

Utility Functions: Two UDFs to Make Spreadsheets Easier

Posted At : April 18, 2011 10:46 AM | Posted By : Jon Hartmann

Related Categories: ColdFusion, Utility Function

I started diving into using ColdFusion's capabilities to work with Excel spreadsheets for the first time today. Even though its been a part of ColdFusion since CF9 dropped in 2009, I've not been called on to use the functionality, so I'm a bit behind the curve. Once I got moving though, I found some annoyances and gotcha's that drove me to come up with a couple of new UDFs to ease my pain. Find `SpreadsheetExpandedInfo()` and `SpreadsheetGetData()` after the jump.

SpreadsheetExpandedInfo()

My first problem was that the basic method of getting information about a spreadsheet seemed to be a little weird... you don't get row counts about specific pages unless you read in a specific page and then look at the spreadsheet object itself; its not listed as part of the spreadsheet "info". I created the following code to get back data that I found a little more palatable.

```
<cffunction name="SpreadsheetExpandedInfo" access="public" output="false" returntype="struct">
    <cfargument name="FilePath" type="string" required="true" />

    <cfset Local.Spreadsheet = SpreadsheetRead(Arguments.FilePath) />
    <cfset Local.Info = SpreadsheetInfo(Local.Spreadsheet) />

    <cfset StructDelete(Local.Info, "SheetNames") />

    <cfset Local.Sheets = ArrayNew(1) />

    <cfloop from="1" to="#Local.Info.Sheets#" index="Local.X">
        <!--- JavaCast() necessary to prevent Java error caused by passing Local.X directly --->
        <cfset Local.SheetData = SpreadsheetRead(Arguments.FilePath, JavaCast("string", Local.X)) />

        <cfset Local.Sheet = StructNew() />

        <cfset Local.Sheet.Name = Local.SheetData.SheetName />
        <cfset Local.Sheet.RowCount = Local.SheetData.RowCount />

        <cfset ArrayAppend(Local.Sheets, Local.Sheet) />
    </cfloop>

    <cfset Local.Info.Sheets = Local.Sheets />

    <cfreturn Local.Info />
</cffunction>
```

And you can use it like this:

```
<cfset filename = ExpandPath("example.xls") />

<cfdump var="#SpreadsheetExpandedInfo(filename)#">
```

The basic difference here is that instead of getting a structure with Sheets and SheetNames values that show how many sheets are in the document, and a list of their names, you now get back just a Sheets value. The new Sheets value is an array of structs giving each sheet's Name and RowCount. This means you can now see all data thats available about the sheets/rows without needing to do multiple calls just to get each sheet's RowCount.

SpreadsheetGetData()

This was a big one... I can't find a built in function that allows you to get actual row data from a spreadsheet. It seems the only way to do that is to use `<cfspreadsheet>` with the query attribute specified. Maybe I missed something... if I did, please point it out! To get around this I made a simple wrapper function, which isn't special, but I wanted to kill two birds with one stone so I built in a fix to the "row-header problem". You can tell CF that one of your sheet's rows contains the headers for the columns, but CF doesn't automatically strip them out. Further, you have to specify which row it is... even though most sane people leave headers in row 1. I've built wrapper so that it has a simple boolean flag to use the first row as header data and to remove it from the result set.

```

<cffunction name="SpreadsheetGetData" access="public" output="false" returntype="query">
    <cfargument name="Src" type="string" required="true" />
    <cfargument name="Sheet" type="string" required="false" default="1" />
    <cfargument name="FirstRowIsHeaders" type="boolean" required="false" default="false" />

    <cfset Local.SpreadsheetAttributes = {
        Action = "read",
        Query = "Local.Result"
    } />

    <!-- Tack on all extra arguments -->
    <cfloop list="Columns,ColumnNames,Format,HeaderRow,Rows,Sheet,Src" index="Local.Argument">
        <cfif StructKeyExists(Arguments, Local.Argument)>
            <cfset Local.SpreadsheetAttributes[Local.Argument] = Arguments[Local.Argument] />
        </cfif>
    </cfloop>

    <!-- Handle custom argument -->
    <cfif Arguments.FirstRowIsHeaders>
        <cfset Local.Info = SpreadsheetRead(Local.SpreadsheetAttributes.Src, Arguments.Sheet) />
        <cfset Local.SpreadsheetAttributes.HeaderRow = 1 />
        <cfset Local.SpreadsheetAttributes.Rows = "2-#Local.Info.RowCount#" />
    </cfif>

    <!-- Handle if a sheet name was passed in via nameless arguments -->
    <cfif NOT IsNumeric(Local.SpreadsheetAttributes.Sheet)>
        <cfset Local.SpreadsheetAttributes.SheetName = Local.SpreadsheetAttributes.Sheet />
        <cfset StructDelete(Local.SpreadsheetAttributes, "Sheet") />
    </cfif>

    <!-- Get the data -->
    <cfspreadsheet attributecollection="#Local.SpreadsheetAttributes#" />

    <cfreturn Local.Result />
</cffunction>

```

And the example:

```

<cfset filename = ExpandPath("example.xls") />

<cfdump var="#SpreadsheetGetData(filename, 1)#">
<cfdump var="#SpreadsheetGetData(filename, 1, true)#">
<cfdump var="#SpreadsheetGetData(filename, 2, true)#">
<cfdump var="#SpreadsheetGetData(filename, 'Departments', true)#">
<cfdump var="#SpreadsheetGetData(Src=filename, sheetname='Employees', columns='1,5')#">

```

Sample Data

I've attached the spreadsheet I am using for testing; download it if you want to test my exact examples.