

ADOBE® ROBOHELP® 10

Scripting Guide



© 2012 Adobe Systems Incorporated. All rights reserved.

Adobe® RoboHelp® 10 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	5
Project	13
Token	20
TokenManager	24
Topic	26
TopicManager	28
MasterPageManager	30
SnippetManager	32
UDVManager	33
FileManager	35
TOCManager	36
IndexManager	37
Glossary	38
GlossaryManager	42
ScreenProfile	43
ScreenProfileManager	44
SkinManager	45
CBT	45
CBTManager	46
SSL	48
SSLManager	55
SSLContent	56
FMFileManager	61
WordFileManager	62
ReportManager	63
Server	65
ServerManager	68
MapIdManager	69
File	72
Skin	74
UDV	77
ContentManager	78
Snippet	79
MasterPage	82
TOC	86
Index	89
FMFile	92

Contents

WordFile	95
HTMLHelpWindow	97
HTMLHelpWindowManager	99
Chapter 3: Constants	
Syntax for using constants	101
AddMethod	101
SSLLayoutType	101
Language	102
ReportType	103
DeleteOption	104
Status	104
SkinType	104
FrameFileFormat	105
TokenType	105
TagType	105
PType	107
FileStatus	108
SyncStatus	109
ServerType	109
WordFileFormat	109
HTMLHelpWindowButtons	110
HTMLHelpWindowAlign	111
HTMLHelpWindowDefaultTab	111
HTMLHelpWindowProperties	111

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 5 and “[Constants](#)” on page 101 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

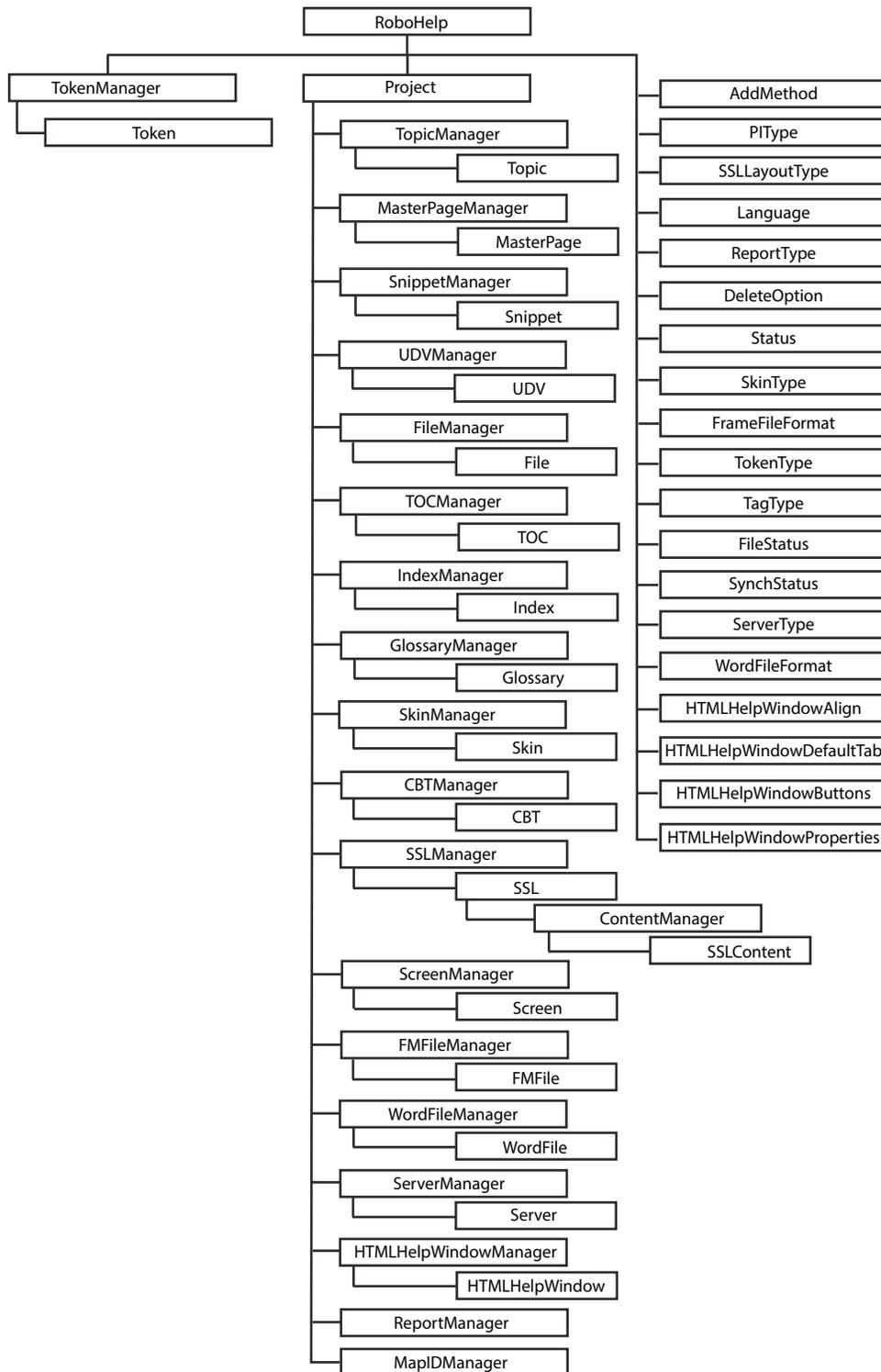
The RoboHelp object model is based on the principles of a document object model (DOM). A DOM is an application programming interface (API) that allows you to programmatically access various components of a document (as defined for that application) through a scripting language. The principle behind a 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
```

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



RoboHelp containment hierarchy

For more information on classes, see “[Classes](#)” on page 5.

For more information on constants, see “[Constants](#)” on page 101.

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
convertToHTMLOutput	Boolean	True if the Convert RoboHelp-edited Topics To HTML Option is selected. This option can be set using the scripting API.
typename	String	Class name of reference object
scriptingVersion	String	Scripting interface version
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
AddMethod	AddMethod	AddMethod constant
DeleteOption	DeleteOption	Read-only DeleteOption constant
Status	Status	Read-only Status constant
SkinType	SkinType	Read-only SkinType constant

Property name	Data type	Description
FrameFileFormat	FrameFileFormat	Read-only FrameFileFormat constant
TokenType	TokenType	Read-only TokenType constant
TagType	TagType	Read-only TagType constant
FileStatus	FileStatus	Read-only FileStatus constant
SyncStatus	SyncStatus	Read-only SyncStatus constant
ServerType	ServerType	Read-only ServerType constant
WordFileFormat	WordFileFormat	Read-only WordFileFormat constant
commandLineArgCount	Integer	Number of command-line arguments passed in the RoboHelp process

RoboHelp methods

getCommandLineArg

Description

Get the command line argument passed in RoboHelp process

Returns

String

Syntax

getCommandLineArg (argCount)

Parameters

Parameter name	Data Type	Optional	Description
argCount	int	No	Argument index in the command line

openProject

Description

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

Classes**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 older version of RoboHelp. Default value is false.If upgrade not required for the project, 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

readWholeFile**Description**

Read the whole file and return the file content. This function will take care of encoding issues for XML and html files as well

Returns

String

Syntax

