

How To Get the List of Files from an FTP Server in SQL Server Integration Services (SSIS) 2005

Written By: Ray Barley -- 12/8/2008

Problem

We use the FTP task in SSIS to process a number of files from an FTP server. We would like to implement a step in our SSIS packages that would retrieve the list of files that are available on the FTP server before we try to process them. The FTP task doesn't have an operation that will retrieve the list of files. Can you provide an example of how to do this?

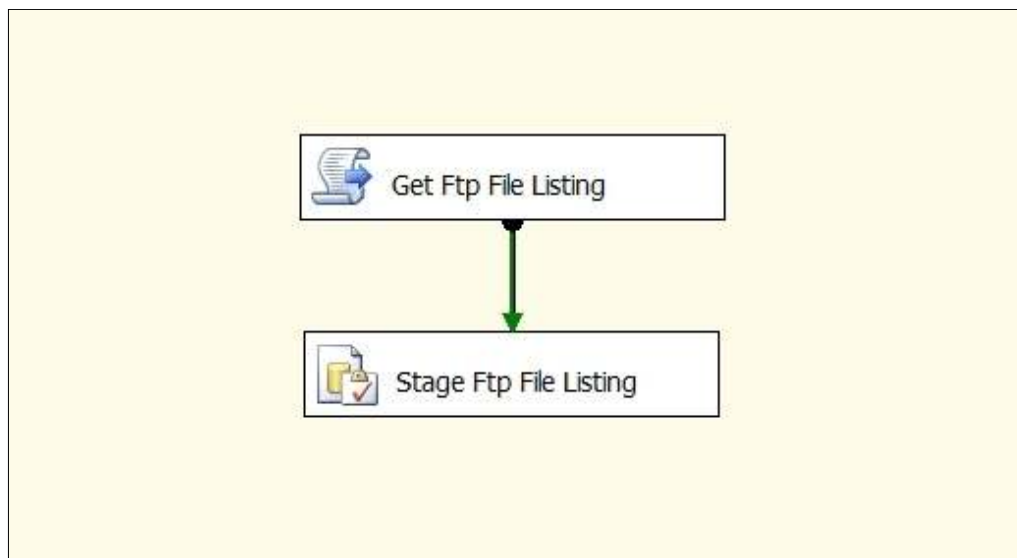
Solution

All of the code for this tip can be [downloaded here](#).

The FTP task provides the following operations as shown below:

<input type="checkbox"/> Remote Parameters	
IsRemotePathVariable	False
RemotePath	
OverwriteFileAtDest	False
<input type="checkbox"/> Local Parameters	
IsLocalPathVariable	False
LocalPath	
<input type="checkbox"/> Operation	
Operation	Send files
IsTransferAscii	Send files
	Receive files
	Create local directory
	Create remote directory
	Remove local directory
	Remove remote directory
	Delete local files
	Delete remote files

As you have noted, there is no operation to retrieve a list of files. You can use the Script task to do this. Here is a sample SSIS package that we will review to get the list of files from an FTP server:







The main points about the above SSIS package are:

- Get Ftp File Listing is a Script task that will retrieve the list of files available on the FTP server. It will return the list of available files as an XML document.
- Stage Ftp File Listing is an Execute SQL task which will call a stored procedure to insert the list of files from the XML document into a table.

SSIS Package Setup

Before digging in to the package tasks shown above, let's discuss the package setup. The following variables are defined in the SSIS package:

Variables			
Name	Scope	Data Type	Value
 FtpFileListXML	GetFtpFileListing	String	
 FtpFileType	GetFtpFileListing	String	
 FtpServer	GetFtpFileListing	String	
 FtpWorkingDirectory	GetFtpFileListing	String	

The variables are used as follows:

- FtpFileListXML will be populated with the list of files available on the FTP server by the Get Ftp File Listing Script task
- FtpFileType is used to specify the type of file
- FtpServer is the URL of the Ftp server; e.g. ftp.yourservername.com
- FtpWorkingDirectory is the directory or folder on the Ftp server to get the list of files from

The FtpFileType, FtpServer, and FtpWorkingDirectory variables will be set on the command line when we run the package. This will allow our package to retrieve the list of files from any or our ftp servers.

