

# ADOBE® ROBOHELP® 8

## Scripting Guide

© 2009 Adobe Systems Incorporated. All rights reserved.

Adobe® RoboHelp® 8 Scripting Guide for Windows®

Adobe, the Adobe logo, Adobe AIR, AIR, FlashHelp, FrameMaker, and RoboHelp are either registered trademarks or trademarks of Adobe Systems Incorporated in the United States and/or other countries. Windows is either a registered trademark or trademark of Microsoft Corporation in the United States and/or other countries. All other trademarks are the property of their respective owners.

This work is licensed under the Creative Commons Attribution Non-Commercial 3.0 License. To view a copy of this license, visit <http://creativecommons.org/licenses/by-nc/3.0/us/>

Adobe Systems Incorporated, 345 Park Avenue, San Jose, California 95110, USA.

# Contents

## Chapter 1: Scripting basics

Scripting support in RoboHelp .....	1
RoboHelp object model .....	2

## Chapter 2: Classes

RoboHelp .....	4
Project .....	9
TokenManager .....	15
Token .....	16
TopicManager .....	19
MasterPageManager .....	21
SnippetManager .....	22
UDVManager .....	24
FileManager .....	25
TOCManager .....	26
IndexManager .....	28
GlossaryManager .....	29
SkinManager .....	30
CBTManager .....	31
SSLManager .....	32
FMFileManager .....	33
WordFileManager .....	34
ReportManager .....	35
ServerManager .....	37
Server .....	38
MapIdManager .....	41
HTMLHelpWindowManager .....	44
File .....	45
Skin .....	47
CBT .....	50
UDV .....	51
SSL .....	52
Snippet .....	56
Topic .....	59
MasterPage .....	62
TOC .....	65
Index .....	68
HTMLHelpWindow .....	71
Glossary .....	73
FMFile .....	76
WordFile .....	79

**Chapter 3: Constants**

Syntax for using constants .....	83
SSLLayoutType .....	83
Language .....	83
ReportType .....	85
DeleteOption .....	86
Status .....	86
SkinType .....	86
FrameFileFormat .....	86
TokenType .....	87
TagType .....	87
FileStatus .....	88
SyncStatus .....	88
ServerType .....	89
WordFileFormat .....	89
HTMLHelpWindowAlign .....	89
HTMLHelpWindowDefaultTab .....	90
HTMLHelpWindowButtons .....	90
HTMLHelpWindowProperties .....	91
<b>Index</b> .....	<b>93</b>

# Chapter 1: Scripting basics

A script is a series of commands that instructs Adobe® RoboHelp® to perform a set of specified actions, such as counting the images used in a RoboHelp project or selecting multiple master pages and importing them. Scripts automate repetitive tasks and are often used as a creative tool to streamline tasks that are time-consuming.

Before you begin scripting, take a few moments to read an overview of the scripting support in RoboHelp and find out how to create and run scripts.

If you are familiar with scripting or programming languages, you can skip this chapter and refer to “[Classes](#)” on page 4 and “[Constants](#)” on page 83 for information about RoboHelp scripting classes and constants.

If you are new to scripting, you can acquaint yourself with the basic scripting concepts provided in the [Introduction to Scripting](#) manual.

## Scripting support in RoboHelp

RoboHelp supports ExtendScript, which is an extended version of JavaScript. As a beginner, the difference between these two languages does not affect you. However, name JavaScript scripts with a .jsx extension, instead of the usual .js extension. ExtendScript files are distinguished by the .jsx extension.

### Creating scripts

Adobe provides ExtendScript Toolkit (ESTK) with its applications as a scripting utility. The ESTK has many features that make it easier to use than a text editor, including a built-in syntax checker that identifies where the problems are in your script and tries to explain how to fix them, and the ability to run your scripts right from the ESTK without saving the file. This second feature can save you a lot of time, especially in the beginning when you may have to test and edit a script several times to get it to work.

### Debug scripts

- ❖ Select the Adobe RoboHelp 8 option as the target application from the drop-down list in the ESTK.

### Edit scripts

- ❖ Double-click a script file in Script Explorer or Windows Explorer to edit it using the ESTK.

## Managing scripts from within RoboHelp

The Script Explorer pod enables you to manage scripts and do any of the following:

**Create a script folder** Create a folder for storing specific scripts. For example, you can store scripts that record data in one folder and scripts that display output in another folder.

**Reorganize folders** You can reorganize folders through simple drag-and-drop operations.


**Import and export** You can export and import script files and folders in different projects and locations. To import or export a folder, it must contain at least one script file.

## Running scripts

- From the command line, enter the following command:

```
Robohtml.exe -x [scriptfilenames]
```

You can specify multiple script filenames, separated by a space, to run one script after another. If the filename path contains a space, enclose the filename path in single or double quotation marks.

- In RoboHelp, choose Tools > Scripts to run sample scripts or create new scripts and run them.
- In Windows Explorer, double-click a script to open it in ExtendScript Toolkit and run it.
- In ExtendScript Toolkit, select Adobe RoboHelp 8 from the ExtendScript Toolkit pop-up menu and click the Click To Connect To Target Application  to run RoboHelp.

## RoboHelp sample scripts

RoboHelp provides a set of sample scripts that you can use or customize. You can use these sample scripts to do the following:

**EclipseHelp** Use this script to convert WebHelp output to EclipseHelp output. See the script for more details.

**Link Converter** Use this script to convert an anchor link href target across all the files in a RoboHelp project. For example, define a link to convert `www.adobe.com` to `www.adobe.com/support/` across all the Help files in a project. See the script file for more information.

**MultiMasterPageImport** Use this script to select multiple master pages and import them.

**SaveAsProjectTemplate** Use this script to save a RoboHelp project as a template for creating similar RoboHelp projects.

**UDV Converter with UI** Use this script to convert a keyword into a user-defined variable and change its value across all files in a project. Enter a keyword, a user-defined variable name, and its value.

**UDV Converter** Use this script to convert a keyword into a user-defined variable and change its value across all the files in a project. See the script for more details.

**Word Count** Use this script to get a word count for an open RoboHelp project. It provides a word count by topic and by project. To run the Word Count script, right-click the script and select Run. The word count summary appears in the Output View pod.

## RoboHelp object model

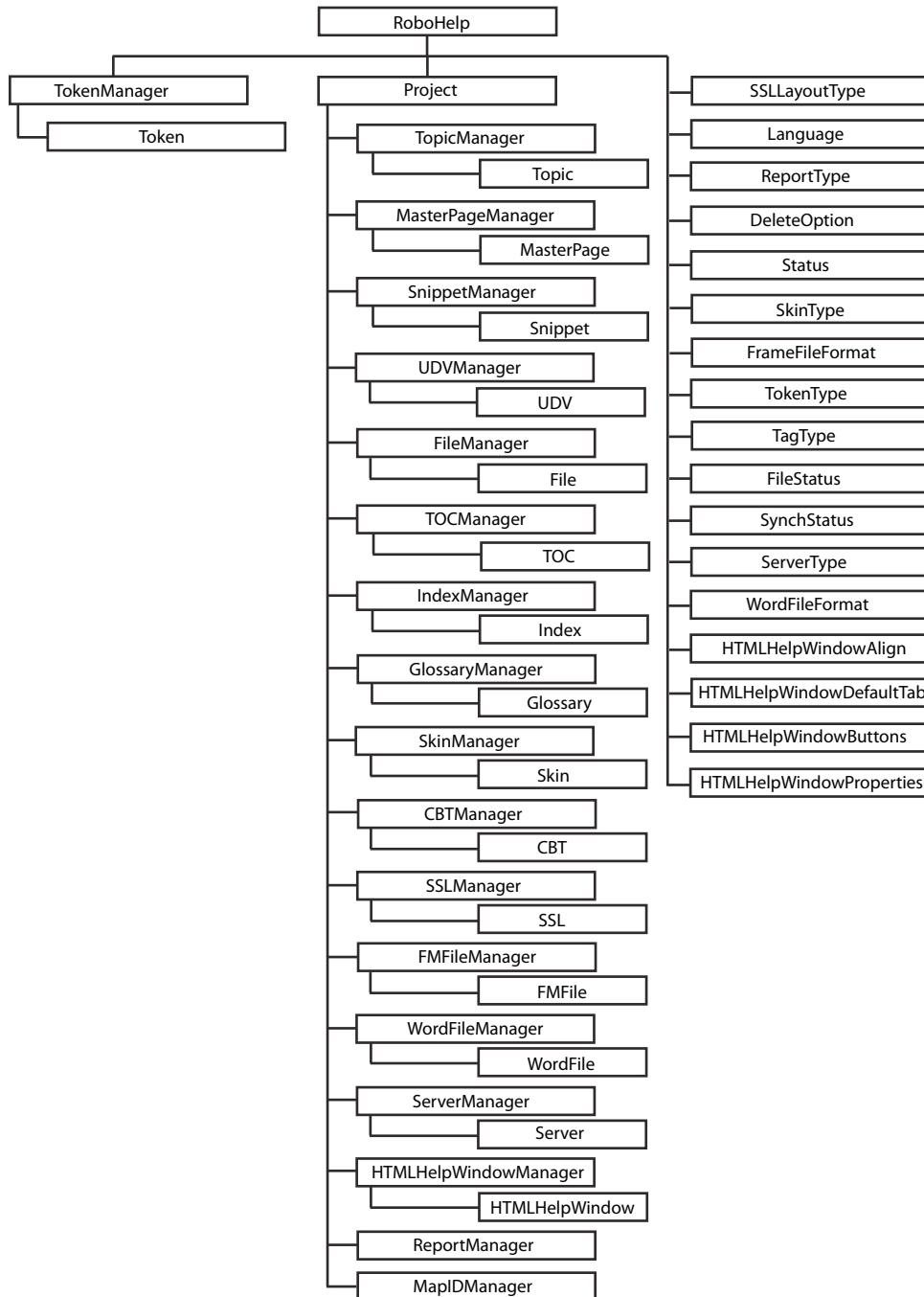
The RoboHelp object model is based on the principles of a document object model (DOM). A DOM is an application programming interface (API) that allows you to programmatically access various components of a document (as defined for that application) through a scripting language. The principle behind a DOM is the containment hierarchy. In other words, top-level objects in a DOM contain next-level objects, which contain the subsequent level of objects, and so on.

Each node in the RoboHelp containment hierarchy is a class. The `RoboHelp` class sits at the top of the hierarchy and is a collection of classes and constants. To accomplish a task through scripting, you need to reference these objects following the parent-child relationship defined in the containment hierarchy.

