Arrays of 2 Doubles.*
removeOverride	*void* removeOverride()	*Removes the override from a previously overridden master page item.*
resize	*void* resize(*variant* In, *variant* From, #ResizeMethods By, *variant* Values, *boolean* ResizeIndividually, *boolean* ConsideringRulerUnits)	*Resize the page item. / In: The bounding box to resize. Type: idCoordinateSpaces enumerator, idBoundingBoxLimits enumerator or Ordered array containing CoordinateSpace:idCoordinateSpaces enumerator, BoundsKind:idBoundingBoxLimits enumerator. / From: The transform origin. Legal specifications: relative to bounding box: anchor | {anchor | {x,y}, bounds kind [, coordinate space]}; relative to coordinate space: {x,y} | {{x,y} [, coordinate space]}; relative to layout window ruler: {{x,y}, page index | bounds kind}. Type: Array of 2 Doubles, idAnchorPoint enumerator or Array of Arrays of 2 Doubles, idCoordinateSpaces enumerators, idAnchorPoint enumerators, idBoundingBoxLimits enumerators or Long Integers. / By: How the current dimensions are affected by the given values. / Values: The width and height values. Legal dimensions specifications: {x, y [, coordinate space]}, {x, resize constraint [, coordinate space]}, or {resize constraint, y [, coordinate space]}; where x and y are real numbers and coordinate space is used to determine _only_ the unit of length for x and y; coordinate space is ignored for the 'current dimensions times' resize method). Type: Array of Doubles, idResizeConstraints enumerators or idCoordinateSpaces enumerators. / ResizeIndividually: If false and multiple page items are targeted, the new dimensions are attained only by moving the individual items rather than resizing them. / ConsideringRulerUnits: If true then a ruler location is interpreted using ruler units rather than points. The default value is false. This parameter has no effect unless the reference point is specified relative to a page.*
resolve	*variant* resolve(*variant* Location, #CoordinateSpaces In, *boolean* ConsideringRulerUnits)	*Get the coordinates of the given location in the specified coordinate system. / Return value: The resolved location for each of the targeted page items. Type: Array of 2 Doubles or Array of Arrays of 2 Doubles. / Location: The location requested. Type: Array of 2 Doubles, idAnchorPoint enumerator or Array of Arrays of 2 Doubles, idCoordinateSpaces enumerators, idAnchorPoint enumerators, idBoundingBoxLimits enumerators or Long Integers. / In: The coordinate space to use. / ConsideringRulerUnits: If true then a ruler location is interpreted using ruler units rather than points. The default value is false. This parameter has no effect unless the reference point is specified relative to a page.*

CHECKBOX CS6 VALUES	PROTOTYPE	DESCRIPTION
select	*void* select(#SelectionOptions ExistingSelection)	*Selects the object. / ExistingSelection: The selection status of the CheckBox in relation to previously selected objects.*
sendBackward	*void* sendBackward()	*Sends the CheckBox back one level in its layer.*
sendToBack	*void* sendToBack(*variant* Reference)	*Sends the CheckBox to the back of its layer or behind a particular item (must have same parent). / Reference: The reference object to send the object behind. Type: PageItem.*
store	{Asset} store(Library Using)	*Stores the object in the specified library. / Return value: The new library asset(s). / Using: The library in which to store the object.*
transform	*void* transform(#CoordinateSpaces In, *variant* From, *variant* WithMatrix, *variant* ReplacingCurrent, *boolean* ConsideringRulerUnits)	*Transform the page item. / In: The coordinate space to use. / From: The temporary origin during the transformation. Type: Array of 2 Doubles, idAnchorPoint enumerator or Array of Arrays of 2 Doubles, idCoordinateSpaces enumerators, idAnchorPoint enumerators, idBoundingBoxLimits enumerators or Long Integers. / WithMatrix: Transform matrix. Type: Array of 6 Doubles or TransformationMatrix. / ReplacingCurrent: Transform components to consider; providing this optional parameter causes the target's existing transform components to be replaced with new values. Without this parameter, the given matrix is concatenated onto the target's existing transform combining the effect of the two. Type: idMatrixContent enumerator, Array of idMatrixContent enumerators or Long Integer. / ConsideringRulerUnits: If true then a ruler based origin is interpreted using ruler units rather than points. The default value is false. This parameter has no effect unless the reference point is specified relative to a page.*
transformAgain	*variant* transformAgain()	*Transforms the CheckBox using the last transformation performed on any object. Transformations include moving, rotating, shearing, scaling, and flipping. . / Return value: Playback errors. Type: Array of Strings.*
transformAgainIndividually	*variant* transformAgainIndividually()	*Transforms the CheckBox using the last transformation performed on any CheckBox. Transformations include moving, rotating, shearing, scaling, and flipping. . / Return value: Playback errors. Type: Array of Strings.*
transformSequenceAgain	*variant* transformSequenceAgain()	*Transforms the CheckBox using the last sequence of transform operations performed on any single object or performed at the same time on any group of objects. Transformations include moving, rotating, shearing, scaling, and flipping. / Return value: Playback errors. Type: Array of Strings.*
transformSequenceAgainIndividually	*variant* transformSequenceAgainIndividually()	*Transforms the CheckBox using the last sequence of transformations performed on any single object or performed at the same time on any group of objects. Transformations include moving, rotating, shearing, scaling, and flipping. / Return value: Playback errors. Type: Array of Strings.*
transformValuesOf	{Objects} transformValuesOf(#CoordinateSpaces In)	*Get the transformation values of the page item. / Return value: A list of the transformation matrices. / In: The coordinate space to use.*
absoluteFlip	#Flip absoluteFlip	*Indicates whether the CheckBox has been flipped independently of its parent object and, if yes, the direction in which the CheckBox was flipped.*
absoluteHorizontalScale	*double* absoluteHorizontalScale	*The horizontal scale of the CheckBox relative to its containing object.*
absoluteRotationAngle	*double* absoluteRotationAngle	*The rotation angle of the CheckBox relative to its containing object. (Range: -360 to 360).*
absoluteShearAngle	*double* absoluteShearAngle	*The skewing angle of the CheckBox relative to its containing object. (Range: -360 to 360).*
absoluteVerticalScale	*double* absoluteVerticalScale	*The vertical scale of the CheckBox relative to its containing object.*
activeStateIndex	*integer* activeStateIndex	*The index of the active state in the object's states collection.*

CHECKBOX CS6 VALUES	PROTOTYPE	DESCRIPTION
allArticles	{Objects} •allArticles	*The list of all articles this page item is part of.*
allGraphics	{Objects} •allGraphics	*Lists all graphics contained by the CheckBox.*
allowOverrides	*boolean* allowOverrides	*If true, the master page item can be overridden.*
allPageItems	{Objects} •allPageItems	*Lists all page items contained by the CheckBox.*
anchoredObjectSettings	{AnchoredObjectSetting} •anchoredObjectSettings	*Anchored object settings.*
animationSettings	{AnimationSetting} •animationSettings	*The page item animation settings.*
appliedObjectStyle	{ObjectStyle} appliedObjectStyle	*The object style applied to the CheckBox.*
associatedXMLElement	{XMLItem} •associatedXMLElement	*The XML element associated with the CheckBox.*
behaviors	{Behaviors} •behaviors	*A collection of behavior objects.*
bottomLeftCornerOption	#CornerOptions bottomLeftCornerOption	*The shape to apply to the bottom left corner of rectangular shapes.*
bottomLeftCornerRadius	*variant* bottomLeftCornerRadius	*The radius in measurement units of the corner effect applied to the bottom left corner of rectangular shapes. Type: Unit (Double or String).*
bottomRightCornerOption	#CornerOptions bottomRightCornerOption	*The shape to apply to the bottom right corner of rectangular shapes.*
bottomRightCornerRadius	*variant* bottomRightCornerRadius	*The radius in measurement units of the corner effect applied to the bottom right corner of rectangular shapes. Type: Unit (Double or String).*
checkedByDefault	*boolean* checkedByDefault	*If true, the check box/radio button is selected by default in the exported PDF.*
clearFormBehaviors	{ClearFormBehaviors} •clearFormBehaviors	*A collection of clear form behavior objects.*
contentTransparencySettings	{ContentTransparencySetting} •contentTransparencySettings	*Transparency settings for the content of the CheckBox.*
description	*string* description	*The description of the CheckBox.*
endCap	#EndCap endCap	*The end shape of an open path.*
endJoin	#EndJoin endJoin	*The corner join applied to the CheckBox.*
epss	{EPSs} •epss	*A collection of EPS files.*
epsTexts	{EPSTexts} •epsTexts	*EPSTexts.*
exportValue	*string* exportValue	*Export value for the check box/radio button in the exported PDF.*
fillColor	*variant* fillColor	*The swatch (color, gradient, tint, or mixed ink) applied to the fill of the CheckBox. . Type: Swatch or String.*
fillTint	*double* fillTint	*The percent of tint to use in the CheckBox's fill color. (To specify a tint percent, use a number in the range of 0 to 100; to use the inherited or overridden value, use -1.).*
fillTransparencySettings	{FillTransparencySetting} •fillTransparencySettings	*Transparency settings for the fill applied to the CheckBox.*
flip	#Flip flip	*The direction in which to flip the printed image.*
gapColor	{Swatch} gapColor	*The swatch (color, gradient, tint, or mixed ink) applied to the gap of a dashed, dotted, or striped stroke. For information, see stroke type.*
gapTint	*double* gapTint	*The tint as a percentage of the gap color. (To specify a tint percent, use a number in the range of 0 to 100; to use the inherited or overridden value, use -1.).*
geometricBounds	*variant* geometricBounds	*The bounds of the CheckBox excluding the stroke width, in the format [y1, x1, y2, x2], which give the coordinates of the top-left and bottom-right corners of the bounding box. Type: Array of 4 Units (Doubles or Strings).*

CHECKBOX CS6 VALUES	PROTOTYPE	DESCRIPTION
gotoAnchorBehaviors	{GotoAnchorBehaviors} •gotoAnchorBehaviors	*A collection of goto anchor behavior objects.*
gotoFirstPageBehaviors	{GotoFirstPageBehaviors} •gotoFirstPageBehaviors	*A collection of goto first page behavior objects.*
gotoLastPageBehaviors	{GotoLastPageBehaviors} •gotoLastPageBehaviors	*A collection of goto last page behavior objects.*
gotoNextPageBehaviors	{GotoNextPageBehaviors} •gotoNextPageBehaviors	*A collection of goto next page behavior objects.*
gotoNextViewBehaviors	{GotoNextViewBehaviors} •gotoNextViewBehaviors	*A collection of goto next view behavior objects.*
gotoPreviousPageBehaviors	{GotoPreviousPageBehaviors} •gotoPreviousPageBehaviors	*A collection of goto previous page behavior objects.*
gotoPreviousViewBehaviors	{GotoPreviousViewBehaviors} •gotoPreviousViewBehaviors	*A collection of goto previous view behavior objects.*
gotoURLBehaviors	{GotoURLBehaviors} •gotoURLBehaviors	*A collection of goto URL behavior objects.*
gradientFillAngle	*double* gradientFillAngle	*The angle of a linear gradient applied to the fill of the CheckBox. (Range: -180 to 180).*
gradientFillLength	*variant* gradientFillLength	*The length (for a linear gradient) or radius (for a radial gradient) applied to the fill of the CheckBox. Type: Unit (Double or String).*
gradientFillStart	*variant* gradientFillStart	*The starting point (in page coordinates) of a gradient applied to the fill of the CheckBox, in the format [x, y]. Type: Array of 2 Units (Doubles or Strings).*
gradientStrokeAngle	*double* gradientStrokeAngle	*The angle of a linear gradient applied to the stroke of the CheckBox. (Range: -180 to 180).*
gradientStrokeLength	*variant* gradientStrokeLength	*The length (for a linear gradient) or radius (for a radial gradient) applied to the stroke of the CheckBox. Type: Unit (Double or String).*
gradientStrokeStart	*variant* gradientStrokeStart	*The starting point (in page coordinates) of a gradient applied to the stroke of the CheckBox, in the format [x, y]. Type: Array of 2 Units (Doubles or Strings).*
graphicLines	{GraphicLines} •graphicLines	*A collection of graphic lines.*
graphics	{Graphics} •graphics	*A collection of imported graphics in any graphic file format (vector, metafile, or bitmap).*
groups	{Groups} •groups	*A collection of groups.*
hiddenUntilTriggered	*boolean* hiddenUntilTriggered	*If true, the form field/push button is hidden until triggered in the exported PDF.*
horizontalLayoutConstraints	*variant* horizontalLayoutConstraints	*The left margin, width, and right margin constraints this item is subject to when using the object-based layout rule. Type: Array of 3 idDimensionsConstraints enumerators.*
horizontalScale	*double* horizontalScale	*The horizontal scaling applied to the CheckBox.*
id	*integer* •id	*The unique ID of the CheckBox.*
images	{Images} •images	*A collection of bitmap images in any bitmap file format (including TIFF, JPEG, or GIF).*
index	*integer* •index	*The index of the CheckBox within its containing object.*
itemLayer	{Layer} itemLayer	*The layer that the CheckBox is on.*
label	*string* label	*A property that can be set to any string.*
leftLineEnd	#ArrowHead leftLineEnd	*The arrowhead applied to the start of the path.*
linkedPageItemOptions	{LinkedPageItemOption} •linkedPageItemOptions	*Linked Page Item options.*
localDisplaySetting	#DisplaySettingOptions localDisplaySetting	*Display performance options for the CheckBox.*

CHECKBOX CS6 VALUES	PROTOTYPE	DESCRIPTION
locked	boolean locked	If true, the CheckBox is locked.
miterLimit	double miterLimit	The limit of the ratio of stroke width to miter length before a miter (pointed) join becomes a bevel (squared-off) join. Type: Double (1 - 500).
movieBehaviors	{MovieBehaviors} •movieBehaviors	A collection of movie behavior objects.
name	string name	The name of the CheckBox.
nonprinting	boolean nonprinting	If true, the CheckBox does not print.
openFileBehaviors	{OpenFileBehaviors} •openFileBehaviors	A collection of open file behavior objects.
ovals	{Ovals} •ovals	A collection of ellipses.
overprintFill	boolean overprintFill	If true, the CheckBox's fill color overprints any underlying objects. If false, the fill color knocks out the underlying colors.
overprintGap	boolean overprintGap	If true, the gap color overprints any underlying colors. If false, the gap color knocks out the underlying colors.
overprintStroke	boolean overprintStroke	If true, the CheckBox's stroke color overprints any underlying objects. If false, the stroke color knocks out the underlying colors.
overridden	boolean •overridden	If true, the object originated on a master spread and was overridden. If false, the object either originated on a master spread and was not overridden, or the object did not originate on a master page.
overriddenMasterPageItem	variant •overriddenMasterPageItem	An object that originated on a master page and has been overridden. Type: PageItem, Guide, Graphic, Movie or Sound.
pageItems	{PageItems} •pageItems	The page items collection, which can be used to process all page items in a container (such as a document, page, or group), regardless of type.
parent	variant •parent	The parent of the CheckBox (a Snippet, PlaceGun, Character, Spread, MasterSpread, SplineItem, Polygon, GraphicLine, Rectangle, Oval or Group). Type: Snippet, PlaceGun, Character, Spread, MasterSpread, SplineItem, Polygon, GraphicLine, Rectangle, Oval or Group.
parentPage	{Page} •parentPage	The page on which this page item appears.
pdfs	{PDFs} •pdfs	A collection of PDF files.
picts	{PICTs} •picts	A collection of PICT graphics.
polygons	{Polygons} •polygons	A collection of polygons.
preferences	{Preferences} •preferences	A collection of preferences objects.
printableInPdf	boolean printableInPdf	If true, the form field/push button is printable in the exported PDF.
printFormBehaviors	{PrintFormBehaviors} •printFormBehaviors	A collection of print form behavior objects.
readOnly	boolean readOnly	If true, the form field is read only in the exported PDF.
rectangles	{Rectangles} •rectangles	A collection of rectangles.
required	boolean required	If true, the form field is required in the exported PDF.
rightLineEnd	#ArrowHead rightLineEnd	The arrowhead applied to the end of the path.
rotationAngle	double rotationAngle	The rotatation angle of the CheckBox. (Range: -360 to 360).
shearAngle	double shearAngle	The skewing angle applied to the CheckBox. (Range: -360 to 360).
showHideFieldsBehaviors	{ShowHideFieldsBehaviors} •showHideFieldsBehaviors	A collection of show/hide fields behavior objects.
soundBehaviors	{SoundBehaviors} •soundBehaviors	A collection of sound behavior objects.

CHECKBOX CS6 VALUES	PROTOTYPE	DESCRIPTION
splineItems	{SplineItems} •splineItems	*The spline items collection.*
states	{States} •states	*A collection of states.*
strokeAlignment	#StrokeAlignment strokeAlignment	*The stroke alignment applied to the CheckBox.*
strokeColor	*variant* strokeColor	*The swatch (color, gradient, tint, or mixed ink) applied to the stroke of the CheckBox. Type: Swatch or String.*
strokeCornerAdjustment	#StrokeCornerAdjustment strokeCornerAdjustment	*The corner adjustment applied to the CheckBox.*
strokeDashAndGap	*variant* strokeDashAndGap	*The dash and gap measurements that define the pattern of a custom dashed line. Define up to six values (in points) in the format [dash1, gap1, dash2, gap2, dash3, gap3]. Type: Array of Units (Doubles or Strings).*
strokeTint	*double* strokeTint	*The percent of tint to use in object's stroke color. (To specify a tint percent, use a number in the range of 0 to 100; to use the inherited or overridden value, use -1.).*
strokeTransparencySettings	{StrokeTransparencySetting} •strokeTransparencySettings	*Transparency settings for the stroke.*
strokeType	*variant* strokeType	*The name of the stroke style to apply. Type: StrokeStyle or String.*
strokeWeight	*variant* strokeWeight	*The weight (in points) to apply to the CheckBox's stroke. Type: Unit (Double or String).*
submitFormBehaviors	{SubmitFormBehaviors} •submitFormBehaviors	*A collection of submit form behavior objects.*
textFrames	{TextFrames} •textFrames	*A collection of text frames.*
textWrapPreferences	{TextWrapPreference} •textWrapPreferences	*The text wrap preference properties that define the default formatting for wrapping text around objects.*
timingSettings	{TimingSetting} •timingSettings	*The object timing settings.*
topLeftCornerOption	#CornerOptions topLeftCornerOption	*The shape to be applied to the top left corner of rectangular shapes and all corners of non-rectangular shapes.Note: corner option differs from end join in which you can set a radius for a corner option, whereas the rounded or beveled effect of an end join depends on the stroke weight.*
topLeftCornerRadius	*variant* topLeftCornerRadius	*The radius in measurement units of the corner effect applied to the top left corner of rectangular shapes and all corners of non-rectangular shapes. Type: Unit (Double or String).*
topRightCornerOption	#CornerOptions topRightCornerOption	*The shape to apply to the top right corner of rectangular shapes.*
topRightCornerRadius	*variant* topRightCornerRadius	*The radius in measurement units of the corner effect applied to the top right corner of rectangular shapes. Type: Unit (Double or String).*
transparencySettings	{TransparencySetting} •transparencySettings	*Transparency settings.*
verticalLayoutConstraints	*variant* verticalLayoutConstraints	*The top margin, height, and bottom margin constraints this item is subject to when using the object-based layout rule. Type: Array of 3 idDimensionsConstraints enumerators.*
verticalScale	*double* verticalScale	*The vertical scaling applied to the CheckBox.*
viewZoomBehaviors	{ViewZoomBehaviors} •viewZoomBehaviors	*A collection of view zoom behavior objects.*
visible	*boolean* visible	*If true, the CheckBox is visible.*
visibleBounds	*variant* visibleBounds	*The bounds of the CheckBox including the stroke width, in the format [y1, x1, y2, x2], which give the coordinates of the top-left and bottom-right corners of the bounding box. Type: Array of 4 Units (Doubles or Strings).*
wmfs	{WMFs} •wmfs	*A collection of WMF graphics.*

CheckBoxes

Object available in ID 8

CHECKBOXES CS6 VALUES	PROTOTYPE	DESCRIPTION
add	{CheckBox} add(*variant* Layer, #LocationOptions At, *variant* Reference)	*Creates a new CheckBox. / Return value: The new CheckBox. / Layer: The layer on which to create the CheckBox. Type: Layer. / At: The location at which to insert the CheckBox relative to the reference object or within the container object. / Reference: The reference object. Note: Required when the at parameter is before or after. Type: Document, Spread, MasterSpread, Page, Layer or PageItem.*
anyItem	IDispatch anyItem()	*Returns any object in the collection. / Return value: The object.*
firstItem	IDispatch firstItem()	*Returns the first object in the collection. / Return value: The first object.*
itemByID	IDispatch itemByID(*integer* Id)	*Returns the object with the specified ID. / Return value: The object. / ID: The ID.*
itemByRange	{Objects} itemByRange(*variant* From, *variant* To)	*Returns the objects within the specified range. / Return value: The objects. / From: The object, index, or name at the beginning of the range. Type: Object, Long Integer or String. / To: The object, index, or name at the end of the range. Type: Object, Long Integer or String.*
lastItem	IDispatch lastItem()	*Returns the last object in the collection. / Return value: The last object.*
middleItem	IDispatch middleItem()	*Returns the middle object in the collection. / Return value: The middle object.*
nextItem	IDispatch nextItem(IDispatch Obj)	*Returns the object whose index follows the specified object in the collection. / Return value: The next object. / Obj: The object whose index comes before the desired object.*
previousItem	IDispatch previousItem(IDispatch Obj)	*Returns the object with the index previous to the specified index. / Return value: The previous object. / Obj: The index of the object that follows the desired object.*
count	*integer* •count	*Displays the number of elements in the CheckBox.*
item	{CheckBox} •item(*variant* Index)	*Returns the object with the specified index or name. Type: Long Integer or String.*

CheckBoxEvents

Object available in ID 8

CHECKBOXEVENTS CS6 VALUES	PROTOTYPE	DESCRIPTION
afterPlace	*const* afterPlace(afterPlace)	*Dispatched after a CheckBox is placed. This event bubbles. This event is not cancelable.*
beforePlace	*const* beforePlace(beforePlace)	*Dispatched before a CheckBox is placed. This event bubbles. This event is cancelable.*

ClearFormBehavior

Object available in ID 8

CLEARFORMBEHAVIOR CS6 VALUES	PROTOTYPE	DESCRIPTION
delete	*void* delete()	*Deletes the ClearFormBehavior.*
extractLabel	*string* extractLabel(*string* Key)	*Gets the label value associated with the specified key. / Return value: The label. / Key: The key.*
insertLabel	*void* insertLabel(*string* Key, *string* Value)	*Sets the label to the value associated with the specified key. / Key: The key. / Value: The value.*
behaviorEvent	#BehaviorEvents behaviorEvent	*The event which triggers the behavior.*
enableBehavior	*boolean* enableBehavior	*If true, the behavior is enabled.*
id	*integer* •id	*The unique ID of the ClearFormBehavior.*
index	*integer* •index	*The index of the ClearFormBehavior within its containing object.*
label	*string* label	*A property that can be set to any string.*
name	*string* •name	*The name of the ClearFormBehavior.*
parent	*variant* •parent	*The parent of the ClearFormBehavior (a Button, CheckBox, ComboBox, ListBox, RadioButton, TextBox or SignatureField). Type: Button, CheckBox, ComboBox, ListBox, RadioButton, TextBox or SignatureField.*

ClearFormBehaviors

Object available in ID 8

CLEARFORMBEHAVIORS CS6 VALUES	PROTOTYPE	DESCRIPTION
add	{ClearFormBehavior} add()	*Creates a new ClearFormBehavior. / Return value: The new ClearFormBehavior.*
anyItem	IDispatch anyItem()	*Returns any object in the collection. / Return value: The object.*
firstItem	IDispatch firstItem()	*Returns the first object in the collection. / Return value: The first object.*
itemByID	IDispatch itemByID(*integer* Id)	*Returns the object with the specified ID. / Return value: The object. / ID: The ID.*
itemByRange	{Objects} itemByRange(*variant* From, *variant* To)	*Returns the objects within the specified range. / Return value: The objects. / From: The object, index, or name at the beginning of the range. Type: Object, Long Integer or String. / To: The object, index, or name at the end of the range. Type: Object, Long Integer or String.*
lastItem	IDispatch lastItem()	*Returns the last object in the collection. / Return value: The last object.*
middleItem	IDispatch middleItem()	*Returns the middle object in the collection. / Return value: The middle object.*
nextItem	IDispatch nextItem(IDispatch Obj)	*Returns the object whose index follows the specified object in the collection. / Return value: The next object. / Obj: The object whose index comes before the desired object.*
previousItem	IDispatch previousItem(IDispatch Obj)	*Returns the object with the index previous to the specified index. / Return value: The previous object. / Obj: The index of the object that follows the desired object.*

CLEARFORMBEHAVIORS CS6 VALUES	PROTOTYPE	DESCRIPTION
count	*integer* •count	*Displays the number of elements in the ClearFormBehavior.*
item	{ClearFormBehavior} •item(*variant* Index)	*Returns the object with the specified index or name. Type: Long Integer or String.*

