
JMP Clinical Review API

The **JMP Clinical Review API** provides functions for manipulation of JMP Clinical Reviews, Reports, and Sections.

The general usage centers around a reference to a Review Viewer/Builder based on a generated ID. The ID may be obtained via one of several functions below, or will be made available in the context of Review Viewer/Builder executing code embedded within a JMP data table within a Report. The variable `JMPClinicalReviewID` will be the ID of the Review in which the code is executing. In this way, the script writer may customize output generated by a JMP Clinical Report by simply adding a script to the data table within the Report.

Descriptions of each function is provided below. Additionally, the installed API/example folder contains example scripts which demonstrate the usage of the functions.

Functions:

JMPClinicalReviewAPI:getReviewBuilder

This function gets an instance of a Review Builder. If an instance doesn't exist, one will be created.

Arguments:

show

{show=1} - shows the window

{show=0} - hides the window

Returns:

A reference to the Review Builder object.

Syntax:

```
JMPClinicalReviewAPI:getReviewBuilder = function({show=1}, {default local},
```

JMPClinicalReviewAPI:closeReviewBuilder

This function closes the current Review Builder instance.

Arguments:

None

Returns:

N/A

Syntax:

```
JMPClinicalReviewAPI:closeReviewBuilder = function({}, {default local},
```

JMPClinicalReviewAPI:openReviewTemplate

This function opens a Review Template

Argument:

templatePath - path to the Review Template to open

Returns:

A reference to the Review Builder object.

Syntax:

```
JMPClinicalReviewAPI:openReviewTemplate = function({templatePath}, {default local},
```

JMPClinicalReviewAPI:addReport

This function adds a Report, as defined in the Report Definitions, to the Review Builder.

Arguments:

reportDefinitionID - the name/ID of the Report to add (e.g. AEDistribution)

Returns:

A reference to the Review Builder object.

Syntax:

```
JMPClinicalReviewAPI:addReport = function({reportDefinitionID}, {default local},
```

JMPClinicalReviewAPI:removeReportByReportTitle

This function removes a Report from the Review Builder.

Arguments:

reportTitle - title of the Report to remove

Returns:

N/A

Syntax:

```
JMPClinicalReviewAPI:removeReportByReportTitle = function({reportTitle},  
{default local},
```

JMPClinicalReviewAPI:runReportByReportTitle

This function runs a Report in the Review.

Arguments:

reportTitle - title of the report to run

completedListener - function to call when the report completes execution. The callback function needs accept a string that will be the Report ID that just ran.

Returns:

N/A

Syntax:

```
JMPClinicalReviewAPI:runReportByReportTitle = function({reportTitle,  
completedListener=empty()}, {default local},
```

JMPClinicalReviewAPI:runAllReports

This function runs all Reports in the Review.

Arguments:

completedListener - function to call when the last report completes execution. The callback function does accept any arguments.

Returns:

N/A

Syntax:

```
JMPClinicalReviewAPI:runAllReports = function({completedListener=empty()},  
{default local},
```

JMPClinicalReviewAPI:saveReview

This function saves the review.

Arguments:

reviewPath - path of the Review to save

Returns:

N/A

Syntax:

```
JMPClinicalReviewAPI:saveReview = function({reviewPath}, {default local},
```

JMPClinicalReviewAPI:openReview

This function opens a review.

Arguments:

reviewPath - path of the Review to save

Returns:

N/A

Syntax:

```
JMPClinicalReviewAPI:openReview = function({reviewPath}, {default local},
```

JMPClinicalReviewAPI:getReviewViewerIDsAndTitles

This function gets the IDs and Titles of all Review Viewers/Builders.

Arguments:

None

Returns:

A list of associative arrays. Keys include:

id - String: the ID of the Review Viewer/Builder

title - String: the title of the Review Viewer/Builder

Syntax:

```
JMPClinicalReviewAPI:getReviewViewerIDsAndTitles = function({}, {default local},
```

JMPClinicalReviewAPI:getReviewViewer

This function gets an instance of the Review Viewer/Builder.

Arguments:

id - ID of the Review Viewer/Builder

Returns:

References to the Review Viewer/Builder, or empty() if not found

Syntax:

```
JMPClinicalReviewAPI:getReviewViewer = function({id}, {default local},
```

JMPClinicalReviewAPI:getReviewTableOfContents

This function gets the table of contents of the Review.

Arguments:

id - ID of the Review Viewer/Builder

Returns:

A list of associative arrays. Keys include:

title - String: title of the Report

sections - List: List of Strings for the names of the Sections within the Report

Syntax:

```
JMPClinicalReviewAPI:getReviewTableOfContents = function({id}, {default local},
```

JMPClinicalReviewAPI:getReportIDList

This function gets the list of Report IDs.

Arguments:

id - ID of the Review Viewer/Builder

Returns:

A list of ids of the Reports within the Review Viewer/Builder

Syntax:

```
JMPClinicalReviewAPI:getReportIDList = function({id}, {default local},
```

JMPClinicalReviewAPI:getReportTitleList

This function gets a list of Report titles.

