What is CFXL?

CFXL lets you quickly and easily create new spreadsheets and modify or read data from existing spreadsheets. This project abstracts and simplifies the POI API while still allowing you to get into the depths of that API.

Support and other licenses available if needed.

Contributors

If you're interested in contributing, just send me an email at jason@delmore.info.

Installation

1 - Move the com folder to your webroot

2 – *(optional – do this if you want to use the tag syntax)* Move xl.cfm and xlparam.cfm from the CustomTags folder into the ColdFusion CustomTags folder (if you want to use the CustomTag)

3 – *(optional – do this if you want to see some examples run)* Move the examples folder into your webroot

License

Documentation

Usage

Check out the examples folder.

- index.cfm – Links to all of the examples and displays code
- parseXLS.cfm - Shows how to parse an existing xls file
- useCFXLCFC.cfm - Shows how to call the functions in cfxl.cfc directly
- useCFXLTag.cfm - Shows how to use the custom tags

Component Docs

Component cfxl - cfxl.com.jasondelmore.cfxl

This Component exposes many functions of the POI API for working with Spreadsheets.

hierarchy:	WEB-INF.cftags.component cfxl.com.jasondelmore.cfxl
path:	/Library/WebServer/Documents/CFXL/com/jasondelmore/cfxl.cfc
properties:	
Methods: (* - indicates private method)	closeWorkbook, convertExcelColumn*, convertExcelRow*, createJavaObject*, getCell, getCellValue, getOutputStream, getRow, getSheet, getWorkbook, init, initializeJavaLoader*, removeRow, setCell, setCellValue, setOutputStream, setRow, setSheet, setWorkbook, viewWorkbook, writeWorkbook

closeWorkbook

```
public closeWorkbook ( )
```

convertExcelColumn*

```
private numeric convertExcelColumn ( required any column )
```

Converts the alphabetical value for column displayed in excel to the numeric POI wants.

Parameters:
column: any, required, column

convertExcelRow*

```
private numeric convertExcelRow ( required numeric row )
```

Converts the value for row displayed in excel to the value POI wants (POI starts with 0 rather than 1).

Parameters:
row: numeric, required, row

createJavaObject*

```
private any createJavaObject ( )
```

getCell

```
public any getCell ( )
```

getCellValue

```
public any getCellValue ( required any columnNum, numeric rowNum )
```

Parameters:
columnNum: any, required, columnNum
rowNum: numeric, optional, rowNum

getOutputStream

```
public any getOutputStream ( )
```

getRow
public any **getRow** ( )

getSheet
public any **getSheet** ( )

getWorkbook
public any **getWorkbook** ( )

init
public any **init** (*string* sourceFile="")

Parameters:
sourceFile: string, optional, sourceFile

initializeJavaLoader*
private void **initializeJavaLoader** (*string* javaloaderpath="com.compundtheory.javaloader.JavaLoader")

Parameters:
javaloaderpath: string, optional, javaloaderpath

removeRow
public void **removeRow** (*required numeric* row)

Parameters:
row: numeric, required, row

setCell
public any **setCell** (*required any* columnNum, *numeric* rowNum, *any* newValue, *any* cellType)

Parameters:
columnNum: any, required, columnNum
rowNum: numeric, optional, rowNum
newValue: any, optional, newValue
cellType: any, optional, cellType

setCellValue
public any **setCellValue** (*required any* newValue, *any* cellType)

Parameters:
newValue: any, required, newValue
cellType: any, optional, cellType - Possible values: numeric, string, formula, blank, boolean

setOutputStream
public any **setOutputStream** ( )

setRow
public any **setRow** (*required numeric* rowNum="1")

Parameters:
rowNum: numeric, required, rowNum

setSheet
public any **setSheet** (sheet="0")

Parameters:
sheet: any, optional, sheet

setWorkbook
public any **setWorkbook** (*string* sourceFile="")

Parameters:
sourceFile: string, optional, sourceFile

viewWorkbook
public void **viewWorkbook** ( )

writeWorkbook
public void **writeWorkbook** (*required string* outputFile)

Parameters:
outputFile: string, required, outputFile