ComboBox

Object available in ID 8

COMBOBOX CS6 VALUES	PROTOTYPE	DESCRIPTION
applyObjectStyle	*void* applyObjectStyle(ObjectStyle Using, *boolean* ClearingOverrides, *boolean* ClearingOverridesThroughRootObjectStyle)	*Applies the specified object style. / Using: The object style to apply. / ClearingOverrides: If true, clears the ComboBox's existing attributes before applying the style. / ClearingOverridesThroughRootObjectStyle: If true, clears attributes and formatting applied to the ComboBox that are not defined in the object style.*
asynchronousExportFile	{BackgroundTask} asynchronousExportFile(*variant* Format, *string* To, *boolean* ShowingOptions, *variant* Using, *variant* VersionComments, *boolean* ForceSave)	*asynchronously exports the object(s) to a file. / Return value: object representing the asynchronous export. / Format: The export format, specified as an enumeration value or as an extension that appears in the Save as type or Format menu in the Export dialog. Type: idExportFormat enumerator or String. / To: The path to the export file. / ShowingOptions: If true, displays the export options dialog. / Using: The export style. Type: PDFExportPreset. / VersionComments: The comment for this version. Type: String. / ForceSave: If true, forcibly saves a version.*
autoTag	*void* autoTag()	*Tag the object or the parent story using default tags defined in XML preference.*
bringForward	*void* bringForward()	*Brings the ComboBox forward one level in its layer.*
bringToFront	*void* bringToFront(*variant* Reference)	*Brings the ComboBox to the front of its layer or in front of a particular item. / Reference: The reference object to bring the object in front of (must have same parent). Type: PageItem.*
changeObject	{Objects} changeObject(*variant* ReverseOrder)	*Finds objects that match the find what value and replace the objects with the change to value. / Return value: The replaced object(s). / ReverseOrder: If true, returns the results in reverse order. Type: Boolean.*
clearObjectStyleOverrides	*void* clearObjectStyleOverrides()	*Clear overrides for object style.*
clearTransformations	*void* clearTransformations()	*Clears transformations from the ComboBox. Transformations include rotation, scaling, flipping, fitting, and shearing.*
contentPlace	*variant* contentPlace(Objects PageItems, *boolean* LinkPageItems, *boolean* LinkStories, *boolean* MapStyles, *boolean* ShowingOptions)	*Duplicate an object and place it into the target page item. / Return value: The placed content. Type: Array of PageItems, Graphics, Stories or Texts. / PageItems: One or more page items to place or load. / LinkPageItems: Whether to link pageItems in content placer (if true it will override link stories value). / LinkStories: Whether to link stories in content placer (only applicable for single story, pageItem links will also be created in case of more than one item). / MapStyles: Whether to map styles in content placer. / ShowingOptions: Whether to display the link options dialog.*
convertShape	*void* convertShape(#ConvertShapeOptions Given, *variant* NumberOfSides, *variant* InsetPercentage, *variant* CornerRadius)	*Converts the ComboBox to a different shape. / Given: The ComboBox's new shape. / NumberOfSides: The number of sides for the resulting polygon. (Range: 3 to 100). Type: Long Integer (3 - 100). / InsetPercentage: The star inset percentage for the resulting polygon. (Range: 0.0 to 100.0) . Type: Long Integer (0 - 100). / CornerRadius: The corner radius of the resulting rectangle. Type: Unit (Double or String) (0 - 800 points).*

COMBOBOX CS6 VALUES	PROTOTYPE	DESCRIPTION
convertToObject	*void* convertToObject()	*Converts the button object to the page item currently in the active state. Page items from other states will be lost.*
delete	*void* delete()	*Deletes the ComboBox.*
detach	*void* detach()	*Detaches an overridden master page item from the master page.*
duplicate	{PageItem} duplicate(*variant* To, *variant* By)	*Duplicates the ComboBox at the specified location or offset. / Return value: The duplicated ComboBox. / To: The location of the new ComboBox, specified in coordinates in the format [x, y]. Type: Array of 2 Units (Doubles or Strings), Spread, Page or Layer. / By: Amount by which to offset the new ComboBox from the original ComboBox's position. Type: Array of 2 Units (Doubles or Strings).*
export	*void* export(*variant* Format, *string* To, *boolean* ShowingOptions, *variant* Using, *variant* VersionComments, *boolean* ForceSave)	*Exports the object(s) to a file. / Format: The export format, specified as an enumeration value or as an extension that appears in the Save as type or Format menu in the Export dialog. Type: idExportFormat enumerator or String. / To: The path to the export file. / ShowingOptions: If true, displays the export options dialog. / Using: The export style. Type: PDFExportPreset. / VersionComments: The comment for this version. Type: String. / ForceSave: If true, forcibly saves a version.*
extractLabel	*string* extractLabel(*string* Key)	*Gets the label value associated with the specified key. / Return value: The label. / Key: The key.*
findObject	{Objects} findObject(*variant* ReverseOrder)	*Finds objects that match the find what value. / Return value: The found object(s). / ReverseOrder: If true, returns the results in reverse order. Type: Boolean.*
fit	*void* fit(#FitOptions Given)	*Applies the specified fit option to content in a frame. / Given: The fit option to use.*
flipItem	*void* flipItem(#Flip Given, *variant* Around)	*Flips the ComboBox. / Given: The axis around which to flip the ComboBox. / Around: The point around which to flip the ComboBox. Type: Array of 2 Units (Doubles or Strings) or idAnchorPoint enumerator.*
insertLabel	*void* insertLabel(*string* Key, *string* Value)	*Sets the label to the value associated with the specified key. / Key: The key. / Value: The value.*
markup	*void* markup(XMLElement Using)	*Associates the page item with the specified XML element while preserving existing content. / Using: The XML element.*
move	*void* move(*variant* To, *variant* By)	*Moves the ComboBox to a new location. Note: Either the 'to' or 'by' parameter is required; if both parameters are defined, only the to value is used. / To: The new location of the ComboBox,in the format (x, y). Type: Array of 2 Units (Doubles or Strings), Spread, Page or Layer. / By: The amount (in measurement units) to move the ComboBox relative to its current position, in the format (x, y). Type: Array of 2 Units (Doubles or Strings).*
override	*variant* override(Page DestinationPage)	*Overrides a master page item and places the item on the document page as a new object. / Return value: The new object. Type: PageItem, Guide, Graphic, Movie or Sound. / DestinationPage: The document page that contains the master page item to override.*
placeXML	*void* placeXML(XMLElement Using)	*Places XML content into the specified object. Note: Replaces any existing content. / Using: The XML element whose content you want to place.*
redefineScaling	*void* redefineScaling(*variant* To)	*Apply an item's scaling to its content if possible. / To: The scale factors to be left on the item. The default is {1.0, 1.0}. Type: Array of 2 Doubles.*

COMBOBOX CS6 VALUES	PROTOTYPE	DESCRIPTION
reframe	*void* reframe(*variant* In, *variant* OpposingCorners)	*Move the bounding box of the page item. / In: The bounding box to resize. Type: idCoordinateSpaces enumerator or Ordered array containing CoordinateSpace:idCoordinateSpaces enumerator, BoundsKind:idBoundingBoxLimits enumerator. / OpposingCorners: Opposing corners of new bounding box in the given coordinate space. Type: Array of 2 Arrays of 2 Doubles.*
removeOverride	*void* removeOverride()	*Removes the override from a previously overridden master page item.*
resize	*void* resize(*variant* In, *variant* From, #ResizeMethods By, *variant* Values, *boolean* ResizeIndividually, *boolean* ConsideringRulerUnits)	*Resize the page item. / In: The bounding box to resize. Type: idCoordinateSpaces enumerator, idBoundingBoxLimits enumerator or Ordered array containing CoordinateSpace:idCoordinateSpaces enumerator, BoundsKind:idBoundingBoxLimits enumerator. / From: The transform origin. Legal specifications: relative to bounding box: anchor | {anchor | {x,y}, bounds kind [, coordinate space]}; relative to coordinate space: {x,y} | {{x,y} [, coordinate space]}; relative to layout window ruler: {{x,y}, page index | bounds kind}. Type: Array of 2 Doubles, idAnchorPoint enumerator or Array of Arrays of 2 Doubles, idCoordinateSpaces enumerators, idAnchorPoint enumerators, idBoundingBoxLimits enumerators or Long Integers. / By: How the current dimensions are affected by the given values. / Values: The width and height values. Legal dimensions specifications: {x, y [, coordinate space]}, {x, resize constraint [, coordinate space]}, or {resize constraint, y [, coordinate space]}; where x and y are real numbers and coordinate space is used to determine _only_ the unit of length for x and y; coordinate space is ignored for the 'current dimensions times' resize method). Type: Array of Doubles, idResizeConstraints enumerators or idCoordinateSpaces enumerators. / ResizeIndividually: If false and multiple page items are targeted, the new dimensions are attained only by moving the individual items rather than resizing them. / ConsideringRulerUnits: If true then a ruler location is interpreted using ruler units rather than points. The default value is false. This parameter has no effect unless the reference point is specified relative to a page.*
resolve	*variant* resolve(*variant* Location, #CoordinateSpaces In, *boolean* ConsideringRulerUnits)	*Get the coordinates of the given location in the specified coordinate system. / Return value: The resolved location for each of the targeted page items. Type: Array of 2 Doubles or Array of Arrays of 2 Doubles. / Location: The location requested. Type: Array of 2 Doubles, idAnchorPoint enumerator or Array of Arrays of 2 Doubles, idCoordinateSpaces enumerators, idAnchorPoint enumerators, idBoundingBoxLimits enumerators or Long Integers. / In: The coordinate space to use. / ConsideringRulerUnits: If true then a ruler location is interpreted using ruler units rather than points. The default value is false. This parameter has no effect unless the reference point is specified relative to a page.*
select	*void* select(#SelectionOptions ExistingSelection)	*Selects the object. / ExistingSelection: The selection status of the ComboBox in relation to previously selected objects.*
sendBackward	*void* sendBackward()	*Sends the ComboBox back one level in its layer.*
sendToBack	*void* sendToBack(*variant* Reference)	*Sends the ComboBox to the back of its layer or behind a particular item (must have same parent). / Reference: The reference object to send the object behind. Type: PageItem.*
store	{Asset} store(Library Using)	*Stores the object in the specified library. / Return value: The new library asset(s). / Using: The library in which to store the object.*

COMBOBOX CS6 VALUES	PROTOTYPE	DESCRIPTION
transform	void transform(#CoordinateSpaces In, variant From, variant WithMatrix, variant ReplacingCurrent, boolean ConsideringRulerUnits)	*Transform the page item. / In: The coordinate space to use. / From: The temporary origin during the transformation. Type: Array of 2 Doubles, idAnchorPoint enumerator or Array of Arrays of 2 Doubles, idCoordinateSpaces enumerators, idAnchorPoint enumerators, idBoundingBoxLimits enumerators or Long Integers. / WithMatrix: Transform matrix. Type: Array of 6 Doubles or TransformationMatrix. / ReplacingCurrent: Transform components to consider; providing this optional parameter causes the target's existing transform components to be replaced with new values. Without this parameter, the given matrix is concatenated onto the target's existing transform combining the effect of the two. Type: idMatrixContent enumerator, Array of idMatrixContent enumerators or Long Integer. / ConsideringRulerUnits: If true then a ruler based origin is interpreted using ruler units rather than points. The default value is false. This parameter has no effect unless the reference point is specified relative to a page.*
transformAgain	variant transformAgain()	*Transforms the ComboBox using the last transformation performed on any object. Transformations include moving, rotating, shearing, scaling, and flipping. . / Return value: Playback errors. Type: Array of Strings.*
transformAgainIndividually	variant transformAgainIndividually()	*Transforms the ComboBox using the last transformation performed on any ComboBox. Transformations include moving, rotating, shearing, scaling, and flipping. . / Return value: Playback errors. Type: Array of Strings.*
transformSequenceAgain	variant transformSequenceAgain()	*Transforms the ComboBox using the last sequence of transform operations performed on any single object or performed at the same time on any group of objects. Transformations include moving, rotating, shearing, scaling, and flipping. / Return value: Playback errors. Type: Array of Strings.*
transformSequenceAgainIndividually	variant transformSequenceAgainIndividually()	*Transforms the ComboBox using the last sequence of transformations performed on any single object or performed at the same time on any group of objects. Transformations include moving, rotating, shearing, scaling, and flipping. / Return value: Playback errors. Type: Array of Strings.*
transformValuesOf	{Objects} transformValuesOf(#CoordinateSpaces In)	*Get the transformation values of the page item. / Return value: A list of the transformation matrices. / In: The coordinate space to use.*
absoluteFlip	#Flip absoluteFlip	*Indicates whether the ComboBox has been flipped independently of its parent object and, if yes, the direction in which the ComboBox was flipped.*
absoluteHorizontalScale	double absoluteHorizontalScale	*The horizontal scale of the ComboBox relative to its containing object.*
absoluteRotationAngle	double absoluteRotationAngle	*The rotation angle of the ComboBox relative to its containing object. (Range: -360 to 360).*
absoluteShearAngle	double absoluteShearAngle	*The skewing angle of the ComboBox relative to its containing object. (Range: -360 to 360).*
absoluteVerticalScale	double absoluteVerticalScale	*The vertical scale of the ComboBox relative to its containing object.*
activeStateIndex	integer activeStateIndex	*The index of the active state in the object's states collection.*
allArticles	{Objects} •allArticles	*The list of all articles this page item is part of.*
allGraphics	{Objects} •allGraphics	*Lists all graphics contained by the ComboBox.*
allowOverrides	boolean allowOverrides	*If true, the master page item can be overridden.*
allPageItems	{Objects} •allPageItems	*Lists all page items contained by the ComboBox.*
anchoredObjectSettings	{AnchoredObjectSetting} •anchoredObjectSettings	*Anchored object settings.*
animationSettings	{AnimationSetting} •animationSettings	*The page item animation settings.*
appliedObjectStyle	{ObjectStyle} appliedObjectStyle	*The object style applied to the ComboBox.*

COMBOBOX CS6 VALUES	PROTOTYPE	DESCRIPTION
associatedXMLElement	{XMLItem} •associatedXMLElement	*The XML element associated with the ComboBox.*
behaviors	{Behaviors} •behaviors	*A collection of behavior objects.*
bottomLeftCornerOption	#CornerOptions bottomLeftCornerOption	*The shape to apply to the bottom left corner of rectangular shapes.*
bottomLeftCornerRadius	*variant* bottomLeftCornerRadius	*The radius in measurement units of the corner effect applied to the bottom left corner of rectangular shapes. Type: Unit (Double or String).*
bottomRightCornerOption	#CornerOptions bottomRightCornerOption	*The shape to apply to the bottom right corner of rectangular shapes.*
bottomRightCornerRadius	*variant* bottomRightCornerRadius	*The radius in measurement units of the corner effect applied to the bottom right corner of rectangular shapes. Type: Unit (Double or String).*
choiceList	*variant* choiceList	*The list of choices for combo and list boxes in the exported PDF. Type: Array of Strings.*
clearFormBehaviors	{ClearFormBehaviors} •clearFormBehaviors	*A collection of clear form behavior objects.*
contentTransparencySettings	{ContentTransparencySetting} •contentTransparencySettings	*Transparency settings for the content of the ComboBox.*
description	*string* description	*The description of the ComboBox.*
endCap	#EndCap endCap	*The end shape of an open path.*
endJoin	#EndJoin endJoin	*The corner join applied to the ComboBox.*
epss	{EPSs} •epss	*A collection of EPS files.*
epsTexts	{EPSTexts} •epsTexts	*EPSTexts.*
fillColor	*variant* fillColor	*The swatch (color, gradient, tint, or mixed ink) applied to the fill of the ComboBox. . Type: Swatch or String.*
fillTint	*double* fillTint	*The percent of tint to use in the ComboBox's fill color. (To specify a tint percent, use a number in the range of 0 to 100; to use the inherited or overridden value, use -1.).*
fillTransparencySettings	{FillTransparencySetting} •fillTransparencySettings	*Transparency settings for the fill applied to the ComboBox.*
flip	#Flip flip	*The direction in which to flip the printed image.*
fontSize	*double* fontSize	*The font size for the form field in the exported PDF.*
gapColor	{Swatch} gapColor	*The swatch (color, gradient, tint, or mixed ink) applied to the gap of a dashed, dotted, or striped stroke. For information, see stroke type.*
gapTint	*double* gapTint	*The tint as a percentage of the gap color. (To specify a tint percent, use a number in the range of 0 to 100; to use the inherited or overridden value, use -1.).*
geometricBounds	*variant* geometricBounds	*The bounds of the ComboBox excluding the stroke width, in the format [y1, x1, y2, x2], which give the coordinates of the top-left and bottom-right corners of the bounding box. Type: Array of 4 Units (Doubles or Strings).*
gotoAnchorBehaviors	{GotoAnchorBehaviors} •gotoAnchorBehaviors	*A collection of goto anchor behavior objects.*
gotoFirstPageBehaviors	{GotoFirstPageBehaviors} •gotoFirstPageBehaviors	*A collection of goto first page behavior objects.*
gotoLastPageBehaviors	{GotoLastPageBehaviors} •gotoLastPageBehaviors	*A collection of goto last page behavior objects.*
gotoNextPageBehaviors	{GotoNextPageBehaviors} •gotoNextPageBehaviors	*A collection of goto next page behavior objects.*
gotoNextViewBehaviors	{GotoNextViewBehaviors} •gotoNextViewBehaviors	*A collection of goto next view behavior objects.*
gotoPreviousPageBehaviors	{GotoPreviousPageBehaviors} •gotoPreviousPageBehaviors	*A collection of goto previous page behavior objects.*
gotoPreviousViewBehaviors	{GotoPreviousViewBehaviors} •gotoPreviousViewBehaviors	*A collection of goto previous view behavior objects.*

COMBOBOX CS6 VALUES	PROTOTYPE	DESCRIPTION
gotoURLBehaviors	{GotoURLBehaviors} •gotoURLBehaviors	*A collection of goto URL behavior objects.*
gradientFillAngle	*double* gradientFillAngle	*The angle of a linear gradient applied to the fill of the ComboBox. (Range: -180 to 180).*
gradientFillLength	*variant* gradientFillLength	*The length (for a linear gradient) or radius (for a radial gradient) applied to the fill of the ComboBox. Type: Unit (Double or String).*
gradientFillStart	*variant* gradientFillStart	*The starting point (in page coordinates) of a gradient applied to the fill of the ComboBox, in the format [x, y]. Type: Array of 2 Units (Doubles or Strings).*
gradientStrokeAngle	*double* gradientStrokeAngle	*The angle of a linear gradient applied to the stroke of the ComboBox. (Range: -180 to 180).*
gradientStrokeLength	*variant* gradientStrokeLength	*The length (for a linear gradient) or radius (for a radial gradient) applied to the stroke of the ComboBox. Type: Unit (Double or String).*
gradientStrokeStart	*variant* gradientStrokeStart	*The starting point (in page coordinates) of a gradient applied to the stroke of the ComboBox, in the format [x, y]. Type: Array of 2 Units (Doubles or Strings).*
graphicLines	{GraphicLines} •graphicLines	*A collection of graphic lines.*
graphics	{Graphics} •graphics	*A collection of imported graphics in any graphic file format (vector, metafile, or bitmap).*
groups	{Groups} •groups	*A collection of groups.*
hiddenUntilTriggered	*boolean* hiddenUntilTriggered	*If true, the form field/push button is hidden until triggered in the exported PDF.*
horizontalLayoutConstraints	*variant* horizontalLayoutConstraints	*The left margin, width, and right margin constraints this item is subject to when using the object-based layout rule. Type: Array of 3 idDimensionsConstraints enumerators.*
horizontalScale	*double* horizontalScale	*The horizontal scaling applied to the ComboBox.*
id	*integer* •id	*The unique ID of the ComboBox.*
images	{Images} •images	*A collection of bitmap images in any bitmap file format (including TIFF, JPEG, or GIF).*
index	*integer* •index	*The index of the ComboBox within its containing object.*
itemLayer	{Layer} itemLayer	*The layer that the ComboBox is on.*
label	*string* label	*A property that can be set to any string.*
leftLineEnd	#ArrowHead leftLineEnd	*The arrowhead applied to the start of the path.*
linkedPageItemOptions	{LinkedPageItemOption} •linkedPageItemOptions	*Linked Page Item options.*
localDisplaySetting	#DisplaySettingOptions localDisplaySetting	*Display performance options for the ComboBox.*
locked	*boolean* locked	*If true, the ComboBox is locked.*
miterLimit	*double* miterLimit	*The limit of the ratio of stroke width to miter length before a miter (pointed) join becomes a bevel (squared-off) join. Type: Double (1 - 500).*
movieBehaviors	{MovieBehaviors} •movieBehaviors	*A collection of movie behavior objects.*
name	*string* name	*The name of the ComboBox.*
nonprinting	*boolean* nonprinting	*If true, the ComboBox does not print.*
openFileBehaviors	{OpenFileBehaviors} •openFileBehaviors	*A collection of open file behavior objects.*

COMBOBOX CS6 VALUES	PROTOTYPE	DESCRIPTION
ovals	{Ovals} •ovals	*A collection of ellipses.*
overprintFill	*boolean* overprintFill	*If true, the ComboBox's fill color overprints any underlying objects. If false, the fill color knocks out the underlying colors.*
overprintGap	*boolean* overprintGap	*If true, the gap color overprints any underlying colors. If false, the gap color knocks out the underlying colors.*
overprintStroke	*boolean* overprintStroke	*If true, the ComboBox's stroke color overprints any underlying objects. If false, the stroke color knocks out the underlying colors.*
overridden	*boolean* •overridden	*If true, the object originated on a master spread and was overridden. If false, the object either originated on a master spread and was not overridden, or the object did not originate on a master page.*
overriddenMasterPageItem	*variant* •overriddenMasterPageItem	*An object that originated on a master page and has been overridden. Type: PageItem, Guide, Graphic, Movie or Sound.*
pageItems	{PageItems} •pageItems	*The page items collection, which can be used to process all page items in a container (such as a document, page, or group), regardless of type.*
parent	*variant* •parent	*The parent of the ComboBox (a Snippet, PlaceGun, Character, Spread, MasterSpread, SplineItem, Polygon, GraphicLine, Rectangle, Oval or Group). Type: Snippet, PlaceGun, Character, Spread, MasterSpread, SplineItem, Polygon, GraphicLine, Rectangle, Oval or Group.*
parentPage	{Page} •parentPage	*The page on which this page item appears.*
pdfs	{PDFs} •pdfs	*A collection of PDF files.*
picts	{PICTs} •picts	*A collection of PICT graphics.*
polygons	{Polygons} •polygons	*A collection of polygons.*
preferences	{Preferences} •preferences	*A collection of preferences objects.*
printableInPdf	*boolean* printableInPdf	*If true, the form field/push button is printable in the exported PDF.*
printFormBehaviors	{PrintFormBehaviors} •printFormBehaviors	*A collection of print form behavior objects.*
readOnly	*boolean* readOnly	*If true, the form field is read only in the exported PDF.*
rectangles	{Rectangles} •rectangles	*A collection of rectangles.*
required	*boolean* required	*If true, the form field is required in the exported PDF.*
rightLineEnd	#ArrowHead rightLineEnd	*The arrowhead applied to the end of the path.*
rightToLeftField	*boolean* rightToLeftField	*If true, the form field has right to left text enabled in the exported PDF.*
rotationAngle	*double* rotationAngle	*The rotatation angle of the ComboBox. (Range: -360 to 360).*
shearAngle	*double* shearAngle	*The skewing angle applied to the ComboBox. (Range: -360 to 360).*
showHideFieldsBehaviors	{ShowHideFieldsBehaviors} •showHideFieldsBehaviors	*A collection of show/hide fields behavior objects.*
sortChoices	*boolean* sortChoices	*If true, the choices would be sorted in the exported PDF.*
soundBehaviors	{SoundBehaviors} •soundBehaviors	*A collection of sound behavior objects.*
splineItems	{SplineItems} •splineItems	*The spline items collection.*
strokeAlignment	#StrokeAlignment strokeAlignment	*The stroke alignment applied to the ComboBox.*
strokeColor	*variant* strokeColor	*The swatch (color, gradient, tint, or mixed ink) applied to the stroke of the ComboBox. Type: Swatch or String.*
strokeCornerAdjustment	#StrokeCornerAdjustment strokeCornerAdjustment	*The corner adjustment applied to the ComboBox.*

COMBOBOX CS6 VALUES	PROTOTYPE	DESCRIPTION
strokeDashAndGap	*variant* strokeDashAndGap	*The dash and gap measurements that define the pattern of a custom dashed line. Define up to six values (in points) in the format [dash1, gap1, dash2, gap2, dash3, gap3]. Type: Array of Units (Doubles or Strings).*
strokeTint	*double* strokeTint	*The percent of tint to use in object's stroke color. (To specify a tint percent, use a number in the range of 0 to 100; to use the inherited or overridden value, use -1.).*
strokeTransparencySettings	{StrokeTransparencySetting} •strokeTransparencySettings	*Transparency settings for the stroke.*
strokeType	*variant* strokeType	*The name of the stroke style to apply. Type: StrokeStyle or String.*
strokeWeight	*variant* strokeWeight	*The weight (in points) to apply to the ComboBox's stroke. Type: Unit (Double or String).*
submitFormBehaviors	{SubmitFormBehaviors} •submitFormBehaviors	*A collection of submit form behavior objects.*
textFrames	{TextFrames} •textFrames	*A collection of text frames.*
textWrapPreferences	{TextWrapPreference} •textWrapPreferences	*The text wrap preference properties that define the default formatting for wrapping text around objects.*
timingSettings	{TimingSetting} •timingSettings	*The object timing settings.*
topLeftCornerOption	#CornerOptions topLeftCornerOption	*The shape to be applied to the top left corner of rectangular shapes and all corners of non-rectangular shapes.Note: corner option differs from end join in which you can set a radius for a corner option, whereas the rounded or beveled effect of an end join depends on the stroke weight.*
topLeftCornerRadius	*variant* topLeftCornerRadius	*The radius in measurement units of the corner effect applied to the top left corner of rectangular shapes and all corners of non-rectangular shapes. Type: Unit (Double or String).*
topRightCornerOption	#CornerOptions topRightCornerOption	*The shape to apply to the top right corner of rectangular shapes.*
topRightCornerRadius	*variant* topRightCornerRadius	*The radius in measurement units of the corner effect applied to the top right corner of rectangular shapes. Type: Unit (Double or String).*
transparencySettings	{TransparencySetting} •transparencySettings	*Transparency settings.*
verticalLayoutConstraints	*variant* verticalLayoutConstraints	*The top margin, height, and bottom margin constraints this item is subject to when using the object-based layout rule. Type: Array of 3 idDimensionsConstraints enumerators.*
verticalScale	*double* verticalScale	*The vertical scaling applied to the ComboBox.*
viewZoomBehaviors	{ViewZoomBehaviors} •viewZoomBehaviors	*A collection of view zoom behavior objects.*
visible	*boolean* visible	*If true, the ComboBox is visible.*
visibleBounds	*variant* visibleBounds	*The bounds of the ComboBox including the stroke width, in the format [y1, x1, y2, x2], which give the coordinates of the top-left and bottom-right corners of the bounding box. Type: Array of 4 Units (Doubles or Strings).*
wmfs	{WMFs} •wmfs	*A collection of WMF graphics.*