For example, if you want to specify Adobe® AIR® Help as the single-source layout for a RoboHelp application, the object reference in your script needs to be:

RoboHelp.SSLLayoutType.AIRHELP

SSLLayout is a child class of the RoboHelp class, and AIRHELP is a property of the SSLLayout class.



*RoboHelp containment hierarchy*

For more information on classes, see “[Classes](#)” on page 4.

For more information on constants, see “[Constants](#)” on page 83.

# Chapter 2: Classes

## RoboHelp

Main class for interacting with a RoboHelp application.

### RoboHelp properties

Property name	Data type	Description
version	String	Read-only RoboHelp product version
name	String	Read-only Product name
path	String	Read-only Full path to the RoboHelp application executable
language	Integer constant	Read-only RoboHelp language settings
project	Project	Project object
SSLLayoutType	SSLLayoutType	Read-only SSLLayoutType constant
Language	Language	Read-only Language constant
ReportType	ReportType	Read-only ReportType constant
DeleteOption	DeleteOption	Read-only DeleteOption constant
Status	Status	Read-only Status constant
SkinType	SkinType	Read-only SkinType constant
FrameFileFormat	FrameFileFormat	Read-only FrameFileFormat constant
TokenType	TokenType	Read-only TokenType constant
TagType	TagType	Read-only TagType constant

Property name	Data type	Description
FileStatus	<a href="#">FileStatus</a>	Read-only FileStatus constant
SyncStatus	<a href="#">SyncStatus</a>	Read-only SyncStatus constant
ServerType	<a href="#">ServerType</a>	Read-only ServerType constant
WordFileFormat	<a href="#">WordFileFormat</a>	Read-only WordFileFormat constant

## RoboHelp methods

### openProject

#### Description

Opens .mpj and .xpj projects in a RoboHelp application.

#### Syntax

```
openProject (projectPath, upgradeIfRequired)
```

#### Parameters

Parameter name	Data type	Optional	Description
projectPath	String	No	Full path to the project file that you want to open
upgradeIfRequired	Boolean	Yes	Upgrade the project if created in an older version of RoboHelp. The default value is false. If the project does not require an upgrade, this parameter is ignored.

### getTokenManager

#### Description

Returns Token Manager for the specified file.

#### Returns

[TokenManager](#)

#### Syntax

```
getTokenManager (fullFilePath)
```

**Parameters**

Parameter name	Data type	Optional	Description
fullFilePath	String	No	Full path to the file that you want to open

**getCurrentProject****Description**

Returns the active project object.

**Returns**

[Project](#)

**Syntax**

```
getCurrentProject ()
```

**closeProject****Description**

Closes the active project.

*Note: Unsaved topic files are not saved.*

**Returns**

String

**Syntax**

```
closeProject ()
```

**newProject****Description**

Creates a project.

**Returns**

[Project](#)

**Syntax**

```
newProject (template, projectName, projectPath, title, firstTopic, language, saveSettings)
```

## Parameters

Parameter name	Data type	Optional	Description
template	String	No	Name of the template or full path to the Template.hpr file. Specify an empty string to create a blank project.
projectName	String	No	Name of the project
projectPath	String	No	Full path to the location where the project will be created
title	String	Yes	Title of the project
firstTopic	String	Yes	Title of the first topic
language	<a href="#">Integer constant</a>	Yes	Language for the project
saveSettings	Boolean	Yes	Saves the specified language as the default language for new projects

## newProjectImport

### Description

Creates a project from a DITA map.

### Returns

[Project](#)

### Syntax

```
newProjectImport (projectName, outputFolder, DITAOTHomeDirectory, ditavalFile, topHeader,
bodyHeader, bodyFooter, xslFile, indexShow)
```

## Parameters

Parameter name	Data type	Optional	Description
projectName	String	No	Full path to the DITA map file
outputFolder	String	No	Full folder path where the project will be created
DITAOTHomeDirectory	String	No	Full path to the DITA Open Toolkit home directory
ditavalFile	String	Yes	Full path to the Ditaval file for filtered processing
topHeader	String	Yes	Full path to the file containing XHTML to place in the header area of the output file
bodyHeader	String	Yes	Full path to the file containing XHTML to place in the body running-header area of the output file

Parameter name	Data type	Optional	Description
bodyFooter	String	Yes	Full path to the file containing XHTML to place in the body running-footer area of the output file.
xslFile	String	Yes	Full path to the XSL file that is used instead of the default XSL file to transform the topics to XHTML
indexShow	Boolean	Yes	Use this option to show or hide index entries

## bringToFront

### Description

Makes the RoboHelp application the active window.

### Syntax

```
bringToFront ()
```

## quit

### Description

Exits the RoboHelp application.

### Syntax

```
quit ()
```

## executeScript

### Description

Runs a script file.

### Returns

String

### Syntax

```
executeScript (scriptFilePath)
```

### Parameters

Parameter name	Data type	Optional	Description
scriptFilePath	String	No	Full path to the script file to run

## closeAllWindows

### Description

Closes all active document windows in the RoboHelp application.

**Syntax**

```
closeAllWindows ()
```

# Project

Main class for handling project-related activities.

## Project properties

Property name	Data type	Description
name	String	Read-only Project name
title	String	Read-only Project title
path	String	Read-only Full path to the project main folder
typename	String	Read-only Class name of the reference object
language	<a href="#">Language</a>	Read-write Project language settings
TopicManager	<a href="#">TopicManager</a>	Read-only TopicManager object
MasterPageManager	<a href="#">MasterPageManager</a>	Read-only MasterPageManager object
SnippetManager	<a href="#">SnippetManager</a>	Read-only SnippetManager object
UDVManager	<a href="#">UDVManager</a>	Read-only UDVManager object
FileManager	<a href="#">FileManager</a>	Read-only FileManager object
TOCManager	<a href="#">TOCManager</a>	Read-only TOCManager object
IndexManager	<a href="#">IndexManager</a>	Read-only IndexManager object
GlossaryManager	<a href="#">GlossaryManager</a>	Read-only GlossaryManager object
SkinManager	<a href="#">SkinManager</a>	Read-only SkinManager object

Property name	Data type	Description
CBTManager	<a href="#">CBTManager</a>	Read-only CBTManager object
SSLManager	<a href="#">SSLManager</a>	Read-only SSLManager object
FMFileManager	<a href="#">FMFileManager</a>	Read-only FMFileManager object
WordFileManager	<a href="#">WorldFileManager</a>	Read-only WordFileManager object
ReportManager	<a href="#">ReportManager</a>	Read-only ReportManager object
ServerManager	<a href="#">ServerManager</a>	Read-only ServerManager object
MapIdManager	<a href="#">MapIdManager</a>	Read-only MapIdManager object
HTMLHelpWindowManager	<a href="#">HTMLHelpWindowManager</a>	Read-only HTMLHelpWindowManager object

## Project methods

### importSettings

#### Description

Imports the settings file (\*.isf) into the current project.

#### Returns

Boolean

#### Syntax

```
importSettings (settingsFilePath)
```

#### Parameters

Parameter name	Data type	Optional	Description
settingsFilePath	String	No	Full path to the .isf settings file

### exportSettings

#### Description

Exports the settings file (\*.isf) from the current project.

**Returns**

Boolean

**Syntax**`exportSettings (settingsFilePath, overwriteFile)`**Parameters**

Parameter name	Data type	Optional	Description
settingsFilePath	String	No	Full path to the .isf settings file
overwriteFile	Boolean	Yes	Flag to overwrite the existing file The default value is true.

**outputMessage****Description**

Displays the message on the Output View pod.

**Returns**

Boolean

**Syntax**`outputMessage (message)`**Parameters**

Parameter name	Data type	Optional	Description
message	String	No	String message to display

**rename****Description**

Renames the current project.

**Returns**

Boolean

**Syntax**`rename (newName)`**Parameters**

Parameter name	Data type	Optional	Description
newName	String	No	New name for the project

## updateAll

### Description

Updates all linked documents (FrameMaker documents and Word documents) in the project.

### Returns

Boolean

### Syntax

```
updateAll (flagForceUpdate)
```

### Parameters

Parameter name	Data type	Optional	Description
flagForceUpdate	Boolean	No	Flag to force the update of all linked documents

## importDITAMap

### Description

Imports a DITA map file into the project.

### Returns

Boolean

### Syntax

```
importDITAMap (projectName, DITAOTHomeDirectory, ditavalFile, topHeader, bodyHeader,  
bodyFooter, xslFile, indexShow)
```

### Parameters

Parameter name	Data type	Optional	Description
projectName	String	No	Full path to the DITA map file
DITAOTHomeDirectory	String	No	Full path to the DITA Open Toolkit home directory
ditavalFile	String	Yes	Full path to the Ditaval file for filtered processing
topHeader	String	Yes	Full path to the file containing XHTML to place in the header area of the output file
bodyHeader	String	Yes	Full path to the file containing XHTML to place in the body running-header area of the output file

Parameter name	Data type	Optional	Description
<code>bodyFooter</code>	String	Yes	Full path to the file containing XHTML to place in the body running-footer area of the output file.
<code>xslFile</code>	String	Yes	Full path to the XSL file that is used instead of the default XSL file to transform topics to XHTML
<code>indexShow</code>	Boolean	Yes	Use this option to show or hide index entries

## autoCreateBrowseSequence

### Description

Creates browse sequences based on a specified TOC.

### Returns

Boolean

### Syntax

```
autoCreateBrowseSequence (TOC object, level)
```

### Parameters

Parameter name	Data type	Optional	Description
<code>TOC object</code>	TOC	Yes	A TOC object on which the browse sequences will be based. If not specified, the default TOC of the project is used.
<code>level</code>	Integer	Yes	Creates a separate sequence for each book down to this level. If not specified, the last set value is used.

## importHtmlFile

### Description

Imports an HTML file into the current project.

### Returns

Boolean

### Syntax

```
importHtmlFile (filePath, folderName)
```

### Parameters

Parameter name	Data type	Optional	Description
<code>filePath</code>	String	No	Absolute path of the HTML file to import
<code>folderName</code>	String	No	Relative path of the folder in the current project in which the HTML file is to be imported

## importWordDoc

### Description

Imports a Word document into the current project.

### Returns

Boolean

### Syntax

```
importWordDoc (filePath, folderName, TOCName, addIndexToTopic, indexName, glossaryName)
```

### Parameters

Parameter name	Data type	Optional	Description
filePath	String	No	Absolute path of the Word document to import
folderName	String	No	Relative path of the folder in the current project in which the Word document is to be imported
TOCName	String	No	Name of the existing or new TOC for the imported Word document
addIndexToTopic	Boolean	No	Whether the index from the Word document is added to the imported topics or not
indexName	String	No	If <code>addIndexToTopic</code> is false, this is the name of the existing or new index for the imported Word document.
glossaryName	String	No	Name of the existing or new glossary for the imported Word document

## importFrameMakerDoc

### Description

Imports a FrameMaker document into the current project.

