DTS' Other Uses

Posted At : March 29, 2007 5:01 PM | Posted By : Mark Kruger
Related Categories: MS SQL Server

One of the nicest things about MS SQL is DTS. If you are a reader who has a visceral reaction to anything nice being said about Microsoft you should grab a paper bag so that you don't hyperventilate as we go forward. Shallow breaths... shallow breaths.... ok - ready? Here we go. I suppose that most developers are exposed to DTS (Data Transformation Services) as an import-export mechanism for Microsoft SQL Server. The most common use is during deployment, rehosting or setting up a development environment. There are a host of other things for which you can use DTS that perhaps you hadn't thought of. For example:

- *Moving Data from one DB platform or format to another* - For example, with 2 ODBC connections to 2 Oracle servers you could import and export data from one Oracle server to another. I know that those of you with experience in Oracle will gasp with horror at this though - but migrating data around in Oracle is unnecessarily difficult (as is just about everything else about Oracle). You could transfer data from a DBase file to an access file. You could move information from a proprietary platform running on Cobol (as long as you had a driver) into a flat file or into MSSQL. It's a very nifty mechanism with all sorts of possibilities.
- *File Drop Import* - We use DTS to "pickup" a file who's name may not be consistent. It is even possible to FTP a file for import.
- *Data Column Validation* - With a minimal amount of scripting you can check values prior to importing them. So, for example, you can verify if a field is a number or is parsable into a date, or is populated or not.
- *Complex SQL Tasks* - You can add SQL task and even COM tasks to a package extending it well beyond just transferring data.

For example, we have a task that imports stock data. It is set to run every weekday. On days that are holidays it would run and sometimes create problems with the Bid and Ask. We needed a way to check and see if it was a holiday. If it *was* a holiday we wanted to terminate the process. Here is what our DTS Guru came up with. Using the package designer, at the beginning of the package add an ActiveX task that looks like this (this is the generic version).

```
Function Main()

Dim cn
Dim rs
Dim Flag

Set cn = CreateObject("ADODB.Command")
Set rs = CreateObject("ADODB.Recordset")
cn.ActiveConnection = "Provider=sqloledb;Data Source=database;Initial Catalog=database;User
Id=username;Password=password;"
cn.CommandType = 1 'adCmdText
cn.ActiveConnection.CursorLocation = 3 'adUseClient
cn.CommandTimeout = 60 'set to 1 min
cn.CommandText = "select * from [table]"

Set rs = cn.Execute()
```

```
If rs.RecordCount > 0 Then
Flag = "F"
Else
Flag = "S"
End If

Set rs = Nothing
Set cn = Nothing
'msgbox "Result=" & Flag
If Flag = "S" Then
Main = DTSTaskExecResult_Success
Else
Main = DTSTaskExecResult_Failure
End If

End Function
```

Make sure and set the task to "continue on success". The last few lines tell the story. If "Main" is set to the constant DTSTaskExecResult_Failure then the task will not continue. The result is a DTS task that is "smarter". We are able to put the execution of the task in the hands of the stakeholder of the site who only needs to be sure and maintain his table of trading holidays.