ComboBoxes

Object available in ID 8

COMBOBOXES CS6 VALUES	PROTOTYPE	DESCRIPTION
add	{ComboBox} add(*variant* Layer, #LocationOptions At, *variant* Reference)	*Creates a new ComboBox. / Return value: The new ComboBox. / Layer: The layer on which to create the ComboBox. Type: Layer. / At: The location at which to insert the ComboBox relative to the reference object or within the container object. / Reference: The reference object. Note: Required when the at parameter is before or after. Type: Document, Spread, MasterSpread, Page, Layer or PageItem.*
anyItem	IDispatch anyItem()	*Returns any object in the collection. / Return value: The object.*
firstItem	IDispatch firstItem()	*Returns the first object in the collection. / Return value: The first object.*
itemByID	IDispatch itemByID(*integer* Id)	*Returns the object with the specified ID. / Return value: The object. / ID: The ID.*
itemByRange	{Objects} itemByRange(*variant* From, *variant* To)	*Returns the objects within the specified range. / Return value: The objects. / From: The object, index, or name at the beginning of the range. Type: Object, Long Integer or String. / To: The object, index, or name at the end of the range. Type: Object, Long Integer or String.*
lastItem	IDispatch lastItem()	*Returns the last object in the collection. / Return value: The last object.*
middleItem	IDispatch middleItem()	*Returns the middle object in the collection. / Return value: The middle object.*
nextItem	IDispatch nextItem(IDispatch Obj)	*Returns the object whose index follows the specified object in the collection. / Return value: The next object. / Obj: The object whose index comes before the desired object.*
previousItem	IDispatch previousItem(IDispatch Obj)	*Returns the object with the index previous to the specified index. / Return value: The previous object. / Obj: The index of the object that follows the desired object.*
count	*integer* •count	*Displays the number of elements in the ComboBox.*
item	{ComboBox} •item(*variant* Index)	*Returns the object with the specified index or name. Type: Long Integer or String.*

ComboBoxEvents

Object available in ID 8

COMBOBOXEVENTS CS6 VALUES	PROTOTYPE	DESCRIPTION
afterPlace	*const* afterPlace(afterPlace)	*Dispatched after a ComboBox is placed. This event bubbles. This event is not cancelable.*
beforePlace	*const* beforePlace(beforePlace)	*Dispatched before a ComboBox is placed. This event bubbles. This event is cancelable.*

ContentPlacerObject

Object available in ID 8

CONTENTPLACEROBJECT CS6 VALUES	PROTOTYPE	DESCRIPTION
load	*void* load(Objects PageItems, *boolean* LinkPageItems, *boolean* LinkStories, *boolean* MapStyles, *boolean* ShowingOptions)	*Load the Content Placer with one or more objects. / PageItems: One or more page items to place or load. / LinkPageItems: Whether to link pageItems in content placer (if true it will override link stories value). / LinkStories: Whether to link stories in content placer (only applicable for single story, pageItem links will also be created in case of more than one item). / MapStyles: Whether to map styles in content placer. / ShowingOptions: Whether to display the link options dialog.*
loaded	*boolean* •loaded	*Whether the Content Placer is currently loaded with content for placing.*
parent	{Application} •parent	*The parent of the ContentPlacerObject (a Application).*

FontLockingPreference

Object available in ID 8

FONTLOCKINGPREFERENCE CS6 VALUES	PROTOTYPE	DESCRIPTION
fontChangeLocking	*boolean* fontChangeLocking	*If true, turns on missing glyph protection during font change.*
fontInputLocking	*boolean* fontInputLocking	*If true, turns on missing glyph protection during typing.*
parent	{Application} •parent	*The parent of the FontLockingPreference (a Application).*

HtmlItem

Object available in ID 8

HTMLITEM CS6 VALUES	PROTOTYPE	DESCRIPTION
applyObjectStyle	*void* applyObjectStyle(ObjectStyle Using, *boolean* ClearingOverrides, *boolean* ClearingOverridesThroughRootObjectStyle)	*Applies the specified object style. / Using: The object style to apply. / ClearingOverrides: If true, clears the HtmlItem's existing attributes before applying the style. / ClearingOverridesThroughRootObjectStyle: If true, clears attributes and formatting applied to the HtmlItem that are not defined in the object style.*
asynchronousExportFile	{BackgroundTask} asynchronousExportFile(*variant* Format, *string* To, *boolean* ShowingOptions, *variant* Using, *variant* VersionComments, *boolean* ForceSave)	*asynchronously exports the object(s) to a file. / Return value: object representing the asynchronous export. / Format: The export format, specified as an enumeration value or as an extension that appears in the Save as type or Format menu in the Export dialog. Type: idExportFormat enumerator or String. / To: The path to the export file. / ShowingOptions: If true, displays the export options dialog. / Using: The export style. Type: PDFExportPreset. / VersionComments: The comment for this version. Type: String. / ForceSave: If true, forcibly saves a version.*
autoTag	*void* autoTag()	*Tag the object or the parent story using default tags defined in XML preference.*
changeObject	{Objects} changeObject(*variant* ReverseOrder)	*Finds objects that match the find what value and replace the objects with the change to value. / Return value: The replaced object(s). / ReverseOrder: If true, returns the results in reverse order. Type: Boolean.*

HTMLITEM CS6 VALUES	PROTOTYPE	DESCRIPTION
clearObjectStyleOverrides	*void* clearObjectStyleOverrides()	*Clear overrides for object style.*
clearTransformations	*void* clearTransformations()	*Clears transformations from the HtmlItem. Transformations include rotation, scaling, flipping, fitting, and shearing.*
contentPlace	*variant* contentPlace(Objects PageItems, *boolean* LinkPageItems, *boolean* LinkStories, *boolean* MapStyles, *boolean* ShowingOptions)	*Duplicate an object and place it into the target page item. / Return value: The placed content. Type: Array of PageItems, Graphics, Stories or Texts. / PageItems: One or more page items to place or load. / LinkPageItems: Whether to link pageItems in content placer (if true it will override link stories value). / LinkStories: Whether to link stories in content placer (only applicable for single story, pageItem links will also be created in case of more than one item). / MapStyles: Whether to map styles in content placer. / ShowingOptions: Whether to display the link options dialog.*
convertShape	*void* convertShape(#ConvertShapeOptions Given, *variant* NumberOfSides, *variant* InsetPercentage, *variant* CornerRadius)	*Converts the HtmlItem to a different shape. / Given: The HtmlItem's new shape. / NumberOfSides: The number of sides for the resulting polygon. (Range: 3 to 100). Type: Long Integer (3 - 100). / InsetPercentage: The star inset percentage for the resulting polygon. (Range: 0.0 to 100.0) . Type: Long Integer (0 - 100). / CornerRadius: The corner radius of the resulting rectangle. Type: Unit (Double or String) (0 - 800 points).*
delete	*void* delete()	*Deletes the HtmlItem.*
detach	*void* detach()	*Detaches an overridden master page item from the master page.*
duplicate	{PageItem} duplicate(*variant* To, *variant* By)	*Duplicates the HtmlItem at the specified location or offset. / Return value: The duplicated HtmlItem. / To: The location of the new HtmlItem, specified in coordinates in the format [x, y]. Type: Array of 2 Units (Doubles or Strings), Spread, Page or Layer. / By: Amount by which to offset the new HtmlItem from the original HtmlItem's position. Type: Array of 2 Units (Doubles or Strings).*
export	*void* export(*variant* Format, *string* To, *boolean* ShowingOptions, *variant* Using, *variant* VersionComments, *boolean* ForceSave)	*Exports the object(s) to a file. / Format: The export format, specified as an enumeration value or as an extension that appears in the Save as type or Format menu in the Export dialog. Type: idExportFormat enumerator or String. / To: The path to the export file. / ShowingOptions: If true, displays the export options dialog. / Using: The export style. Type: PDFExportPreset. / VersionComments: The comment for this version. Type: String. / ForceSave: If true, forcibly saves a version.*
extractLabel	*string* extractLabel(*string* Key)	*Gets the label value associated with the specified key. / Return value: The label. / Key: The key.*
findObject	{Objects} findObject(*variant* ReverseOrder)	*Finds objects that match the find what value. / Return value: The found object(s). / ReverseOrder: If true, returns the results in reverse order. Type: Boolean.*
fit	*void* fit(#FitOptions Given)	*Applies the specified fit option to content in a frame. / Given: The fit option to use.*
flipItem	*void* flipItem(#Flip Given, *variant* Around)	*Flips the HtmlItem. / Given: The axis around which to flip the HtmlItem. / Around: The point around which to flip the HtmlItem. Type: Array of 2 Units (Doubles or Strings) or idAnchorPoint enumerator.*
insertLabel	*void* insertLabel(*string* Key, *string* Value)	*Sets the label to the value associated with the specified key. / Key: The key. / Value: The value.*
markup	*void* markup(XMLElement Using)	*Associates the page item with the specified XML element while preserving existing content. / Using: The XML element.*

HTMLITEM CS6 VALUES	PROTOTYPE	DESCRIPTION
move	*void* move(*variant* To, *variant* By)	*Moves the HtmlItem to a new location. Note: Either the 'to' or 'by' parameter is required; if both parameters are defined, only the to value is used. / To: The new location of the HtmlItem,in the format (x, y). Type: Array of 2 Units (Doubles or Strings), Spread, Page or Layer. / By: The amount (in measurement units) to move the HtmlItem relative to its current position, in the format (x, y). Type: Array of 2 Units (Doubles or Strings).*
override	*variant* override(Page DestinationPage)	*Overrides a master page item and places the item on the document page as a new object. / Return value: The new object. Type: PageItem, Guide, Graphic, Movie or Sound. / DestinationPage: The document page that contains the master page item to override.*
place	*variant* place(*string* FileName, *boolean* ShowingOptions)	*Places the file. / Return value: The placed file. Type: Array of PageItems, Graphics, Movies, Sounds, Stories or Texts. / FileName: The file to place. / ShowingOptions: Whether to display the import options dialog.*
placeXML	*void* placeXML(XMLElement Using)	*Places XML content into the specified object. Note: Replaces any existing content. / Using: The XML element whose content you want to place.*
redefineScaling	*void* redefineScaling(*variant* To)	*Apply an item's scaling to its content if possible. / To: The scale factors to be left on the item. The default is {1.0, 1.0}. Type: Array of 2 Doubles.*
reframe	*void* reframe(*variant* In, *variant* OpposingCorners)	*Move the bounding box of the page item. / In: The bounding box to resize. Type: idCoordinateSpaces enumerator or Ordered array containing CoordinateSpace:idCoordinateSpaces enumerator, BoundsKind:idBoundingBoxLimits enumerator. / OpposingCorners: Opposing corners of new bounding box in the given coordinate space. Type: Array of 2 Arrays of 2 Doubles.*
removeOverride	*void* removeOverride()	*Removes the override from a previously overridden master page item.*
resize	*void* resize(*variant* In, *variant* From, #ResizeMethods By, *variant* Values, *boolean* ResizeIndividually, *boolean* ConsideringRulerUnits)	*Resize the page item. / In: The bounding box to resize. Type: idCoordinateSpaces enumerator, idBoundingBoxLimits enumerator or Ordered array containing CoordinateSpace:idCoordinateSpaces enumerator, BoundsKind:idBoundingBoxLimits enumerator. / From: The transform origin. Legal specifications: relative to bounding box: anchor | {anchor | {x,y}, bounds kind [, coordinate space]}; relative to coordinate space: {x,y} | {{x,y} [, coordinate space]}; relative to layout window ruler: {{x,y}, page index | bounds kind}. Type: Array of 2 Doubles, idAnchorPoint enumerator or Array of Arrays of 2 Doubles, idCoordinateSpaces enumerators, idAnchorPoint enumerators, idBoundingBoxLimits enumerators or Long Integers. / By: How the current dimensions are affected by the given values. / Values: The width and height values. Legal dimensions specifications: {x, y [, coordinate space]}, {x, resize constraint [, coordinate space]}, or {resize constraint, y [, coordinate space]}; where x and y are real numbers and coordinate space is used to determine _only_ the unit of length for x and y; coordinate space is ignored for the 'current dimensions times' resize method). Type: Array of Doubles, idResizeConstraints enumerators or idCoordinateSpaces enumerators. / ResizeIndividually: If false and multiple page items are targeted, the new dimensions are attained only by moving the individual items rather than resizing them. / ConsideringRulerUnits: If true then a ruler location is interpreted using ruler units rather than points. The default value is false. This parameter has no effect unless the reference point is specified relative to a page.*

HTMLITEM CS6 VALUES	PROTOTYPE	DESCRIPTION
resolve	*variant* resolve(*variant* Location, #CoordinateSpaces In, *boolean* ConsideringRulerUnits)	*Get the coordinates of the given location in the specified coordinate system. / Return value: The resolved location for each of the targeted page items. Type: Array of 2 Doubles or Array of Arrays of 2 Doubles. / Location: The location requested. Type: Array of 2 Doubles, idAnchorPoint enumerator or Array of Arrays of 2 Doubles, idCoordinateSpaces enumerators, idAnchorPoint enumerators, idBoundingBoxLimits enumerators or Long Integers. / In: The coordinate space to use. / ConsideringRulerUnits: If true then a ruler location is interpreted using ruler units rather than points. The default value is false. This parameter has no effect unless the reference point is specified relative to a page.*
select	*void* select(#SelectionOptions ExistingSelection)	*Selects the object. / ExistingSelection: The selection status of the HtmlItem in relation to previously selected objects.*
store	{Asset} store(Library Using)	*Stores the object in the specified library. / Return value: The new library asset(s). / Using: The library in which to store the object.*
transform	*void* transform(#CoordinateSpaces In, *variant* From, *variant* WithMatrix, *variant* ReplacingCurrent, *boolean* ConsideringRulerUnits)	*Transform the page item. / In: The coordinate space to use. / From: The temporary origin during the transformation. Type: Array of 2 Doubles, idAnchorPoint enumerator or Array of Arrays of 2 Doubles, idCoordinateSpaces enumerators, idAnchorPoint enumerators, idBoundingBoxLimits enumerators or Long Integers. / WithMatrix: Transform matrix. Type: Array of 6 Doubles or TransformationMatrix. / ReplacingCurrent: Transform components to consider; providing this optional parameter causes the target's existing transform components to be replaced with new values. Without this parameter, the given matrix is concatenated onto the target's existing transform combining the effect of the two. Type: idMatrixContent enumerator, Array of idMatrixContent enumerators or Long Integer. / ConsideringRulerUnits: If true then a ruler based origin is interpreted using ruler units rather than points. The default value is false. This parameter has no effect unless the reference point is specified relative to a page.*
transformAgain	*variant* transformAgain()	*Transforms the HtmlItem using the last transformation performed on any object. Transformations include moving, rotating, shearing, scaling, and flipping. . / Return value: Playback errors. Type: Array of Strings.*
transformAgainIndividually	*variant* transformAgainIndividually()	*Transforms the HtmlItem using the last transformation performed on any HtmlItem. Transformations include moving, rotating, shearing, scaling, and flipping. . / Return value: Playback errors. Type: Array of Strings.*
transformSequenceAgain	*variant* transformSequenceAgain()	*Transforms the HtmlItem using the last sequence of transform operations performed on any single object or performed at the same time on any group of objects. Transformations include moving, rotating, shearing, scaling, and flipping. / Return value: Playback errors. Type: Array of Strings.*
transformSequenceAgainIndividually	*variant* transformSequenceAgainIndividually()	*Transforms the HtmlItem using the last sequence of transformations performed on any single object or performed at the same time on any group of objects. Transformations include moving, rotating, shearing, scaling, and flipping. / Return value: Playback errors. Type: Array of Strings.*
transformValuesOf	{Objects} transformValuesOf(#CoordinateSpaces In)	*Get the transformation values of the page item. / Return value: A list of the transformation matrices. / In: The coordinate space to use.*
absoluteFlip	#Flip absoluteFlip	*Indicates whether the HtmlItem has been flipped independently of its parent object and, if yes, the direction in which the HtmlItem was flipped.*
absoluteHorizontalScale	*double* absoluteHorizontalScale	*The horizontal scale of the HtmlItem relative to its containing object.*
absoluteRotationAngle	*double* absoluteRotationAngle	*The rotation angle of the HtmlItem relative to its containing object. (Range: -360 to 360).*

HTMLITEM CS6 VALUES	PROTOTYPE	DESCRIPTION
absoluteShearAngle	*double* absoluteShearAngle	*The skewing angle of the HtmlItem relative to its containing object. (Range: -360 to 360).*
absoluteVerticalScale	*double* absoluteVerticalScale	*The vertical scale of the HtmlItem relative to its containing object.*
allArticles	{Objects} •allArticles	*The list of all articles this page item is part of.*
allGraphics	{Objects} •allGraphics	*Lists all graphics contained by the HtmlItem.*
allowOverrides	*boolean* allowOverrides	*If true, the master page item can be overridden.*
allPageItems	{Objects} •allPageItems	*Lists all page items contained by the HtmlItem.*
animationSettings	{AnimationSetting} •animationSettings	*The page item animation settings.*
appliedObjectStyle	{ObjectStyle} appliedObjectStyle	*The object style applied to the HtmlItem.*
associatedXMLElement	{XMLItem} •associatedXMLElement	*The XML element associated with the HtmlItem.*
bottomLeftCornerOption	#CornerOptions bottomLeftCornerOption	*The shape to apply to the bottom left corner of rectangular shapes.*
bottomLeftCornerRadius	*variant* bottomLeftCornerRadius	*The radius in measurement units of the corner effect applied to the bottom left corner of rectangular shapes. Type: Unit (Double or String).*
bottomRightCornerOption	#CornerOptions bottomRightCornerOption	*The shape to apply to the bottom right corner of rectangular shapes.*
bottomRightCornerRadius	*variant* bottomRightCornerRadius	*The radius in measurement units of the corner effect applied to the bottom right corner of rectangular shapes. Type: Unit (Double or String).*
contentTransparencySettings	{ContentTransparencySetting} •contentTransparencySettings	*Transparency settings for the content of the HtmlItem.*
endCap	#EndCap endCap	*The end shape of an open path.*
endJoin	#EndJoin endJoin	*The corner join applied to the HtmlItem.*
fillColor	*variant* fillColor	*The swatch (color, gradient, tint, or mixed ink) applied to the fill of the HtmlItem. . Type: Swatch or String.*
fillTint	*double* fillTint	*The percent of tint to use in the HtmlItem's fill color. (To specify a tint percent, use a number in the range of 0 to 100; to use the inherited or overridden value, use -1.).*
fillTransparencySettings	{FillTransparencySetting} •fillTransparencySettings	*Transparency settings for the fill applied to the HtmlItem.*
fixedDimensions	*boolean* fixedDimensions	*Is this HTML able to adapt its width and height based on changes to the parent div or does it have fixed dimensions?.*
flip	#Flip flip	*The direction in which to flip the printed image.*
gapColor	{Swatch} gapColor	*The swatch (color, gradient, tint, or mixed ink) applied to the gap of a dashed, dotted, or striped stroke. For information, see stroke type.*
gapTint	*double* gapTint	*The tint as a percentage of the gap color. (To specify a tint percent, use a number in the range of 0 to 100; to use the inherited or overridden value, use -1.).*
geometricBounds	*variant* geometricBounds	*The bounds of the HtmlItem excluding the stroke width, in the format [y1, x1, y2, x2], which give the coordinates of the top-left and bottom-right corners of the bounding box. Type: Array of 4 Units (Doubles or Strings).*
gradientFillAngle	*double* gradientFillAngle	*The angle of a linear gradient applied to the fill of the HtmlItem. (Range: -180 to 180).*
gradientFillLength	*variant* gradientFillLength	*The length (for a linear gradient) or radius (for a radial gradient) applied to the fill of the HtmlItem. Type: Unit (Double or String).*

HTMLITEM CS6 VALUES	PROTOTYPE	DESCRIPTION
gradientFillStart	*variant* gradientFillStart	*The starting point (in page coordinates) of a gradient applied to the fill of the HtmlItem, in the format [x, y]. Type: Array of 2 Units (Doubles or Strings).*
gradientStrokeAngle	*double* gradientStrokeAngle	*The angle of a linear gradient applied to the stroke of the HtmlItem. (Range: -180 to 180).*
gradientStrokeLength	*variant* gradientStrokeLength	*The length (for a linear gradient) or radius (for a radial gradient) applied to the stroke of the HtmlItem. Type: Unit (Double or String).*
gradientStrokeStart	*variant* gradientStrokeStart	*The starting point (in page coordinates) of a gradient applied to the stroke of the HtmlItem, in the format [x, y]. Type: Array of 2 Units (Doubles or Strings).*
horizontalLayoutConstraints	*variant* horizontalLayoutConstraints	*The left margin, width, and right margin constraints this item is subject to when using the object-based layout rule. Type: Array of 3 idDimensionsConstraints enumerators.*
horizontalScale	*double* horizontalScale	*The horizontal scaling applied to the HtmlItem.*
htmlContent	*string* htmlContent	*The embedded HTML text.*
id	*integer* •id	*The unique ID of the HtmlItem.*
index	*integer* •index	*The index of the HtmlItem within its containing object.*
itemLayer	{Layer} itemLayer	*The layer that the HtmlItem is on.*
label	*string* label	*A property that can be set to any string.*
leftLineEnd	#ArrowHead leftLineEnd	*The arrowhead applied to the start of the path.*
linkedPageItemOptions	{LinkedPageItemOption} •linkedPageItemOptions	*Linked Page Item options.*
localDisplaySetting	#DisplaySettingOptions localDisplaySetting	*Display performance options for the HtmlItem.*
locked	*boolean* locked	*If true, the HtmlItem is locked.*
miterLimit	*double* miterLimit	*The limit of the ratio of stroke width to miter length before a miter (pointed) join becomes a bevel (squared-off) join. Type: Double (1 - 500).*
name	*string* name	*The name of the HtmlItem.*
nonprinting	*boolean* nonprinting	*If true, the HtmlItem does not print.*
overprintFill	*boolean* overprintFill	*If true, the HtmlItem's fill color overprints any underlying objects. If false, the fill color knocks out the underlying colors.*
overprintGap	*boolean* overprintGap	*If true, the gap color overprints any underlying colors. If false, the gap color knocks out the underlying colors.*
overprintStroke	*boolean* overprintStroke	*If true, the HtmlItem's stroke color overprints any underlying objects. If false, the stroke color knocks out the underlying colors.*
overridden	*boolean* •overridden	*If true, the object originated on a master spread and was overridden. If false, the object either originated on a master spread and was not overridden, or the object did not originate on a master page.*
overriddenMasterPageItem	*variant* •overriddenMasterPageItem	*An object that originated on a master page and has been overridden. Type: PageItem, Guide, Graphic, Movie or Sound.*
parent	*variant* •parent	*The parent of the HtmlItem (a SplineItem, Polygon, GraphicLine, Rectangle or Oval). Type: SplineItem, Polygon, GraphicLine, Rectangle or Oval.*
parentPage	{Page} •parentPage	*The page on which this page item appears.*
preferences	{Preferences} •preferences	*A collection of preferences objects.*

HTMLITEM CS6 VALUES	PROTOTYPE	DESCRIPTION
rightLineEnd	#ArrowHead rightLineEnd	*The arrowhead applied to the end of the path.*
rotationAngle	*double* rotationAngle	*The rotatation angle of the HtmlItem. (Range: -360 to 360).*
shearAngle	*double* shearAngle	*The skewing angle applied to the HtmlItem. (Range: -360 to 360).*
strokeAlignment	#StrokeAlignment strokeAlignment	*The stroke alignment applied to the HtmlItem.*
strokeColor	*variant* strokeColor	*The swatch (color, gradient, tint, or mixed ink) applied to the stroke of the HtmlItem. Type: Swatch or String.*
strokeCornerAdjustment	#StrokeCornerAdjustment strokeCornerAdjustment	*The corner adjustment applied to the HtmlItem.*
strokeDashAndGap	*variant* strokeDashAndGap	*The dash and gap measurements that define the pattern of a custom dashed line. Define up to six values (in points) in the format [dash1, gap1, dash2, gap2, dash3, gap3]. Type: Array of Units (Doubles or Strings).*
strokeTint	*double* strokeTint	*The percent of tint to use in object's stroke color. (To specify a tint percent, use a number in the range of 0 to 100; to use the inherited or overridden value, use -1.).*
strokeTransparencySettings	{StrokeTransparencySetting} •strokeTransparencySettings	*Transparency settings for the stroke.*
strokeType	*variant* strokeType	*The name of the stroke style to apply. Type: StrokeStyle or String.*
strokeWeight	*variant* strokeWeight	*The weight (in points) to apply to the HtmlItem's stroke. Type: Unit (Double or String).*
textWrapPreferences	{TextWrapPreference} •textWrapPreferences	*The text wrap preference properties that define the default formatting for wrapping text around objects.*
timingSettings	{TimingSetting} •timingSettings	*The object timing settings.*
topLeftCornerOption	#CornerOptions topLeftCornerOption	*The shape to be applied to the top left corner of rectangular shapes and all corners of non-rectangular shapes.Note: corner option differs from end join in which you can set a radius for a corner option, whereas the rounded or beveled effect of an end join depends on the stroke weight.*
topLeftCornerRadius	*variant* topLeftCornerRadius	*The radius in measurement units of the corner effect applied to the top left corner of rectangular shapes and all corners of non-rectangular shapes. Type: Unit (Double or String).*
topRightCornerOption	#CornerOptions topRightCornerOption	*The shape to apply to the top right corner of rectangular shapes.*
topRightCornerRadius	*variant* topRightCornerRadius	*The radius in measurement units of the corner effect applied to the top right corner of rectangular shapes. Type: Unit (Double or String).*
transparencySettings	{TransparencySetting} •transparencySettings	*Transparency settings.*
verticalLayoutConstraints	*variant* verticalLayoutConstraints	*The top margin, height, and bottom margin constraints this item is subject to when using the object-based layout rule. Type: Array of 3 idDimensionsConstraints enumerators.*
verticalScale	*double* verticalScale	*The vertical scaling applied to the HtmlItem.*
visible	*boolean* visible	*If true, the HtmlItem is visible.*
visibleBounds	*variant* visibleBounds	*The bounds of the HtmlItem including the stroke width, in the format [y1, x1, y2, x2], which give the coordinates of the top-left and bottom-right corners of the bounding box. Type: Array of 4 Units (Doubles or Strings).*

HtmlItemEvents

Object available in ID 8

HTMLITEMEVENTS CS6 VALUES	PROTOTYPE	DESCRIPTION
afterPlace	const afterPlace(afterPlace)	*Dispatched after a HtmlItem is placed. This event bubbles. This event is not cancelable.*
beforePlace	const beforePlace(beforePlace)	*Dispatched before a HtmlItem is placed. This event bubbles. This event is cancelable.*