### Returns

Boolean

### Syntax

```
importFrameMakerDoc (filePath, folderName, frameMakerTOCFilePath, TOCName, addIndexToTopic, indexName, glossaryName)
```

## Parameters

Parameter name	Data type	Optional	Description
filePath	String	No	Absolute path of the FrameMaker document to import
folderName	String	No	Relative path of the folder the in the current project in which the FrameMaker document is to be imported
frameMakerTOCFilePath	String	No	Absolute path of the FrameMaker TOC file for the FrameMaker document to import
TOCName	String	No	Name of the existing or new TOC for the imported FrameMaker document
addIndexToTopic	Boolean	No	Whether the index from the FrameMaker document is added to the imported topics or not
indexName	String	No	If <code>addIndexToTopic</code> is false, this is the name of the existing or new index for the imported FrameMaker document.
glossaryName	String	No	Name of the existing or new glossary for the imported FrameMaker document

# TokenManager

Handles tokens for a specific file.

## TokenManager properties

Property name	Data type	Description
parent	<a href="#">RoboHelp</a>	Read-only RoboHelp application object
typename	String	Read-only Class name of the reference object
count	Integer	Read-only Number of tokens in the file
valid	Boolean	Read-only Checks if the token file is valid or not

## TokenManager methods

### item

#### Description

Returns the token specified by the index.

**Returns**[Token](#)**Syntax**`item (index)`**Parameters**

Parameter name	Data type	Optional	Description
<code>index</code>	Integer	No	Index for the token

**save****Description**

Saves the specified token file.

**Returns**

Boolean

**Syntax**`save ()`

## Token

Handles a specific token.

**Token properties**

Property name	Data type	Description
<code>parent</code>	<a href="#">TokenManager</a>	Read-only Parent TokenManager object
<code>name</code>	String	Read-only Text from the token object
<code>typename</code>	String	Read-only Class name of the reference object
<code>previous</code>	<a href="#">Token</a>	Read-only Returns the previous token object

Property name	Data type	Description
next	Token	Read-only Returns the next object
tokenType	Integer constant	Read-only Type of token
tagType	Integer constant	Read-only Returns the type of tag if tokenType is TOKENTAG

## Token methods

### update

#### Description

Updates the text of the token if tokenType is TOKENTEXT.

#### Returns

Boolean

#### Syntax

```
update (newText)
```

#### Parameters

Parameter name	Data type	Optional	Description
newText	String	No	New text string to update for the token

### getAttribute

#### Description

Returns the specific attribute value.

#### Returns

String

#### Syntax

```
getAttribute (attributeName)
```

#### Parameters

Parameter name	Data type	Optional	Description
attributeName	String	No	Name of the attribute

## setAttribute

### Description

Sets the specific attribute value.

### Returns

Boolean

### Syntax

```
setAttribute (attributeName, attributeValue)
```

### Parameters

Parameter name	Data type	Optional	Description
attributeName	String	No	Name of the attribute
attributeValue	String	No	Value of the attribute

## removeAttribute

### Description

Removes the specific attribute value.

### Returns

Boolean

### Syntax

```
removeAttribute (attributeName)
```

### Parameters

Parameter name	Data type	Optional	Description
attributeName	String	No	Name of the attribute

## insertText

### Description

Inserts a new text token before or after the specified token.

### Returns

Boolean

### Syntax

```
insertText (newText, afterCurrentTag)
```

**Parameters**

Parameter name	Data type	Optional	Description
newText	String	No	New text to insert
afterCurrentTag	Boolean	Yes	Inserts a new text token before a specified token By default, the new token is inserted after the current tag.

**delete****Description**

Removes the token.

**Returns**

Boolean

**Syntax**

```
delete ()
```

## TopicManager

Handles all topics in the current project.

**TopicManager properties**

Property name	Data type	Description
parent	<a href="#">Project</a>	Read-only Project object
typename	String	Read-only Class name of the reference object
count	Integer	Read-only Number of topics present in the project

**TopicManager methods****item****Description**

Returns the topic specified by the index.

**Returns**

[Topic](#)

**Syntax**

```
item (index)
```

**Parameters**

Parameter name	Data type	Optional	Description
index	Integer	No	Index for the topic

**newItem****Description**

Creates a topic in the current project.

**Returns**

[Topic](#)

**Syntax**

```
newItem (topicTitle, fileName, language)
```

**Parameters**

Parameter name	Data type	Optional	Description
topicTitle	String	No	Title of the topic
fileName	String	No	Filename of the topic
language	<a href="#">Integer constant</a>	No	Language of topic

**importItem****Description**

Imports an HTML file into the current project.

**Returns**

[Topic](#)

**Syntax**

```
importItem (filePath, folderName)
```

**Parameters**

Parameter name	Data type	Optional	Description
filePath	String	No	Absolute path of the HTML file to import
folderName	String	No	Relative path of the folder in the current project in which the HTML file is to be imported

# MasterPageManager

Handles all master pages in the current project.

## MasterPageManager properties

Property name	Data type	Description
parent	<a href="#">Project</a>	Read-only Project object
typename	String	Read-only Class name of the reference object
count	Integer	Read-only Number of master pages present in the project

## MasterPageManager methods

### item

#### Description

Returns the master page specified by the index.

#### Returns

[MasterPage](#)

#### Syntax

```
item (index)
```

#### Parameters

Parameter name	Data type	Optional	Description
index	Integer	No	Index for the master page

### newItem

#### Description

Creates a master page in the current project.

#### Returns

[MasterPage](#)

#### Syntax

```
newItem (title, css, desc)
```

### Parameters

Parameter name	Data type	Optional	Description
title	String	Yes	Title of the master page The filename for the master page is also same as the title.
css	String	Yes	CSS to attach The path of the CSS file must be relative to the project path. If this parameter is null, no CSS is attached. If this parameter is not specified or is empty, the default CSS of the project is attached.
desc	String	Yes	Description

### importItem

#### Description

Imports a master page into the current project.

#### Returns

[MasterPage](#)

#### Syntax

```
importItem (filePath)
```

### Parameters

Parameter name	Data type	Optional	Description
filePath	String	No	Absolute path of the master page to import

## SnippetManager

Handles all snippets in the current project.

### SnippetManager properties

Property name	Data type	Description
parent	<a href="#">Project</a>	Read-only Project object
typename	String	Read-only Class name of the reference object
count	Integer	Read-only Number of snippets present in the project

## SnippetManager methods

### item

#### Description

Returns the snippet specified by the index.

#### Returns

[Snippet](#)

#### Syntax

```
item (index)
```

#### Parameters

Parameter name	Data type	Optional	Description
index	Integer	No	Index for the snippet

### newItem

#### Description

Creates a snippet in the current project.

#### Returns

[Snippet](#)

#### Syntax

```
newItem (snippetName)
```

#### Parameters

Parameter name	Data type	Optional	Description
snippetName	String	No	Name of the snippet

### importFile

#### Description

Imports an existing snippet into the current project.

#### Returns

[Snippet](#)

#### Syntax

```
importFile (snippetFilePath)
```

**Parameters**

Parameter name	Data type	Optional	Description
snippetFilePath	String	No	Full path of the snippet file

## UDVManager

Handles all user-defined variables (UDVs) in the current project.

**UDVManager properties**

Property name	Data type	Description
parent	<a href="#">Project</a>	Read-only Project object
typename	String	Read-only Class name of the reference object
count	Integer	Read-only Number of UDVs present in the project

**UDVManager methods****item****Description**

Returns the UDV specified by the index.

**Returns**

[UDV](#)

**Syntax**

```
item (index)
```

**Parameters**

Parameter name	Data type	Optional	Description
index	Integer	No	Index for the UDV

**newItem****Description**

Creates a UDV in the current project.

**Returns**[UDV](#)**Syntax**

newItem (UDVName, UDVValue)

**Parameter**

Parameter name	Data type	Optional	Description
UDVName	String	No	Name of the new UDV
UDVValue	String	No	Value of the new UDV

## FileManager

Handles all files in the current project.

**FileManager properties**

Property name	Data type	Description
parent	<a href="#">Project</a>	Read-only Project object
typename	String	Read-only Class name of the reference object
count	Integer	Read-only Number of files present in the project

**FileManager methods****item****Description**

Returns the file specified by the index.

**Returns**[File](#)**Syntax**

item (index)

**Parameters**

Parameter name	Data type	Optional	Description
index	Integer	No	Index for the file

## **checkoutAll**

### **Description**

Checks out all files in the current project from the version control system.

### **Returns**

Boolean

### **Syntax**

```
checkoutAll ()
```

## **checkinAll**

### **Description**

Checks in all files in the current project to the version control system.

### **Returns**

Boolean

### **Syntax**

```
checkinAll ()
```

## **getLatestAll**

### **Description**

Gets the latest version of all the files from the version control system into the current project.

### **Returns**

Boolean

### **Syntax**

```
getLatestAll ()
```

## **TOCManager**

Handles all TOCs in the current project.

## TOCManager properties

Property name	Data type	Description
parent	<a href="#">Project</a>	Read-only Project object
typename	String	Read-only Class name of the reference object
count	Integer	Read-only Number of TOC files present in the project

## TOCManager methods

### item

#### Description

Returns the TOC specified by the index.

#### Returns

[TOC](#)

#### Syntax

```
item (index)
```

#### Parameters

Parameter name	Data type	Optional	Description
index	Integer	No	Index for the TOC

### newItem

#### Description

Creates a TOC in the current project.

#### Returns

[TOC](#)

#### Syntax

```
newItem (newTOCName, existingTOCFilePath)
```

#### Parameters

Parameter name	Data type	Optional	Description
newTOCName	String	No	Name of the new TOC
existingTOCFilePath	String	Yes	Path of an existing TOC file, which is copied into the new TOC

# IndexManager

Handles all index files in the current project.

## IndexManager properties

Property name	Data type	Description
parent	<a href="#">Project</a>	Read-only Project object
typename	String	Read-only Class name of the reference object
count	Integer	Read-only Number of index files present in the project

## IndexManager methods

### item

#### Description

Returns the index specified by the index.

#### Returns

[Index](#)

#### Syntax

```
item (index)
```

#### Parameters

Parameter name	Data type	Optional	Description
index	Integer	No	Index for the index file

### newItem

#### Description

Creates an index in the current project.