The package uses the following Connection Managers:

- OLE DB Connection called Staging which is the SQL Server database where the list of files from the FTP server will be written
- FTP Connection which defines the FTP Server; the ServerName property is set to the FtpServer package variable

Get Ftp File Listing

Let's take a look at the Script task that is used to get the list of files from the FTP server. The Script

page of the editor is shown below:

Script	
ScriptLanguage	Microsoft Visual Basic .NET
PrecompileScriptIntoBinaryCode	True
EntryPoint	ScriptMain
ReadOnlyVariables	FtpWorkingDirectory
ReadWriteVariables	FtpFileListXML

The FtpWorkingDirectory variable allows the package to specify the folder on the FTP server to query for the list of files. The script code will populate the FtpFileListXML variable with the list of files available on the FTP server as an XML document.

Now let's review the actual code in the Script task:

```
' STEP 1
Dim ftpFileNameListXML As New StringBuilder
ftpFileNameListXML.AppendLine("<filelist>")

' STEP 2
Dim ftpcm As ConnectionManager = Dts.Connections("FTP")
Dim ftp As FtpClientConnection = _
    New FtpClientConnection(ftpcm.AcquireConnection(Nothing))
Dim ftpFileNames() As String
Dim ftpFolderNames() As String

' STEP 3
ftp.Connect()
ftp.SetWorkingDirectory(Dts.Variables("FtpWorkingDirectory").Value.ToString())
ftp.GetListing(ftpFolderNames, ftpFileNames)
ftp.Close()

' STEP 4
Dim i As Integer
For i = 0 To ftpFileNames.GetUpperBound(0)
    ' add xml element
    ftpFileNameListXML.Append("<file name='")
    ftpFileNameListXML.Append(ftpFileNames(i))
    ftpFileNameListXML.AppendLine("'>")
Next i

' STEP 5
ftpFileNameListXML.AppendLine("</filelist>")
Dts.Variables("FtpFileListXML").Value = ftpFileNameListXML.ToString()

Dts.TaskResult = Dts.Results.Success
```

The main points about the above code are as follows:

- Step 1: declare a StringBuilder variable to build up the XML document containing the list of available files on the FTP server
- Step 2: setup the FtpClientConnection object using an FTP Connection Manager defined in the

package; the FtpClientConnection class is provided in the Microsoft.SqlServer.Dts.Runtime namespace; the ftpFileNames array will be populated with the list of files; the ftpFolderNames array will be populated with the list of subfolders

- Step 3: connect to the FTP server and get the list of files and subfolders from the folder specified by the FtpWorkingDirectory variable
- Step 4: iterate through the ftpFileNames array and build an XML document with the list of file names
- Step 5: set the FtpFileListXML package variable to the XML document containing the list of file names

Stage Ftp File Listing

Let's take a look at the Execute SQL task that inserts the list of files available on the FTP server into a table. The create table script is as follows:

```
CREATE TABLE dbo.stg_FtpFileList
(
  FileType nvarchar(50) NOT NULL,
  FileName nvarchar(50) NOT NULL
)
```

The following stored procedure is used to shred the XML document created in the Get Ftp File Listing task and insert the available files into the table above:

```
CREATE PROCEDURE [dbo].[usp_PutFtpFileList]
  @fileType nvarchar(50)
, @xml xml
AS
BEGIN
  SET NOCOUNT ON;

  DELETE FROM [dbo].[stg_FtpFileList]
  WHERE [FileType] = @fileType

  INSERT INTO [dbo].[stg_FtpFileList] (
    [FileType]
    , [FileName]
  )
  SELECT
    @fileType
  , doc.col.value('@name', 'nvarchar(50)') filename
  FROM @xml.nodes('//file') doc(col)
END
```

The above stored procedure takes advantage of new XML capabilities provided in SQL Server 2005; take a look at our previous tip [Replacing OPENXML with the XML nodes\(\) Function in SQL Server 2005](http://www.mssqltips.com/tipprint.asp?tip=1641) for the details.