HtmlItems

Object available in ID 8

HTMLITEMS CS6 VALUES	PROTOTYPE	DESCRIPTION
add	{HtmlItem} add(variant Layer, #LocationOptions At, variant Reference)	*Creates a new HtmlItem. / Return value: The new HtmlItem. / Layer: The layer on which to create the HtmlItem. Type: Layer. / At: The location at which to insert the HtmlItem relative to the reference object or within the container object. / Reference: The reference object. Note: Required when the at parameter is before or after. Type: Document, Spread, MasterSpread, Page, Layer or PageItem.*
anyItem	IDispatch anyItem()	*Returns any object in the collection. / Return value: The object.*
firstItem	IDispatch firstItem()	*Returns the first object in the collection. / Return value: The first object.*
itemByID	IDispatch itemByID(integer Id)	*Returns the object with the specified ID. / Return value: The object. / ID: The ID.*
itemByRange	{Objects} itemByRange(variant From, variant To)	*Returns the objects within the specified range. / Return value: The objects. / From: The object, index, or name at the beginning of the range. Type: Object, Long Integer or String. / To: The object, index, or name at the end of the range. Type: Object, Long Integer or String.*
lastItem	IDispatch lastItem()	*Returns the last object in the collection. / Return value: The last object.*
middleItem	IDispatch middleItem()	*Returns the middle object in the collection. / Return value: The middle object.*
nextItem	IDispatch nextItem(IDispatch Obj)	*Returns the object whose index follows the specified object in the collection. / Return value: The next object. / Obj: The object whose index comes before the desired object.*
previousItem	IDispatch previousItem(IDispatch Obj)	*Returns the object with the index previous to the specified index. / Return value: The previous object. / Obj: The index of the object that follows the desired object.*
count	integer •count	*Displays the number of elements in the HtmlItem.*
item	{HtmlItem} •item(variant Index)	*Returns the object with the specified index or name. Type: Long Integer or String.*

LinkedPageItemOption

Object available in ID 8

LINKEDPAGEITEMOPTION CS6 VALUES	PROTOTYPE	DESCRIPTION
parent	*variant* •parent	*The parent of the LinkedPageItemOption (a Application, Document, PageItem, HtmlItem, FormField, SignatureField, TextBox, RadioButton, ListBox, ComboBox, CheckBox, MultiStateObject, Button, MediaItem, Sound, Movie, EPSText, SplineItem, Polygon, GraphicLine, Rectangle, Oval, Graphic, ImportedPage, PICT, WMF, PDF, EPS, Image, Group or TextFrame). Type: Application, Document, PageItem, HtmlItem, FormField, SignatureField, TextBox, RadioButton, ListBox, ComboBox, CheckBox, MultiStateObject, Button, MediaItem, Sound, Movie, EPSText, SplineItem, Polygon, GraphicLine, Rectangle, Oval, Graphic, ImportedPage, PICT, WMF, PDF, EPS, Image, Group or TextFrame.*
preserveAppearance	*boolean* preserveAppearance	*If true, appearance edits will be preserved during update.*
preserveFrameContent	*boolean* preserveFrameContent	*If true, frame content edits will be preserved during update.*
preserveInteractivity	*boolean* preserveInteractivity	*If true, interactivity edits will be preserved during update.*
preserveOthers	*boolean* preserveOthers	*If true, text wrap, hyperLinks, text frame options, object export settings will be preserved during update.*
preserveSizeAndShape	*boolean* preserveSizeAndShape	*If true, size and shape edits will be preserved during update.*
updateLinkWhileSaving	*boolean* updateLinkWhileSaving	*If true, the linked Page Item will be updated while saving.*
warnOnUpdateOfEditedPageItem	*boolean* warnOnUpdateOfEditedPageItem	*If true, a warning will be shown if the update link operation will override local edits.*

ListBox

Object available in ID 8

LISTBOX CS6 VALUES	PROTOTYPE	DESCRIPTION
applyObjectStyle	*void* applyObjectStyle(ObjectStyle Using, *boolean* ClearingOverrides, *boolean* ClearingOverridesThroughRootObjectStyle)	*Applies the specified object style. / Using: The object style to apply. / ClearingOverrides: If true, clears the ListBox's existing attributes before applying the style. / ClearingOverridesThroughRootObjectStyle: If true, clears attributes and formatting applied to the ListBox that are not defined in the object style.*
asynchronousExportFile	{BackgroundTask} asynchronousExportFile(*variant* Format, *string* To, *boolean* ShowingOptions, *variant* Using, *variant* VersionComments, *boolean* ForceSave)	*asynchronously exports the object(s) to a file. / Return value: object representing the asynchronous export. / Format: The export format, specified as an enumeration value or as an extension that appears in the Save as type or Format menu in the Export dialog. Type: idExportFormat enumerator or String. / To: The path to the export file. / ShowingOptions: If true, displays the export options dialog. / Using: The export style. Type: PDFExportPreset. / VersionComments: The comment for this version. Type: String. / ForceSave: If true, forcibly saves a version.*
autoTag	*void* autoTag()	*Tag the object or the parent story using default tags defined in XML preference.*
bringForward	*void* bringForward()	*Brings the ListBox forward one level in its layer.*

LISTBOX CS6 VALUES	PROTOTYPE	DESCRIPTION
bringToFront	*void* bringToFront(*variant* Reference)	*Brings the ListBox to the front of its layer or in front of a particular item. / Reference: The reference object to bring the object in front of (must have same parent). Type: PageItem.*
changeObject	{Objects} changeObject(*variant* ReverseOrder)	*Finds objects that match the find what value and replace the objects with the change to value. / Return value: The replaced object(s). / ReverseOrder: If true, returns the results in reverse order. Type: Boolean.*
clearObjectStyleOverrides	*void* clearObjectStyleOverrides()	*Clear overrides for object style.*
clearTransformations	*void* clearTransformations()	*Clears transformations from the ListBox. Transformations include rotation, scaling, flipping, fitting, and shearing.*
contentPlace	*variant* contentPlace(Objects PageItems, *boolean* LinkPageItems, *boolean* LinkStories, *boolean* MapStyles, *boolean* ShowingOptions)	*Duplicate an object and place it into the target page item. / Return value: The placed content. Type: Array of PageItems, Graphics, Stories or Texts. / PageItems: One or more page items to place or load. / LinkPageItems: Whether to link pageItems in content placer (if true it will override link stories value). / LinkStories: Whether to link stories in content placer (only applicable for single story, pageItem links will also be created in case of more than one item). / MapStyles: Whether to map styles in content placer. / ShowingOptions: Whether to display the link options dialog.*
convertShape	*void* convertShape(#ConvertShapeOptions Given, *variant* NumberOfSides, *variant* InsetPercentage, *variant* CornerRadius)	*Converts the ListBox to a different shape. / Given: The ListBox's new shape. / NumberOfSides: The number of sides for the resulting polygon. (Range: 3 to 100). Type: Long Integer (3 - 100). / InsetPercentage: The star inset percentage for the resulting polygon. (Range: 0.0 to 100.0) . Type: Long Integer (0 - 100). / CornerRadius: The corner radius of the resulting rectangle. Type: Unit (Double or String) (0 - 800 points).*
convertToObject	*void* convertToObject()	*Converts the button object to the page item currently in the active state. Page items from other states will be lost.*
delete	*void* delete()	*Deletes the ListBox.*
detach	*void* detach()	*Detaches an overridden master page item from the master page.*
duplicate	{PageItem} duplicate(*variant* To, *variant* By)	*Duplicates the ListBox at the specified location or offset. / Return value: The duplicated ListBox. / To: The location of the new ListBox, specified in coordinates in the format [x, y]. Type: Array of 2 Units (Doubles or Strings), Spread, Page or Layer. / By: Amount by which to offset the new ListBox from the original ListBox's position. Type: Array of 2 Units (Doubles or Strings).*
export	*void* export(*variant* Format, *string* To, *boolean* ShowingOptions, *variant* Using, *variant* VersionComments, *boolean* ForceSave)	*Exports the object(s) to a file. / Format: The export format, specified as an enumeration value or as an extension that appears in the Save as type or Format menu in the Export dialog. Type: idExportFormat enumerator or String. / To: The path to the export file. / ShowingOptions: If true, displays the export options dialog. / Using: The export style. Type: PDFExportPreset. / VersionComments: The comment for this version. Type: String. / ForceSave: If true, forcibly saves a version.*
extractLabel	*string* extractLabel(*string* Key)	*Gets the label value associated with the specified key. / Return value: The label. / Key: The key.*
findObject	{Objects} findObject(*variant* ReverseOrder)	*Finds objects that match the find what value. / Return value: The found object(s). / ReverseOrder: If true, returns the results in reverse order. Type: Boolean.*
fit	*void* fit(#FitOptions Given)	*Applies the specified fit option to content in a frame. / Given: The fit option to use.*
flipItem	*void* flipItem(#Flip Given, *variant* Around)	*Flips the ListBox. / Given: The axis around which to flip the ListBox. / Around: The point around which to flip the ListBox. Type: Array of 2 Units (Doubles or Strings) or idAnchorPoint enumerator.*

LISTBOX CS6 VALUES	PROTOTYPE	DESCRIPTION
insertLabel	*void* insertLabel(*string* Key, *string* Value)	*Sets the label to the value associated with the specified key. / Key: The key. / Value: The value.*
markup	*void* markup(XMLElement Using)	*Associates the page item with the specified XML element while preserving existing content. / Using: The XML element.*
move	*void* move(*variant* To, *variant* By)	*Moves the ListBox to a new location. Note: Either the 'to' or 'by' parameter is required; if both parameters are defined, only the to value is used. / To: The new location of the ListBox,in the format (x, y). Type: Array of 2 Units (Doubles or Strings), Spread, Page or Layer. / By: The amount (in measurement units) to move the ListBox relative to its current position, in the format (x, y). Type: Array of 2 Units (Doubles or Strings).*
override	*variant* override(Page DestinationPage)	*Overrides a master page item and places the item on the document page as a new object. / Return value: The new object. Type: PageItem, Guide, Graphic, Movie or Sound. / DestinationPage: The document page that contains the master page item to override.*
placeXML	*void* placeXML(XMLElement Using)	*Places XML content into the specified object. Note: Replaces any existing content. / Using: The XML element whose content you want to place.*
redefineScaling	*void* redefineScaling(*variant* To)	*Apply an item's scaling to its content if possible. / To: The scale factors to be left on the item. The default is {1.0, 1.0}. Type: Array of 2 Doubles.*
reframe	*void* reframe(*variant* In, *variant* OpposingCorners)	*Move the bounding box of the page item. / In: The bounding box to resize. Type: idCoordinateSpaces enumerator or Ordered array containing CoordinateSpace:idCoordinateSpaces enumerator, BoundsKind:idBoundingBoxLimits enumerator. / OpposingCorners: Opposing corners of new bounding box in the given coordinate space. Type: Array of 2 Arrays of 2 Doubles.*
removeOverride	*void* removeOverride()	*Removes the override from a previously overridden master page item.*
resize	*void* resize(*variant* In, *variant* From, #ResizeMethods By, *variant* Values, *boolean* ResizeIndividually, *boolean* ConsideringRulerUnits)	*Resize the page item. / In: The bounding box to resize. Type: idCoordinateSpaces enumerator, idBoundingBoxLimits enumerator or Ordered array containing CoordinateSpace:idCoordinateSpaces enumerator, BoundsKind:idBoundingBoxLimits enumerator. / From: The transform origin. Legal specifications: relative to bounding box: anchor | {anchor | {x,y}, bounds kind [, coordinate space]}; relative to coordinate space: {x,y} | {{x,y} [, coordinate space]}; relative to layout window ruler: {{x,y}, page index | bounds kind}. Type: Array of 2 Doubles, idAnchorPoint enumerator or Array of Arrays of 2 Doubles, idCoordinateSpaces enumerators, idAnchorPoint enumerators, idBoundingBoxLimits enumerators or Long Integers. / By: How the current dimensions are affected by the given values. / Values: The width and height values. Legal dimensions specifications: {x, y [, coordinate space]}, {x, resize constraint [, coordinate space]}, or {resize constraint, y [, coordinate space]}; where x and y are real numbers and coordinate space is used to determine _only_ the unit of length for x and y; coordinate space is ignored for the 'current dimensions times' resize method). Type: Array of Doubles, idResizeConstraints enumerators or idCoordinateSpaces enumerators. / ResizeIndividually: If false and multiple page items are targeted, the new dimensions are attained only by moving the individual items rather than resizing them. / ConsideringRulerUnits: If true then a ruler location is interpreted using ruler units rather than points. The default value is false. This parameter has no effect unless the reference point is specified relative to a page.*

LISTBOX CS6 VALUES	PROTOTYPE	DESCRIPTION
resolve	*variant* resolve(*variant* Location, #CoordinateSpaces In, *boolean* ConsideringRulerUnits)	*Get the coordinates of the given location in the specified coordinate system. / Return value: The resolved location for each of the targeted page items. Type: Array of 2 Doubles or Array of Arrays of 2 Doubles. / Location: The location requested. Type: Array of 2 Doubles, idAnchorPoint enumerator or Array of Arrays of 2 Doubles, idCoordinateSpaces enumerators, idAnchorPoint enumerators, idBoundingBoxLimits enumerators or Long Integers. / In: The coordinate space to use. / ConsideringRulerUnits: If true then a ruler location is interpreted using ruler units rather than points. The default value is false. This parameter has no effect unless the reference point is specified relative to a page.*
select	*void* select(#SelectionOptions ExistingSelection)	*Selects the object. / ExistingSelection: The selection status of the ListBox in relation to previously selected objects.*
sendBackward	*void* sendBackward()	*Sends the ListBox back one level in its layer.*
sendToBack	*void* sendToBack(*variant* Reference)	*Sends the ListBox to the back of its layer or behind a particular item (must have same parent). / Reference: The reference object to send the object behind. Type: PageItem.*
store	{Asset} store(Library Using)	*Stores the object in the specified library. / Return value: The new library asset(s). / Using: The library in which to store the object.*
transform	*void* transform(#CoordinateSpaces In, *variant* From, *variant* WithMatrix, *variant* ReplacingCurrent, *boolean* ConsideringRulerUnits)	*Transform the page item. / In: The coordinate space to use. / From: The temporary origin during the transformation. Type: Array of 2 Doubles, idAnchorPoint enumerator or Array of Arrays of 2 Doubles, idCoordinateSpaces enumerators, idAnchorPoint enumerators, idBoundingBoxLimits enumerators or Long Integers. / WithMatrix: Transform matrix. Type: Array of 6 Doubles or TransformationMatrix. / ReplacingCurrent: Transform components to consider; providing this optional parameter causes the target's existing transform components to be replaced with new values. Without this parameter, the given matrix is concatenated onto the target's existing transform combining the effect of the two. Type: idMatrixContent enumerator, Array of idMatrixContent enumerators or Long Integer. / ConsideringRulerUnits: If true then a ruler based origin is interpreted using ruler units rather than points. The default value is false. This parameter has no effect unless the reference point is specified relative to a page.*
transformAgain	*variant* transformAgain()	*Transforms the ListBox using the last transformation performed on any object. Transformations include moving, rotating, shearing, scaling, and flipping. . / Return value: Playback errors. Type: Array of Strings.*
transformAgainIndividually	*variant* transformAgainIndividually()	*Transforms the ListBox using the last transformation performed on any ListBox. Transformations include moving, rotating, shearing, scaling, and flipping. . / Return value: Playback errors. Type: Array of Strings.*
transformSequenceAgain	*variant* transformSequenceAgain()	*Transforms the ListBox using the last sequence of transform operations performed on any single object or performed at the same time on any group of objects. Transformations include moving, rotating, shearing, scaling, and flipping. / Return value: Playback errors. Type: Array of Strings.*
transformSequenceAgainIndividually	*variant* transformSequenceAgainIndividually()	*Transforms the ListBox using the last sequence of transformations performed on any single object or performed at the same time on any group of objects. Transformations include moving, rotating, shearing, scaling, and flipping. / Return value: Playback errors. Type: Array of Strings.*
transformValuesOf	{Objects} transformValuesOf(#CoordinateSpaces In)	*Get the transformation values of the page item. / Return value: A list of the transformation matrices. / In: The coordinate space to use.*
absoluteFlip	#Flip absoluteFlip	*Indicates whether the ListBox has been flipped independently of its parent object and, if yes, the direction in which the ListBox was flipped.*

LISTBOX CS6 VALUES	PROTOTYPE	DESCRIPTION
absoluteHorizontalScale	*double* absoluteHorizontalScale	*The horizontal scale of the ListBox relative to its containing object.*
absoluteRotationAngle	*double* absoluteRotationAngle	*The rotation angle of the ListBox relative to its containing object. (Range: -360 to 360).*
absoluteShearAngle	*double* absoluteShearAngle	*The skewing angle of the ListBox relative to its containing object. (Range: -360 to 360).*
absoluteVerticalScale	*double* absoluteVerticalScale	*The vertical scale of the ListBox relative to its containing object.*
activeStateIndex	*integer* activeStateIndex	*The index of the active state in the object's states collection.*
allArticles	{Objects} •allArticles	*The list of all articles this page item is part of.*
allGraphics	{Objects} •allGraphics	*Lists all graphics contained by the ListBox.*
allowOverrides	*boolean* allowOverrides	*If true, the master page item can be overridden.*
allPageItems	{Objects} •allPageItems	*Lists all page items contained by the ListBox.*
anchoredObjectSettings	{AnchoredObjectSetting} •anchoredObjectSettings	*Anchored object settings.*
animationSettings	{AnimationSetting} •animationSettings	*The page item animation settings.*
appliedObjectStyle	{ObjectStyle} appliedObjectStyle	*The object style applied to the ListBox.*
associatedXMLElement	{XMLItem} •associatedXMLElement	*The XML element associated with the ListBox.*
behaviors	{Behaviors} •behaviors	*A collection of behavior objects.*
bottomLeftCornerOption	#CornerOptions bottomLeftCornerOption	*The shape to apply to the bottom left corner of rectangular shapes.*
bottomLeftCornerRadius	*variant* bottomLeftCornerRadius	*The radius in measurement units of the corner effect applied to the bottom left corner of rectangular shapes. Type: Unit (Double or String).*
bottomRightCornerOption	#CornerOptions bottomRightCornerOption	*The shape to apply to the bottom right corner of rectangular shapes.*
bottomRightCornerRadius	*variant* bottomRightCornerRadius	*The radius in measurement units of the corner effect applied to the bottom right corner of rectangular shapes. Type: Unit (Double or String).*
choiceList	*variant* choiceList	*The list of choices for combo and list boxes in the exported PDF. Type: Array of Strings.*
clearFormBehaviors	{ClearFormBehaviors} •clearFormBehaviors	*A collection of clear form behavior objects.*
contentTransparencySettings	{ContentTransparencySetting} •contentTransparencySettings	*Transparency settings for the content of the ListBox.*
description	*string* description	*The description of the ListBox.*
endCap	#EndCap endCap	*The end shape of an open path.*
endJoin	#EndJoin endJoin	*The corner join applied to the ListBox.*
epss	{EPSs} •epss	*A collection of EPS files.*
epsTexts	{EPSTexts} •epsTexts	*EPSTexts.*
fillColor	*variant* fillColor	*The swatch (color, gradient, tint, or mixed ink) applied to the fill of the ListBox. . Type: Swatch or String.*
fillTint	*double* fillTint	*The percent of tint to use in the ListBox's fill color. (To specify a tint percent, use a number in the range of 0 to 100; to use the inherited or overridden value, use -1.).*
fillTransparencySettings	{FillTransparencySetting} •fillTransparencySettings	*Transparency settings for the fill applied to the ListBox.*
flip	#Flip flip	*The direction in which to flip the printed image.*

LISTBOX CS6 VALUES	PROTOTYPE	DESCRIPTION
fontSize	*double* fontSize	*The font size for the form field in the exported PDF.*
gapColor	{Swatch} gapColor	*The swatch (color, gradient, tint, or mixed ink) applied to the gap of a dashed, dotted, or striped stroke. For information, see stroke type.*
gapTint	*double* gapTint	*The tint as a percentage of the gap color. (To specify a tint percent, use a number in the range of 0 to 100; to use the inherited or overridden value, use -1.).*
geometricBounds	*variant* geometricBounds	*The bounds of the ListBox excluding the stroke width, in the format [y1, x1, y2, x2], which give the coordinates of the top-left and bottom-right corners of the bounding box. Type: Array of 4 Units (Doubles or Strings).*
gotoAnchorBehaviors	{GotoAnchorBehaviors} •gotoAnchorBehaviors	*A collection of goto anchor behavior objects.*
gotoFirstPageBehaviors	{GotoFirstPageBehaviors} •gotoFirstPageBehaviors	*A collection of goto first page behavior objects.*
gotoLastPageBehaviors	{GotoLastPageBehaviors} •gotoLastPageBehaviors	*A collection of goto last page behavior objects.*
gotoNextPageBehaviors	{GotoNextPageBehaviors} •gotoNextPageBehaviors	*A collection of goto next page behavior objects.*
gotoNextViewBehaviors	{GotoNextViewBehaviors} •gotoNextViewBehaviors	*A collection of goto next view behavior objects.*
gotoPreviousPageBehaviors	{GotoPreviousPageBehaviors} •gotoPreviousPageBehaviors	*A collection of goto previous page behavior objects.*
gotoPreviousViewBehaviors	{GotoPreviousViewBehaviors} •gotoPreviousViewBehaviors	*A collection of goto previous view behavior objects.*
gotoURLBehaviors	{GotoURLBehaviors} •gotoURLBehaviors	*A collection of goto URL behavior objects.*
gradientFillAngle	*double* gradientFillAngle	*The angle of a linear gradient applied to the fill of the ListBox. (Range: -180 to 180).*
gradientFillLength	*variant* gradientFillLength	*The length (for a linear gradient) or radius (for a radial gradient) applied to the fill of the ListBox. Type: Unit (Double or String).*
gradientFillStart	*variant* gradientFillStart	*The starting point (in page coordinates) of a gradient applied to the fill of the ListBox, in the format [x, y]. Type: Array of 2 Units (Doubles or Strings).*
gradientStrokeAngle	*double* gradientStrokeAngle	*The angle of a linear gradient applied to the stroke of the ListBox. (Range: -180 to 180).*
gradientStrokeLength	*variant* gradientStrokeLength	*The length (for a linear gradient) or radius (for a radial gradient) applied to the stroke of the ListBox. Type: Unit (Double or String).*
gradientStrokeStart	*variant* gradientStrokeStart	*The starting point (in page coordinates) of a gradient applied to the stroke of the ListBox, in the format [x, y]. Type: Array of 2 Units (Doubles or Strings).*
graphicLines	{GraphicLines} •graphicLines	*A collection of graphic lines.*
graphics	{Graphics} •graphics	*A collection of imported graphics in any graphic file format (vector, metafile, or bitmap).*
groups	{Groups} •groups	*A collection of groups.*
hiddenUntilTriggered	*boolean* hiddenUntilTriggered	*If true, the form field/push button is hidden until triggered in the exported PDF.*
horizontalLayoutConstraints	*variant* horizontalLayoutConstraints	*The left margin, width, and right margin constraints this item is subject to when using the object-based layout rule. Type: Array of 3 idDimensionsConstraints enumerators.*
horizontalScale	*double* horizontalScale	*The horizontal scaling applied to the ListBox.*
id	*integer* •id	*The unique ID of the ListBox.*
images	{Images} •images	*A collection of bitmap images in any bitmap file format (including TIFF, JPEG, or GIF).*

LISTBOX CS6 VALUES	PROTOTYPE	DESCRIPTION
index	*integer* •index	*The index of the ListBox within its containing object.*
itemLayer	{Layer} itemLayer	*The layer that the ListBox is on.*
label	*string* label	*A property that can be set to any string.*
leftLineEnd	#ArrowHead leftLineEnd	*The arrowhead applied to the start of the path.*
linkedPageItemOptions	{LinkedPageItemOption} •linkedPageItemOptions	*Linked Page Item options.*
localDisplaySetting	#DisplaySettingOptions localDisplaySetting	*Display performance options for the ListBox.*
locked	*boolean* locked	*If true, the ListBox is locked.*
miterLimit	*double* miterLimit	*The limit of the ratio of stroke width to miter length before a miter (pointed) join becomes a bevel (squared-off) join. Type: Double (1 - 500).*
movieBehaviors	{MovieBehaviors} •movieBehaviors	*A collection of movie behavior objects.*
multipleSelection	*boolean* multipleSelection	*If true, the list box can have multiple items selected simultaneously in the exported PDF.*
name	*string* name	*The name of the ListBox.*
nonprinting	*boolean* nonprinting	*If true, the ListBox does not print.*
openFileBehaviors	{OpenFileBehaviors} •openFileBehaviors	*A collection of open file behavior objects.*
ovals	{Ovals} •ovals	*A collection of ellipses.*
overprintFill	*boolean* overprintFill	*If true, the ListBox's fill color overprints any underlying objects. If false, the fill color knocks out the underlying colors.*
overprintGap	*boolean* overprintGap	*If true, the gap color overprints any underlying colors. If false, the gap color knocks out the underlying colors.*
overprintStroke	*boolean* overprintStroke	*If true, the ListBox's stroke color overprints any underlying objects. If false, the stroke color knocks out the underlying colors.*
overridden	*boolean* •overridden	*If true, the object originated on a master spread and was overridden. If false, the object either originated on a master spread and was not overridden, or the object did not originate on a master page.*
overriddenMasterPageItem	*variant* •overriddenMasterPageItem	*An object that originated on a master page and has been overridden. Type: PageItem, Guide, Graphic, Movie or Sound.*
pageItems	{PageItems} •pageItems	*The page items collection, which can be used to process all page items in a container (such as a document, page, or group), regardless of type.*
parent	*variant* •parent	*The parent of the ListBox (a Snippet, PlaceGun, Character, Spread, MasterSpread, SplineItem, Polygon, GraphicLine, Rectangle, Oval or Group). Type: Snippet, PlaceGun, Character, Spread, MasterSpread, SplineItem, Polygon, GraphicLine, Rectangle, Oval or Group.*
parentPage	{Page} •parentPage	*The page on which this page item appears.*
pdfs	{PDFs} •pdfs	*A collection of PDF files.*
picts	{PICTs} •picts	*A collection of PICT graphics.*
polygons	{Polygons} •polygons	*A collection of polygons.*
preferences	{Preferences} •preferences	*A collection of preferences objects.*
printableInPdf	*boolean* printableInPdf	*If true, the form field/push button is printable in the exported PDF.*
printFormBehaviors	{PrintFormBehaviors} •printFormBehaviors	*A collection of print form behavior objects.*