```
readWholeFile (fullFilePath)
```

Parameters

Parameter name	Data Type	Optional	Description
fullFilePath	String	No	Full path of file which user wants to read

Classes**writeWholeFile****Description**

Creates a new file if required and write the content to this file. This function will take care of encoding issues for XML and html files. If the file is HTML or XML then it will check for encoding type and encode the file accordingly, otherwise it will write the content in UTF-8 and write BOM as well.

Returns

Boolean

Syntax

```
writeWholeFile(fullFilePath, stringBuffer)
```

Parameters

Parameter name	Data Type	Optional	Description
fullFilePath	String	No	Full path of file which user wants to read
stringBuffer	String	No	String Content which needs to be written to file

createZipFile**Description**

Creates a zipped file. This function requires a input folder as well as output zip file path

Returns

Boolean

Syntax

```
createZipFile(inputFolderPath, outputZipFilePath, mimeType)
```

Parameters

Parameter name	Data Type	Optional	Description
inputFolderPath	String	No	Full path of folder which need to be zipped
outputZipFilePath	String	No	Full path of output zip file.
mimeType	String	Yes	Mimetype for zip file. Empty Mimetype will not create any mimetype file.

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	Integer constant	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
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

executeCommandline

Description

Execute any commandline and wait for that process to complete

Returns

Boolean

Syntax

```
executeCommandline(commandLine, currentDirectory, outputLogFilePath, waitForProcess)
```

Parameters

Parameter name	Data Type	Optional	Description
commandLine	String	No	CommandLine string to execute
currentDirectory	String	No	Path to current directory where this process will be executed
outputLogFilePath	String	No	Log file path, which will contain all the stdout and error logs
waitForProcess	Boolean	No	Boolean to allow waiting for that process to complete execution

closeAllWindows

Description

Closes all active document windows in the RoboHelp application.

Syntax

```
closeAllWindows ()
```

registerEvent

Description

User can register any function in a script file along with an event. Function will be called each time the event is triggered in RoboHelp

Returns

Boolean

Syntax

```
registerEvent(eventName, scriptFilePath, functionName)
```

Parameters

Parameter name	Data Type	Optional	Description
eventName	String	No	Name of Event
scriptFilePath	String	No	Full path to the script file to execute
functionName	String	No	Name of function defined in the script file

unregisterEvent

Description

User can unregister any function in a script file along with an event

Returns

Boolean

Syntax

```
unregisterEvent(eventName, scriptFilePath, functionName)
```

Parameters

Parameter name	Data Type	Optional	Description
eventName	String	No	Name of Event
scriptFilePath	String	No	Full path to the script file to execute
functionName	String	No	Name of function defined in the script file

isEventRegistered

Description

Checks whether the given function in a script file along with an event is already registered

Returns

Boolean

Syntax