#### Returns

[Index](#)

#### Syntax

```
newItem (newIndexName, existingIndexPath)
```

**Parameters**

Parameter name	Data type	Optional	Description
newIndexName	String	No	Name of the new index
existingIndexFilePath	String	Yes	Path of an existing index file, which is copied into the new index

## GlossaryManager

Handles all glossary files in the current project.

**GlossaryManager properties**

Property name	Data type	Description
parent	<a href="#">Project</a>	Read-only Project object
typename	String	Read-only Class name of the reference object
count	Integer	Read-only Number of glossary files present in the project

**GlossaryManager methods****item****Description**

Returns the glossary file specified by the index.

**Returns**

[Glossary](#)

**Parameters**

Parameter name	Data type	Optional	Description
index	Integer	No	Index for the glossary

**newItem****Description**

Creates a glossary file in the current project.

**Returns**

[Glossary](#)

**Syntax**

```
newItem (newGlossaryName, existingGlossaryFilePath)
```

**Parameters**

Parameter name	Data type	Optional	Description
newGlossaryName	String	No	Name of new glossary
existingGlossaryFilePath	String	Yes	Path of an existing glossary file, which is copied into the new glossary

## SkinManager

Handles all skins in the current project.

**SkinManager properties**

Property name	Data type	Description
parent	<a href="#">Project</a>	Read-only Project object
typename	String	Read-only Class name of the reference object
count	Integer	Read-only Number of skin files present in the project

**SkinManager methods****item****Description**

Returns the skin specified by the index.

**Returns**

[Skin](#)

**Syntax**

```
item (index)
```

**Parameters**

Parameter name	Data type	Optional	Description
index	Integer	No	Index for the skin

# CBTManager

Handles all conditional build tags (CBTs) in the current project.

## CBTManager properties

Property name	Data type	Description
parent	<a href="#">Project</a>	Read-only Project object
typename	String	Read-only Class name of the reference object
count	Integer	Read-only Number of CBTs present in the project

## CBTManager methods

### item

#### Description

Returns the CBT specified by the index.

#### Returns

[CBT](#)

#### Syntax

```
item (index)
```

#### Parameters

Parameter name	Data type	Optional	Description
index	Integer	No	Index for the CBT

### newItem

#### Description

Creates a CBT in the current project.

#### Returns

[CBT](#)

#### Syntax

```
newItem (CBTName, redColor, greenColor, blueColor)
```

**Parameters**

Parameter name	Data type	Optional	Description
CBTName	String	No	Name of the new CBT
redColor	Integer	Yes	Specify the red color value for CBT, range (0-255)
greenColor	Integer	Yes	Specify the green color value for CBT, range (0-255)
blueColor	Integer	Yes	Specify the blue color value for CBT, range (0-255)

## SSLManager

Handles all single-source layouts (SSLs) in the current project.

**SSLManager properties**

Property name	Data type	Description
parent	<a href="#">Project</a>	Read-only Project object
typename	String	Read-only Class name of the reference object
count	Integer	Read-only Number of single-source layouts present in the project

**SSLManager methods****item****Description**

Returns the SSL object specified by the index.

**Returns**

[SSL](#)

**Syntax**

```
item (index)
```

**Parameters**

Parameter name	Data type	Optional	Description
index	Integer	No	Index for the SSL

# FMFileManager

Handles all FrameMaker files in the current project.

*Note: You can implement the `FMFileManager` class if you have Technical Communication Suite installed.*

## FMFileManager properties

Property name	Data type	Description
parent	<a href="#">Project</a>	Read-only Project object
typename	String	Read-only Class name of the reference object
count	Integer	Read-only Number of FrameMaker files present in the project

## FMFileManager methods

### item

#### Description

Returns the FrameMaker file specified by the index.

#### Returns

[FMFile](#)

#### Syntax

```
item (index)
```

#### Parameters

Parameter name	Data type	Optional	Description
index	Integer	No	Index for the FrameMaker file

### newItem

#### Description

Adds a new FrameMaker File, `FMFile`, in the current project.

#### Returns

[FMFile](#)

#### Syntax

```
newItem (FrameMakerFilePath, folderName)
```

**Parameters**

Parameter name	Data type	Optional	Description
FrameMakerFilePath	String	No	Path to the existing FrameMaker file
folderName	String	Yes	Relative path of the folder in the current project in which the FrameMaker file is to be linked

## WordFileManager

Handles all Word files in the current project.

**WordFileManager properties**

Property name	Data type	Description
parent	<a href="#">Project</a>	Read-only Project object
typename	String	Read-only Class name of the reference object
count	Integer	Read-only Number of Word files present in the project

**WordFileManager methods****item****Description**

Returns the Word file specified by the index.

**Returns**

[WordFile](#)

**Syntax**

```
item (index)
```

**Parameters**

Parameter name	Data type	Optional	Description
index	Integer	No	Index for the Word file

**newItem****Description**

Adds a new Word file, WordFile, in the current project.

**Returns**[WordFile](#)**Syntax**`newItem (WordFilePath, folderName)`**Parameters**

Parameter name	Data type	Optional	Description
WordFilePath	String	No	Path to the existing Word file
folderName	String	Yes	Relative path of the folder in the current project in which the Word File is to be linked

## ReportManager

Handles all types of reports in the current project.

**ReportManager properties**

Property name	Data type	Description
parent	<a href="#">Project</a>	Read-only Project object
typename	String	Read-only Class name of the reference object

**ReportManager methods****save****Description**

Saves a specified report type.

**Returns**

Boolean

**Syntax**`save (reportType, fileNamePath)`**Parameters**

Parameter name	Data type	Optional	Description
reportType	<a href="#">ReportType</a>	No	Report type
fileNamePath	String	No	Full path to identify the location of the report

## print

### Description

Prints a specified report type.

### Returns

Boolean

### Syntax

```
print (reportType)
```

### Parameters

Parameter name	Data type	Optional	Description
reportType	ReportType	No	Report type

## copy

### Description

Copies a specified report type to the Clipboard.

### Returns

Boolean

### Syntax

```
copy (reportType)
```

### Parameters

Parameter name	Data type	Optional	Description
reportType	ReportType	No	Report type

## mail

### Description

Sends a specified report type to specified e-mail addresses.

### Returns

Boolean

### Syntax

```
mail (reportType, emailIDs)
```

**Parameters**

Parameter name	Data type	Optional	Description
reportType	<a href="#">ReportType</a>	No	Report type
emailIDs	String	No	E-mail addresses, in semicolon-separated format

## ServerManager

Handles all servers specified in the current project

**ServerManager properties**

Property name	Data type	Description
parent	<a href="#">Project</a>	Read-only Project object
typename	String	Read-only Class name of the reference object
count	Integer	Read-only Number of servers present in the project

**ServerManager methods****item****Description**

Returns the server specified by the index.

**Returns**

[Server](#)

**Syntax**

```
item (index)
```

**Parameters**

Parameter name	Data type	Optional	Description
index	Integer	No	Index for the server

**newItem****Description**

Creates a server item in the current project.

**Returns**[Server](#)**Syntax**`newItem (serverDescription, serverType)`**Parameters**

Parameter name	Data type	Optional	Description
<code>serverDescription</code>	String	No	Description of the new server item
<code>serverType</code>	<a href="#">ServerType</a>	No	Type of server that you want to create

## Server

Handles specific server items.

**Server properties**

Property name	Data type	Description
<code>parent</code>	<a href="#">ServerManager</a>	Read-only Parent ServerManager object
<code>name</code>	String	Read-write Description of a server item
<code>typename</code>	String	Read-only Class name of the reference object
<code>previous</code>	<a href="#">Server</a>	Read-only Returns the previous server object
<code>next</code>	<a href="#">Server</a>	Read-only Returns the next server object
<code>serverType</code>	<a href="#">ServerType</a>	Read-only Server type
<code>hostName</code>	String	Read-write Host name of the server This will not apply for FILESYSTEM ServerType.

Property name	Data type	Description
port	Integer	Read-write Port number of the server This will apply only for FTP ServerType.
serverDirectory	String	Read-write Server directory This will not apply for FILESYSTEM ServerType.
destinationPath	String	Read-write Destination path This will apply only for FILESYSTEM ServerType.

## Server methods

### changeToFTP

#### Description

Changes the existing server to the FTP server type.

#### Returns

Boolean

#### Syntax

changeToFTP (hostName, portNumber, serverDirectory, username)

#### Parameters

Parameter name	Data type	Optional	Description
hostName	String	No	Host name for the server
portNumber	Integer	No	Port number for the server
serverDirectory	String	No	Server directory
username	String	Yes	Username for the FTP server

### changeToHTTP

#### Description

Changes the existing server to the HTTP server type.

#### Returns

Boolean

#### Syntax

changeToHTTP (hostName, serverDirectory)

**Parameters**

Parameter name	Data type	Optional	Description
hostName	String	No	Host name for the server
serverDirectory	String	No	Server directory

**changeToFrontPage****Description**

Changes the existing server to the FrontPage server type.

**Returns**

Boolean

**Syntax**

`changeToFrontPage (hostName, serverDirectory, webname)`

**Parameters**

Parameter name	Data type	Optional	Description
hostName	String	No	Host name for the server
serverDirectory	String	No	Server directory
webname	String	Yes	Web name for the FrontPage server

**changeToFileSystem****Description**

Changes the existing server to the FileSystem server type.

**Returns**

Boolean

**Syntax**

`changeToFileSystem (destinationPath)`

**Parameters**

Parameter name	Data type	Optional	Description
destinationPath	String	No	Destination path for FileSystem ServerType

**delete****Description**

Deletes the server.

**Returns**

Boolean

**Syntax**`delete ()`

## MapIdManager

Handles map ID manipulation in the current project.

### MapIdManager properties

Property name	Data type	Description
parent	<a href="#">Project</a>	Read-only Project object
typename	String	Read-only Class name of the reference object

### MapIdManager methods

#### newMapId

**Description**

Creates a map ID.

**Returns**

Boolean

**Syntax**`newMapId (mapId, mapNumber)`**Parameters**

Parameter name	Data type	Optional	Description
mapId	String	No	Topic ID
mapNumber	Integer	No	Map number

#### assign

**Description**

Assigns a map ID to a topic

**Returns**

Boolean

**Syntax**`assign (mapId, topicPath)`**Parameters**

Parameter name	Data type	Optional	Description
mapId	String	No	Topic ID
topicPath	String	No	File name of the topic

**unassign****Description**

Unassigns a map ID from a topic.