LISTBOX CS6 VALUES	PROTOTYPE	DESCRIPTION
readOnly	*boolean* readOnly	*If true, the form field is read only in the exported PDF.*
rectangles	{Rectangles} •rectangles	*A collection of rectangles.*
required	*boolean* required	*If true, the form field is required in the exported PDF.*
rightLineEnd	#ArrowHead rightLineEnd	*The arrowhead applied to the end of the path.*
rightToLeftField	*boolean* rightToLeftField	*If true, the form field has right to left text enabled in the exported PDF.*
rotationAngle	*double* rotationAngle	*The rotatation angle of the ListBox. (Range: -360 to 360).*
shearAngle	*double* shearAngle	*The skewing angle applied to the ListBox. (Range: -360 to 360).*
showHideFieldsBehaviors	{ShowHideFieldsBehaviors} •showHideFieldsBehaviors	*A collection of show/hide fields behavior objects.*
sortChoices	*boolean* sortChoices	*If true, the choices would be sorted in the exported PDF.*
soundBehaviors	{SoundBehaviors} •soundBehaviors	*A collection of sound behavior objects.*
splineItems	{SplineItems} •splineItems	*The spline items collection.*
strokeAlignment	#StrokeAlignment strokeAlignment	*The stroke alignment applied to the ListBox.*
strokeColor	*variant* strokeColor	*The swatch (color, gradient, tint, or mixed ink) applied to the stroke of the ListBox. Type: Swatch or String.*
strokeCornerAdjustment	#StrokeCornerAdjustment strokeCornerAdjustment	*The corner adjustment applied to the ListBox.*
strokeDashAndGap	*variant* strokeDashAndGap	*The dash and gap measurements that define the pattern of a custom dashed line. Define up to six values (in points) in the format [dash1, gap1, dash2, gap2, dash3, gap3]. Type: Array of Units (Doubles or Strings).*
strokeTint	*double* strokeTint	*The percent of tint to use in object's stroke color. (To specify a tint percent, use a number in the range of 0 to 100; to use the inherited or overridden value, use -1.).*
strokeTransparencySettings	{StrokeTransparencySetting} •strokeTransparencySettings	*Transparency settings for the stroke.*
strokeType	*variant* strokeType	*The name of the stroke style to apply. Type: StrokeStyle or String.*
strokeWeight	*variant* strokeWeight	*The weight (in points) to apply to the ListBox's stroke. Type: Unit (Double or String).*
submitFormBehaviors	{SubmitFormBehaviors} •submitFormBehaviors	*A collection of submit form behavior objects.*
textFrames	{TextFrames} •textFrames	*A collection of text frames.*
textWrapPreferences	{TextWrapPreference} •textWrapPreferences	*The text wrap preference properties that define the default formatting for wrapping text around objects.*
timingSettings	{TimingSetting} •timingSettings	*The object timing settings.*
topLeftCornerOption	#CornerOptions topLeftCornerOption	*The shape to be applied to the top left corner of rectangular shapes and all corners of non-rectangular shapes.Note: corner option differs from end join in which you can set a radius for a corner option, whereas the rounded or beveled effect of an end join depends on the stroke weight.*
topLeftCornerRadius	*variant* topLeftCornerRadius	*The radius in measurement units of the corner effect applied to the top left corner of rectangular shapes and all corners of non-rectangular shapes. Type: Unit (Double or String).*
topRightCornerOption	#CornerOptions topRightCornerOption	*The shape to apply to the top right corner of rectangular shapes.*
topRightCornerRadius	*variant* topRightCornerRadius	*The radius in measurement units of the corner effect applied to the top right corner of rectangular shapes. Type: Unit (Double or String).*

LISTBOX CS6 VALUES	PROTOTYPE	DESCRIPTION
transparencySettings	{TransparencySetting} •transparencySettings	*Transparency settings.*
verticalLayoutConstraints	*variant* verticalLayoutConstraints	*The top margin, height, and bottom margin constraints this item is subject to when using the object-based layout rule. Type: Array of 3 idDimensionsConstraints enumerators.*
verticalScale	*double* verticalScale	*The vertical scaling applied to the ListBox.*
viewZoomBehaviors	{ViewZoomBehaviors} •viewZoomBehaviors	*A collection of view zoom behavior objects.*
visible	*boolean* visible	*If true, the ListBox is visible.*
visibleBounds	*variant* visibleBounds	*The bounds of the ListBox including the stroke width, in the format [y1, x1, y2, x2], which give the coordinates of the top-left and bottom-right corners of the bounding box. Type: Array of 4 Units (Doubles or Strings).*
wmfs	{WMFs} •wmfs	*A collection of WMF graphics.*

ListBoxes

Object available in ID 8

LISTBOXES CS6 VALUES	PROTOTYPE	DESCRIPTION
add	{ListBox} add(*variant* Layer, #LocationOptions At, *variant* Reference)	*Creates a new ListBox. / Return value: The new ListBox. / Layer: The layer on which to create the ListBox. Type: Layer. / At: The location at which to insert the ListBox relative to the reference object or within the container object. / Reference: The reference object. Note: Required when the at parameter is before or after. Type: Document, Spread, MasterSpread, Page, Layer or PageItem.*
anyItem	IDispatch anyItem()	*Returns any object in the collection. / Return value: The object.*
firstItem	IDispatch firstItem()	*Returns the first object in the collection. / Return value: The first object.*
itemByID	IDispatch itemByID(*integer* Id)	*Returns the object with the specified ID. / Return value: The object. / ID: The ID.*
itemByRange	{Objects} itemByRange(*variant* From, *variant* To)	*Returns the objects within the specified range. / Return value: The objects. / From: The object, index, or name at the beginning of the range. Type: Object, Long Integer or String. / To: The object, index, or name at the end of the range. Type: Object, Long Integer or String.*
lastItem	IDispatch lastItem()	*Returns the last object in the collection. / Return value: The last object.*
middleItem	IDispatch middleItem()	*Returns the middle object in the collection. / Return value: The middle object.*
nextItem	IDispatch nextItem(IDispatch Obj)	*Returns the object whose index follows the specified object in the collection. / Return value: The next object. / Obj: The object whose index comes before the desired object.*
previousItem	IDispatch previousItem(IDispatch Obj)	*Returns the object with the index previous to the specified index. / Return value: The previous object. / Obj: The index of the object that follows the desired object.*
count	*integer* •count	*Displays the number of elements in the ListBox.*
item	{ListBox} •item(*variant* Index)	*Returns the object with the specified index or name. Type: Long Integer or String.*

ListBoxEvents

Object available in ID 8

LISTBOXEVENTS CS6 VALUES	PROTOTYPE	DESCRIPTION
afterPlace	*const* afterPlace(afterPlace)	*Dispatched after a ListBox is placed. This event bubbles. This event is not cancelable.*
beforePlace	*const* beforePlace(beforePlace)	*Dispatched before a ListBox is placed. This event bubbles. This event is cancelable.*

ParaStyleMapping

Object available in ID 8

PARASTYLEMAPPING CS6 VALUES	PROTOTYPE	DESCRIPTION
delete	*void* delete()	*deletes a style mapping.*
destinationStyleName	*string* destinationStyleName	*The destination style name property.*
index	*integer* •index	*The index of the ParaStyleMapping within its containing object.*
mappingRuleType	#MapType mappingRuleType	*The type of the mapping rule.*
parent	*variant* •parent	*The parent of the ParaStyleMapping (a Application, Document, Story or XmlStory). Type: Application, Document, Story or XmlStory.*
sourceStyleName	*string* sourceStyleName	*The source style name property.*

ParaStyleMappings

Object available in ID 8

PARASTYLEMAPPINGS CS6 VALUES	PROTOTYPE	DESCRIPTION
add	{ParaStyleMapping} add(*string* SourceStyleName, *string* DestinationStyleName, #MapType MappingRuleType)	*Adds a style mapping. / Return value: The added ParaStyleMapping. / SourceStyleName: The source style name. / DestinationStyleName: The destination style name. / MappingRuleType: The mapping type.*
anyItem	IDispatch anyItem()	*Returns any object in the collection. / Return value: The object.*
firstItem	IDispatch firstItem()	*Returns the first object in the collection. / Return value: The first object.*
itemByRange	{Objects} itemByRange(*variant* From, *variant* To)	*Returns the objects within the specified range. / Return value: The objects. / From: The object, index, or name at the beginning of the range. Type: Object, Long Integer or String. / To: The object, index, or name at the end of the range. Type: Object, Long Integer or String.*
lastItem	IDispatch lastItem()	*Returns the last object in the collection. / Return value: The last object.*
middleItem	IDispatch middleItem()	*Returns the middle object in the collection. / Return value: The middle object.*
nextItem	IDispatch nextItem(IDispatch Obj)	*Returns the object whose index follows the specified object in the collection. / Return value: The next object. / Obj: The object whose index comes before the desired object.*

PARASTYLEMAPPINGS CS6 VALUES	PROTOTYPE	DESCRIPTION
previousItem	IDispatch previousItem(IDispatch Obj)	*Returns the object with the index previous to the specified index. / Return value: The previous object. / Obj: The index of the object that follows the desired object.*
count	*integer* •count	*Displays the number of elements in the ParaStyleMapping.*
item	{ParaStyleMapping} •item(*variant* Index)	*Returns the object with the specified index or name. Type: Long Integer or String.*

PNGExportPreference

Object available in ID 8

PNGEXPORTPREFERENCE CS6 VALUES	PROTOTYPE	DESCRIPTION
antiAlias	*boolean* antiAlias	*If true, use anti-aliasing for text and vectors during export.*
exportingSpread	*boolean* exportingSpread	*If true, exports each spread as a single PNG file. If false, exports facing pages as separate files and appends sequential numbers to each file name.*
exportResolution	*double* exportResolution	*The export resolution expressed as a real number instead of an integer. (Range: 1.0 to 2400.0). Type: Double (1 - 2400).*
pageString	*string* pageString	*The page(s) to export, specified as a page number or an array of page numbers. Note: Valid when PNG export range is not all.*
parent	{Application} •parent	*The parent of the PNGExportPreference (a Application).*
pngColorSpace	#PNGColorSpaceEnum pngColorSpace	*RGB or Gray.*
pngExportRange	#PNGExportRangeEnum pngExportRange	*The page range to export.*
pngQuality	#PNGQualityEnum pngQuality	*The compression quality.*
simulateOverprint	*boolean* simulateOverprint	*If true, simulates the effects of overprinting spot and process colors in the same way they would occur when printing.*
transparentBackground	*boolean* transparentBackground	*If true, use a transparent background for the exported PNG.*
useDocumentBleeds	*boolean* useDocumentBleeds	*If true, uses the document's bleed settings in the exported PNG.*

PrintFormBehavior

Object available in ID 8

PRINTFORMBEHAVIOR CS6 VALUES	PROTOTYPE	DESCRIPTION
delete	*void* delete()	*Deletes the PrintFormBehavior.*
extractLabel	*string* extractLabel(*string* Key)	*Gets the label value associated with the specified key. / Return value: The label. / Key: The key.*
insertLabel	*void* insertLabel(*string* Key, *string* Value)	*Sets the label to the value associated with the specified key. / Key: The key. / Value: The value.*
behaviorEvent	#BehaviorEvents behaviorEvent	*The event which triggers the behavior.*
enableBehavior	*boolean* enableBehavior	*If true, the behavior is enabled.*

PRINTFORMBEHAVIOR CS6 VALUES	PROTOTYPE	DESCRIPTION
id	*integer* •id	*The unique ID of the PrintFormBehavior.*
index	*integer* •index	*The index of the PrintFormBehavior within its containing object.*
label	*string* label	*A property that can be set to any string.*
name	*string* •name	*The name of the PrintFormBehavior.*
parent	*variant* •parent	*The parent of the PrintFormBehavior (a Button, CheckBox, ComboBox, ListBox, RadioButton, TextBox or SignatureField). Type: Button, CheckBox, ComboBox, ListBox, RadioButton, TextBox or SignatureField.*

PrintFormBehaviors

Object available in ID 8

PRINTFORMBEHAVIORS CS6 VALUES	PROTOTYPE	DESCRIPTION
add	{PrintFormBehavior} add()	*Creates a new PrintFormBehavior. / Return value: The new PrintFormBehavior.*
anyItem	IDispatch anyItem()	*Returns any object in the collection. / Return value: The object.*
firstItem	IDispatch firstItem()	*Returns the first object in the collection. / Return value: The first object.*
itemByID	IDispatch itemByID(*integer* Id)	*Returns the object with the specified ID. / Return value: The object. / ID: The ID.*
itemByRange	{Objects} itemByRange(*variant* From, *variant* To)	*Returns the objects within the specified range. / Return value: The objects. / From: The object, index, or name at the beginning of the range. Type: Object, Long Integer or String. / To: The object, index, or name at the end of the range. Type: Object, Long Integer or String.*
lastItem	IDispatch lastItem()	*Returns the last object in the collection. / Return value: The last object.*
middleItem	IDispatch middleItem()	*Returns the middle object in the collection. / Return value: The middle object.*
nextItem	IDispatch nextItem(IDispatch Obj)	*Returns the object whose index follows the specified object in the collection. / Return value: The next object. / Obj: The object whose index comes before the desired object.*
previousItem	IDispatch previousItem(IDispatch Obj)	*Returns the object with the index previous to the specified index. / Return value: The previous object. / Obj: The index of the object that follows the desired object.*
count	*integer* •count	*Displays the number of elements in the PrintFormBehavior.*
item	{PrintFormBehavior} •item(*variant* Index)	*Returns the object with the specified index or name. Type: Long Integer or String.*

RadioButton

Object available in ID 8

RADIOBUTTON CS6 VALUES	PROTOTYPE	DESCRIPTION
applyObjectStyle	void applyObjectStyle(ObjectStyle Using, boolean ClearingOverrides, boolean ClearingOverridesThroughRootObjectStyle)	*Applies the specified object style. / Using: The object style to apply. / ClearingOverrides: If true, clears the RadioButton's existing attributes before applying the style. / ClearingOverridesThroughRootObjectStyle: If true, clears attributes and formatting applied to the RadioButton that are not defined in the object style.*
asynchronousExportFile	{BackgroundTask} asynchronousExportFile(variant Format, string To, boolean ShowingOptions, variant Using, variant VersionComments, boolean ForceSave)	*asynchronously exports the object(s) to a file. / Return value: object representing the asynchronous export. / Format: The export format, specified as an enumeration value or as an extension that appears in the Save as type or Format menu in the Export dialog. Type: idExportFormat enumerator or String. / To: The path to the export file. / ShowingOptions: If true, displays the export options dialog. / Using: The export style. Type: PDFExportPreset. / VersionComments: The comment for this version. Type: String. / ForceSave: If true, forcibly saves a version.*
autoTag	void autoTag()	*Tag the object or the parent story using default tags defined in XML preference.*
bringForward	void bringForward()	*Brings the RadioButton forward one level in its layer.*
bringToFront	void bringToFront(variant Reference)	*Brings the RadioButton to the front of its layer or in front of a particular item. / Reference: The reference object to bring the object in front of (must have same parent). Type: PageItem.*
changeObject	{Objects} changeObject(variant ReverseOrder)	*Finds objects that match the find what value and replace the objects with the change to value. / Return value: The replaced object(s). / ReverseOrder: If true, returns the results in reverse order. Type: Boolean.*
clearObjectStyleOverrides	void clearObjectStyleOverrides()	*Clear overrides for object style.*
clearTransformations	void clearTransformations()	*Clears transformations from the RadioButton. Transformations include rotation, scaling, flipping, fitting, and shearing.*
contentPlace	variant contentPlace(Objects PageItems, boolean LinkPageItems, boolean LinkStories, boolean MapStyles, boolean ShowingOptions)	*Duplicate an object and place it into the target page item. / Return value: The placed content. Type: Array of PageItems, Graphics, Stories or Texts. / PageItems: One or more page items to place or load. / LinkPageItems: Whether to link pageItems in content placer (if true it will override link stories value). / LinkStories: Whether to link stories in content placer (only applicable for single story, pageItem links will also be created in case of more than one item). / MapStyles: Whether to map styles in content placer. / ShowingOptions: Whether to display the link options dialog.*
convertShape	void convertShape(#ConvertShapeOptions Given, variant NumberOfSides, variant InsetPercentage, variant CornerRadius)	*Converts the RadioButton to a different shape. / Given: The RadioButton's new shape. / NumberOfSides: The number of sides for the resulting polygon. (Range: 3 to 100). Type: Long Integer (3 - 100). / InsetPercentage: The star inset percentage for the resulting polygon. (Range: 0.0 to 100.0) . Type: Long Integer (0 - 100). / CornerRadius: The corner radius of the resulting rectangle. Type: Unit (Double or String) (0 - 800 points).*
convertToObject	void convertToObject()	*Converts the button object to the page item currently in the active state. Page items from other states will be lost.*
delete	void delete()	*Deletes the RadioButton.*
detach	void detach()	*Detaches an overridden master page item from the master page.*

RADIOBUTTON CS6 VALUES	PROTOTYPE	DESCRIPTION
duplicate	{PageItem} duplicate(*variant* To, *variant* By)	*Duplicates the RadioButton at the specified location or offset. / Return value: The duplicated RadioButton. / To: The location of the new RadioButton, specified in coordinates in the format [x, y]. Type: Array of 2 Units (Doubles or Strings), Spread, Page or Layer. / By: Amount by which to offset the new RadioButton from the original RadioButton's position. Type: Array of 2 Units (Doubles or Strings).*
export	*void* export(*variant* Format, *string* To, *boolean* ShowingOptions, *variant* Using, *variant* VersionComments, *boolean* ForceSave)	*Exports the object(s) to a file. / Format: The export format, specified as an enumeration value or as an extension that appears in the Save as type or Format menu in the Export dialog. Type: idExportFormat enumerator or String. / To: The path to the export file. / ShowingOptions: If true, displays the export options dialog. / Using: The export style. Type: PDFExportPreset. / VersionComments: The comment for this version. Type: String. / ForceSave: If true, forcibly saves a version.*
extractLabel	*string* extractLabel(*string* Key)	*Gets the label value associated with the specified key. / Return value: The label. / Key: The key.*
findObject	{Objects} findObject(*variant* ReverseOrder)	*Finds objects that match the find what value. / Return value: The found object(s). / ReverseOrder: If true, returns the results in reverse order. Type: Boolean.*
fit	*void* fit(#FitOptions Given)	*Applies the specified fit option to content in a frame. / Given: The fit option to use.*
flipItem	*void* flipItem(#Flip Given, *variant* Around)	*Flips the RadioButton. / Given: The axis around which to flip the RadioButton. / Around: The point around which to flip the RadioButton. Type: Array of 2 Units (Doubles or Strings) or idAnchorPoint enumerator.*
insertLabel	*void* insertLabel(*string* Key, *string* Value)	*Sets the label to the value associated with the specified key. / Key: The key. / Value: The value.*
markup	*void* markup(XMLElement Using)	*Associates the page item with the specified XML element while preserving existing content. / Using: The XML element.*
move	*void* move(*variant* To, *variant* By)	*Moves the RadioButton to a new location. Note: Either the 'to' or 'by' parameter is required; if both parameters are defined, only the to value is used. / To: The new location of the RadioButton,in the format (x, y). Type: Array of 2 Units (Doubles or Strings), Spread, Page or Layer. / By: The amount (in measurement units) to move the RadioButton relative to its current position, in the format (x, y). Type: Array of 2 Units (Doubles or Strings).*
override	*variant* override(Page DestinationPage)	*Overrides a master page item and places the item on the document page as a new object. / Return value: The new object. Type: PageItem, Guide, Graphic, Movie or Sound. / DestinationPage: The document page that contains the master page item to override.*
placeXML	*void* placeXML(XMLElement Using)	*Places XML content into the specified object. Note: Replaces any existing content. / Using: The XML element whose content you want to place.*
redefineScaling	*void* redefineScaling(*variant* To)	*Apply an item's scaling to its content if possible. / To: The scale factors to be left on the item. The default is {1.0, 1.0}. Type: Array of 2 Doubles.*
reframe	*void* reframe(*variant* In, *variant* OpposingCorners)	*Move the bounding box of the page item. / In: The bounding box to resize. Type: idCoordinateSpaces enumerator or Ordered array containing CoordinateSpace:idCoordinateSpaces enumerator, BoundsKind:idBoundingBoxLimits enumerator. / OpposingCorners: Opposing corners of new bounding box in the given coordinate space. Type: Array of 2 Arrays of 2 Doubles.*
removeOverride	*void* removeOverride()	*Removes the override from a previously overridden master page item.*

RADIOBUTTON CS6 VALUES	PROTOTYPE	DESCRIPTION
resize	*void* resize(*variant* In, *variant* From, #ResizeMethods By, *variant* Values, *boolean* ResizeIndividually, *boolean* ConsideringRulerUnits)	*Resize the page item. / In: The bounding box to resize. Type: idCoordinateSpaces enumerator, idBoundingBoxLimits enumerator or Ordered array containing CoordinateSpace:idCoordinateSpaces enumerator, BoundsKind:idBoundingBoxLimits enumerator. / From: The transform origin. Legal specifications: relative to bounding box: anchor | {anchor | {x,y}, bounds kind [, coordinate space]}; relative to coordinate space: {x,y} | {{x,y} [, coordinate space]}; relative to layout window ruler: {{x,y}, page index | bounds kind}. Type: Array of 2 Doubles, idAnchorPoint enumerator or Array of Arrays of 2 Doubles, idCoordinateSpaces enumerators, idAnchorPoint enumerators, idBoundingBoxLimits enumerators or Long Integers. / By: How the current dimensions are affected by the given values. / Values: The width and height values. Legal dimensions specifications: {x, y [, coordinate space]}, {x, resize constraint [, coordinate space]}, or {resize constraint, y [, coordinate space]}; where x and y are real numbers and coordinate space is used to determine _only_ the unit of length for x and y; coordinate space is ignored for the 'current dimensions times' resize method). Type: Array of Doubles, idResizeConstraints enumerators or idCoordinateSpaces enumerators. / ResizeIndividually: If false and multiple page items are targeted, the new dimensions are attained only by moving the individual items rather than resizing them. / ConsideringRulerUnits: If true then a ruler location is interpreted using ruler units rather than points. The default value is false. This parameter has no effect unless the reference point is specified relative to a page.*
resolve	*variant* resolve(*variant* Location, #CoordinateSpaces In, *boolean* ConsideringRulerUnits)	*Get the coordinates of the given location in the specified coordinate system. / Return value: The resolved location for each of the targeted page items. Type: Array of 2 Doubles or Array of Arrays of 2 Doubles. / Location: The location requested. Type: Array of 2 Doubles, idAnchorPoint enumerator or Array of Arrays of 2 Doubles, idCoordinateSpaces enumerators, idAnchorPoint enumerators, idBoundingBoxLimits enumerators or Long Integers. / In: The coordinate space to use. / ConsideringRulerUnits: If true then a ruler location is interpreted using ruler units rather than points. The default value is false. This parameter has no effect unless the reference point is specified relative to a page.*
select	*void* select(#SelectionOptions ExistingSelection)	*Selects the object. / ExistingSelection: The selection status of the RadioButton in relation to previously selected objects.*
sendBackward	*void* sendBackward()	*Sends the RadioButton back one level in its layer.*
sendToBack	*void* sendToBack(*variant* Reference)	*Sends the RadioButton to the back of its layer or behind a particular item (must have same parent). / Reference: The reference object to send the object behind. Type: PageItem.*
store	{Asset} store(Library Using)	*Stores the object in the specified library. / Return value: The new library asset(s). / Using: The library in which to store the object.*

RADIOBUTTON CS6 VALUES	PROTOTYPE	DESCRIPTION
transform	*void* transform(#CoordinateSpaces In, *variant* From, *variant* WithMatrix, *variant* ReplacingCurrent, *boolean* ConsideringRulerUnits)	*Transform the page item. / In: The coordinate space to use. / From: The temporary origin during the transformation. Type: Array of 2 Doubles, idAnchorPoint enumerator or Array of Arrays of 2 Doubles, idCoordinateSpaces enumerators, idAnchorPoint enumerators, idBoundingBoxLimits enumerators or Long Integers. / WithMatrix: Transform matrix. Type: Array of 6 Doubles or TransformationMatrix. / ReplacingCurrent: Transform components to consider; providing this optional parameter causes the target's existing transform components to be replaced with new values. Without this parameter, the given matrix is concatenated onto the target's existing transform combining the effect of the two. Type: idMatrixContent enumerator, Array of idMatrixContent enumerators or Long Integer. / ConsideringRulerUnits: If true then a ruler based origin is interpreted using ruler units rather than points. The default value is false. This parameter has no effect unless the reference point is specified relative to a page.*
transformAgain	*variant* transformAgain()	*Transforms the RadioButton using the last transformation performed on any object. Transformations include moving, rotating, shearing, scaling, and flipping. . / Return value: Playback errors. Type: Array of Strings.*
transformAgainIndividually	*variant* transformAgainIndividually()	*Transforms the RadioButton using the last transformation performed on any RadioButton. Transformations include moving, rotating, shearing, scaling, and flipping. . / Return value: Playback errors. Type: Array of Strings.*
transformSequenceAgain	*variant* transformSequenceAgain()	*Transforms the RadioButton using the last sequence of transform operations performed on any single object or performed at the same time on any group of objects. Transformations include moving, rotating, shearing, scaling, and flipping. / Return value: Playback errors. Type: Array of Strings.*
transformSequenceAgainIndividually	*variant* transformSequenceAgainIndividually()	*Transforms the RadioButton using the last sequence of transformations performed on any single object or performed at the same time on any group of objects. Transformations include moving, rotating, shearing, scaling, and flipping. / Return value: Playback errors. Type: Array of Strings.*
transformValuesOf	{Objects} transformValuesOf(#CoordinateSpaces In)	*Get the transformation values of the page item. / Return value: A list of the transformation matrices. / In: The coordinate space to use.*
absoluteFlip	#Flip absoluteFlip	*Indicates whether the RadioButton has been flipped independently of its parent object and, if yes, the direction in which the RadioButton was flipped.*
absoluteHorizontalScale	*double* absoluteHorizontalScale	*The horizontal scale of the RadioButton relative to its containing object.*
absoluteRotationAngle	*double* absoluteRotationAngle	*The rotation angle of the RadioButton relative to its containing object. (Range: -360 to 360).*
absoluteShearAngle	*double* absoluteShearAngle	*The skewing angle of the RadioButton relative to its containing object. (Range: -360 to 360).*
absoluteVerticalScale	*double* absoluteVerticalScale	*The vertical scale of the RadioButton relative to its containing object.*
activeStateIndex	*integer* activeStateIndex	*The index of the active state in the object's states collection.*
allArticles	{Objects} •allArticles	*The list of all articles this page item is part of.*
allGraphics	{Objects} •allGraphics	*Lists all graphics contained by the RadioButton.*
allowOverrides	*boolean* allowOverrides	*If true, the master page item can be overridden.*
allPageItems	{Objects} •allPageItems	*Lists all page items contained by the RadioButton.*
anchoredObjectSettings	{AnchoredObjectSetting} •anchoredObjectSettings	*Anchored object settings.*
animationSettings	{AnimationSetting} •animationSettings	*The page item animation settings.*
appliedObjectStyle	{ObjectStyle} appliedObjectStyle	*The object style applied to the RadioButton.*