```
isEventRegistered(eventName, scriptFilePath, functionName)
```

Parameters

Parameter name	Data Type	Optional	Description
eventName	String	No	Name of Event
scriptFilePath	String	No	Full path to the script file to execute
functionName	String	No	Name of function defined in the script file

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	Language	Read-write Project language settings
TopicManager	TopicManager	Read-only TopicManager object
MasterPageManager	MasterPageManager	Read-only MasterPageManager object
SnippetManager	SnippetManager	Read-only SnippetManager object
UDVManager	UDVManager	Read-only UDVManager object
FileManager	FileManager	Read-only FileManager object
TOCManager	TOCManager	Read-only TOCManager object

Property name	Data type	Description
IndexManager	IndexManager	Read-only IndexManager object
GlossaryManager	GlossaryManager	Read-only GlossaryManager object
SkinManager	SkinManager	Read-only SkinManager object
CBTManager	CBTManager	Read-only CBTManager object
SSLManager	SSLManager	Read-only SSLManager object
FMFileManager	FMFileManager	Read-only FMFileManager object
WordFileManager	WorldFileManager	Read-only WordFileManager object
ReportManager	ReportManager	Read-only ReportManager object
ServerManager	ServerManager	Read-only ServerManager object
MapIdManager	MapIdManager	Read-only MapIdManager object
HTMLHelpWindowManager	HTMLHelpWindowManager	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

Classes**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

Classes

Parameter name	Data type	Optional	Description
bodyHeader	String	Yes	Full path to the file containing XHTML to place in the body running-header area of the output file
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 topics to XHTML
indexShow	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
TOC object	TOC	Yes	A TOC object on which the browse sequences will be based. If not specified, the default TOC of the project is used.
level	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
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

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

saveOutputViewLog

Description

Save Output View Pod logs to the given file

Returns

Boolean

Syntax

`saveOutputViewLog (logFilePath)`

Parameters

Parameter name	Data Type	Optional	Description
logFilePath	String	No	File path to save the logs

clearOutputViewLog

Description

Clear the Output View Pod logs

Returns

Boolean

Syntax

```
clearOutputViewLog()
```

compileCHMFile

Description

Compile and generate CHM file

Returns

Boolean

Syntax

```
compileCHMFile(HHPfilePath)
```

Parameters

Parameter name	Data Type	Optional	Description
HHPfilePath	String	No	Absolute path of the HHP File, This file is used for CHM file compilation

saveAll

Description

Save all the unsaved items in the project and return true on success

Returns

Boolean

Syntax

```
saveAll(forceSave)
```

Parameters

Parameter name	Data Type	Optional	Description
forceSave	Boolean	Yes	Flag to force save project

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

Token

Handles a specific token.

Token properties

Property name	Data type	Description
parent	TokenManager	Read-only Parent TokenManager object
name	String	Read-only Text from the token object
typename	String	Read-only Class name of the reference object
previous	Token	Read-only Returns the previous token object
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
attributeCount	int	In case of TOKENTAG, returns the number of attributes present in the token

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

isEndTagofType

Description

In case of TOKENTAG, it returns true if the token is and end tag

Returns

Boolean

Syntax

```
isEndTagOfType (startTagType)
```

Parameters

Parameter name	Data Type	Optional	Description
startTagType	TagType	No	

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

getAttributeName

Description

Returns the attribute name specified by the index

Returns

String

Syntax

```
getAttributeName (index)
```

Parameters

Parameter name	Data Type	Optional	Description
index	int	No	Index for the attribute Name (starts with 1)

hasAttribute

Description

In case of TOKENTAG, Returns true if the specific attribute is present in the token, otherwise false

Returns

Boolean

Syntax

```
hasAttribute(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 ()
```

TokenManager

Handles tokens for a specific file.

TokenManager properties

Property name	Data type	Description
parent	RoboHelp	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
index	Integer	No	Index for the token

save

Description

Saves the specified token file.

Returns

Boolean

Syntax

```
save ()
```

close

Description

Closes the specified token file and frees all the resources consumed by it.

Returns

Syntax

`close()`

Topic

Handles specific topic functions in the project.

Topic properties

Property name	Data type	Description
parent	TopicManager	Read-only Parent TopicManager object
name	String	Read-write Title of the topic file
filename	String	Read-only Topic file name
typename	String	Read-only Class name of the reference object
previous	Topic	Read-only Returns the previous topic object
next	Topic	Read-only Returns the next topic object
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

Property name	Data type	Description
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 ()
```

TopicManager

Handles all topics in the current project.

TopicManager properties

Property name	Data type	Description
parent	Project	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	Integer constant	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	Project	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	Project	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	Project	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	Project	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	Project	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	Project	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
existingIndexPath	String	Yes	Path of an existing index file, which is copied into the new index

Glossary

Handles specific glossary functions in the project.

Glossary properties

Property name	Data type	Description
parent	GlossaryManager	Read-only Parent GlossaryManager object
name	String	Read-write Name of the glossary
typename	String	Read-only Class name of the reference object
previous	Glossary	Read-only Returns the previous glossary object
next	Glossary	Read-only Returns next Glossary object
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 file in (mm-dd-yyyy) format

Glossary methods

checkout

Description

Checks out the specified file from the version control system.

Returns

Boolean

Classes**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
newName	String	No	New name for the glossary

setAsDefault

Description

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

Returns

Boolean

Syntax

```
setAsDefault ()
```

GlossaryManager

Handles all glossary files in the current project.

GlossaryManager properties

Property name	Data type	Description
parent	Project	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

ScreenProfile

Handles specific Screen Profile functions in the project.