**Returns**

Boolean

**Syntax**`unassign (mapId)`**Parameters**

Parameter name	Data type	Optional	Description
mapId	String	No	Topic ID

**delete****Description**

Deletes a map ID.

**Returns**

Boolean

**Syntax**`delete (mapId)`**Parameters**

Parameter name	Data type	Optional	Description
mapId	String	No	Topic ID

## edit

### Description

Edits a map ID.

### Returns

Boolean

### Syntax

```
edit (mapId, mapNumber)
```

### Parameters

Parameter name	Data type	Optional	Description
mapId	String	No	Topic ID
mapNumber	Integer	No	Map number

## newMapFile

### Description

Creates a map file.

### Returns

Boolean

### Syntax

```
newMapFile (mapFilename)
```

### Parameters

Parameter name	Data type	Optional	Description
mapFilename	String	No	Name of the map file

## importMapFile

### Description

Imports an existing map file (\*.h, \*.hh, \*.hm).

### Returns

Boolean

### Syntax

```
importMapFile (mapFilename, destinationFolderInProject)
```

**Parameters**

Parameter name	Data type	Optional	Description
mapFilename	String	No	Full path of the map file
destinationFolderInProject	String	Yes	Path of the folder where the map file is imported. By default, the map file is imported in the project folder.

**setCurMapFile****Description**

Sets the current map file. After this function is called, new map IDs are created in this file. If the file does not exist, a default file is used.

**Returns**

Boolean

**Syntax**

```
setCurMapFile (mapFilename)
```

**Parameters**

Parameter name	Data type	Optional	Description
mapFilename	String	No	Name of the map file

## HTMLHelpWindowManager

Class to create HTMLHelp windows and manipulate window settings.

**HTMLHelpWindowManager properties**

Property name	Data type	Description
parent	<a href="#">Project</a>	Read-only Project object
typename	String	Read-only Class name of the reference object
count	Integer	Read-only Number of HTMLHelp windows present in the project

## HTMLHelpWindowManager methods

### item

#### Description

Returns the HTMLHelp window specified by the index.

#### Returns

[HTMLHelpWindow](#)

#### Syntax

```
item (index)
```

#### Parameters

Parameter name	Data type	Optional	Description
index	Integer	No	Index for the window

### newItem

#### Description

Creates an HTMLHelp window.

#### Returns

[HTMLHelpWindow](#)

#### Syntax

```
newItem (windowName)
```

#### Parameters

Parameter name	Data type	Optional	Description
windowName	String	No	Name of the window

## File

Specifies basic file functions in the project.

## File properties

Property name	Data type	Description
parent	<a href="#">FileManager</a>	Read-only Parent FileManager object
name	String	Read-only Name of the file
typename	String	Read-only Class name of the reference object
previous	<a href="#">File</a>	Read-only Returns the previous file object
next	<a href="#">File</a>	Read-only Returns the next file object
fileStatus	<a href="#">FileStatus</a>	Read-only Status of the file
valid	Boolean	Read-only Returns true if the file is present physically
extension	String	Read-only File extension
location	String	Read-only Folder location path where the file is present
path	String	Read-only Absolute file path
size	Integer	Read-only File size in bytes
creationDate	String	Read-only Creation date of the file in (mm-dd-yyyy) format
modifyDate	String	Read-only Last modified date of the file in (mm-dd-yyyy) format

## File methods

### checkout

#### Description

Checks out the specified file from the version control system.

#### Returns

Boolean

**Syntax**`checkout ()`**checkin****Description**

Checks in the specified file to the version control system.

**Returns**

Boolean

**Syntax**`checkin ()`**getLatestVersion****Description**

Gets the latest version of the specified file from the version control system.

**Returns**

Boolean

**Syntax**`getLatestVersion ()`**undoCheckout****Description**

Cancels the checkout of the specified file from the version control system.

**Returns**

Boolean

**Syntax**`undoCheckout ()`**Skin**

Handles specific skin functions in the project.

## Skin properties

Property name	Data type	Description
parent	<a href="#">SkinManager</a>	Read-only Parent SkinManager object
name	String	Read-write Name of the skin
typename	String	Read-only Class name of the reference object
previous	<a href="#">Skin</a>	Read-only Returns the previous skin object
next	<a href="#">Skin</a>	Read-only Returns the next skin object
skinType	<a href="#">SkinType</a>	Read-only Skin type
canDelete	Boolean	Read-only Indicates whether you can delete a skin file or not
fileStatus	<a href="#">FileStatus</a>	Read-only Status of the file
valid	Boolean	Read-only Returns true if the file is present physically
extension	String	Read-only File extension
location	String	Read-only Folder location path where the file is present
path	String	Read-only Absolute file path
size	Integer	Read-only File size in bytes
creationDate	String	Read-only Creation date of the file in (mm-dd-yyyy) format
modifyDate	String	Read-only Last modified date of the file in (mm-dd-yyyy) format

## Skin methods

### checkout

#### Description

Checks out the specified file from the version control system.

#### Returns

Boolean

#### Syntax

```
checkout ()
```

### checkin

#### Description

Checks in the specified file to the version control system.

#### Returns

Boolean

#### Syntax

```
checkin ()
```

### getLatestVersion

#### Description

Gets the latest version of the specified file from the version control system.

#### Returns

Boolean

#### Syntax

```
getLatestVersion ()
```

### undoCheckout

#### Description

Cancels the checkout of the specified file from the version control system.

#### Returns

Boolean

#### Syntax

```
undoCheckout ()
```

## delete

### Description

Deletes the specified skin.

### Returns

Boolean

## exportFile

### Description

Exports the specified skin.

### Returns

Boolean

### Syntax

```
exportFile (newFileLocation)
```

### Parameters

Parameter name	Data type	Optional	Description
newFileLocation	String	No	Full path to the new skin file

## CBT

Handles specific CBT functions in the project.

### CBT properties

Property name	Data type	Description
parent	<a href="#">CBTManager</a>	Read-only Parent CBTManager object
name	String	Read-write Name of the CBT
typename	String	Read-only Class name of the reference object
previous	<a href="#">CBT</a>	Read-only Returns the previous CBT object
next	<a href="#">CBT</a>	Read-only Returns the next CBT object

Property name	Data type	Description
red	Integer	Read-write Red color value
green	Integer	Read-write Green color value
blue	Integer	Read-write Blue color value

## CBT methods

### delete

#### Description

Deletes the specified CBT.

#### Returns

Boolean

#### Syntax

```
delete ()
```

## UDV

Handles specific UDV functions in the project.

## UDV properties

Property name	Data type	Description
parent	<a href="#">UDVManager</a>	Read-only Parent UDVManager object
name	String	Read-write Name of UDV
value	String	Read-write Value of UDV
typename	String	Read-only Class name of the reference object
previous	<a href="#">UDV</a>	Read-only Returns the previous UDV object

Property name	Data type	Description
next	UDV	Read-only Returns the next UDV object
htmlText	String	Read-only HTML Text tag, which can be inserted in any html file
usedIn	String	Read-only Returns the list of topics in which the UDV is referenced

## UDV methods

### delete

#### Description

Deletes the specified CBT.

#### Returns

Boolean

#### Syntax

```
delete ()
```

## SSL

Handles specific SSL functions in the project.

### SSL properties

Property name	Data Type	Description
parent	SSLManager	Read-only Parent SSLManager object
name	String	Read-write Name of SSL
typename	String	Read-only Class name of the reference object
previous	SSL	Read-only Returns the previous SSL object
next	SSL	Read-only Returns the next SSL object

Property name	Data Type	Description
layoutType	<a href="#">SSLLayoutType</a>	Read-only Returns the layout type of the SSL object
canPublish	Boolean	Read-only Returns true if you can publish the specific SSL
defaultLayout	Boolean	Read-write Specifies whether the SSL is the default layout or not
Section508	Boolean	Read-write Specifies section 508 settings
W3Ccompliance	Boolean	Read-write Specifies W3C compliance settings
addMarkOfWeb	Boolean	Read-write Specifies the setting for adding Mark of the Web
addBreadcrumb	Boolean	Read-write Specifies the setting for adding breadcrumbs
aboutBox	Boolean	Read-write Enables the About box
highlightSearch	Boolean	Read-write Enables highlight search
outputFileName	String	Read-write Specifies the output filename
defaultTopic	<a href="#">Topic</a>	Read-write Specifies the default topic
TOC	String	Read-write Specifies the TOC for SSL
index	String	Read-write Specifies the index for SSL
glossary	String	Read-write Specifies the glossary for SSL
window	String	Read-write Specifies the window for SSL
skin	<a href="#">Skin</a>	Read-write Specifies the skin for SSL

Property name	Data Type	Description
CBT	String	Read-write Specifies the CBT for SSL
server	String	Read-write Specifies the server list in SSL, server names with \n separation
serverProjectName	String	Read-write Allows read/write to the project name setting in the SSL. This setting is used when publishing to the RoboHelp server.

## SSL methods

### delete

#### Description

Deletes the specified SSL.

#### Returns

Boolean

#### Syntax

```
delete ()
```

### createDuplicate

#### Description

Creates a duplicate SSL from the specified SSL.

#### Returns

[SSL](#)

#### Syntax

```
createDuplicate (newName)
```

#### Parameters

Parameter name	Data type	Optional	Description
newName	String	No	Name of new SSL

### generate

#### Description

Generates the specified SSL.

**Returns**

Boolean

**Syntax**

```
generate ()
```

**view****Description**

Opens the output of the specified SSL for viewing.

**Returns**

Boolean

**Syntax**

```
view ()
```

**publish**

Generates the specified SSL and publishes the output to the set server

**Returns:** Boolean

**clearAllOverrides****Description**

Clears overridden master pages or CSSs.

**Returns**

Boolean

**Syntax**

```
clearAllOverrides ()
```

**applyMasterPageToAllTopics****Description**

Applies the specified master page to all topics during SSL generation.

**Returns**

Boolean

**Syntax**

```
applyMasterPageToAllTopics (MasterPage)
```

**Parameters**

Parameter name	Data type	Optional	Description
MasterPage	String	No	Name of the master page to apply. A master page object itself can also be passed.

**applyCSSToAllTopics****Description**

Path of the CSS to apply. The path is relative to the project.

**Returns**

Boolean

**Syntax**

```
applyCSSToAllTopics (css)
```

**Parameters**

Parameter name	Data type	Optional	Description
css	String	No	Filename of CSS

# Snippet

Handles specific snippet functions in the project.