RADIOBUTTON CS6 VALUES	PROTOTYPE	DESCRIPTION
associatedXMLElement	{XMLItem} •associatedXMLElement	*The XML element associated with the RadioButton.*
behaviors	{Behaviors} •behaviors	*A collection of behavior objects.*
bottomLeftCornerOption	#CornerOptions bottomLeftCornerOption	*The shape to apply to the bottom left corner of rectangular shapes.*
bottomLeftCornerRadius	*variant* bottomLeftCornerRadius	*The radius in measurement units of the corner effect applied to the bottom left corner of rectangular shapes. Type: Unit (Double or String).*
bottomRightCornerOption	#CornerOptions bottomRightCornerOption	*The shape to apply to the bottom right corner of rectangular shapes.*
bottomRightCornerRadius	*variant* bottomRightCornerRadius	*The radius in measurement units of the corner effect applied to the bottom right corner of rectangular shapes. Type: Unit (Double or String).*
checkedByDefault	*boolean* checkedByDefault	*If true, the check box/radio button is selected by default in the exported PDF.*
clearFormBehaviors	{ClearFormBehaviors} •clearFormBehaviors	*A collection of clear form behavior objects.*
contentTransparencySettings	{ContentTransparencySetting} •contentTransparencySettings	*Transparency settings for the content of the RadioButton.*
description	*string* description	*The description of the RadioButton.*
endCap	#EndCap endCap	*The end shape of an open path.*
endJoin	#EndJoin endJoin	*The corner join applied to the RadioButton.*
epss	{EPSs} •epss	*A collection of EPS files.*
epsTexts	{EPSTexts} •epsTexts	*EPSTexts.*
exportValue	*string* exportValue	*Export value for the check box/radio button in the exported PDF.*
fillColor	*variant* fillColor	*The swatch (color, gradient, tint, or mixed ink) applied to the fill of the RadioButton. . Type: Swatch or String.*
fillTint	*double* fillTint	*The percent of tint to use in the RadioButton's fill color. (To specify a tint percent, use a number in the range of 0 to 100; to use the inherited or overridden value, use -1.).*
fillTransparencySettings	{FillTransparencySetting} •fillTransparencySettings	*Transparency settings for the fill applied to the RadioButton.*
flip	#Flip flip	*The direction in which to flip the printed image.*
gapColor	{Swatch} gapColor	*The swatch (color, gradient, tint, or mixed ink) applied to the gap of a dashed, dotted, or striped stroke. For information, see stroke type.*
gapTint	*double* gapTint	*The tint as a percentage of the gap color. (To specify a tint percent, use a number in the range of 0 to 100; to use the inherited or overridden value, use -1.).*
geometricBounds	*variant* geometricBounds	*The bounds of the RadioButton excluding the stroke width, in the format [y1, x1, y2, x2], which give the coordinates of the top-left and bottom-right corners of the bounding box. Type: Array of 4 Units (Doubles or Strings).*
gotoAnchorBehaviors	{GotoAnchorBehaviors} •gotoAnchorBehaviors	*A collection of goto anchor behavior objects.*
gotoFirstPageBehaviors	{GotoFirstPageBehaviors} •gotoFirstPageBehaviors	*A collection of goto first page behavior objects.*
gotoLastPageBehaviors	{GotoLastPageBehaviors} •gotoLastPageBehaviors	*A collection of goto last page behavior objects.*
gotoNextPageBehaviors	{GotoNextPageBehaviors} •gotoNextPageBehaviors	*A collection of goto next page behavior objects.*
gotoNextViewBehaviors	{GotoNextViewBehaviors} •gotoNextViewBehaviors	*A collection of goto next view behavior objects.*
gotoPreviousPageBehaviors	{GotoPreviousPageBehaviors} •gotoPreviousPageBehaviors	*A collection of goto previous page behavior objects.*
gotoPreviousViewBehaviors	{GotoPreviousViewBehaviors} •gotoPreviousViewBehaviors	*A collection of goto previous view behavior objects.*

RADIOBUTTON CS6 VALUES	PROTOTYPE	DESCRIPTION
gotoURLBehaviors	{GotoURLBehaviors} •gotoURLBehaviors	*A collection of goto URL behavior objects.*
gradientFillAngle	*double* gradientFillAngle	*The angle of a linear gradient applied to the fill of the RadioButton. (Range: -180 to 180).*
gradientFillLength	*variant* gradientFillLength	*The length (for a linear gradient) or radius (for a radial gradient) applied to the fill of the RadioButton. Type: Unit (Double or String).*
gradientFillStart	*variant* gradientFillStart	*The starting point (in page coordinates) of a gradient applied to the fill of the RadioButton, in the format [x, y]. Type: Array of 2 Units (Doubles or Strings).*
gradientStrokeAngle	*double* gradientStrokeAngle	*The angle of a linear gradient applied to the stroke of the RadioButton. (Range: -180 to 180).*
gradientStrokeLength	*variant* gradientStrokeLength	*The length (for a linear gradient) or radius (for a radial gradient) applied to the stroke of the RadioButton. Type: Unit (Double or String).*
gradientStrokeStart	*variant* gradientStrokeStart	*The starting point (in page coordinates) of a gradient applied to the stroke of the RadioButton, in the format [x, y]. Type: Array of 2 Units (Doubles or Strings).*
graphicLines	{GraphicLines} •graphicLines	*A collection of graphic lines.*
graphics	{Graphics} •graphics	*A collection of imported graphics in any graphic file format (vector, metafile, or bitmap).*
groups	{Groups} •groups	*A collection of groups.*
hiddenUntilTriggered	*boolean* hiddenUntilTriggered	*If true, the form field/push button is hidden until triggered in the exported PDF.*
horizontalLayoutConstraints	*variant* horizontalLayoutConstraints	*The left margin, width, and right margin constraints this item is subject to when using the object-based layout rule. Type: Array of 3 idDimensionsConstraints enumerators.*
horizontalScale	*double* horizontalScale	*The horizontal scaling applied to the RadioButton.*
id	*integer* •id	*The unique ID of the RadioButton.*
images	{Images} •images	*A collection of bitmap images in any bitmap file format (including TIFF, JPEG, or GIF).*
index	*integer* •index	*The index of the RadioButton within its containing object.*
itemLayer	{Layer} itemLayer	*The layer that the RadioButton is on.*
label	*string* label	*A property that can be set to any string.*
leftLineEnd	#ArrowHead leftLineEnd	*The arrowhead applied to the start of the path.*
linkedPageItemOptions	{LinkedPageItemOption} •linkedPageItemOptions	*Linked Page Item options.*
localDisplaySetting	#DisplaySettingOptions localDisplaySetting	*Display performance options for the RadioButton.*
locked	*boolean* locked	*If true, the RadioButton is locked.*
miterLimit	*double* miterLimit	*The limit of the ratio of stroke width to miter length before a miter (pointed) join becomes a bevel (squared-off) join. Type: Double (1 - 500).*
movieBehaviors	{MovieBehaviors} •movieBehaviors	*A collection of movie behavior objects.*
name	*string* name	*The name of the RadioButton.*
nonprinting	*boolean* nonprinting	*If true, the RadioButton does not print.*
openFileBehaviors	{OpenFileBehaviors} •openFileBehaviors	*A collection of open file behavior objects.*

RADIOBUTTON CS6 VALUES	PROTOTYPE	DESCRIPTION
ovals	{Ovals} •ovals	*A collection of ellipses.*
overprintFill	*boolean* overprintFill	*If true, the RadioButton's fill color overprints any underlying objects. If false, the fill color knocks out the underlying colors.*
overprintGap	*boolean* overprintGap	*If true, the gap color overprints any underlying colors. If false, the gap color knocks out the underlying colors.*
overprintStroke	*boolean* overprintStroke	*If true, the RadioButton's stroke color overprints any underlying objects. If false, the stroke color knocks out the underlying colors.*
overridden	*boolean* •overridden	*If true, the object originated on a master spread and was overridden. If false, the object either originated on a master spread and was not overridden, or the object did not originate on a master page.*
overriddenMasterPageItem	*variant* •overriddenMasterPageItem	*An object that originated on a master page and has been overridden. Type: PageItem, Guide, Graphic, Movie or Sound.*
pageItems	{PageItems} •pageItems	*The page items collection, which can be used to process all page items in a container (such as a document, page, or group), regardless of type.*
parent	*variant* •parent	*The parent of the RadioButton (a Snippet, PlaceGun, Character, Spread, MasterSpread, SplineItem, Polygon, GraphicLine, Rectangle, Oval or Group). Type: Snippet, PlaceGun, Character, Spread, MasterSpread, SplineItem, Polygon, GraphicLine, Rectangle, Oval or Group.*
parentPage	{Page} •parentPage	*The page on which this page item appears.*
pdfs	{PDFs} •pdfs	*A collection of PDF files.*
picts	{PICTs} •picts	*A collection of PICT graphics.*
polygons	{Polygons} •polygons	*A collection of polygons.*
preferences	{Preferences} •preferences	*A collection of preferences objects.*
printableInPdf	*boolean* printableInPdf	*If true, the form field/push button is printable in the exported PDF.*
printFormBehaviors	{PrintFormBehaviors} •printFormBehaviors	*A collection of print form behavior objects.*
readOnly	*boolean* readOnly	*If true, the form field is read only in the exported PDF.*
rectangles	{Rectangles} •rectangles	*A collection of rectangles.*
required	*boolean* required	*If true, the form field is required in the exported PDF.*
rightLineEnd	#ArrowHead rightLineEnd	*The arrowhead applied to the end of the path.*
rotationAngle	*double* rotationAngle	*The rotatation angle of the RadioButton. (Range: -360 to 360).*
shearAngle	*double* shearAngle	*The skewing angle applied to the RadioButton. (Range: -360 to 360).*
showHideFieldsBehaviors	{ShowHideFieldsBehaviors} •showHideFieldsBehaviors	*A collection of show/hide fields behavior objects.*
soundBehaviors	{SoundBehaviors} •soundBehaviors	*A collection of sound behavior objects.*
splineItems	{SplineItems} •splineItems	*The spline items collection.*
states	{States} •states	*A collection of states.*
strokeAlignment	#StrokeAlignment strokeAlignment	*The stroke alignment applied to the RadioButton.*
strokeColor	*variant* strokeColor	*The swatch (color, gradient, tint, or mixed ink) applied to the stroke of the RadioButton. Type: Swatch or String.*
strokeCornerAdjustment	#StrokeCornerAdjustment strokeCornerAdjustment	*The corner adjustment applied to the RadioButton.*

RADIOBUTTON CS6 VALUES	PROTOTYPE	DESCRIPTION
strokeDashAndGap	variant strokeDashAndGap	*The dash and gap measurements that define the pattern of a custom dashed line. Define up to six values (in points) in the format [dash1, gap1, dash2, gap2, dash3, gap3]. Type: Array of Units (Doubles or Strings).*
strokeTint	double strokeTint	*The percent of tint to use in object's stroke color. (To specify a tint percent, use a number in the range of 0 to 100; to use the inherited or overridden value, use -1.).*
strokeTransparencySettings	{StrokeTransparencySetting} •strokeTransparencySettings	*Transparency settings for the stroke.*
strokeType	variant strokeType	*The name of the stroke style to apply. Type: StrokeStyle or String.*
strokeWeight	variant strokeWeight	*The weight (in points) to apply to the RadioButton's stroke. Type: Unit (Double or String).*
submitFormBehaviors	{SubmitFormBehaviors} •submitFormBehaviors	*A collection of submit form behavior objects.*
textFrames	{TextFrames} •textFrames	*A collection of text frames.*
textWrapPreferences	{TextWrapPreference} •textWrapPreferences	*The text wrap preference properties that define the default formatting for wrapping text around objects.*
timingSettings	{TimingSetting} •timingSettings	*The object timing settings.*
topLeftCornerOption	#CornerOptions topLeftCornerOption	*The shape to be applied to the top left corner of rectangular shapes and all corners of non-rectangular shapes.Note: corner option differs from end join in which you can set a radius for a corner option, whereas the rounded or beveled effect of an end join depends on the stroke weight.*
topLeftCornerRadius	variant topLeftCornerRadius	*The radius in measurement units of the corner effect applied to the top left corner of rectangular shapes and all corners of non-rectangular shapes. Type: Unit (Double or String).*
topRightCornerOption	#CornerOptions topRightCornerOption	*The shape to apply to the top right corner of rectangular shapes.*
topRightCornerRadius	variant topRightCornerRadius	*The radius in measurement units of the corner effect applied to the top right corner of rectangular shapes. Type: Unit (Double or String).*
transparencySettings	{TransparencySetting} •transparencySettings	*Transparency settings.*
verticalLayoutConstraints	variant verticalLayoutConstraints	*The top margin, height, and bottom margin constraints this item is subject to when using the object-based layout rule. Type: Array of 3 idDimensionsConstraints enumerators.*
verticalScale	double verticalScale	*The vertical scaling applied to the RadioButton.*
viewZoomBehaviors	{ViewZoomBehaviors} •viewZoomBehaviors	*A collection of view zoom behavior objects.*
visible	boolean visible	*If true, the RadioButton is visible.*
visibleBounds	variant visibleBounds	*The bounds of the RadioButton including the stroke width, in the format [y1, x1, y2, x2], which give the coordinates of the top-left and bottom-right corners of the bounding box. Type: Array of 4 Units (Doubles or Strings).*
wmfs	{WMFs} •wmfs	*A collection of WMF graphics.*

RadioButtonEvents

Object available in ID 8

RADIOBUTTONEVENTS CS6 VALUES	PROTOTYPE	DESCRIPTION
afterPlace	*const* afterPlace(afterPlace)	*Dispatched after a RadioButton is placed. This event bubbles. This event is not cancelable.*
beforePlace	*const* beforePlace(beforePlace)	*Dispatched before a RadioButton is placed. This event bubbles. This event is cancelable.*

RadioButtons

Object available in ID 8

RADIOBUTTONS CS6 VALUES	PROTOTYPE	DESCRIPTION
add	{RadioButton} add(*variant* Layer, #LocationOptions At, *variant* Reference)	*Creates a new RadioButton. / Return value: The new RadioButton. / Layer: The layer on which to create the RadioButton. Type: Layer. / At: The location at which to insert the RadioButton relative to the reference object or within the container object. / Reference: The reference object. Note: Required when the at parameter is before or after. Type: Document, Spread, MasterSpread, Page, Layer or PageItem.*
anyItem	IDispatch anyItem()	*Returns any object in the collection. / Return value: The object.*
firstItem	IDispatch firstItem()	*Returns the first object in the collection. / Return value: The first object.*
itemByID	IDispatch itemByID(*integer* Id)	*Returns the object with the specified ID. / Return value: The object. / ID: The ID.*
itemByRange	{Objects} itemByRange(*variant* From, *variant* To)	*Returns the objects within the specified range. / Return value: The objects. / From: The object, index, or name at the beginning of the range. Type: Object, Long Integer or String. / To: The object, index, or name at the end of the range. Type: Object, Long Integer or String.*
lastItem	IDispatch lastItem()	*Returns the last object in the collection. / Return value: The last object.*
middleItem	IDispatch middleItem()	*Returns the middle object in the collection. / Return value: The middle object.*
nextItem	IDispatch nextItem(IDispatch Obj)	*Returns the object whose index follows the specified object in the collection. / Return value: The next object. / Obj: The object whose index comes before the desired object.*
previousItem	IDispatch previousItem(IDispatch Obj)	*Returns the object with the index previous to the specified index. / Return value: The previous object. / Obj: The index of the object that follows the desired object.*
count	*integer* •count	*Displays the number of elements in the RadioButton.*
item	{RadioButton} •item(*variant* Index)	*Returns the object with the specified index or name. Type: Long Integer or String.*

SignatureField

Object available in ID 8

SIGNATUREFIELD CS6 VALUES	PROTOTYPE	DESCRIPTION
applyObjectStyle	*void* applyObjectStyle(ObjectStyle Using, *boolean* ClearingOverrides, *boolean* ClearingOverridesThroughRootObjectStyle)	*Applies the specified object style. / Using: The object style to apply. / ClearingOverrides: If true, clears the SignatureField's existing attributes before applying the style. / ClearingOverridesThroughRootObjectStyle: If true, clears attributes and formatting applied to the SignatureField that are not defined in the object style.*
asynchronousExportFile	{BackgroundTask} asynchronousExportFile(*variant* Format, *string* To, *boolean* ShowingOptions, *variant* Using, *variant* VersionComments, *boolean* ForceSave)	*asynchronously exports the object(s) to a file. / Return value: object representing the asynchronous export. / Format: The export format, specified as an enumeration value or as an extension that appears in the Save as type or Format menu in the Export dialog. Type: idExportFormat enumerator or String. / To: The path to the export file. / ShowingOptions: If true, displays the export options dialog. / Using: The export style. Type: PDFExportPreset. / VersionComments: The comment for this version. Type: String. / ForceSave: If true, forcibly saves a version.*
autoTag	*void* autoTag()	*Tag the object or the parent story using default tags defined in XML preference.*
bringForward	*void* bringForward()	*Brings the SignatureField forward one level in its layer.*
bringToFront	*void* bringToFront(*variant* Reference)	*Brings the SignatureField to the front of its layer or in front of a particular item. / Reference: The reference object to bring the object in front of (must have same parent). Type: PageItem.*
changeObject	{Objects} changeObject(*variant* ReverseOrder)	*Finds objects that match the find what value and replace the objects with the change to value. / Return value: The replaced object(s). / ReverseOrder: If true, returns the results in reverse order. Type: Boolean.*
clearObjectStyleOverrides	*void* clearObjectStyleOverrides()	*Clear overrides for object style.*
clearTransformations	*void* clearTransformations()	*Clears transformations from the SignatureField. Transformations include rotation, scaling, flipping, fitting, and shearing.*
contentPlace	*variant* contentPlace(Objects PageItems, *boolean* LinkPageItems, *boolean* LinkStories, *boolean* MapStyles, *boolean* ShowingOptions)	*Duplicate an object and place it into the target page item. / Return value: The placed content. Type: Array of PageItems, Graphics, Stories or Texts. / PageItems: One or more page items to place or load. / LinkPageItems: Whether to link pageItems in content placer (if true it will override link stories value). / LinkStories: Whether to link stories in content placer (only applicable for single story, pageItem links will also be created in case of more than one item). / MapStyles: Whether to map styles in content placer. / ShowingOptions: Whether to display the link options dialog.*
convertShape	*void* convertShape(#ConvertShapeOptions Given, *variant* NumberOfSides, *variant* InsetPercentage, *variant* CornerRadius)	*Converts the SignatureField to a different shape. / Given: The SignatureField's new shape. / NumberOfSides: The number of sides for the resulting polygon. (Range: 3 to 100). Type: Long Integer (3 - 100). / InsetPercentage: The star inset percentage for the resulting polygon. (Range: 0.0 to 100.0) . Type: Long Integer (0 - 100). / CornerRadius: The corner radius of the resulting rectangle. Type: Unit (Double or String) (0 - 800 points).*
convertToObject	*void* convertToObject()	*Converts the button object to the page item currently in the active state. Page items from other states will be lost.*
delete	*void* delete()	*Deletes the SignatureField.*
detach	*void* detach()	*Detaches an overridden master page item from the master page.*

SIGNATUREFIELD CS6 VALUES	PROTOTYPE	DESCRIPTION
duplicate	{PageItem} duplicate(*variant* To, *variant* By)	*Duplicates the SignatureField at the specified location or offset. / Return value: The duplicated SignatureField. / To: The location of the new SignatureField, specified in coordinates in the format [x, y]. Type: Array of 2 Units (Doubles or Strings), Spread, Page or Layer. / By: Amount by which to offset the new SignatureField from the original SignatureField's position. Type: Array of 2 Units (Doubles or Strings).*
export	*void* export(*variant* Format, *string* To, *boolean* ShowingOptions, *variant* Using, *variant* VersionComments, *boolean* ForceSave)	*Exports the object(s) to a file. / Format: The export format, specified as an enumeration value or as an extension that appears in the Save as type or Format menu in the Export dialog. Type: idExportFormat enumerator or String. / To: The path to the export file. / ShowingOptions: If true, displays the export options dialog. / Using: The export style. Type: PDFExportPreset. / VersionComments: The comment for this version. Type: String. / ForceSave: If true, forcibly saves a version.*
extractLabel	*string* extractLabel(*string* Key)	*Gets the label value associated with the specified key. / Return value: The label. / Key: The key.*
findObject	{Objects} findObject(*variant* ReverseOrder)	*Finds objects that match the find what value. / Return value: The found object(s). / ReverseOrder: If true, returns the results in reverse order. Type: Boolean.*
fit	*void* fit(#FitOptions Given)	*Applies the specified fit option to content in a frame. / Given: The fit option to use.*
flipItem	*void* flipItem(#Flip Given, *variant* Around)	*Flips the SignatureField. / Given: The axis around which to flip the SignatureField. / Around: The point around which to flip the SignatureField. Type: Array of 2 Units (Doubles or Strings) or idAnchorPoint enumerator.*
insertLabel	*void* insertLabel(*string* Key, *string* Value)	*Sets the label to the value associated with the specified key. / Key: The key. / Value: The value.*
markup	*void* markup(XMLElement Using)	*Associates the page item with the specified XML element while preserving existing content. / Using: The XML element.*
move	*void* move(*variant* To, *variant* By)	*Moves the SignatureField to a new location. Note: Either the 'to' or 'by' parameter is required; if both parameters are defined, only the to value is used. / To: The new location of the SignatureField,in the format (x, y). Type: Array of 2 Units (Doubles or Strings), Spread, Page or Layer. / By: The amount (in measurement units) to move the SignatureField relative to its current position, in the format (x, y). Type: Array of 2 Units (Doubles or Strings).*
override	*variant* override(Page DestinationPage)	*Overrides a master page item and places the item on the document page as a new object. / Return value: The new object. Type: PageItem, Guide, Graphic, Movie or Sound. / DestinationPage: The document page that contains the master page item to override.*
placeXML	*void* placeXML(XMLElement Using)	*Places XML content into the specified object. Note: Replaces any existing content. / Using: The XML element whose content you want to place.*
redefineScaling	*void* redefineScaling(*variant* To)	*Apply an item's scaling to its content if possible. / To: The scale factors to be left on the item. The default is {1.0, 1.0}. Type: Array of 2 Doubles.*
reframe	*void* reframe(*variant* In, *variant* OpposingCorners)	*Move the bounding box of the page item. / In: The bounding box to resize. Type: idCoordinateSpaces enumerator or Ordered array containing CoordinateSpace:idCoordinateSpaces enumerator, BoundsKind:idBoundingBoxLimits enumerator. / OpposingCorners: Opposing corners of new bounding box in the given coordinate space. Type: Array of 2 Arrays of 2 Doubles.*
removeOverride	*void* removeOverride()	*Removes the override from a previously overridden master page item.*

SIGNATUREFIELD CS6 VALUES	PROTOTYPE	DESCRIPTION
resize	*void* resize(*variant* In, *variant* From, #ResizeMethods By, *variant* Values, *boolean* ResizeIndividually, *boolean* ConsideringRulerUnits)	*Resize the page item. / In: The bounding box to resize. Type: idCoordinateSpaces enumerator, idBoundingBoxLimits enumerator or Ordered array containing CoordinateSpace:idCoordinateSpaces enumerator, BoundsKind:idBoundingBoxLimits enumerator. / From: The transform origin. Legal specifications: relative to bounding box: anchor | {anchor | {x,y}, bounds kind [, coordinate space]}; relative to coordinate space: {x,y} | {{x,y} [, coordinate space]}; relative to layout window ruler: {{x,y}, page index | bounds kind}. Type: Array of 2 Doubles, idAnchorPoint enumerator or Array of Arrays of 2 Doubles, idCoordinateSpaces enumerators, idAnchorPoint enumerators, idBoundingBoxLimits enumerators or Long Integers. / By: How the current dimensions are affected by the given values. / Values: The width and height values. Legal dimensions specifications: {x, y [, coordinate space]}, {x, resize constraint [, coordinate space]}, or {resize constraint, y [, coordinate space]}; where x and y are real numbers and coordinate space is used to determine _only_ the unit of length for x and y; coordinate space is ignored for the 'current dimensions times' resize method). Type: Array of Doubles, idResizeConstraints enumerators or idCoordinateSpaces enumerators. / ResizeIndividually: If false and multiple page items are targeted, the new dimensions are attained only by moving the individual items rather than resizing them. / ConsideringRulerUnits: If true then a ruler location is interpreted using ruler units rather than points. The default value is false. This parameter has no effect unless the reference point is specified relative to a page.*
resolve	*variant* resolve(*variant* Location, #CoordinateSpaces In, *boolean* ConsideringRulerUnits)	*Get the coordinates of the given location in the specified coordinate system. / Return value: The resolved location for each of the targeted page items. Type: Array of 2 Doubles or Array of Arrays of 2 Doubles. / Location: The location requested. Type: Array of 2 Doubles, idAnchorPoint enumerator or Array of Arrays of 2 Doubles, idCoordinateSpaces enumerators, idAnchorPoint enumerators, idBoundingBoxLimits enumerators or Long Integers. / In: The coordinate space to use. / ConsideringRulerUnits: If true then a ruler location is interpreted using ruler units rather than points. The default value is false. This parameter has no effect unless the reference point is specified relative to a page.*
select	*void* select(#SelectionOptions ExistingSelection)	*Selects the object. / ExistingSelection: The selection status of the SignatureField in relation to previously selected objects.*
sendBackward	*void* sendBackward()	*Sends the SignatureField back one level in its layer.*
sendToBack	*void* sendToBack(*variant* Reference)	*Sends the SignatureField to the back of its layer or behind a particular item (must have same parent). / Reference: The reference object to send the object behind. Type: PageItem.*
store	{Asset} store(Library Using)	*Stores the object in the specified library. / Return value: The new library asset(s). / Using: The library in which to store the object.*