ScreenProfile properties

Property name	Data Type	Description
parent	ScreenProfileManager	Parent ScreenProfileManager object
name	String	Name of Screen Profile
typename	String	Class name of reference object
previous	ScreenProfile	Returns previous ScreenProfile object
next	ScreenProfile	Returns next ScreenProfile object
minWidth	int	minimum Width
maxWidth	int	Maximum Width
minHeight	int	Minimum Height
maxHeight	int	Maximum Height
previewWidth	int	Preview Width
previewHeight	int	Preview Height
bgTop	int	Background Image Top position
bgLeft	int	Background Image Left position
browserAgent	String	Browser User agent
bglImage	String	Background image file path

ScreenProfile methods

delete

Description

Deletes the specified ScreenProfile

Returns

Boolean

Syntax

```
delete()
```

ScreenProfileManager

Handles all Screen Profiles in the current project.

ScreenProfileManager properties

Property name	Data Type	Description
parent	Project	Project object
typename	String	Class name of reference object
count	int	Number of Screen Profiles present in the project

ScreenProfileManager methods

item

Description

Returns the Screen Profile specified by the index

Returns

[ScreenProfile](#)

Syntax

```
item(index)
```

Parameters

Parameter name	Data Type	Optional	Description
index	int	No	Index for the Screen Profile

newItem

Description

Creates a new Screen Profile in the current project

Returns

[ScreenProfile](#)

Syntax

```
newItem(Name)
```

Parameters

Parameter name	Data Type	Optional	Description
Name	String	No	Name of new Screen Profile

SkinManager

Handles all skins in the current project.

SkinManager properties

Property name	Data type	Description
parent	Project	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

CBT

Handles specific CBT functions in the project.

CBT properties

Property name	Data type	Description
parent	CBTManager	Read-only Parent CBTManager object
name	String	Read-write Name of the CBT
typename	String	Read-only Class name of the reference object
previous	CBT	Read-only Returns the previous CBT object
next	CBT	Read-only Returns the next CBT object
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 ()
```

CBTManager

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

CBTManager properties

Property name	Data type	Description
parent	Project	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)

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
layoutType	SSLLayoutType	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
ContentManager	ContentManager	Returns the ContentManager object only for Web Help and AIR Help. For other SSLs, this property is undefined.
Section508	Boolean	Read-write Specifies section 508 settings

Property name	Data Type	Description
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
favicon	String	Set the favicon file path. Provide the absolute path to the favicon image(*.ico,*.png,*.gif).
helpTitle	String	Set the Help Title, such that it is displayed in the browser title bar/tab bar
defaultTopic	Topic	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	Skin	Read-write Specifies the skin for SSL
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.

Property name	Data Type	Description
language	String	<p>Set the language for the current SSL content.</p> <p>The supported values are:</p> <p>da_DK(Danish), de_DE(German), es_ES(Spanish), fr_FR(French), it_IT(Italian), nl_NL(Dutch), nb_NO(Norwegian-Bolkan), pt_BR(Brazilian), sv_SE(Swedish), pt_PT(Portuguese), ja_JP(Japanese), el_GR(Greek), fi_FI(Finnish), ca_ES(Catalan), ru_RU(Russian), bg_BG(Bulgarian), cs_CZ(Czech), pl_PL(Polish), ro_RO(Romanian), tr_TR(Turkish), hu_HU(Hungarian), hr_HR(Croatian), et_EE(Estonian), lv_LV(Latvian), lt_LT(Lithuanian), sl_SI(Slovenian), ko_KP(Korean), zh_TW(Chinese-Traditional), zh_CN(Chinese-Simplified), th_TH(Thai), vi_VN(Vietnamese), nn_NO(Norwegian), fr_CA(French-Canadian), de_CH(Swiss-German), en_US(English US), en_UK(English UK)</p>