**Snippet properties**

Property name	Data type	Description
parent	<a href="#">SnippetManager</a>	Read-only Parent SnippetManager object
name	String	Read-write Name of the snippet
typename	String	Read-only Class name of the reference object
previous	<a href="#">Snippet</a>	Read-only Returns the previous snippet object
next	<a href="#">Snippet</a>	Read-only Returns the next snippet object
htmlText	String	Read-only HTML Text tag, which can be inserted in any HTML file

Property name	Data type	Description
fileStatus	FileStatus	Read-only Status of the file
valid	Boolean	Read-only Returns true if the file is present physically
extension	String	Read-only File extension
location	String	Read-only Path to the folder in which the file is present
path	String	Read-only Absolute file path
size	Integer	Read-only File size in bytes
creationDate	String	Read-only Creation date of the file in (mm-dd-yyyy) format
modifyDate	String	Read-only Last modified date of the file in (mm-dd-yyyy) format

## Snippet methods

### checkout

#### Description

Checks out the specified file from the version control system.

#### Returns

Boolean

#### Syntax

```
checkout ()
```

### checkin

#### Description

Checks in the specified file to the version control system.

#### Returns

Boolean

**Syntax**

```
checkin ()
```

**getLatestVersion****Description**

Gets the latest version of the specified file from the version control system.

**Returns**

Boolean

**Syntax**

```
getLatestVersion ()
```

**undoCheckout****Description**

Cancels the checkout of the specified file from the version control system.

**Returns**

Boolean

**Syntax**

```
undoCheckout ()
```

**delete****Description**

Deletes the specified snippet.

**Returns**

Boolean

**Syntax**

```
delete (deleteOption)
```

**Parameters**

Parameter name	Data type	Optional	Description
deleteOption	DeleteOption	No	Specifies the way to delete the snippet

**exportFile****Description**

Exports the specified snippet.

**Returns**

Boolean

**Syntax**`exportFile (newFileLocation)`**Parameters**

Parameter name	Data type	Optional	Description
<code>newFileLocation</code>	String	No	Full path to the new snippet file

**createDuplicate****Description**

Creates a duplicate snippet from the specified snippet.

**Returns**[Snippet](#)**Syntax**`createDuplicate ()`

# Topic

Handles specific topic functions in the project.

**Topic properties**

Property name	Data type	Description
<code>parent</code>	<a href="#">TopicManager</a>	Read-only Parent TopicManager object
<code>name</code>	String	Read-write Title of the topic file
<code>filename</code>	String	Read-only Topic file name
<code>typename</code>	String	Read-only Class name of the reference object
<code>previous</code>	<a href="#">Topic</a>	Read-only Returns the previous topic object
<code>next</code>	<a href="#">Topic</a>	Read-only Returns the next topic object

Property name	Data type	Description
language	Language	Read-only Language of the specified topic
status	Status	Read-only Returns the status of the topic file
CBT	String	Read-write List of CBTs applied on the topic
fileStatus	FileStatus	Read-only Status of the file
valid	Boolean	Read-only Returns true if the file is present physically
extension	String	Read-only File extension
location	String	Read-only Path to the folder in which the file is present
path	String	Read-only Absolute file path
size	Integer	Read-only File size in bytes
creationDate	String	Read-only Creation date of the file in (mm-dd-yyyy) format
modifyDate	String	Read-only Last modified date of the file in (mm-dd-yyyy) format

## Topic methods

### checkout

#### Description

Checks out the specified file from the version control system.

#### Returns

Boolean

#### Syntax

```
checkout ( )
```

## **checkin**

### **Description**

Checks in the specified file to the version control system.

### **Returns**

Boolean

### **Syntax**

```
checkin ()
```

## **getLatestVersion**

### **Description**

Gets the latest version of the specified file from the version control system.

### **Returns**

Boolean

### **Syntax**

```
getLatestVersion ()
```

## **undoCheckout**

### **Description**

Cancels the checkout of the specified file from the version control system.

### **Returns**

Boolean

### **Syntax**

```
undoCheckout ()
```

## **delete**

### **Description**

Deletes the specified topic.

### **Returns**

Boolean

### **Syntax**

```
delete ()
```

# MasterPage

Handles specific master page functions in the project.

## MasterPage properties

Property name	Data type	Description
parent	<a href="#">MasterPageManager</a>	Read-only Parent MasterPageManager object
name	String	Read-only Title of the master page
filename	String	Read-only Master page file name
typename	String	Read-only Class name of the reference object
previous	<a href="#">MasterPage</a>	Read-only Returns the previous master page object
next	<a href="#">MasterPage</a>	Read-only Returns the next master page object
language	<a href="#">Language</a>	Read-only Language of the specified master page
status	<a href="#">Status</a>	Read-only Return status of the master page file
CBT	String	Read-only List of CBTs applied on the master page
fileStatus	<a href="#">FileStatus</a>	Read-only Status of the file
valid	Boolean	Read-only Returns true if the file is present physically
extension	String	Read-only File extension
location	String	Read-only Path to the folder in which the file is present
path	String	Read-only Absolute file path

Property name	Data type	Description
size	Integer	Read-only File size in bytes
creationDate	String	Read-only Creation date of the file in (mm-dd-yyyy) format
modifyDate	String	Read-only Last modified date of the file in (mm-dd-yyyy) format

## MasterPage methods

### checkout

#### Description

Checks out the specified file from the version control system.

#### Returns

Boolean

#### Syntax

```
checkout ()
```

### checkin

#### Description

Checks in the specified file to the version control system.

#### Returns

Boolean

#### Syntax

```
checkin ()
```

### getLatestVersion

#### Description

Gets the latest version of the specified file from the version control system.

#### Returns

Boolean

#### Syntax

```
getLatestVersion ()
```

## undoCheckout

### Description

Cancels the checkout of the specified file from the version control system.

### Returns

Boolean

### Syntax

```
undoCheckout ()
```

## delete

### Description

Deletes the specified master page.

### Returns

Boolean

### Syntax

```
delete ()
```

## rename

### Description

Renames a master page in the current project.

### Returns

Boolean

### Syntax

```
rename (filename)
```

### Parameters

Parameter name	Data type	Optional	Description
filename	String	No	New filename without the extension and the path

## applyMasterPageToTopics

### Description

Applies the master page to all the topics contained in an input array.

### Returns

Boolean

## Syntax

```
applyMasterPageToTopics (arrayOfTopics)
```

## Parameters

Parameter name	Data type	Optional	Description
arrayOfTopics	Array	No	A JavaScript array. Each element of this array must be a topic object.

# TOC

Handles specific TOC functions in the project.

## TOC properties

Property name	Data type	Description
parent	<a href="#">TOCManager</a>	Read-only Parent TOCManager object
name	String	Read-write Name of the TOC
typename	String	Read-only Class name of the reference object
previous	<a href="#">TOC</a>	Read-only Returns the previous TOC object
next	<a href="#">TOC</a>	Read-only Returns the next TOC object
fileStatus	<a href="#">FileStatus</a>	Read-only Status of the file
valid	Boolean	Read-only Returns true if the file is present physically
extension	String	Read-only File extension
location	String	Read-only Path to the folder in which the file is present
path	String	Read-only Absolute file path

Property name	Data type	Description
size	Integer	Read-only File size in bytes
creationDate	String	Read-only Creation date of the file in (mm-dd-yyyy) format
modifyDate	String	Read-only Last modified date of the file in (mm-dd-yyyy) format

## TOC methods

### checkin

#### Description

Checks in the specified file to the version control system.

#### Returns

Boolean

#### Syntax

```
checkin ()
```

### checkout

#### Description

Checks out the specified file from the version control system.

#### Returns

Boolean

#### Syntax

```
checkout ()
```

### getLatestVersion

#### Description

Gets the latest version of the specified file from the version control system.

#### Returns

Boolean

#### Syntax

```
getLatestVersion ()
```

## undoCheckout

### Description

Cancels the checkout of the specified file from the version control system.

### Returns

Boolean

### Syntax

```
undoCheckout ()
```

## delete

### Description

Deletes the specified TOC.

### Returns

Boolean

### Syntax

```
delete ()
```

## autoCreateTOC

### Description

Auto-creates the TOC for the specified TOC object.

### Returns

Boolean

### Syntax

```
autoCreateTOC (deleteCurrentTOC, bookMark)
```

### Parameters

Parameter name	Data type	Optional	Description
deleteCurrentTOC	Boolean	No	Flag to delete the current TOC content during auto-generation
bookMark	Boolean	No	Flag to set the bookmark flag for the current TOC

## rename

### Description

Renames the TOC.

**Returns**

Boolean

**Syntax**`rename (newName)`**Parameters**

Parameter name	Data type	Optional	Description
<code>newName</code>	String	No	New name for the TOC

**setAsDefault****Description**

Sets the TOC object as the default TOC of the project.

**Returns**

Boolean

**Syntax**`setAsDefault ()`

# Index

Handles specific index functions in the project.

**Index properties**

Property name	Data type	Description
<code>parent</code>	<a href="#">IndexManager</a>	Read-only Parent IndexManager object
<code>name</code>	String	Read-write Name of the index
<code>typename</code>	String	Read-only Class name of the reference object
<code>previous</code>	<a href="#">Index</a>	Read-only Returns the previous index object
<code>next</code>	<a href="#">Index</a>	Read-only Returns the next index object
<code>fileStatus</code>	<a href="#">FileStatus</a>	Read-only Status of the file

Property name	Data type	Description
valid	Boolean	Read-only Returns true if the file is present physically
extension	String	Read-only File extension
location	String	Read-only Path to the folder in which the file is present
path	String	Read-only Absolute file path
size	Integer	Read-only File size in bytes
creationDate	String	Read-only Creation date of the file in (mm-dd-yyyy) format
modifyDate	String	Read-only Last modified date of the file in (mm-dd-yyyy) format

## Index methods

### checkout

#### Description

Checks out the specified file from the version control system.

#### Returns

Boolean

#### Syntax

```
checkout ( )
```

### checkin

#### Description

Checks in the specified file to the version control system.

#### Returns

Boolean

#### Syntax

```
checkin ( )
```

## **getLatestVersion**

### **Description**

Gets the latest version of the specified file from the version control system.

### **Returns**

Boolean

### **Syntax**

```
getLatestVersion ()
```

## **undoCheckout**

### **Description**

Cancels the checkout of the specified file from the version control system.

### **Returns**

Boolean

### **Syntax**

```
undoCheckout ()
```

## **delete**

### **Description**

Deletes the specified index.

### **Returns**

Boolean

### **Syntax**

```
delete ()
```

## **rename**

### **Description**

Renames the index.

### **Returns**

Boolean

### **Syntax**

```
rename (newName)
```

**Parameters**

Parameter name	Data type	Optional	Description
newName	String	No	New name for the index

**setAsDefault****Description**

Sets the index object as the default index for the project.