Arguments:

id - ID of the Review Viewer/Builder

Returns:

A list of titles of the Reports within the Review Viewer/Builder

Syntax:

```
JMPClinicalReviewAPI:getReportTitleList = function({id}, {default local},
```

JMPClinicalReviewAPI:getReportByID

This function gets a reference to a Report.

Arguments:

reportID - ID of the Report (see JMPClinicalReviewAPI:getReportIDList)

Returns:

A reference to the Report object

Syntax:

```
JMPClinicalReviewAPI:getReportByID = function({reportID}, {default local},
```

JMPClinicalReviewAPI:getReportByTitle

This function gets a reference to a Report.

Arguments:

id - ID of the Review Viewer/Builder

reportTitle - title of the Report

Returns:

A reference to the Report object

Syntax:

```
JMPClinicalReviewAPI:getReportByTitle = function({id, reportTitle}, {default local},
```

JMPClinicalReviewAPI:getReportDetailsByTitle

This function gets the Report Definition associative array for a Report

Arguments:

id - ID of the Review Viewer/Builder

reportTitle - title of the Report

Returns:

An associative array describing the Report. Keys include:

1 - String: CDISC category

2 - String: Statistical category

label - String: Label of the Report

name - String: Name/unique id of the Report

icon - String: Path to the icon for the Report (using installation.path.preferences notation)

Syntax:

```
JMPClinicalReviewAPI:getReportDetailsByTitle = function({id, reportTitle},  
{default local},
```

JMPClinicalReviewAPI:getReportResultsByTitle

This function gets a reference to the results namespace of the Report.

Arguments:

id - ID of the Review Viewer/Builder

reportTitle - title of the Report

Returns:

A namespace with the contents of the Report, empty() if title doesn't exist or Report hasn't been run.

Syntax:

```
JMPClinicalReviewAPI:getReportResultsByTitle = function({id, reportTitle},  
{default local},
```

JMPClinicalReviewAPI:getReportSectionByTitleAndSectionName

This function gets a reference to Section of a Report.

Arguments:

id - ID of the Review Viewer/Builder
reportTitle - title of the Report
sectionName - name of the Report Section

Returns:

A reference to the OutlineBox the Report Section, empty() if title or Section name doesn't exist or Report hasn't been run.

Syntax:

```
JMPClinicalReviewAPI:getReportSectionByTitleAndSectionName = function({id,  
reportTitle, sectionName}, {default local},
```

JMPClinicalReviewAPI:getDataTablesForSectionByTitleAndSectionName

This function gets a list of data tables for a Report Section.

Arguments:

id - ID of the Review Viewer/Builder
reportTitle - title of the Report
sectionName - name of the Report Section

Returns:

A list of data tables used by the Report Section, empty list {} if title or Section name doesn't exist or Report hasn't been run

Syntax:

```
JMPClinicalReviewAPI:getDataTablesForSectionByTitleAndSectionName =  
function({id, reportTitle, sectionName}, {default local},
```


JMPClinicalReviewAPI:insertSectionIntoReportByTitleAndSectionName

This function insert a new Section into a Report.

Arguments:

id - ID of the Review Viewer/Builder

reportTitle - title of the Report

sectionName - name of the new Report Section

displayObject - reference to a DisplayObject to be inserted

insertionIndex - index of where to insert the new Section 0 (default)

Returns:

N/A

Syntax:

```
JMPClinicalReviewAPI:insertSectionIntoReportByTitleAndSectionName =  
function({id, reportTitle, sectionName, displayObject, insertionIndex=0},  
{default local},
```

JMPClinicalReviewAPI:removeSectionFromReportByTitleAndSection- Name

This function removes a Report Section

Arguments:

id - ID of the Review Viewer/Builder

reportTitle - title of the Report

sectionName - name of the new Report Section

Returns:

N/A

Syntax:

```
JMPClinicalReviewAPI:removeSectionFromReportByTitleAndSectionName =  
function({id, reportTitle, sectionName}, {default local},
```

JMPClinicalReviewAPI:appendDisplayObjectToReportSectionByTitleAndSectionName

This function appends a DisplayObject to the end of a Report Section.

Arguments:

id - ID of the Review Viewer/Builder

reportTitle - title of the Report

sectionName - name of the new Report Section

displayObject - reference to a DisplayObject to be inserted

Returns:

N/A

Syntax:

```
JMPClinicalReviewAPI:appendDisplayObjectToReportSectionByTitleAndSectionName  
= function({id, reportTitle, sectionName, displayObject}, {default local},
```

JMPClinicalReviewAPI:prependDisplayObjectToReportSectionByTitleAndSectionName

This function prepends a DisplayObject to the beginning of a Report Section

Arguments:

id - ID of the Review Viewer/Builder

reportTitle - title of the Report

sectionName - name of the new Report Section

displayObject - reference to a DisplayObject to be inserted

Returns:

N/A

Syntax:

```
JMPClinicalReviewAPI:prependDisplayObjectToReportSectionByTitleAndSectionName  
= function({id, reportTitle, sectionName, displayObject}, {default local},
```