Property name	Data Type	Description
encoding	String	<p>Set the encoding for the current SSL content.</p> <p>Supported values:</p> <p>asmo-708(Arabic(ASMO708)), big5(ChineseTraditional(Big5)), cp866(Cyrillic(DOS)), csiso2022jp(Japanese(JIS-Allow1byteKana)), dos-720(Arabic(DOS)), dos-862(Hebrew(DOS)), euc-kr(Korean(EUC)), gb2312(ChineseSimplified(GB2312)), hz-gb-2312(ChineseSimplified(HZ)), ibm437(OEMUnitedStates)), ibm737(Greek(DOS)), ibm775(Baltic(DOS)), ibm850(WesternEuropean(DOS)), ibm852(CentralEuropean(DOS)), ibm855(OEMCyrillic), ibm857(Turkish(DOS)), ibm860(Portuguese(DOS)), ibm861(Icelandic(DOS)), ibm863(FrenchCanadian(DOS)), ibm864(Arabic-864), ibm865(Nordic(DOS)), ibm869(Greek,Modern(DOS)), iso-2022-jp(Japanese(JIS)), iso-2022-kr(Korean(ISO)), iso-8859-1(WesternEuropean(ISO)), iso-8859-13(Estonian(ISO)), iso-8859-15(Latin9(ISO)), iso-8859-2(CentralEuropean(ISO)), iso-8859-4(Baltic(ISO)), iso-8859-5(Cyrillic(ISO)), iso-8859-6(Arabic(ISO)), iso-8859-7(Greek(ISO)), iso-8859-8(Hebrew(ISO-Visual)), iso-8859-8-i(Hebrew(ISO-Logical)), iso-8859-9(Turkish(ISO)), johab(Korean(Johab)), koi8-r(Cyrillic(KOI8-R)), koi8-u(Cyrillic(KOI8-U)), ks_c_5601-1987(Korean), macintosh(WesternEuropean(Mac)), shift_jis(Japanese(Shift-JIS)), us-ascii(US-ASCII), utf-16(Unicode), utf-8(Unicode(UTF-8)), windows-1250(CentralEuropean(Windows)), windows-1251(Cyrillic(Windows)), windows-1252(WesternEuropean(Windows)), windows-1253(Greek(Windows)), windows-1254(Turkish(Windows)), windows-1255(Hebrew(Windows)), windows-1256(Arabic(Windows)), windows-1257(Baltic(Windows)), windows-1258(Vietnamese(Windows)), windows-874(Thai(Windows)), x-chinese-cns(ChineseTraditional(CNS)), x-cp20261(T.61), x-cp20936(ChineseSimplified(GB2312-80)), x-cp20949(KoreanWansung), x-cp50227(ChineseSimplified(ISO-2022)), x-iscii-as(ISCIIAssamese), x-iscii-be(ISCII Bengali), x-iscii-de(ISCII Devanagari), x-iscii-gu(ISCII Gujarati), x-iscii-ka(ISCII Kannada)</p>

Property name	Data Type	Description
encoding(<i>continued...</i>)		x-iscii-ma(ISCII Malayalam), x-iscii-or(ISCII Oriya), x-iscii-pa(ISCII Punjabi), x-iscii-ta(ISCII Tamil), x-iscii-te(ISCII Telugu), x-mac-arabic(Arabic(Mac)), x-mac-ce(Central European(Mac)), x-mac-chinesesimp(Chinese Simplified(Mac)), x-mac-chinesetrad(Chinese Traditional(Mac)), x-mac-croatian(Croatian(Mac)), x-mac-cyrillic(Cyrillic(Mac)), x-mac-gr x-mac-cyrillic(Cyrillic(Mac)), x-mac-greek(Greek(Mac)), x-mac-hebrew(Hebrew(Mac)), x-mac-icelandic(Icelandic(Mac)), x-mac-japanese(Japanese(Mac)), x-mac-korean(Korean(Mac)), x-mac-romanian(Romanian(Mac)), x-mac-thai(Thai(Mac)), x-mac-turkish(Turkish(Mac)), x-mac-ukrainian(Ukrainian(Mac))
excludeUnrefTopics	Boolean	Set whether to include/exclude unreferenced topics from output
excludeUnrefBaggages	Boolean	Set whether to include/exclude unreferenced baggage files from search
enableExternalContentSearch	Boolean	Set setting Enable External Content Search
enableSubstringSearch	Boolean	Set setting Enable Substring Search
hideRankColumnInSearchResults	Boolean	Set setting Hide Rank Column in Search Results
showContextInSearchResults	Boolean	Set setting Show Context in Search Results
numberOfResultsOnAPage	int	Set setting Number of Results on a Page
highlightSearchResultColor	String	Set Highlight Search Result color, Please provide color in Hex value like #0000FF, #00FF00 or color constants like red, green

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

getSpecificProperty

Description

Get any specific String property from the SSL file.

Returns

String

Syntax