**Returns**

Boolean

**Syntax**

```
setAsDefault ()
```

## HTMLHelpWindow

Gets or configures HTMLHelp window settings.

**HTMLHelpWindow properties**

Property name	Data type	Description
parent	<a href="#">HTMLHelpWindowManager</a>	Read-only Parent HTMLHelpWindowManager object
name	String	Read-write Name of the window
typename	String	Read-only Class name of the reference object
previous	<a href="#">HTMLHelpWindow</a>	Read-only Returns the previous HTMLHelpWindow object
next	<a href="#">HTMLHelpWindow</a>	Read-only Returns the next HTMLHelpWindow object
windowCaption	String	Read-write Caption of the window
navPaneWidth	Integer	Read-write Navigation pane width
tabPosition	<a href="#">HTMLHelpWindowAlign</a>	Read-write Tripane Tab position

Property name	Data type	Description
defaultTab	<a href="#">HTMLHelpWindowDefaultTab</a>	Read-write Tripane default Tab
buttons	<a href="#">HTMLHelpWindowButtons</a>	Read-write Flags to show different buttons in the HTMLHelp window
windowProperties	<a href="#">HTMLHelpWindowProperties</a>	Read-write Used to set properites of the HTMLHelp window
windowStyles	Integer	Read-write Window styles
windowExtendedStyles	Integer	Read-write Window extended styles
defaultFileName	String	Read-write Advanced property "Default"
homeFileName	String	Read-write Advanced property "Home"
tocFileName	String	Read-write Advanced property "Table of Contents"
indexFileName	String	Read-write Advanced property "Index"
button1Label	String	Read-write Advanced property "Button 1 Label"
button1Url	String	Read-write Advanced property "Custom Button 1 URL"
button2Label	String	Read-write Advanced property "Button 2 Label"
button2Url	String	Read-write Advanced property "Custom Button 2 URL"

## HTMLHelpWindow methods

### setWindowPosition

#### Description

Sets window placement properties.

#### Returns

Boolean

## Syntax

`setWindowPosition (top, left, width, height)`

## Parameters

Parameter name	Data type	Optional	Description
top	Integer	No	Top position
left	Integer	No	Left position
width	Integer	No	Width of the window
height	Integer	No	Height of the window

# Glossary

Handles specific glossary functions in the project.

## Glossary properties

Property name	Data type	Description
parent	<a href="#">GlossaryManager</a>	Read-only Parent GlossaryManager object
name	String	Read-write Name of the glossary
typename	String	Read-only Class name of the reference object
previous	<a href="#">Glossary</a>	Read-only Returns the previous glossary object
next	<a href="#">Glossary</a>	Read-only Returns next Glossary object
fileStatus	<a href="#">FileStatus</a>	Read-only Status of the file
valid	Boolean	Read-only Returns true if the file is present physically
extension	String	Read-only File extension
location	String	Read-only Path to the folder in which the file is present
path	String	Read-only Absolute file path

Property name	Data type	Description
size	Integer	Read-only File size in bytes
creationDate	String	Read-only Creation date of the file in (mm-dd-yyyy) format
modifyDate	String	Read-only Last modified date of file in (mm-dd-yyyy) format

## Glossary methods

### checkout

#### Description

Checks out the specified file from the version control system.

#### Returns

Boolean

#### Syntax

```
checkout ()
```

### checkin

#### Description

Checks in the specified file to the version control system.

#### Returns

Boolean

#### Syntax

```
checkin ()
```

### getLatestVersion

#### Description

Gets the latest version of the specified file from the version control system.

#### Returns

Boolean

#### Syntax

```
getLatestVersion ()
```

## undoCheckout

### Description

Cancels the checkout of the specified file from the version control system.

### Returns

Boolean

### Syntax

```
undoCheckout ()
```

## delete

### Description

Deletes the specified glossary.

### Returns

Boolean

### Syntax

```
delete ()
```

## newItem

### Description

Adds a new glossary term to the object.

### Returns

Boolean

### Syntax

```
newItem (name, value, overWriteFlag)
```

### Parameters

Parameter name	Data type	Optional	Description
name	String	No	Glossary term name
value	String	No	Glossary term value
overWriteFlag	Boolean	No	Flag to overwrite the existing term value The default value of the flag is false.

## rename

### Description

Renames the glossary.

**Returns**

Boolean

**Syntax**`rename (newName)`**Parameters**

Parameter name	Data type	Optional	Description
<code>newName</code>	String	No	New name for the glossary

**setAsDefault****Description**

Sets the glossary object as the default glossary for the project.

**Returns**

Boolean

**Syntax**`setAsDefault ()`

## FMFile

Handles specific FrameMaker file functions in the project.

**FMFile properties**

Property name	Data type	Description
<code>parent</code>	<a href="#">FMFileManager</a>	Read-only Parent FMFileManager object
<code>name</code>	String	Read-only Name of the FrameMaker file
<code>typename</code>	String	Read-only Class name of the reference object
<code>previous</code>	<a href="#">FMFile</a>	Read-only Returns the previous FMFile object
<code>next</code>	<a href="#">FMFile</a>	Read-only Returns the next FMFile object
<code>frameFileFormat</code>	<a href="#">FrameFileFormat</a>	Read-only FrameMaker file format type (read/write)

Property name	Data type	Description
frameTocFilePath	String	Read-write FrameMaker TOC file path (read/write)
associatedTocName	String	Read-write RoboHelp associated TOC name (read/write)
associatedIndexName	String	Read-write RoboHelp associated index name (read/write)
associatedGlossaryName	String	Read-write RoboHelp associated glossary name (read/write)
syncStatus	<a href="#">SyncStatus</a>	Read-only Synchronization status of the FrameMaker file
fileStatus	<a href="#">FileStatus</a>	Read-only Status of the file
valid	Boolean	Read-only Returns true if the file is present physically
extension	String	Read-only File extension
location	String	Read-only Path to the folder in which the file is present
path	String	Read-only Absolute file path
size	Integer	Read-only File size in bytes
creationDate	String	Read-only Creation date of the file in (mm-dd-yyyy) format
modifyDate	String	Read-only Last modified date of the file in (mm-dd-yyyy) format

## FMFile methods

### checkout

#### Description

Checks out the specified file from the version control system.

#### Returns

Boolean

**Syntax**`checkout ()`**checkin****Description**

Checks in the specified file to the version control system.

**Returns**

Boolean

**Syntax**`checkin ()`**getLatestVersion****Description**

Gets the latest version of the specified file from the version control system.

**Returns**

Boolean

**Syntax**`getLatestVersion ()`**undoCheckout****Description**

Cancels the checkout of the specified file from the version control system.

**Returns**

Boolean

**Syntax**`undoCheckout ()`**delete****Description**

Removes the specified FrameMaker file from the project.

**Returns**

Boolean

**Syntax**`delete ()`

## update

### Description

Updates the FrameMaker file and generates specific topics in the RoboHelp project.

### Returns

Boolean

### Syntax

```
update (flagForceUpdate)
```

### Parameters

Parameter name	Data type	Optional	Description
flagForceUpdate	Boolean	No	Flag to force the update of the FrameMaker file

## WordFile

Handles specific Word file functions in the project.

### WordFile properties

Property name	Data type	Description
parent	<a href="#">WordFileManager</a>	Read-only Parent WordFileManager object
name	String	Read-only Name of the Word file
typename	String	Read-only Class name of the reference object
previous	<a href="#">WordFile</a>	Read-only Returns the previous WordFile object
next	<a href="#">WordFile</a>	Read-only Returns the next WordFile object
wordFileFormat	<a href="#">WordFileFormat</a>	Read-only Word file format type
associatedTocName	String	Read-write RoboHelp associated TOC name (read/write)
associatedIndexName	String	Read-write RoboHelp associated index name (read/write)

Property name	Data type	Description
associatedGlossaryName	String	Read-write RoboHelp associated glossary name (read/write)
syncStatus	<a href="#">SyncStatus</a>	Read-only Synchronization status of the Word file
fileStatus	<a href="#">FileStatus</a>	Read-only Status of the file
valid	Boolean	Read-only Returns true if the file is present physically
extension	String	Read-only File extension
location	String	Read-only Path to the folder in which the file is present
path	String	Read-only Absolute file path
size	Integer	Read-only File size in bytes
creationDate	String	Read-only Creation date of the file in (mm-dd-yyyy) format
modifyDate	String	Read-only Last modified date of the file in (mm-dd-yyyy) format

## WordFile methods

### checkout

#### Description

Checks out the specified file from the version control system.

#### Returns

Boolean

#### Syntax

```
checkout ( )
```

### checkin

#### Description

Checks in the specified file to the version control system.

**Returns**

Boolean

**Syntax**

```
checkin ()
```

**getLatestVersion****Description**

Gets the latest version of the specified file from the version control system.

**Returns**

Boolean

**Syntax**

```
getLatestVersion ()
```

**undoCheckout****Description**

Cancels the checkout of the specified file from the version control system.

**Returns**

Boolean

**Syntax**

```
undoCheckout ()
```

**delete****Description**

Removes the specified Word file from the project.

**Returns**

Boolean

**Syntax**

```
delete ()
```

**update****Description**

Updates the Word file and generates specific topics in the RoboHelp project.

**Returns**

Boolean

**Syntax**

```
update (flagForceUpdate)
```

**Parameters**

Parameter name	Data type	Optional	Description
flagForceUpdate	Boolean	No	Flag to force the update of the Word file

# Chapter 3: Constants

## Syntax for using constants

A constant is a class with predefined values. The syntax for specifying a constant is:

```
RoboHelp.ConstantClass.constantname
```

For example:

```
RoboHelp.SSLLayoutType.AIRHELP
```

## SSLLayoutType

Provides constants for different single-source layouts in RoboHelp.

### SSLLayoutType properties

Property name	Data type
AIRHELP	Integer
FLASHHELP	Integer
FLASHHELPPRO	Integer
ORACLEHELP	Integer
HTMLHELP	Integer
JAVAHELP	Integer
PRINTEDDOC	Integer
WEBHELP	Integer
WEBHELPPRO	Integer
XMLOUTPUT	Integer

## Language

Provides constants for different languages in RoboHelp.