SIGNATUREFIELD CS6 VALUES	PROTOTYPE	DESCRIPTION
transform	*void* transform(#CoordinateSpaces In, *variant* From, *variant* WithMatrix, *variant* ReplacingCurrent, *boolean* ConsideringRulerUnits)	*Transform the page item. / In: The coordinate space to use. / From: The temporary origin during the transformation. Type: Array of 2 Doubles, idAnchorPoint enumerator or Array of Arrays of 2 Doubles, idCoordinateSpaces enumerators, idAnchorPoint enumerators, idBoundingBoxLimits enumerators or Long Integers. / WithMatrix: Transform matrix. Type: Array of 6 Doubles or TransformationMatrix. / ReplacingCurrent: Transform components to consider; providing this optional parameter causes the target's existing transform components to be replaced with new values. Without this parameter, the given matrix is concatenated onto the target's existing transform combining the effect of the two. Type: idMatrixContent enumerator, Array of idMatrixContent enumerators or Long Integer. / ConsideringRulerUnits: If true then a ruler based origin is interpreted using ruler units rather than points. The default value is false. This parameter has no effect unless the reference point is specified relative to a page.*
transformAgain	*variant* transformAgain()	*Transforms the SignatureField using the last transformation performed on any object. Transformations include moving, rotating, shearing, scaling, and flipping. . / Return value: Playback errors. Type: Array of Strings.*
transformAgainIndividually	*variant* transformAgainIndividually()	*Transforms the SignatureField using the last transformation performed on any SignatureField. Transformations include moving, rotating, shearing, scaling, and flipping. . / Return value: Playback errors. Type: Array of Strings.*
transformSequenceAgain	*variant* transformSequenceAgain()	*Transforms the SignatureField using the last sequence of transform operations performed on any single object or performed at the same time on any group of objects. Transformations include moving, rotating, shearing, scaling, and flipping. / Return value: Playback errors. Type: Array of Strings.*
transformSequenceAgainIndividually	*variant* transformSequenceAgainIndividually()	*Transforms the SignatureField using the last sequence of transformations performed on any single object or performed at the same time on any group of objects. Transformations include moving, rotating, shearing, scaling, and flipping. / Return value: Playback errors. Type: Array of Strings.*
transformValuesOf	{Objects} transformValuesOf(#CoordinateSpaces In)	*Get the transformation values of the page item. / Return value: A list of the transformation matrices. / In: The coordinate space to use.*
absoluteFlip	#Flip absoluteFlip	*Indicates whether the SignatureField has been flipped independently of its parent object and, if yes, the direction in which the SignatureField was flipped.*
absoluteHorizontalScale	*double* absoluteHorizontalScale	*The horizontal scale of the SignatureField relative to its containing object.*
absoluteRotationAngle	*double* absoluteRotationAngle	*The rotation angle of the SignatureField relative to its containing object. (Range: -360 to 360).*
absoluteShearAngle	*double* absoluteShearAngle	*The skewing angle of the SignatureField relative to its containing object. (Range: -360 to 360).*
absoluteVerticalScale	*double* absoluteVerticalScale	*The vertical scale of the SignatureField relative to its containing object.*
activeStateIndex	*integer* activeStateIndex	*The index of the active state in the object's states collection.*
allArticles	{Objects} •allArticles	*The list of all articles this page item is part of.*
allGraphics	{Objects} •allGraphics	*Lists all graphics contained by the SignatureField.*
allowOverrides	*boolean* allowOverrides	*If true, the master page item can be overridden.*
allPageItems	{Objects} •allPageItems	*Lists all page items contained by the SignatureField.*
anchoredObjectSettings	{AnchoredObjectSetting} •anchoredObjectSettings	*Anchored object settings.*
animationSettings	{AnimationSetting} •animationSettings	*The page item animation settings.*

SIGNATUREFIELD CS6 VALUES	PROTOTYPE	DESCRIPTION
appliedObjectStyle	{ObjectStyle} appliedObjectStyle	*The object style applied to the SignatureField.*
associatedXMLElement	{XMLItem} •associatedXMLElement	*The XML element associated with the SignatureField.*
behaviors	{Behaviors} •behaviors	*A collection of behavior objects.*
bottomLeftCornerOption	#CornerOptions bottomLeftCornerOption	*The shape to apply to the bottom left corner of rectangular shapes.*
bottomLeftCornerRadius	*variant* bottomLeftCornerRadius	*The radius in measurement units of the corner effect applied to the bottom left corner of rectangular shapes. Type: Unit (Double or String).*
bottomRightCornerOption	#CornerOptions bottomRightCornerOption	*The shape to apply to the bottom right corner of rectangular shapes.*
bottomRightCornerRadius	*variant* bottomRightCornerRadius	*The radius in measurement units of the corner effect applied to the bottom right corner of rectangular shapes. Type: Unit (Double or String).*
clearFormBehaviors	{ClearFormBehaviors} •clearFormBehaviors	*A collection of clear form behavior objects.*
contentTransparencySettings	{ContentTransparencySetting} •contentTransparencySettings	*Transparency settings for the content of the SignatureField.*
description	*string* description	*The description of the SignatureField.*
endCap	#EndCap endCap	*The end shape of an open path.*
endJoin	#EndJoin endJoin	*The corner join applied to the SignatureField.*
epss	{EPSs} •epss	*A collection of EPS files.*
epsTexts	{EPSTexts} •epsTexts	*EPSTexts.*
fillColor	*variant* fillColor	*The swatch (color, gradient, tint, or mixed ink) applied to the fill of the SignatureField. . Type: Swatch or String.*
fillTint	*double* fillTint	*The percent of tint to use in the SignatureField's fill color. (To specify a tint percent, use a number in the range of 0 to 100; to use the inherited or overridden value, use -1.).*
fillTransparencySettings	{FillTransparencySetting} •fillTransparencySettings	*Transparency settings for the fill applied to the SignatureField.*
flip	#Flip flip	*The direction in which to flip the printed image.*
gapColor	{Swatch} gapColor	*The swatch (color, gradient, tint, or mixed ink) applied to the gap of a dashed, dotted, or striped stroke. For information, see stroke type.*
gapTint	*double* gapTint	*The tint as a percentage of the gap color. (To specify a tint percent, use a number in the range of 0 to 100; to use the inherited or overridden value, use -1.).*
geometricBounds	*variant* geometricBounds	*The bounds of the SignatureField excluding the stroke width, in the format [y1, x1, y2, x2], which give the coordinates of the top-left and bottom-right corners of the bounding box. Type: Array of 4 Units (Doubles or Strings).*
gotoAnchorBehaviors	{GotoAnchorBehaviors} •gotoAnchorBehaviors	*A collection of goto anchor behavior objects.*
gotoFirstPageBehaviors	{GotoFirstPageBehaviors} •gotoFirstPageBehaviors	*A collection of goto first page behavior objects.*
gotoLastPageBehaviors	{GotoLastPageBehaviors} •gotoLastPageBehaviors	*A collection of goto last page behavior objects.*
gotoNextPageBehaviors	{GotoNextPageBehaviors} •gotoNextPageBehaviors	*A collection of goto next page behavior objects.*
gotoNextViewBehaviors	{GotoNextViewBehaviors} •gotoNextViewBehaviors	*A collection of goto next view behavior objects.*
gotoPreviousPageBehaviors	{GotoPreviousPageBehaviors} •gotoPreviousPageBehaviors	*A collection of goto previous page behavior objects.*
gotoPreviousViewBehaviors	{GotoPreviousViewBehaviors} •gotoPreviousViewBehaviors	*A collection of goto previous view behavior objects.*
gotoURLBehaviors	{GotoURLBehaviors} •gotoURLBehaviors	*A collection of goto URL behavior objects.*

SIGNATUREFIELD CS6 VALUES	PROTOTYPE	DESCRIPTION
gradientFillAngle	*double* gradientFillAngle	*The angle of a linear gradient applied to the fill of the SignatureField. (Range: -180 to 180).*
gradientFillLength	*variant* gradientFillLength	*The length (for a linear gradient) or radius (for a radial gradient) applied to the fill of the SignatureField. Type: Unit (Double or String).*
gradientFillStart	*variant* gradientFillStart	*The starting point (in page coordinates) of a gradient applied to the fill of the SignatureField, in the format [x, y]. Type: Array of 2 Units (Doubles or Strings).*
gradientStrokeAngle	*double* gradientStrokeAngle	*The angle of a linear gradient applied to the stroke of the SignatureField. (Range: -180 to 180).*
gradientStrokeLength	*variant* gradientStrokeLength	*The length (for a linear gradient) or radius (for a radial gradient) applied to the stroke of the SignatureField. Type: Unit (Double or String).*
gradientStrokeStart	*variant* gradientStrokeStart	*The starting point (in page coordinates) of a gradient applied to the stroke of the SignatureField, in the format [x, y]. Type: Array of 2 Units (Doubles or Strings).*
graphicLines	{GraphicLines} •graphicLines	*A collection of graphic lines.*
graphics	{Graphics} •graphics	*A collection of imported graphics in any graphic file format (vector, metafile, or bitmap).*
groups	{Groups} •groups	*A collection of groups.*
hiddenUntilTriggered	*boolean* hiddenUntilTriggered	*If true, the form field/push button is hidden until triggered in the exported PDF.*
horizontalLayoutConstraints	*variant* horizontalLayoutConstraints	*The left margin, width, and right margin constraints this item is subject to when using the object-based layout rule. Type: Array of 3 idDimensionsConstraints enumerators.*
horizontalScale	*double* horizontalScale	*The horizontal scaling applied to the SignatureField.*
id	*integer* •id	*The unique ID of the SignatureField.*
images	{Images} •images	*A collection of bitmap images in any bitmap file format (including TIFF, JPEG, or GIF).*
index	*integer* •index	*The index of the SignatureField within its containing object.*
itemLayer	{Layer} itemLayer	*The layer that the SignatureField is on.*
label	*string* label	*A property that can be set to any string.*
leftLineEnd	#ArrowHead leftLineEnd	*The arrowhead applied to the start of the path.*
linkedPageItemOptions	{LinkedPageItemOption} •linkedPageItemOptions	*Linked Page Item options.*
localDisplaySetting	#DisplaySettingOptions localDisplaySetting	*Display performance options for the SignatureField.*
locked	*boolean* locked	*If true, the SignatureField is locked.*
miterLimit	*double* miterLimit	*The limit of the ratio of stroke width to miter length before a miter (pointed) join becomes a bevel (squared-off) join. Type: Double (1 - 500).*
movieBehaviors	{MovieBehaviors} •movieBehaviors	*A collection of movie behavior objects.*
name	*string* name	*The name of the SignatureField.*
nonprinting	*boolean* nonprinting	*If true, the SignatureField does not print.*
openFileBehaviors	{OpenFileBehaviors} •openFileBehaviors	*A collection of open file behavior objects.*
ovals	{Ovals} •ovals	*A collection of ellipses.*

SIGNATUREFIELD CS6 VALUES	PROTOTYPE	DESCRIPTION
overprintFill	*boolean* overprintFill	*If true, the SignatureField's fill color overprints any underlying objects. If false, the fill color knocks out the underlying colors.*
overprintGap	*boolean* overprintGap	*If true, the gap color overprints any underlying colors. If false, the gap color knocks out the underlying colors.*
overprintStroke	*boolean* overprintStroke	*If true, the SignatureField's stroke color overprints any underlying objects. If false, the stroke color knocks out the underlying colors.*
overridden	*boolean* •overridden	*If true, the object originated on a master spread and was overridden. If false, the object either originated on a master spread and was not overridden, or the object did not originate on a master page.*
overriddenMasterPageItem	*variant* •overriddenMasterPageItem	*An object that originated on a master page and has been overridden. Type: PageItem, Guide, Graphic, Movie or Sound.*
pageItems	{PageItems} •pageItems	*The page items collection, which can be used to process all page items in a container (such as a document, page, or group), regardless of type.*
parent	*variant* •parent	*The parent of the SignatureField (a Snippet, PlaceGun, Character, Spread, MasterSpread, SplineItem, Polygon, GraphicLine, Rectangle, Oval or Group). Type: Snippet, PlaceGun, Character, Spread, MasterSpread, SplineItem, Polygon, GraphicLine, Rectangle, Oval or Group.*
parentPage	{Page} •parentPage	*The page on which this page item appears.*
pdfs	{PDFs} •pdfs	*A collection of PDF files.*
picts	{PICTs} •picts	*A collection of PICT graphics.*
polygons	{Polygons} •polygons	*A collection of polygons.*
preferences	{Preferences} •preferences	*A collection of preferences objects.*
printableInPdf	*boolean* printableInPdf	*If true, the form field/push button is printable in the exported PDF.*
printFormBehaviors	{PrintFormBehaviors} •printFormBehaviors	*A collection of print form behavior objects.*
readOnly	*boolean* readOnly	*If true, the form field is read only in the exported PDF.*
rectangles	{Rectangles} •rectangles	*A collection of rectangles.*
required	*boolean* required	*If true, the form field is required in the exported PDF.*
rightLineEnd	#ArrowHead rightLineEnd	*The arrowhead applied to the end of the path.*
rotationAngle	*double* rotationAngle	*The rotatation angle of the SignatureField. (Range: -360 to 360).*
shearAngle	*double* shearAngle	*The skewing angle applied to the SignatureField. (Range: -360 to 360).*
showHideFieldsBehaviors	{ShowHideFieldsBehaviors} •showHideFieldsBehaviors	*A collection of show/hide fields behavior objects.*
soundBehaviors	{SoundBehaviors} •soundBehaviors	*A collection of sound behavior objects.*
splineItems	{SplineItems} •splineItems	*The spline items collection.*
strokeAlignment	#StrokeAlignment strokeAlignment	*The stroke alignment applied to the SignatureField.*
strokeColor	*variant* strokeColor	*The swatch (color, gradient, tint, or mixed ink) applied to the stroke of the SignatureField. Type: Swatch or String.*
strokeCornerAdjustment	#StrokeCornerAdjustment strokeCornerAdjustment	*The corner adjustment applied to the SignatureField.*
strokeDashAndGap	*variant* strokeDashAndGap	*The dash and gap measurements that define the pattern of a custom dashed line. Define up to six values (in points) in the format [dash1, gap1, dash2, gap2, dash3, gap3]. Type: Array of Units (Doubles or Strings).*

SIGNATUREFIELD CS6 VALUES	PROTOTYPE	DESCRIPTION
strokeTint	*double* strokeTint	*The percent of tint to use in object’s stroke color. (To specify a tint percent, use a number in the range of 0 to 100; to use the inherited or overridden value, use -1.).*
strokeTransparencySettings	{StrokeTransparencySetting} •strokeTransparencySettings	*Transparency settings for the stroke.*
strokeType	*variant* strokeType	*The name of the stroke style to apply. Type: StrokeStyle or String.*
strokeWeight	*variant* strokeWeight	*The weight (in points) to apply to the SignatureField’s stroke. Type: Unit (Double or String).*
submitFormBehaviors	{SubmitFormBehaviors} •submitFormBehaviors	*A collection of submit form behavior objects.*
textFrames	{TextFrames} •textFrames	*A collection of text frames.*
textWrapPreferences	{TextWrapPreference} •textWrapPreferences	*The text wrap preference properties that define the default formatting for wrapping text around objects.*
timingSettings	{TimingSetting} •timingSettings	*The object timing settings.*
topLeftCornerOption	#CornerOptions topLeftCornerOption	*The shape to be applied to the top left corner of rectangular shapes and all corners of non-rectangular shapes.Note: corner option differs from end join in which you can set a radius for a corner option, whereas the rounded or beveled effect of an end join depends on the stroke weight.*
topLeftCornerRadius	*variant* topLeftCornerRadius	*The radius in measurement units of the corner effect applied to the top left corner of rectangular shapes and all corners of non-rectangular shapes. Type: Unit (Double or String).*
topRightCornerOption	#CornerOptions topRightCornerOption	*The shape to apply to the top right corner of rectangular shapes.*
topRightCornerRadius	*variant* topRightCornerRadius	*The radius in measurement units of the corner effect applied to the top right corner of rectangular shapes. Type: Unit (Double or String).*
transparencySettings	{TransparencySetting} •transparencySettings	*Transparency settings.*
verticalLayoutConstraints	*variant* verticalLayoutConstraints	*The top margin, height, and bottom margin constraints this item is subject to when using the object-based layout rule. Type: Array of 3 idDimensionsConstraints enumerators.*
verticalScale	*double* verticalScale	*The vertical scaling applied to the SignatureField.*
viewZoomBehaviors	{ViewZoomBehaviors} •viewZoomBehaviors	*A collection of view zoom behavior objects.*
visible	*boolean* visible	*If true, the SignatureField is visible.*
visibleBounds	*variant* visibleBounds	*The bounds of the SignatureField including the stroke width, in the format [y1, x1, y2, x2], which give the coordinates of the top-left and bottom-right corners of the bounding box. Type: Array of 4 Units (Doubles or Strings).*
wmfs	{WMFs} •wmfs	*A collection of WMF graphics.*

SignatureFieldEvents

Object available in ID 8

SIGNATUREFIELDEVENTS CS6 VALUES	PROTOTYPE	DESCRIPTION
afterPlace	*const* afterPlace(afterPlace)	*Dispatched after a SignatureField is placed. This event bubbles. This event is not cancelable.*

SIGNATUREFIELDEVENTS CS6 VALUES	PROTOTYPE	DESCRIPTION
beforePlace	*const* beforePlace(beforePlace)	*Dispatched before a SignatureField is placed. This event bubbles. This event is cancelable.*

SignatureFields

Object available in ID 8

SIGNATUREFIELDS CS6 VALUES	PROTOTYPE	DESCRIPTION
add	{SignatureField} add(*variant* Layer, #LocationOptions At, *variant* Reference)	*Creates a new SignatureField. / Return value: The new SignatureField. / Layer: The layer on which to create the SignatureField. Type: Layer. / At: The location at which to insert the SignatureField relative to the reference object or within the container object. / Reference: The reference object. Note: Required when the at parameter is before or after. Type: Document, Spread, MasterSpread, Page, Layer or PageItem.*
anyItem	IDispatch anyItem()	*Returns any object in the collection. / Return value: The object.*
firstItem	IDispatch firstItem()	*Returns the first object in the collection. / Return value: The first object.*
itemByID	IDispatch itemByID(*integer* Id)	*Returns the object with the specified ID. / Return value: The object. / ID: The ID.*
itemByRange	{Objects} itemByRange(*variant* From, *variant* To)	*Returns the objects within the specified range. / Return value: The objects. / From: The object, index, or name at the beginning of the range. Type: Object, Long Integer or String. / To: The object, index, or name at the end of the range. Type: Object, Long Integer or String.*
lastItem	IDispatch lastItem()	*Returns the last object in the collection. / Return value: The last object.*
middleItem	IDispatch middleItem()	*Returns the middle object in the collection. / Return value: The middle object.*
nextItem	IDispatch nextItem(IDispatch Obj)	*Returns the object whose index follows the specified object in the collection. / Return value: The next object. / Obj: The object whose index comes before the desired object.*
previousItem	IDispatch previousItem(IDispatch Obj)	*Returns the object with the index previous to the specified index. / Return value: The previous object. / Obj: The index of the object that follows the desired object.*
count	*integer* •count	*Displays the number of elements in the SignatureField.*
item	{SignatureField} •item(*variant* Index)	*Returns the object with the specified index or name. Type: Long Integer or String.*

SubmitFormBehavior

Object available in ID 8

SUBMITFORMBEHAVIOR CS6 VALUES	PROTOTYPE	DESCRIPTION
delete	*void* delete()	*Deletes the SubmitFormBehavior.*
extractLabel	*string* extractLabel(*string* Key)	*Gets the label value associated with the specified key. / Return value: The label. / Key: The key.*

SUBMITFORMBEHAVIOR CS6 VALUES	PROTOTYPE	DESCRIPTION
insertLabel	*void* insertLabel(*string* Key, *string* Value)	*Sets the label to the value associated with the specified key. / Key: The key. / Value: The value.*
behaviorEvent	#BehaviorEvents behaviorEvent	*The event which triggers the behavior.*
enableBehavior	*boolean* enableBehavior	*If true, the behavior is enabled.*
id	*integer* •id	*The unique ID of the SubmitFormBehavior.*
index	*integer* •index	*The index of the SubmitFormBehavior within its containing object.*
label	*string* label	*A property that can be set to any string.*
name	*string* •name	*The name of the SubmitFormBehavior.*
parent	*variant* •parent	*The parent of the SubmitFormBehavior (a Button, CheckBox, ComboBox, ListBox, RadioButton, TextBox or SignatureField). Type: Button, CheckBox, ComboBox, ListBox, RadioButton, TextBox or SignatureField.*
url	*string* url	*The URL.*

SubmitFormBehaviors

Object available in ID 8

SUBMITFORMBEHAVIORS CS6 VALUES	PROTOTYPE	DESCRIPTION
add	{SubmitFormBehavior} add()	*Creates a new SubmitFormBehavior. / Return value: The new SubmitFormBehavior.*
anyItem	IDispatch anyItem()	*Returns any object in the collection. / Return value: The object.*
firstItem	IDispatch firstItem()	*Returns the first object in the collection. / Return value: The first object.*
itemByID	IDispatch itemByID(*integer* Id)	*Returns the object with the specified ID. / Return value: The object. / ID: The ID.*
itemByRange	{Objects} itemByRange(*variant* From, *variant* To)	*Returns the objects within the specified range. / Return value: The objects. / From: The object, index, or name at the beginning of the range. Type: Object, Long Integer or String. / To: The object, index, or name at the end of the range. Type: Object, Long Integer or String.*
lastItem	IDispatch lastItem()	*Returns the last object in the collection. / Return value: The last object.*
middleItem	IDispatch middleItem()	*Returns the middle object in the collection. / Return value: The middle object.*
nextItem	IDispatch nextItem(IDispatch Obj)	*Returns the object whose index follows the specified object in the collection. / Return value: The next object. / Obj: The object whose index comes before the desired object.*
previousItem	IDispatch previousItem(IDispatch Obj)	*Returns the object with the index previous to the specified index. / Return value: The previous object. / Obj: The index of the object that follows the desired object.*
count	*integer* •count	*Displays the number of elements in the SubmitFormBehavior.*
item	{SubmitFormBehavior} •item(*variant* Index)	*Returns the object with the specified index or name. Type: Long Integer or String.*

TableStyleMapping

Object available in ID 8

TABLESTYLEMAPPING CS6 VALUES	PROTOTYPE	DESCRIPTION
delete	*void* delete()	*deletes a style mapping.*
destinationStyleName	*string* destinationStyleName	*The destination style name property.*
index	*integer* •index	*The index of the TableStyleMapping within its containing object.*
mappingRuleType	#MapType mappingRuleType	*The type of the mapping rule.*
parent	*variant* •parent	*The parent of the TableStyleMapping (a Application, Document, Story or XmlStory). Type: Application, Document, Story or XmlStory.*
sourceStyleName	*string* sourceStyleName	*The source style name property.*

TableStyleMappings

Object available in ID 8

TABLESTYLEMAPPINGS CS6 VALUES	PROTOTYPE	DESCRIPTION
add	{TableStyleMapping} add(*string* SourceStyleName, *string* DestinationStyleName, #MapType MappingRuleType)	*Adds a style mapping. / Return value: The added TableStyleMapping. / SourceStyleName: The source style name. / DestinationStyleName: The destination style name. / MappingRuleType: The mapping type.*
anyItem	IDispatch anyItem()	*Returns any object in the collection. / Return value: The object.*
firstItem	IDispatch firstItem()	*Returns the first object in the collection. / Return value: The first object.*
itemByRange	{Objects} itemByRange(*variant* From, *variant* To)	*Returns the objects within the specified range. / Return value: The objects. / From: The object, index, or name at the beginning of the range. Type: Object, Long Integer or String. / To: The object, index, or name at the end of the range. Type: Object, Long Integer or String.*
lastItem	IDispatch lastItem()	*Returns the last object in the collection. / Return value: The last object.*
middleItem	IDispatch middleItem()	*Returns the middle object in the collection. / Return value: The middle object.*
nextItem	IDispatch nextItem(IDispatch Obj)	*Returns the object whose index follows the specified object in the collection. / Return value: The next object. / Obj: The object whose index comes before the desired object.*
previousItem	IDispatch previousItem(IDispatch Obj)	*Returns the object with the index previous to the specified index. / Return value: The previous object. / Obj: The index of the object that follows the desired object.*
count	*integer* •count	*Displays the number of elements in the TableStyleMapping.*
item	{TableStyleMapping} •item(*variant* Index)	*Returns the object with the specified index or name. Type: Long Integer or String.*

TextBox

Object available in ID 8

TEXTBOX CS6 VALUES	PROTOTYPE	DESCRIPTION
applyObjectStyle	*void* applyObjectStyle(ObjectStyle Using, *boolean* ClearingOverrides, *boolean* ClearingOverridesThroughRootObjectStyle)	*Applies the specified object style. / Using: The object style to apply. / ClearingOverrides: If true, clears the TextBox's existing attributes before applying the style. / ClearingOverridesThroughRootObjectStyle: If true, clears attributes and formatting applied to the TextBox that are not defined in the object style.*
asynchronousExportFile	{BackgroundTask} asynchronousExportFile(*variant* Format, *string* To, *boolean* ShowingOptions, *variant* Using, *variant* VersionComments, *boolean* ForceSave)	*asynchronously exports the object(s) to a file. / Return value: object representing the asynchronous export. / Format: The export format, specified as an enumeration value or as an extension that appears in the Save as type or Format menu in the Export dialog. Type: idExportFormat enumerator or String. / To: The path to the export file. / ShowingOptions: If true, displays the export options dialog. / Using: The export style. Type: PDFExportPreset. / VersionComments: The comment for this version. Type: String. / ForceSave: If true, forcibly saves a version.*
autoTag	*void* autoTag()	*Tag the object or the parent story using default tags defined in XML preference.*
bringForward	*void* bringForward()	*Brings the TextBox forward one level in its layer.*
bringToFront	*void* bringToFront(*variant* Reference)	*Brings the TextBox to the front of its layer or in front of a particular item. / Reference: The reference object to bring the object in front of (must have same parent). Type: PageItem.*
changeObject	{Objects} changeObject(*variant* ReverseOrder)	*Finds objects that match the find what value and replace the objects with the change to value. / Return value: The replaced object(s). / ReverseOrder: If true, returns the results in reverse order. Type: Boolean.*
clearObjectStyleOverrides	*void* clearObjectStyleOverrides()	*Clear overrides for object style.*
clearTransformations	*void* clearTransformations()	*Clears transformations from the TextBox. Transformations include rotation, scaling, flipping, fitting, and shearing.*
contentPlace	*variant* contentPlace(Objects PageItems, *boolean* LinkPageItems, *boolean* LinkStories, *boolean* MapStyles, *boolean* ShowingOptions)	*Duplicate an object and place it into the target page item. / Return value: The placed content. Type: Array of PageItems, Graphics, Stories or Texts. / PageItems: One or more page items to place or load. / LinkPageItems: Whether to link pageItems in content placer (if true it will override link stories value). / LinkStories: Whether to link stories in content placer (only applicable for single story, pageItem links will also be created in case of more than one item). / MapStyles: Whether to map styles in content placer. / ShowingOptions: Whether to display the link options dialog.*
convertShape	*void* convertShape(#ConvertShapeOptions Given, *variant* NumberOfSides, *variant* InsetPercentage, *variant* CornerRadius)	*Converts the TextBox to a different shape. / Given: The TextBox's new shape. / NumberOfSides: The number of sides for the resulting polygon. (Range: 3 to 100). Type: Long Integer (3 - 100). / InsetPercentage: The star inset percentage for the resulting polygon. (Range: 0.0 to 100.0) . Type: Long Integer (0 - 100). / CornerRadius: The corner radius of the resulting rectangle. Type: Unit (Double or String) (0 - 800 points).*
convertToObject	*void* convertToObject()	*Converts the button object to the page item currently in the active state. Page items from other states will be lost.*
delete	*void* delete()	*Deletes the TextBox.*
detach	*void* detach()	*Detaches an overridden master page item from the master page.*