```
getSpecificProperty(propertyName, sectionName)
```

Parameters

Parameter name	Data Type	Optional	Description
propertyName	String	No	Name of the property to be saved
sectionName	String	Yes	Name of Section Where this property value resides

setSpecificProperty

Description

Set any specific String property in the SSL file.

Returns

Boolean

Syntax

```
setSpecificProperty(propertyName, propertyValue, sectionName)
```

Parameters

Parameter name	Data Type	Optional	Description
propertyName	String	No	Name of the property to be saved
propertyValue	String	No	Value of the property to be saved
sectionName	String	Yes	Name of Section Where this property value need to be saved

SSLManager

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

SSLManager properties

Property name	Data type	Description
parent	Project	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

SSLContent

Gets/sets SSL content properties.

SSLContent properties

Property name	Data Type	Description
parent	ContentManager	ContentManager object
name	String	Name of SSL Content
typename	String	Class name of reference object
defaultContent	Boolean	Set current content as default content
title	String	Set title of the current content
TOC	String	Set setting TOC for current SSL content
index	String	Set setting Index for current SSL content
glossary	String	Set setting Glossary for current SSL content
defaultTopic	Topic	Set setting default Topic for current SSL content
CBT	String	Set setting CBT for current SSL content
variablesSet	String	Set variables set for current SSL content

Property name	Data Type	Description
language	String	<p>Set setting language for current SSL content.</p> <p>Possible values are: da_DK(Danish), de_DE(German), es_ES(Spanish), fr_FR(French), it_IT(Italian), nl_NL(Dutch), nb_NO(Norwegian-Bolkan), pt_BR(Brazilian), sv_SE(Swedish), pt_PT(Portuguese), ja_JP(Japanese), el_GR(Greek), fi_FI(Finnish), ca_ES(Catalan), ru_RU(Russian), bg_BG(Bulgarian), cs_CZ(Czech), pl_PL(Polish), ro_RO(Romanian), tr_TR(Turkish), hu_HU(Hungarian), hr_HR(Croatian), et_EE(Estonian), lv_LV(Latvian), lt_LT(Lithuanian), sl_SI(Slovenian), ko_KP(Korean), zh_TW(Chinese-Traditional), zh_CN(Chinese-Simplified), th_TH(Thai), vi_VN(Vietnamese), nn_NO(Norwegian), fr_CA(French-Canadian), de_CH(Swiss-German), en_US(English US), en_UK(English UK)</p>

Property name	Data Type	Description
encoding	String	<p>Set encoding for current SSL content.</p> <p>Supported Values are:</p> <p>asmo-708(Arabic(ASMO708)), big5(ChineseTraditional(Big5)), cp866(Cyrillic(DOS)), csiso2022jp(Japanese(JIS-Allow1byteKana)), dos-720(Arabic(DOS)), dos-862(Hebrew(DOS)), euc-kr(Korean(EUC)), gb2312(ChineseSimplified(GB2312)), hz-gb-2312(ChineseSimplified(HZ)), ibm437(OEMUnitedStates), ibm737(Greek(DOS)), ibm775(Baltic(DOS)), ibm850(WesternEuropean(DOS)), ibm852(CentralEuropean(DOS)), ibm855(OEMCyrillic), ibm857(Turkish(DOS)), ibm860(Portuguese(DOS)), ibm861(Icelandic(DOS)), ibm863(FrenchCanadian(DOS)), ibm864(Arabic-864), ibm865(Nordic(DOS)), ibm869(Greek,Modern(DOS)), iso-2022-jp(Japanese(JIS)), iso-2022-kr(Korean(ISO)), iso-8859-1(WesternEuropean(ISO)), iso-8859-13(Estonian(ISO)), iso-8859-15(Latin9(ISO)), iso-8859-2(CentralEuropean(ISO)), iso-8859-4(Baltic(ISO)), iso-8859-5(Cyrillic(ISO)), iso-8859-6(Arabic(ISO)), iso-8859-7(Greek(ISO)), iso-8859-8(Hebrew(ISO-Visual)), iso-8859-8-i(Hebrew(ISO-Logical)), iso-8859-9(Turkish(ISO)), johab(Korean(Johab)), koi8-r(Cyrillic(KOI8-R)), koi8-u(Cyrillic(KOI8-U)), ks_c_5601-1987(Korean), macintosh(WesternEuropean(Mac)), shift_jis(Japanese(Shift-JIS)), us-ascii(US-ASCII), utf-16(Unicode), utf-8(Unicode(UTF-8)), windows-1250(CentralEuropean(Windows)), windows-1251(Cyrillic(Windows)), windows-1252(WesternEuropean(Windows)), windows-1253(Greek(Windows)), windows-1254(Turkish(Windows)), windows-1255(Hebrew(Windows)), windows-1256(Arabic(Windows)), windows-1257(Baltic(Windows)), windows-1258(Vietnamese(Windows)), windows-874(Thai(Windows)), x-chinese-cns(ChineseTraditional(CNS)), x-cp20261(T.61), x-cp20936(ChineseSimplified(GB2312-80)), x-cp20949(KoreanWansung), x-cp50227(ChineseSimplified(ISO-2022)), x-iscii-as(ISCIIAssamese), x-iscii-be(ISCII Bengali), x-iscii-de(ISCII Devanagari), x-iscii-gu(ISCII Gujarati), x-iscii-ka(ISCII Kannada), x-iscii-ma(ISCII Malayalam), x-iscii-or(ISCII Oriya), x-iscii-pa(ISCII Punjabi), x-iscii-ta(ISCII Tamil), x-iscii-te(ISCII Telugu), x-mac-arabic(Arabic(Mac)), x-mac-ce(CentralEuropean(Mac)), x-mac-chinesesimp(ChineseSimplified(Mac)), x-mac-chinesetrad(ChineseTraditional(Mac)), x-mac-croatian(Croatian(Mac)), x-mac-cyrillic(Cyrillic(Mac)), x-mac-greek(Greek(Mac)), x-mac-hebrew(Hebrew(Mac)), x-mac-icelandic(Icelandic(Mac)), x-mac-japanese(Japanese(Mac)), x-mac-korean(Korean(Mac)), x-mac-romanian(Romanian(Mac)), x-mac-thai(Thai(Mac)), x-mac-turkish(Turkish(Mac)), x-mac-ukrainian(Ukrainian(Mac))</p>

Property name	Data Type	Description
excludeUnrefTopics	Boolean	Set whether to include/exclude unreferenced topics from output
excludeUnrefBaggages	Boolean	Set whether to include/exclude unreferenced baggage files from search

SSLContent methods

selectMapFile

Description

Select/Deselect a map file for the current content

Returns

Boolean

Syntax

```
selectMapFile(MapFile, Select)
```

Parameters

Parameter name	Data Type	Optional	Description
MapFile	String	No	Name of Map file
Select	Boolean	No	Boolean value to select or deselect the map file given

selectBrs

Description

Select/Deselect a browse sequence for the current content

Returns

Boolean

Syntax

```
selectBrs(BrowseSequenceName, Select)
```

Parameters

Parameter name	Data Type	Optional	Description
BrowseSequenceName	String	No	Name of a browse sequence
Select	Boolean	No	Boolean value to select or deselect the browse sequence given

clearAllOverrides

Description

Clears overridden Master Page or CSS for the current content

Returns

Boolean

Syntax

```
clearAllOverrides()
```

applyMasterPageToAllTopics

Description

Applies the specified Master Page to all topics during SSL generation for the current content

Returns

Boolean

Syntax

