CSV Files and Coldfusion - a New Approach

Posted At : February 5, 2007 12:45 PM | Posted By : Mark Kruger
Related Categories: Coldfusion Tips and Techniques

Many times we have been tasked with working out a way for an ordinary user to "upload" data into a system. This task is fraught with peril because users are not well versed in the idea of "normalized" data. We've tried the Excel file approach. Excel gives the user so many neat options they can't resist adding formatting, charts, graphs and fiddling with the headers. Even with protected files, clever users find ways to make the data incomplete or unusable. We've also tried access. This is workable but it takes more setup and it requires extra expense on the part of the user (namely, they must have excel). Enter the famed low-tech "csv" or "comma separated value" file.

This file is not rocket science by any means. It is a way to represent tabular data in a "plain text" fashion using commas and quotation marks. The commas separate the fields and the quotation marks identify the values (where needed). When are the quotation marks needed? Well, they are particularly needed when a comma might creep into a value. For example:

```
"Joe","Smith","112 Test Street"

"Sam","Jones, Jr.",114 test street"
```

Notice that Mr. Jones is a "Jones, Jr". Without the quotations marks as *text qualifier* that row of data would have 4 pieces of data instead of 3, breaking normalization.

So CSV is a common "lowest denominator" format we can use to get data into our system. In fact, we work with data feeds for financial and analyst data quite often and many of them provide their data in this format by default. So learning how to work with CSV is a good choice for upgrading your skills.

The problem

Coldfusion does not provide some easy way to read this data into an array or query. Working with it as a set of lists is ok, but how do you get around the "text qualifier" issue? Consider **this post** by Ben Nadel (who is no CF_slouch). His function reads a CSV string into an array. It's a good function, but long and involved. Another approach (one which I've used a few times) is to place the csv file at a URL and use CFHTTP to read it into a query object. There is nothing wrong with either of these approaches, but if you are on a windows box I've found an *easier* way.

Use the MS Text Driver

You can use the MS text driver to create a proxy for your file. Here are the steps.

1. **Setup a Data source** - Set up a data source pointing to the MS Text driver. In CFMX this is done by creating an ODBC alias in the ODBC control panel and then using the "ODBC Socket" driver to connect to it. On a CF 5 box it will be in the drop down list.
2. **Use CF Query To Connect To Your File** - The text driver serves as a container for any csv file to which you wish to connect. The code is easy.

```
<Cfset FilePath = "C:\userfiles\upload.csv">
```

```
<cfquery name="CSVData" datasource="GenericCSV">
    SELECT *
    FROM #filePath#
</cfquery>
```

This call will return a query object with your normalized data.

There are a couple of things to note. The first row of your CSV file will contain the column names. I'm sure there are ways around that and researching the text driver a little will unearth them. Also remember that the cleverest technique can be foiled by the least clever user. This technique works when the CSV file conforms to the proper standard.

Finally, you might note that this entry is similar to a popular technique I wrote about in this post on **Connection DSN in CFMX**. In the case of CSV files I suspect it will make life a lot easier.