TEXTBOX CS6 VALUES	PROTOTYPE	DESCRIPTION
duplicate	{PageItem} duplicate(variant To, variant By)	Duplicates the TextBox at the specified location or offset. / Return value: The duplicated TextBox. / To: The location of the new TextBox, specified in coordinates in the format [x, y]. Type: Array of 2 Units (Doubles or Strings), Spread, Page or Layer. / By: Amount by which to offset the new TextBox from the original TextBox's position. Type: Array of 2 Units (Doubles or Strings).
export	void export(variant Format, string To, boolean ShowingOptions, variant Using, variant VersionComments, boolean ForceSave)	Exports the object(s) to a file. / Format: The export format, specified as an enumeration value or as an extension that appears in the Save as type or Format menu in the Export dialog. Type: idExportFormat enumerator or String. / To: The path to the export file. / ShowingOptions: If true, displays the export options dialog. / Using: The export style. Type: PDFExportPreset. / VersionComments: The comment for this version. Type: String. / ForceSave: If true, forcibly saves a version.
extractLabel	string extractLabel(string Key)	Gets the label value associated with the specified key. / Return value: The label. / Key: The key.
findObject	{Objects} findObject(variant ReverseOrder)	Finds objects that match the find what value. / Return value: The found object(s). / ReverseOrder: If true, returns the results in reverse order. Type: Boolean.
fit	void fit(#FitOptions Given)	Applies the specified fit option to content in a frame. / Given: The fit option to use.
flipItem	void flipItem(#Flip Given, variant Around)	Flips the TextBox. / Given: The axis around which to flip the TextBox. / Around: The point around which to flip the TextBox. Type: Array of 2 Units (Doubles or Strings) or idAnchorPoint enumerator.
insertLabel	void insertLabel(string Key, string Value)	Sets the label to the value associated with the specified key. / Key: The key. / Value: The value.
markup	void markup(XMLElement Using)	Associates the page item with the specified XML element while preserving existing content. / Using: The XML element.
move	void move(variant To, variant By)	Moves the TextBox to a new location. Note: Either the 'to' or 'by' parameter is required; if both parameters are defined, only the to value is used. / To: The new location of the TextBox,in the format (x, y). Type: Array of 2 Units (Doubles or Strings), Spread, Page or Layer. / By: The amount (in measurement units) to move the TextBox relative to its current position, in the format (x, y). Type: Array of 2 Units (Doubles or Strings).
override	variant override(Page DestinationPage)	Overrides a master page item and places the item on the document page as a new object. / Return value: The new object. Type: PageItem, Guide, Graphic, Movie or Sound. / DestinationPage: The document page that contains the master page item to override.
placeXML	void placeXML(XMLElement Using)	Places XML content into the specified object. Note: Replaces any existing content. / Using: The XML element whose content you want to place.
redefineScaling	void redefineScaling(variant To)	Apply an item's scaling to its content if possible. / To: The scale factors to be left on the item. The default is {1.0, 1.0}. Type: Array of 2 Doubles.
reframe	void reframe(variant In, variant OpposingCorners)	Move the bounding box of the page item. / In: The bounding box to resize. Type: idCoordinateSpaces enumerator or Ordered array containing CoordinateSpace:idCoordinateSpaces enumerator, BoundsKind:idBoundingBoxLimits enumerator. / OpposingCorners: Opposing corners of new bounding box in the given coordinate space. Type: Array of 2 Arrays of 2 Doubles.
removeOverride	void removeOverride()	Removes the override from a previously overridden master page item.

TEXTBOX CS6 VALUES	PROTOTYPE	DESCRIPTION
resize	*void* resize(*variant* In, *variant* From, #ResizeMethods By, *variant* Values, *boolean* ResizeIndividually, *boolean* ConsideringRulerUnits)	*Resize the page item. / In: The bounding box to resize. Type: idCoordinateSpaces enumerator, idBoundingBoxLimits enumerator or Ordered array containing CoordinateSpace:idCoordinateSpaces enumerator, BoundsKind:idBoundingBoxLimits enumerator. / From: The transform origin. Legal specifications: relative to bounding box: anchor | {anchor | {x,y}, bounds kind [, coordinate space]}; relative to coordinate space: {x,y} | {{x,y} [, coordinate space]}; relative to layout window ruler: {{x,y}, page index | bounds kind}. Type: Array of 2 Doubles, idAnchorPoint enumerator or Array of Arrays of 2 Doubles, idCoordinateSpaces enumerators, idAnchorPoint enumerators, idBoundingBoxLimits enumerators or Long Integers. / By: How the current dimensions are affected by the given values. / Values: The width and height values. Legal dimensions specifications: {x, y [, coordinate space]}, {x, resize constraint [, coordinate space]}, or {resize constraint, y [, coordinate space]}; where x and y are real numbers and coordinate space is used to determine _only_ the unit of length for x and y; coordinate space is ignored for the 'current dimensions times' resize method). Type: Array of Doubles, idResizeConstraints enumerators or idCoordinateSpaces enumerators. / ResizeIndividually: If false and multiple page items are targeted, the new dimensions are attained only by moving the individual items rather than resizing them. / ConsideringRulerUnits: If true then a ruler location is interpreted using ruler units rather than points. The default value is false. This parameter has no effect unless the reference point is specified relative to a page.*
resolve	*variant* resolve(*variant* Location, #CoordinateSpaces In, *boolean* ConsideringRulerUnits)	*Get the coordinates of the given location in the specified coordinate system. / Return value: The resolved location for each of the targeted page items. Type: Array of 2 Doubles or Array of Arrays of 2 Doubles. / Location: The location requested. Type: Array of 2 Doubles, idAnchorPoint enumerator or Array of Arrays of 2 Doubles, idCoordinateSpaces enumerators, idAnchorPoint enumerators, idBoundingBoxLimits enumerators or Long Integers. / In: The coordinate space to use. / ConsideringRulerUnits: If true then a ruler location is interpreted using ruler units rather than points. The default value is false. This parameter has no effect unless the reference point is specified relative to a page.*
select	*void* select(#SelectionOptions ExistingSelection)	*Selects the object. / ExistingSelection: The selection status of the TextBox in relation to previously selected objects.*
sendBackward	*void* sendBackward()	*Sends the TextBox back one level in its layer.*
sendToBack	*void* sendToBack(*variant* Reference)	*Sends the TextBox to the back of its layer or behind a particular item (must have same parent). / Reference: The reference object to send the object behind. Type: PageItem.*
store	{Asset} store(Library Using)	*Stores the object in the specified library. / Return value: The new library asset(s). / Using: The library in which to store the object.*

TEXTBOX CS6 VALUES	PROTOTYPE	DESCRIPTION
transform	*void* transform(#CoordinateSpaces In, *variant* From, *variant* WithMatrix, *variant* ReplacingCurrent, *boolean* ConsideringRulerUnits)	*Transform the page item. / In: The coordinate space to use. / From: The temporary origin during the transformation. Type: Array of 2 Doubles, idAnchorPoint enumerator or Array of Arrays of 2 Doubles, idCoordinateSpaces enumerators, idAnchorPoint enumerators, idBoundingBoxLimits enumerators or Long Integers. / WithMatrix: Transform matrix. Type: Array of 6 Doubles or TransformationMatrix. / ReplacingCurrent: Transform components to consider; providing this optional parameter causes the target's existing transform components to be replaced with new values. Without this parameter, the given matrix is concatenated onto the target's existing transform combining the effect of the two. Type: idMatrixContent enumerator, Array of idMatrixContent enumerators or Long Integer. / ConsideringRulerUnits: If true then a ruler based origin is interpreted using ruler units rather than points. The default value is false. This parameter has no effect unless the reference point is specified relative to a page.*
transformAgain	*variant* transformAgain()	*Transforms the TextBox using the last transformation performed on any object. Transformations include moving, rotating, shearing, scaling, and flipping. . / Return value: Playback errors. Type: Array of Strings.*
transformAgainIndividually	*variant* transformAgainIndividually()	*Transforms the TextBox using the last transformation performed on any TextBox. Transformations include moving, rotating, shearing, scaling, and flipping. . / Return value: Playback errors. Type: Array of Strings.*
transformSequenceAgain	*variant* transformSequenceAgain()	*Transforms the TextBox using the last sequence of transform operations performed on any single object or performed at the same time on any group of objects. Transformations include moving, rotating, shearing, scaling, and flipping. / Return value: Playback errors. Type: Array of Strings.*
transformSequenceAgainIndividually	*variant* transformSequenceAgainIndividually()	*Transforms the TextBox using the last sequence of transformations performed on any single object or performed at the same time on any group of objects. Transformations include moving, rotating, shearing, scaling, and flipping. / Return value: Playback errors. Type: Array of Strings.*
transformValuesOf	{Objects} transformValuesOf(#CoordinateSpaces In)	*Get the transformation values of the page item. / Return value: A list of the transformation matrices. / In: The coordinate space to use.*
absoluteFlip	#Flip absoluteFlip	*Indicates whether the TextBox has been flipped independently of its parent object and, if yes, the direction in which the TextBox was flipped.*
absoluteHorizontalScale	*double* absoluteHorizontalScale	*The horizontal scale of the TextBox relative to its containing object.*
absoluteRotationAngle	*double* absoluteRotationAngle	*The rotation angle of the TextBox relative to its containing object. (Range: -360 to 360).*
absoluteShearAngle	*double* absoluteShearAngle	*The skewing angle of the TextBox relative to its containing object. (Range: -360 to 360).*
absoluteVerticalScale	*double* absoluteVerticalScale	*The vertical scale of the TextBox relative to its containing object.*
activeStateIndex	*integer* activeStateIndex	*The index of the active state in the object's states collection.*
allArticles	{Objects} •allArticles	*The list of all articles this page item is part of.*
allGraphics	{Objects} •allGraphics	*Lists all graphics contained by the TextBox.*
allowOverrides	*boolean* allowOverrides	*If true, the master page item can be overridden.*
allPageItems	{Objects} •allPageItems	*Lists all page items contained by the TextBox.*
anchoredObjectSettings	{AnchoredObjectSetting} •anchoredObjectSettings	*Anchored object settings.*
animationSettings	{AnimationSetting} •animationSettings	*The page item animation settings.*
appliedObjectStyle	{ObjectStyle} appliedObjectStyle	*The object style applied to the TextBox.*

TEXTBOX CS6 VALUES	PROTOTYPE	DESCRIPTION
associatedXMLElement	{XMLItem} •associatedXMLElement	*The XML element associated with the TextBox.*
behaviors	{Behaviors} •behaviors	*A collection of behavior objects.*
bottomLeftCornerOption	#CornerOptions bottomLeftCornerOption	*The shape to apply to the bottom left corner of rectangular shapes.*
bottomLeftCornerRadius	*variant* bottomLeftCornerRadius	*The radius in measurement units of the corner effect applied to the bottom left corner of rectangular shapes. Type: Unit (Double or String).*
bottomRightCornerOption	#CornerOptions bottomRightCornerOption	*The shape to apply to the bottom right corner of rectangular shapes.*
bottomRightCornerRadius	*variant* bottomRightCornerRadius	*The radius in measurement units of the corner effect applied to the bottom right corner of rectangular shapes. Type: Unit (Double or String).*
clearFormBehaviors	{ClearFormBehaviors} •clearFormBehaviors	*A collection of clear form behavior objects.*
contentTransparencySettings	{ContentTransparencySetting} •contentTransparencySettings	*Transparency settings for the content of the TextBox.*
description	*string* description	*The description of the TextBox.*
endCap	#EndCap endCap	*The end shape of an open path.*
endJoin	#EndJoin endJoin	*The corner join applied to the TextBox.*
epss	{EPSs} •epss	*A collection of EPS files.*
epsTexts	{EPSTexts} •epsTexts	*EPSTexts.*
fillColor	*variant* fillColor	*The swatch (color, gradient, tint, or mixed ink) applied to the fill of the TextBox. . Type: Swatch or String.*
fillTint	*double* fillTint	*The percent of tint to use in the TextBox's fill color. (To specify a tint percent, use a number in the range of 0 to 100; to use the inherited or overridden value, use -1.).*
fillTransparencySettings	{FillTransparencySetting} •fillTransparencySettings	*Transparency settings for the fill applied to the TextBox.*
flip	#Flip flip	*The direction in which to flip the printed image.*
fontSize	*double* fontSize	*The font size for the form field in the exported PDF.*
gapColor	{Swatch} gapColor	*The swatch (color, gradient, tint, or mixed ink) applied to the gap of a dashed, dotted, or striped stroke. For information, see stroke type.*
gapTint	*double* gapTint	*The tint as a percentage of the gap color. (To specify a tint percent, use a number in the range of 0 to 100; to use the inherited or overridden value, use -1.).*
geometricBounds	*variant* geometricBounds	*The bounds of the TextBox excluding the stroke width, in the format [y1, x1, y2, x2], which give the coordinates of the top-left and bottom-right corners of the bounding box. Type: Array of 4 Units (Doubles or Strings).*
gotoAnchorBehaviors	{GotoAnchorBehaviors} •gotoAnchorBehaviors	*A collection of goto anchor behavior objects.*
gotoFirstPageBehaviors	{GotoFirstPageBehaviors} •gotoFirstPageBehaviors	*A collection of goto first page behavior objects.*
gotoLastPageBehaviors	{GotoLastPageBehaviors} •gotoLastPageBehaviors	*A collection of goto last page behavior objects.*
gotoNextPageBehaviors	{GotoNextPageBehaviors} •gotoNextPageBehaviors	*A collection of goto next page behavior objects.*
gotoNextViewBehaviors	{GotoNextViewBehaviors} •gotoNextViewBehaviors	*A collection of goto next view behavior objects.*
gotoPreviousPageBehaviors	{GotoPreviousPageBehaviors} •gotoPreviousPageBehaviors	*A collection of goto previous page behavior objects.*
gotoPreviousViewBehaviors	{GotoPreviousViewBehaviors} •gotoPreviousViewBehaviors	*A collection of goto previous view behavior objects.*
gotoURLBehaviors	{GotoURLBehaviors} •gotoURLBehaviors	*A collection of goto URL behavior objects.*

TEXTBOX CS6 VALUES	PROTOTYPE	DESCRIPTION
gradientFillAngle	*double* gradientFillAngle	*The angle of a linear gradient applied to the fill of the TextBox. (Range: -180 to 180).*
gradientFillLength	*variant* gradientFillLength	*The length (for a linear gradient) or radius (for a radial gradient) applied to the fill of the TextBox. Type: Unit (Double or String).*
gradientFillStart	*variant* gradientFillStart	*The starting point (in page coordinates) of a gradient applied to the fill of the TextBox, in the format [x, y]. Type: Array of 2 Units (Doubles or Strings).*
gradientStrokeAngle	*double* gradientStrokeAngle	*The angle of a linear gradient applied to the stroke of the TextBox. (Range: -180 to 180).*
gradientStrokeLength	*variant* gradientStrokeLength	*The length (for a linear gradient) or radius (for a radial gradient) applied to the stroke of the TextBox. Type: Unit (Double or String).*
gradientStrokeStart	*variant* gradientStrokeStart	*The starting point (in page coordinates) of a gradient applied to the stroke of the TextBox, in the format [x, y]. Type: Array of 2 Units (Doubles or Strings).*
graphicLines	{GraphicLines} •graphicLines	*A collection of graphic lines.*
graphics	{Graphics} •graphics	*A collection of imported graphics in any graphic file format (vector, metafile, or bitmap).*
groups	{Groups} •groups	*A collection of groups.*
hiddenUntilTriggered	*boolean* hiddenUntilTriggered	*If true, the form field/push button is hidden until triggered in the exported PDF.*
horizontalLayoutConstraints	*variant* horizontalLayoutConstraints	*The left margin, width, and right margin constraints this item is subject to when using the object-based layout rule. Type: Array of 3 idDimensionsConstraints enumerators.*
horizontalScale	*double* horizontalScale	*The horizontal scaling applied to the TextBox.*
id	*integer* •id	*The unique ID of the TextBox.*
images	{Images} •images	*A collection of bitmap images in any bitmap file format (including TIFF, JPEG, or GIF).*
index	*integer* •index	*The index of the TextBox within its containing object.*
itemLayer	{Layer} itemLayer	*The layer that the TextBox is on.*
label	*string* label	*A property that can be set to any string.*
leftLineEnd	#ArrowHead leftLineEnd	*The arrowhead applied to the start of the path.*
linkedPageItemOptions	{LinkedPageItemOption} •linkedPageItemOptions	*Linked Page Item options.*
localDisplaySetting	#DisplaySettingOptions localDisplaySetting	*Display performance options for the TextBox.*
locked	*boolean* locked	*If true, the TextBox is locked.*
miterLimit	*double* miterLimit	*The limit of the ratio of stroke width to miter length before a miter (pointed) join becomes a bevel (squared-off) join. Type: Double (1 - 500).*
movieBehaviors	{MovieBehaviors} •movieBehaviors	*A collection of movie behavior objects.*
multiline	*boolean* multiline	*If true, the text field is multiline in the exported PDF.*
name	*string* name	*The name of the TextBox.*
nonprinting	*boolean* nonprinting	*If true, the TextBox does not print.*
openFileBehaviors	{OpenFileBehaviors} •openFileBehaviors	*A collection of open file behavior objects.*
ovals	{Ovals} •ovals	*A collection of ellipses.*

TEXTBOX CS6 VALUES	PROTOTYPE	DESCRIPTION
overprintFill	*boolean* overprintFill	*If true, the TextBox's fill color overprints any underlying objects. If false, the fill color knocks out the underlying colors.*
overprintGap	*boolean* overprintGap	*If true, the gap color overprints any underlying colors. If false, the gap color knocks out the underlying colors.*
overprintStroke	*boolean* overprintStroke	*If true, the TextBox's stroke color overprints any underlying objects. If false, the stroke color knocks out the underlying colors.*
overridden	*boolean* •overridden	*If true, the object originated on a master spread and was overridden. If false, the object either originated on a master spread and was not overridden, or the object did not originate on a master page.*
overriddenMasterPageItem	*variant* •overriddenMasterPageItem	*An object that originated on a master page and has been overridden. Type: PageItem, Guide, Graphic, Movie or Sound.*
pageItems	{PageItems} •pageItems	*The page items collection, which can be used to process all page items in a container (such as a document, page, or group), regardless of type.*
parent	*variant* •parent	*The parent of the TextBox (a Snippet, PlaceGun, Character, Spread, MasterSpread, SplineItem, Polygon, GraphicLine, Rectangle, Oval or Group). Type: Snippet, PlaceGun, Character, Spread, MasterSpread, SplineItem, Polygon, GraphicLine, Rectangle, Oval or Group.*
parentPage	{Page} •parentPage	*The page on which this page item appears.*
password	*boolean* password	*If true, the text field is a password field in the exported PDF.*
pdfs	{PDFs} •pdfs	*A collection of PDF files.*
picts	{PICTs} •picts	*A collection of PICT graphics.*
polygons	{Polygons} •polygons	*A collection of polygons.*
preferences	{Preferences} •preferences	*A collection of preferences objects.*
printableInPdf	*boolean* printableInPdf	*If true, the form field/push button is printable in the exported PDF.*
printFormBehaviors	{PrintFormBehaviors} •printFormBehaviors	*A collection of print form behavior objects.*
readOnly	*boolean* readOnly	*If true, the form field is read only in the exported PDF.*
rectangles	{Rectangles} •rectangles	*A collection of rectangles.*
required	*boolean* required	*If true, the form field is required in the exported PDF.*
rightLineEnd	#ArrowHead rightLineEnd	*The arrowhead applied to the end of the path.*
rightToLeftField	*boolean* rightToLeftField	*If true, the form field has right to left text enabled in the exported PDF.*
rotationAngle	*double* rotationAngle	*The rotatation angle of the TextBox. (Range: -360 to 360).*
scrollable	*boolean* scrollable	*If true, the text field is scrollable in the exported PDF.*
shearAngle	*double* shearAngle	*The skewing angle applied to the TextBox. (Range: -360 to 360).*
showHideFieldsBehaviors	{ShowHideFieldsBehaviors} •showHideFieldsBehaviors	*A collection of show/hide fields behavior objects.*
soundBehaviors	{SoundBehaviors} •soundBehaviors	*A collection of sound behavior objects.*
splineItems	{SplineItems} •splineItems	*The spline items collection.*
strokeAlignment	#StrokeAlignment strokeAlignment	*The stroke alignment applied to the TextBox.*
strokeColor	*variant* strokeColor	*The swatch (color, gradient, tint, or mixed ink) applied to the stroke of the TextBox. Type: Swatch or String.*
strokeCornerAdjustment	#StrokeCornerAdjustment strokeCornerAdjustment	*The corner adjustment applied to the TextBox.*

TEXTBOX CS6 VALUES	PROTOTYPE	DESCRIPTION
strokeDashAndGap	*variant* strokeDashAndGap	*The dash and gap measurements that define the pattern of a custom dashed line. Define up to six values (in points) in the format [dash1, gap1, dash2, gap2, dash3, gap3]. Type: Array of Units (Doubles or Strings).*
strokeTint	*double* strokeTint	*The percent of tint to use in object's stroke color. (To specify a tint percent, use a number in the range of 0 to 100; to use the inherited or overridden value, use -1.).*
strokeTransparencySettings	{StrokeTransparencySetting} •strokeTransparencySettings	*Transparency settings for the stroke.*
strokeType	*variant* strokeType	*The name of the stroke style to apply. Type: StrokeStyle or String.*
strokeWeight	*variant* strokeWeight	*The weight (in points) to apply to the TextBox's stroke. Type: Unit (Double or String).*
submitFormBehaviors	{SubmitFormBehaviors} •submitFormBehaviors	*A collection of submit form behavior objects.*
textFrames	{TextFrames} •textFrames	*A collection of text frames.*
textWrapPreferences	{TextWrapPreference} •textWrapPreferences	*The text wrap preference properties that define the default formatting for wrapping text around objects.*
timingSettings	{TimingSetting} •timingSettings	*The object timing settings.*
topLeftCornerOption	#CornerOptions topLeftCornerOption	*The shape to be applied to the top left corner of rectangular shapes and all corners of non-rectangular shapes.Note: corner option differs from end join in which you can set a radius for a corner option, whereas the rounded or beveled effect of an end join depends on the stroke weight.*
topLeftCornerRadius	*variant* topLeftCornerRadius	*The radius in measurement units of the corner effect applied to the top left corner of rectangular shapes and all corners of non-rectangular shapes. Type: Unit (Double or String).*
topRightCornerOption	#CornerOptions topRightCornerOption	*The shape to apply to the top right corner of rectangular shapes.*
topRightCornerRadius	*variant* topRightCornerRadius	*The radius in measurement units of the corner effect applied to the top right corner of rectangular shapes. Type: Unit (Double or String).*
transparencySettings	{TransparencySetting} •transparencySettings	*Transparency settings.*
verticalLayoutConstraints	*variant* verticalLayoutConstraints	*The top margin, height, and bottom margin constraints this item is subject to when using the object-based layout rule. Type: Array of 3 idDimensionsConstraints enumerators.*
verticalScale	*double* verticalScale	*The vertical scaling applied to the TextBox.*
viewZoomBehaviors	{ViewZoomBehaviors} •viewZoomBehaviors	*A collection of view zoom behavior objects.*
visible	*boolean* visible	*If true, the TextBox is visible.*
visibleBounds	*variant* visibleBounds	*The bounds of the TextBox including the stroke width, in the format [y1, x1, y2, x2], which give the coordinates of the top-left and bottom-right corners of the bounding box. Type: Array of 4 Units (Doubles or Strings).*
wmfs	{WMFs} •wmfs	*A collection of WMF graphics.*

TextBoxes

Object available in ID 8

TEXTBOXES CS6 VALUES	PROTOTYPE	DESCRIPTION
add	{TextBox} add(*variant* Layer, #LocationOptions At, *variant* Reference)	*Creates a new TextBox. / Return value: The new TextBox. / Layer: The layer on which to create the TextBox. Type: Layer. / At: The location at which to insert the TextBox relative to the reference object or within the container object. / Reference: The reference object. Note: Required when the at parameter is before or after. Type: Document, Spread, MasterSpread, Page, Layer or PageItem.*
anyItem	IDispatch anyItem()	*Returns any object in the collection. / Return value: The object.*
firstItem	IDispatch firstItem()	*Returns the first object in the collection. / Return value: The first object.*
itemByID	IDispatch itemByID(*integer* Id)	*Returns the object with the specified ID. / Return value: The object. / ID: The ID.*
itemByRange	{Objects} itemByRange(*variant* From, *variant* To)	*Returns the objects within the specified range. / Return value: The objects. / From: The object, index, or name at the beginning of the range. Type: Object, Long Integer or String. / To: The object, index, or name at the end of the range. Type: Object, Long Integer or String.*
lastItem	IDispatch lastItem()	*Returns the last object in the collection. / Return value: The last object.*
middleItem	IDispatch middleItem()	*Returns the middle object in the collection. / Return value: The middle object.*
nextItem	IDispatch nextItem(IDispatch Obj)	*Returns the object whose index follows the specified object in the collection. / Return value: The next object. / Obj: The object whose index comes before the desired object.*
previousItem	IDispatch previousItem(IDispatch Obj)	*Returns the object with the index previous to the specified index. / Return value: The previous object. / Obj: The index of the object that follows the desired object.*
count	*integer* •count	*Displays the number of elements in the TextBox.*
item	{TextBox} •item(*variant* Index)	*Returns the object with the specified index or name. Type: Long Integer or String.*

TextBoxEvents

Object available in ID 8

TEXTBOXEVENTS CS6 VALUES	PROTOTYPE	DESCRIPTION
afterPlace	*const* afterPlace(afterPlace)	*Dispatched after a TextBox is placed. This event bubbles. This event is not cancelable.*
beforePlace	*const* beforePlace(beforePlace)	*Dispatched before a TextBox is placed. This event bubbles. This event is cancelable.*