```
applyMasterPageToAllTopics (MasterPage)
```

Parameters

Parameter name	Data Type	Optional	Description
MasterPage	String	No	Name of Master Page to be applied. A Master Page object itself can also be passed

applyCSSToAllTopics

Description

Path of CSS to be applied for the current content. Path should be relative to project.

Returns

Boolean

Syntax

```
applyCSSToAllTopics (css)
```

Parameters

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

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	Project	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
isDitaMap	Boolean	Yes	True if the FrameMakerFilePath is a DitaMap

WordFileManager

Handles all Word files in the current project.

WordFileManager properties

Property name	Data type	Description
parent	Project	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	Project	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	ReportType	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	ReportType	No	Report type
emailIDs	String	No	E-mail addresses, in semicolon-separated format

Server

Handles specific server items.

Server properties

Property name	Data type	Description
parent	ServerManager	Read-only Parent ServerManager object
name	String	Read-write Description of a server item
typename	String	Read-only Class name of the reference object
previous	Server	Read-only Returns the previous server object
next	Server	Read-only Returns the next server object
serverType	ServerType	Read-only Server type
hostName	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 ()
```

ServerManager

Handles all servers specified in the current project.

ServerManager properties

Property name	Data type	Description
parent	Project	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
serverDescription	String	No	Description of the new server item
serverType	ServerType	No	Type of server that you want to create

MapIdManager

Handles map ID manipulation in the current project.

MapIdManager properties

Property name	Data type	Description
parent	Project	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

File

Specifies basic file functions in the project.

File properties

Property name	Data type	Description
parent	FileManager	Read-only Parent FileManager object
name	String	Read-only Name of the file
typename	String	Read-only Class name of the reference object
previous	File	Read-only Returns the previous file object

Property name	Data type	Description
next	File	Read-only Returns the next file object
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 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	SkinManager	Read-only Parent SkinManager object
name	String	Read-write Name of the skin
typename	String	Read-only Class name of the reference object
previous	Skin	Read-only Returns the previous skin object
next	Skin	Read-only Returns the next skin object

Property name	Data type	Description
skinType	SkinType	Read-only Skin type
canDelete	Boolean	Read-only Indicates whether you can delete a skin file or not
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 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
<code>newFileLocation</code>	String	No	Full path to the new skin file

UDV

Handles specific UDV functions in the project.

UDV properties

Property name	Data type	Description
<code>parent</code>	UDVManager	Read-only Parent UDVManager object
<code>name</code>	String	Read-write Name of UDV
<code>value</code>	String	Read-write Value of UDV
<code>typename</code>	String	Read-only Class name of the reference object
<code>previous</code>	UDV	Read-only Returns the previous UDV object
<code>next</code>	UDV	Read-only Returns the next UDV object
<code>htmlText</code>	String	Read-only HTML Text tag, which can be inserted in any html file
<code>usedIn</code>	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 ()
```

ContentManager

Manages content categories in current SSL.

ContentManager properties

Property name	Data Type	Description
parent	SSL	SSL object
typename	String	Class name of reference object
count	int	Number of Content categories present in the SSL

ContentManager methods

item

Description

Returns the SSLContent object specified by the index

Returns

[SSLContent](#)

Syntax

```
item(index)
```

Parameters

Parameter name	Data Type	Optional	Description
index	int	No	Index for the SSLContent

newItem

Description

Creates a new SSLContent object with the given name.

Returns

[SSLContent](#)

Syntax

```
newItem(name)
```

Parameters

Parameter name	Data Type	Optional	Description
name	String	No	Name for the SSLContent

delete

Description

Deletes an existing SSLContent object with the given name. It does not delete a default content.

Returns

Boolean

Syntax

```
delete (name)
```

Parameters

Parameter name	Data Type	Optional	Description
name	String	No	Name for the SSLContent

save

Description

Saves all the changes in all the content categories. User must call this method to save any changes done.

Returns

Boolean

Syntax

```
save ()
```

Snippet

Handles specific snippet functions in the project.

Snippet properties

Property name	Data type	Description
parent	SnippetManager	Read-only Parent SnippetManager object
name	String	Read-write Name of the snippet
typename	String	Read-only Class name of the reference object
previous	Snippet	Read-only Returns the previous snippet object
next	Snippet	Read-only Returns the next snippet object
htmlText	String	Read-only HTML Text tag, which can be inserted in any HTML file
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
newFileLocation	String	No	Full path to the new snippet file

createDuplicate

Description

Creates a duplicate snippet from the specified snippet.

Returns

[Snippet](#)

Syntax

