Importing Excel Spreadsheets. The Easy Way

Posted At : November 5, 2009 7:20 PM | Posted By : TJ Downes
Related Categories: ColdFusion

Recently I was asked to import data into our system using a spreadsheet provided by the client. There are, of course, numerous ways to go about this, including using tools built into SQL Server (DTS/SSI). However, I wanted to provide tool that the client could use to perform these imports themselves, in the future.

The obvious choice was a web-based tool. The client is used to web-based interfaces and I could also leverage some new functionality in ColdFusion 9. I knew that ColdFusion now has the cfspreadsheet tag, and it was a great opportunity to test this.

The spreadsheet provided to me was over 42,000 rows of data. This seemed like a lot for a spreadsheet! After reading through the documentation on the cfspreadsheet tag, I determined, for my use, that converting the contents to a query object was probably going to make it easiest to work with. I simply provided a form for the client to upload the Excel file to the server. Once the file was uploaded, I used the following to process the file and convert it to a query object:

```
<cfspreadsheet action="read"
   src="#variables.uploadedFile#"
   query="mySpreadsheet"
   headerrow="1"/>
```

I'll explain each of these attributes in minor detail:

action - This is what type of action we want to perform on the spreadsheet. The choice are read, update and write.

src - The source of the Excel spreadsheet. In my case, this is a dynamic value, but does require an absolute path!

query - The name I am giving to my query object for future reference headerrow - the row number in the spreadsheet which contains the column names

I'd like to point out that if you do not define headerrow you will need to reference your column names as such: col_1, col_2, etc. If you do provide the headerrow attribute you MUST reference the columns by name.

Now referencing your spreadsheet data is simple! If you want to output the data to the screen, simply use cfoutput like you normally would for a query!

```
<cfoutput query="mySpreadsheet">
#column1# #column2#<br />
</cfoutput>
```

It's that simple!

Instead of creating a query object you also have the ability to create a csv or html from your spreadsheet. to do this, using the following

```
<cfspreadsheet action="read"
   src="#variables.uploadedFile#"
   name="mySpreadsheet"
   headerrow="1"
format="html/csv" />
```

By leaving out the format attribute, it will return a string. Regardless of which format you choose, you can output the data to the screen like this:

```
<cfoutput>#mySpreadsheet#</cfoutput>
```

As you can see, the cfspreadsheet tag has made working with Excel files much easier!

I would like to point out a couple of caveats I experienced while working with the cfspreadsheet tag:

1. There's a bug, which I have yet to report, but will do soon. Essentially, if your spreadsheet contains pound signs (#), the read will fail. Hopefully this will get fixed soon.

2. Spreadsheets with large amounts of data, such as my project with 42,000+ rows, may completely fail when using HTML or CSV format. I have a fairly beefy machine (multi-core, 6GB RAM) and could not get mine to process the file in HTML or CSV format. Your mileage may vary.

Overall, it looks like cfspreadsheet is a very valuable addition to the CFML language. Hopefully it will get updated over time to include other spreadsheet formats too, like Open Office! It certainly saved me considerable time in my project and will prove a valuable tool for the client for future use. Also, I know how to leverage this for future projects now, time well spent!