Office Document Module
User Guide

Phone: 1-800-470-2302
Fax: 1-587-400-2746
info@kymerasystems.com

Table of Contents

Phone: 1-800-470-2302
Fax: 1-587-400-2746
info@kymerasystems.com

Introduction

The Kymera Office Document module allows Dataset objects in Ignition to be exported to real Excel files, both in .xls and .xlsx format. It contains several scripting functions, allows users to access the Apache POI Workbook and access to the Apache POI DateUtil class. This access to Workbook and DateUtil also allows users to access their associated methods.

Documentation on the Apache POI Workbook can be found here:
https://poi.apache.org/apidocs/org/apache/poi/ss/usermodel/Workbook.html

Documentation on the differences in the type of Workbook generated from the .xlsx and .xls formats can be found here:
https://poi.apache.org/spreadsheet/

Documentation on the Apache POI DateUtil class can be found here:
https://poi.apache.org/apidocs/org/apache/poi/ss/usermodel/DateUtil.html

Phone: 1-800-470-2302
Fax: 1-587-400-2746
info@kymerasystems.com

Setup

Install the module as you would any other module in the Gateway, by navigating to the **System>Modules** menu in the Configure section.

Once installed, the Office Document scripting functions will be available for access in the Designer. The scripting functions are scoped to work in the Client, Designer and Gateway scopes.

Support

Support is only provided to license modules with Kymera Extended Support.

- 1 year of technical support during regular business hours - Monday - Friday 8AM to 4PM MST.
- Free Upgrades with Ignition updates (I.E. 8.0 to 8.01).
- Support may be contacted via email, at support@kymerasystems.com, or via phone, at 1-800-470-2302. Please allow up to 24 hours for a response from our support team.

Change Log

v1.6.0: Initial release

Scripting Functions

The following scripting functions are available with the module.

Phone: 1-800-470-2302
Fax: 1-587-400-2746
info@kymerasystems.com

system.poi.getExcelBuilder()

Description

This method returns an instance of ExcelBuilder, which will decrease the amount of work required to make an Excel sheet. This method provides a streaming mode for creating the document. This method is only capable of creating .xlsx files, and not .xls files.

Syntax

```
system.poi.getExcelBuilder()
```

Parameters

none

Returns

ExcelBuilder – The method used to create an Excel “xlsx” document.

Scope

Designer, Client, Gateway

ExcelBuilder Methods

createSheet(String name**)** – Creates a new worksheet with the name specified. This will become the active worksheet.

addDataset(Dataset data**)** – Appends the dataset into the current worksheet. Note that this accepts both the Ignition Dataset object and the PyDataSet object.

getBytes() – Returns the byte array of the Excel file.

setCompressTempFiles(boolean compress**)** – Sets whether or note the temporary files should be compressed. Save storage IO in exchange for CPU utilization.

setAutosizeColumns(boolean autosize**)** – Set whether or not columns should be autosized based off their content.

Phone: 1-800-470-2302
Fax: 1-587-400-2746
info@kymerasystems.com

Examples

The following example demonstrates how to export 2 datasets into 2 Excel worksheets when called from a button sharing a window with a power table.

```
queryData = system.db.runQuery("SELECT col1, col2 FROM my_table")

powerTableData = event.source.parent.getComponent('Power Table').data
# to create a header for the Excel, file, add a Dataset representing the header
header = system.dataset.toDataSet([col for col in powerTableData.getColumnNames()], [[col for col in
powerTableData.getColumnNames()]])

# create a builder instance
builder = system.poi.getExcelBuilder()
# create the name of the working sheet
builder.createSheet("Query")
# append the data
builder.addDataset(queryData)

# create a new sheet
builder.createSheet("Power Table")
# add the header
builder.addDataset(header)
builder.addDataset(powerTableData)

bytes = builder.getBytes()
path = system.file.saveFile("myFile", "xlsx", "An Excel File")
if path:
	system.file.writeFile(path, bytes)
```

Phone: 1-800-470-2302
Fax: 1-587-400-2746
info@kymerasystems.com

system.poi.export.saveDataSet()

Description

Exports the contents of multiple datasets as an Excel spreadsheet, and returns the byte array of the files contents.