## Language properties

Property name	Data type
NONE	Integer
BRAZILIAN	Integer
BULGARIAN	Integer
CATALAN	Integer
CORATIAN	Integer
CZECH	Integer
DANISH	Integer
DUTCH	Integer
ENGLISHUK	Integer
ENGLISHUS	Integer
ESTONIAN	Integer
FINNISH	Integer
FRENCH	Integer
FRENCH_CANADA	Integer
GERMAN	Integer
GREEK	Integer
GREMAN_SWISS	Integer
HUNGARIAN	Integer
ITALIAN	Integer
JAPANESE	Integer
KOREAN	Integer
LATVIAN	Integer
LITHUANIAN	Integer
NORWEGIAN	Integer
NYNORSK	Integer
POLISH	Integer
PORTUGUESE	Integer
ROMANIAN	Integer
RUSSIAN	Integer
SIMPLIFIED_CHINESE	Integer
SLOVENIAN	Integer
SPANISH	Integer
SWEDISH	Integer

Property name	Data type
THAI	Integer
TRADITIONAL_CHINESE	Integer
TURKISH	Integer
VIETNAMESE	Integer

## ReportType

Provides constants for different report types in RoboHelp.

### ReportType properties

Property name	Data type
BROKENLINKS	Integer
CONDITIONALBUILDTAGS	Integer
DUPLICATEMAPIDS	Integer
EXTERNALTOPICS	Integer
GLOSSARY	Integer
IMAGES	Integer
INDEX	Integer
MAPIDS	Integer
MISSINGTOPICREFERENCES	Integer
PROJECTSTATUS	Integer
SEEALSO	Integer
SNIPPETS	Integer
STYLESHEETS	Integer
TABLEOFCONTENTS	Integer
TOPICPROPERTIES	Integer
TOPICREFERENCES	Integer
TOPICSBPRIORITY	Integer
UNREFERENCEDTOPICS	Integer
UNUSEDINDEXKEYWORDS	Integer
USEDFILES	Integer
VARIABLES	Integer

## DeleteOption

Provides constants for different delete options in RoboHelp.

### DeleteOption properties

Property name	Data type
DELETEANDCLEARREFERENCE	Integer
DELETEONLY	Integer
DELETEANDREPLACECONTENT	Integer

## Status

Provides constants for various topic status in RoboHelp.

### Status properties

Property name	Data type
INPROGRESS	Integer
READYFORREVIEW	Integer
COMPLETE	Integer

## SkinType

Provides constants for different skin types in RoboHelp.

### SkinType properties

Property name	Data type
SKINUNKNOWN	Integer
SKINWEBHELP	Integer
SKINFLASHHEP	Integer
SKINWEBFLASHHELP	Integer

## FrameFileFormat

Provides constants for different FrameMaker file formats in RoboHelp.

## FrameFileFormat properties

Property name	Data type
FRAMEMIFFORMAT	Integer
FRAMEFMFORMAT	Integer
FRAMEBOOKFORMAT	Integer
FRAMETOCFORMAT	Integer
FRAMEMIXEDFORMAT	Integer

## TokenType

Provides constants for different token types for token objects in RoboHelp.

### TokenType properties

Property name	Data type
TOKENTEXT	Integer
TOKENTAG	Integer

## TagType

Provides constants for different tag types when tokenType is TOKENTAG.

### TagType properties

Property name	Data type
TAGNONE	Integer
TAGIMAGE	Integer
TAGHTML	Integer
TAGHEAD	Integer
TAGBODY	Integer
TAGTITLE	Integer
TAGMETA	Integer
TAGLINK	Integer
TAGANCHOR	Integer
TAGPARAGRAPH	Integer
TAGTABLE	Integer

## FileStatus

Provides constants for different file statuses in RoboHelp.

### FileStatus properties

Property name	Data type
FILEUNKNOWN	Integer
FILECHECKEDIN	Integer
FILECHECKEDOUT	Integer
FILECHECKEDOUTEXCLUSIVE	Integer
FILECHECKOUTBYYOUANDOTHER	Integer
FILECHECKEDOUTTOANOTHERFOLDER	Integer
FILECHECKEDOUTBYANOTHER	Integer
FILEREMOVED	Integer
FILENOTCONTROLLED	Integer
FILEREADONLY	Integer
FILEWRITABLE	Integer
FILENOTEXIST	Integer
FILELOCKED	Integer
FILEOUTOFDATE	Integer
FILEMODIFIED	Integer
FILEMERGED	Integer
FILEPINNED	Integer

## SyncStatus

Provides constants for different synchronization statuses in the case of FrameMaker or Word files in RoboHelp.

### SyncStatus properties

Property name	Data type
STATUSNONE	Integer
STATUSSYNC	Integer
STATUSNEW	Integer
STATUSMISSING	Integer
STATUSINTERNALUNSYNCED	Integer

Property name	Data type
STATUSOUTDATEDFILE	Integer
STATUSMODIFIED	Integer
STATUSDELETED	Integer

## ServerType

Provides constants for different server items in RoboHelp.

### ServerType properties

Property name	Data type
FTP	Integer
HTTP	Integer
FRONTPAGE	Integer
FILESYSTEM	Integer
ROBOHELPSERVER	Integer

## WordFileFormat

Provides constants for different Word file formats in RoboHelp.

### WordFileFormat properties

Property name	Data type
FORMATDOCNONE	Integer
FORMATDOC	Integer
FORMATDOCX	Integer
FORMATDOCM	Integer
FORMATRTF	Integer

## HTMLHelpWindowAlign

Provides constants for the Window Tab position.

## HTMLHelpWindowAlign properties

Property name	Data type
ALIGN_TOP	Integer
ALIGN_LEFT	Integer
ALIGN_BOTTOM	Integer

## HTMLHelpWindowDefaultTab

Provides constants for the Window Tab position.

### HTMLHelpWindowDefaultTab properties

Property name	Data type
TAB_CONTENTS	Integer
TAB_INDEX	Integer
TAB_SEARCH	Integer
TAB_HISTORY	Integer
TAB_FAVORITES	Integer

## HTMLHelpWindowButtons

Provides constants for the Window Tab position.

### HTMLHelpWindowButtons properties

Property name	Data type
BTN_EXPAND	Integer
BTN_BACK	Integer
BTN_FORWARD	Integer
BTN_STOP	Integer
BTN_REFRESH	Integer
BTN_HOME	Integer
BTN_SYNC	Integer
BTN_OPTIONS	Integer
BTN_PRINT	Integer
BTN_JUMP1	Integer
BTN_JUMP2	Integer

Property name	Data type
BTN_INDEX	Integer
BTN_SEARCH	Integer
BTN_HISTORY	Integer
BTN_FAVORITES	Integer
BTN_BROWSE_FWD	Integer
BTN_BROWSE_BCK	Integer
BTN_NOTES	Integer
BTN_CONTENTS	Integer

## HTMLHelpWindowProperties

Provides constants for the Window Tab position.

### HTMLHelpWindowProperties properties

Property name	Data type
PROP_ONTOP	Integer
PROP_NOTITLEBAR	Integer
PROP_NODEF_STYLES	Integer
PROP_NODEF_EXSTYLES	Integer
PROP_TRI_PANE	Integer
PROP_NOTB_TEXT	Integer
PROP_POST_QUIT	Integer
PROP_AUTO_SYNC	Integer
PROP_TRACKING	Integer
PROP_TAB_SEARCH	Integer
PROP_TAB_HISTORY	Integer
PROP_TAB_FAVORITES	Integer
PROP_CHANGE_TITLE	Integer
PROP_NAV_ONLY_WIN	Integer
PROP_NO_TOOLBAR	Integer
PROP_TAB_ADVSEARCH	Integer
PROP_TAB_AUTOHIDEDSHOW	Integer
PROP_TAB_GLOSSARY	Integer

Property name	Data type
PROP_EHELP	Integer
PROP_AVENUE	Integer
PROP_USER_POS	Integer

# Index

## A

applyCSSToAllTopics 56  
 applyMasterPageToAllTopics 55  
 applyMasterPageToTopics 64  
 assign 41  
 autoCreateBrowseSequence 13  
 autoCreateTOC 67

## B

bringToFront 8

## C

CBT 50  
 CBTManager 31  
 changeToFileSystem 40  
 changeToFrontPage 40  
 changeToFTP 39  
 changeToHTTP 39  
 checkinAll 26  
 checkoutAll 26  
 clearAllOverrides 55  
 closeAllWindows 8  
 closeProject 6  
 copy 36  
 createDuplicate 54, 59

## D

delete 19, 40, 42  
 DeleteOption 86

## E

edit 43  
 executeScript 8  
 exportFile 50, 58  
 exportSettings 10

## F

File 45  
 FileManager 25  
 FileStatus 88  
 FMFile 76  
 FMFileManager 33  
 FrameFileFormat 86

## G

generate 54  
 getAttribute 17  
 getCurrentProject 6  
 getLatestAll 26  
 getTokenManager 5  
 Glossary 73  
 GlossaryManager 29

## H

HTMLHelpWindow 71  
 HTMLHelpWindowAlign 89  
 HTMLHelpWindowButtons 90  
 HTMLHelpWindowDefaultTab 90  
 HTMLHelpWindowManager 44  
 HTMLHelpWindowProperties 91

## I

importDITAMap 12  
 importFile 23  
 importFrameMakerDoc 14  
 importHtmlFile 13  
 importItem 20, 22  
 importMapFile 43  
 importSettings 10  
 importWordDoc 14  
 Index 68  
 IndexManager 28  
 insertText 18  
 item 15, 19, 21, 23, 24, 27, 28, 29, 30, 31, 32, 33, 34, 37, 45

## L

Language 83

## M

mail 36  
 MapIdManager 41  
 MasterPage 62  
 MasterPageManager 21

## N

newItem 20, 21, 23, 24, 27, 28, 29, 31, 33, 34, 37, 45, 75  
 newMapFile 43  
 newMapId 41

newProject 6  
 newProjectImport 7

## O

openProject 5  
 outputMessage 11

## P

print 36  
 Project 9  
 publish 55

## Q

quit 8

## R

removeAttribute 18  
 rename 11, 64  
 ReportManager 35  
 ReportType 85

## S

save 16, 35  
 Server 38  
 ServerManager 37  
 ServerType 89  
 setAsDefault 68  
 setAttribute 18  
 setCurMapFile 44  
 setWindowPosition 72  
 Skin 47  
 SkinManager 30  
 SkinType 86  
 Snippet 56  
 SnippetManager 22  
 SSL 52  
 SSLLayoutType 83  
 SSLManager 32  
 Status 86  
 SyncStatus 88

## T

TagType 87  
 TOC 65  
 TOCManager 26

Token 16  
TokenManager 15  
TokenType 87  
Topic 59  
TopicManager 19

**U**

UDV 51  
UDVManager 24  
unassign 42  
update 17, 79  
updateAll 12

**V**

view 55

**W**

WordFile 79  
WordFileFormat 89  
WordFileManager 34