Running the Package

To run the package use DTEXEC and specify values for the FtpFileType, FtpServer, and FtpWorkingDirectory variables on the command line; e.g. here is a batch file that you could use:

```
SET FTPSERVER=ftp.yourserver.com
SET FTPFILETYPE=XXX
SET FTPWORKINGDIRECTORY=/yourftpfolder
```

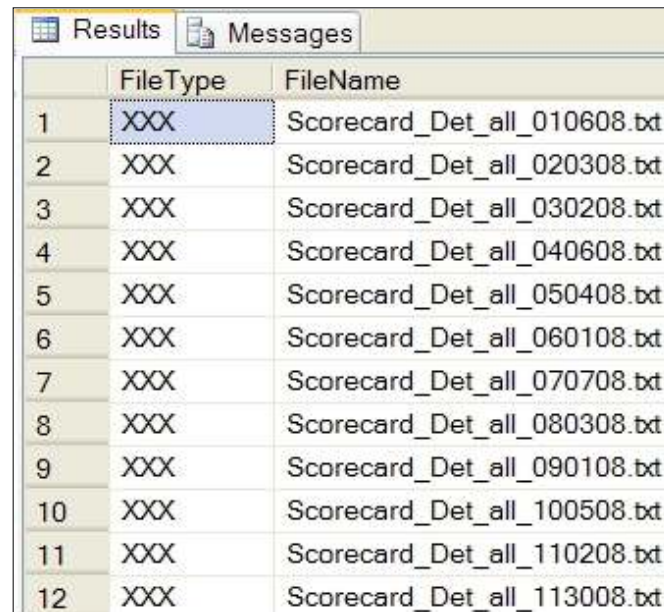
```
DTEXEC /FILE GetFtpFileListing.dtsx [variable assignments go here]
```

The entire DTEXEC command must be specified on a single line; replace [variable assignments go here] with the following:

```
/SET \Package.Variables[FtpFileType].Properties[Value];%FTPFILETYPE%
```

```
/SET \Package.Variables[FtpServer].Properties[Value];%FTPSERVER%  
/SET \Package.Variables[FtpWorkingDirectory].Properties[Value];%FTPWORKINGDIRECTORY%
```

After running the package you can query the stg_FtpFileList table to get the list of files available in a particular folder on an FTP server; e.g.:



The screenshot shows the 'Results' pane in SQL Server Enterprise Manager. It displays a table with two columns: 'FileType' and 'FileName'. There are 12 rows of data, all with 'FileType' as 'XXX' and 'FileName' as 'Scorecard_Det_all_XXXXXX.txt', where XXXXX represents a 5-digit number from 010608 to 113008. The first row is highlighted with a blue selection bar.

	FileType	FileName
1	XXX	Scorecard_Det_all_010608.txt
2	XXX	Scorecard_Det_all_020308.txt
3	XXX	Scorecard_Det_all_030208.txt
4	XXX	Scorecard_Det_all_040608.txt
5	XXX	Scorecard_Det_all_050408.txt
6	XXX	Scorecard_Det_all_060108.txt
7	XXX	Scorecard_Det_all_070708.txt
8	XXX	Scorecard_Det_all_080308.txt
9	XXX	Scorecard_Det_all_090108.txt
10	XXX	Scorecard_Det_all_100508.txt
11	XXX	Scorecard_Det_all_110208.txt
12	XXX	Scorecard_Det_all_113008.txt

Next Steps

- While SSIS provides quite a few useful tasks, occasionally you need something more. The Script task is likely a good alternative as it allows you to execute any VB .NET code inside of your SSIS package.
- Take a look at the sample code [here](#) to experiment on your own.
- The FtpClientConnection class used in the above example has other properties and methods that you may find useful; you can get the details [here](#).
- The FTP task also has other properties that you may need to set such as, user name, password, etc.

Copyright (c) 2006-2008 [Edgewood Solutions, LLC](#) All rights reserved
[privacy statement](#) | [disclaimer](#) | [copyright](#)

Some names and products listed are the registered trademarks of their respective owners.