Syntax

system.poi.export.saveDataSet(datasets, sheetNames, format, autoSizeColumns**)**

Parameters

PyList datasets – A List of Lists of datasets. The outer list represents all the data that will be in the new spreadsheet, and each list in it represents the datasets that will be in each sheet.

PyList sheetNames – A list of the worksheet names

String format – The type of Excel file to make. Valid options are “xls” and “xlsx”.

boolean autoSizeColumns – The script will try to autosize each column allowing for a nicer export.

Returns

byte[] – Returns the byte array containing the Excel file data.

Scope

Client, Designer, Gateway

Examples

The following code will export 2 datasets, into 2 Excel worksheets.

```
data1 = system.db.runQuery("SELECT 1, 2, 3")
data2 = system.db.runQuery("SELECT 4, 5, 6")

bytes = system.poi.export.saveDataSet([data1, data2], [data2, data1]], ['forward', 'reverse'], "xlsx")
path = system.file.saveFile("myFile", "xlsx", "An Excel File")
if path:
	system.file.writeFile(path, bytes)
```

Phone: 1-800-470-2302
Fax: 1-587-400-2746
info@kymerasystems.com

system.poi.file.openWorkbook()

Description

Loads an Apache POI Workbook instance into memory from the provided path.

Syntax

system.poi.file.openWorkbook(path**)**

Parameters

String path – The string representing the path of the spread sheet file. .xlsx and .xls filetypes are the only two supported. If the path doesn't contain one of these two file extensions, then a null is returned.

Returns

Workbook – Returns an instance of an Apache POI Workbook.

Scope

Client, Designer, Gateway

Examples

The following example will open the file from the previous system.poi.export.saveDatasSet() example, and create a Workbook instance.

```
path = system.file.openFile("xlsx")
if path:
	book = system.poi.file.openWorkbook(path)

	sheet = book.getSheet('forward')
```

Phone: 1-800-470-2302
Fax: 1-587-400-2746
info@kymerasystems.com

system.poi.file.saveWorkbook()

Description

Saves a Workbook instance to a given String path.

Syntax

```
system.poi.file.saveWorkbook(workbook, path)
```

Parameters

Workbook workbook – An instance of a POI Workbook.

String path – The path where the output file would be saved.

Returns

none

Scope

Client, Designer, Gateway

Examples

The following example will save myWorkbook as “myFile” with the “xlsx” extension using the given String.

```
path = “C:\Users\Username\FolderName\myFile.xlsx”

book = system.poi.file.saveWorkbook(myWorkbook, path)
```

Phone: 1-800-470-2302
Fax: 1-587-400-2746
info@kymerasystems.com

system.poi.file.writeWorkbookToBytes()

Description

Returns a byte array of a given Workbook.

Syntax

system.poi.file.writeWorkbookToBytes(workbook**)**

Parameters

Workbook workbook – An instance of an Apache POI Workbook.

Returns

byte[] – Returns the byte array containing the Workbook data.

Scope

Client, Designer, Gateway

Phone: 1-800-470-2302
Fax: 1-587-400-2746
info@kymerasystems.com

system.poi.file.openDBWorkbook()

Description

Returns an instance of the Apache POI Workbook. The specific instance is governed by the fileType argument.

Syntax

system.poi.file.openDBWorkbook(bytes, fileType**)**

Parameters

byte[] bytes – A byte array representing the Workbook data.
String fileType – A String representing a specific implementation of Workbook. The only two acceptable inputs are “XSSFWorkbook” and “HSSFWorkbook”; any other arguments will return a null.

Returns

Workbook – An instance of an Apache Workbook.

Scope

Client, Designer, Gateway

Phone: 1-800-470-2302
Fax: 1-587-400-2746
info@kymerasystems.com

system.poi.utils.DateUtil()

Description

Allows access to the Apache POI DateUtil class.

Syntax

system.poi.utils.DateUtil()

Parameters

none

Returns

none

Scope

Client, Designer, Gateway

Examples

The following example will demonstrate how to access and use DateUtil.

```
# a static field
seconds = system.poi.utils.DateUtil.SECONDS_PER_MINUTE

# converts the given String to a java.util.Date object
date = system.poi.utils.DateUtil.parseYYYYMMDDDate("2018/01/01")
```