```
createDuplicate ()
```

MasterPage

Handles specific master page functions in the project.

MasterPage properties

Property name	Data type	Description
parent	MasterPageManager	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	MasterPage	Read-only Returns the previous master page object
next	MasterPage	Read-only Returns the next master page object
language	Language	Read-only Language of the specified master page
status	Status	Read-only Return status of the master page file
CBT	String	Read-only List of CBTs applied on the master page
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

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	TOCManager	Read-only Parent TOCManager object
name	String	Read-write Name of the TOC
typename	String	Read-only Class name of the reference object
previous	TOC	Read-only Returns the previous TOC object
next	TOC	Read-only Returns the next TOC object
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

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
newName	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
parent	IndexManager	Read-only Parent IndexManager object
name	String	Read-write Name of the index
typename	String	Read-only Class name of the reference object
previous	Index	Read-only Returns the previous index object
next	Index	Read-only Returns the next index object
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

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

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 ()

FMFile

Handles specific FrameMaker file functions in the project.

FMFile properties

Property name	Data type	Description
parent	FMFileManager	Read-only Parent FMFileManager object
name	String	Read-only Name of the FrameMaker file
typename	String	Read-only Class name of the reference object
previous	FMFile	Read-only Returns the previous FMFile object
next	FMFile	Read-only Returns the next FMFile object
frameFileFormat	FrameFileFormat	Read-only FrameMaker file format type (read/write)
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	SyncStatus	Read-only Synchronization status of the FrameMaker file
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

Property name	Data type	Description
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	WordFileManager	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	WordFile	Read-only Returns the previous WordFile object
next	WordFile	Read-only Returns the next WordFile object
wordFileFormat	WordFileFormat	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)
associatedGlossaryName	String	Read-write RoboHelp associated glossary name (read/write)
syncStatus	SyncStatus	Read-only Synchronization status of the Word file
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

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

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

HTMLHelpWindow

Gets or configures HTMLHelp window settings.

HTMLHelpWindow properties

Property name	Data type	Description
parent	HTMLHelpWindowManager	Read-only Parent HTMLHelpWindowManager object
name	String	Read-write Name of the window
typename	String	Read-only Class name of the reference object
previous	HTMLHelpWindow	Read-only Returns the previous HTMLHelpWindow object
next	HTMLHelpWindow	Read-only Returns the next HTMLHelpWindow object
windowCaption	String	Read-write Caption of the window
navPaneWidth	Integer	Read-write Navigation pane width
tabPosition	HTMLHelpWindowAlign	Read-write Tripane Tab position
defaultTab		Read-write Tripane default Tab
buttons	HTMLHelpWindowButtons	Read-write Flags to show different buttons in the HTMLHelp window
top	int	Window rectangle top position
left	int	Window rectangle left position
right	int	Window rectangle right position
bottom	int	Window rectangle bottom position
windowProperties	HTMLHelpWindowProperties	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"

Property name	Data type	Description
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

HTMLHelpWindowManager

Creates HTMLHelp windows and manipulate window settings.

HTMLHelpWindowManager properties

Property name	Data type	Description
parent	Project	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

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
```

AddMethod

Provides constants for different add methods in RoboHelp.

AddMethod properties

Property name	Data Type	Description
ADDBYCOPY	int	ADD BY COPY Constant
ADDBYREF	int	ADD BY REF Constant

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
JVAHELP	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

Property name	Data type
SLOVENIAN	Integer
SPANISH	Integer
SWEDISH	Integer
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
TOPICSBYPRIORITY	Integer
UNREFERENCEDTOPICS	Integer

Property name	Data type
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
SKINFLASHHELP	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
FRAMETOCTOCFORMAT	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 in case of TOKENTAG for token objects in RoboHelp.

TagType properties

Property name	Data Type
TAGNONE	int
TAGIMAGE	int
TAGHTML	int
TAGHEAD	int
TAGBODY	int
TAGTITLE	int
TAGMETA	int
TAGLINK	int

Property name	Data Type
TAGANCHOR	int
TAGPARAGRAPH	int
TAGTABLE	int
TAGDIV	int
TAGH1	int
TAGH2	int
TAGH3	int
TAGH4	int
TAGH5	int
TAGH6	int
TAGTABLECELL	int
TAGTABLEHEADER	int
TAGTABLEROW	int
TAGTABLEBODY	int
TAGTABLEFOOT	int
TAGTABLEHEAD	int
TAGCOL	int
TAGHR	int
TAGUL	int
TAGOL	int
TAGLI	int
TAGSPAN	int
TAGSCRIPT	int
TAGDOCTYPE	int
TAGFRAMESET	int
TAGSTYLE	int
TAGINDENT	int
TAGUNDERLINE	int
TAGBOLD	int
TAGOBJECT	int
TAGLINEBREAK	int
TAGIFRAME	int
TAGMARQUEE	int
TAGAPPLET	int

Property name	Data Type
TAGAREA	int
TAGMAP	int
TAGSELECT	int
TAGBUTTON	int
TAGPARAM	int
TAGNOSCRIPT	int
TAGFONT	int
TAGCAPTION	int

PIType

Provides constants for different PI Tag types in case of TOKENXMLPI for token object in RoboHelp.

PIType properties

Property name	Data Type
PINONE	int
PIUDVSTART	int
PIUDVEND	int
PIVARIABLESTART	int
PIVARIABLEEND	int
PICBTSTART	int
PICBTEND	int
PIIGNORED	int
PIPLACEHOLDER	int
PISCRIPSTART	int
PISCRIPEND	int
PISYMBOLSTART	int
PISYMBOLEND	int
PIGLOSSARYSTART	int
PIGLOSSARYEND	int
PIREGIONSTART	int
PIREGIONEND	int
PIALIGNSTART	int
PIALIGNEND	int

Property name	Data Type
PILISTSTART	int
PILISTEND	int
PIIMPLICIT	int
PICOMMENT	int
PICOMMENTSTART	int
PICOMMENTEND	int
PICOMMENTDATASTART	int
PICOMMENTDATAEND	int
PITRACKSTART	int
PITRACKEND	int

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
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

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
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

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

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

Property name	Data type
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
PROP_EHELP	Integer
PROP_AVENUE	Integer
PROP_USER_POS	Integer