Using ActiveX Scripts in a DTS Workflow

New Information - SQL Server 2000 SP3.

You can use Microsoft® ActiveX® scripts to customize the execution of steps in a Data Transformation Services (DTS) package. Because the code is run before the steps executes, you can use an ActiveX script in a workflow to:

- Restart a workflow.
- Turn off a step under certain conditions.
- Initiate retries of connections and other operations.
- Implement loop conditions.

> **SECURITY NOTE** Scripts can be the source of security vulnerabilities; they can invoke system functions without user knowledge or intervention and may contain security credentials in plain text. Review the script for security issues before use. For more information, see Security and Scripting.

You can also use a step ActiveX script to initialize or reference global variables. For more information, see Using Global Variables with DTS Packages.

To add ActiveX workflow scripts in DTS Designer

⊞ Enterprise Manager

The following examples show how to implement ActiveX workflow scripts in DTS Designer. You can also add workflow ActiveX scripts programmatically. For more information, see Adding DTS ActiveX Scripts.

Turning a Step On and Off

The following example, written in Microsoft Visual Basic® Scripting Edition (VBScript), uses an ActiveX script to turn a step on or off based on the existence of a file:

```
Function Main()

      Dim fso 'File system object
      Set fso = CreateObject("Scripting.FileSystemObject")
      IF (fso.FileExists("C:\temp\download.tmp")) THEN
         Main = DTSStepScriptResult_ExecuteTask
      ELSE
         Main = DTSStepScriptResult_DontExecuteTask
      END IF

End Function
```

Retries

The following example, written in VBScript, checks for the presence of a file five times before terminating the step. The global variable *retries* stores the number of attempted file checks:

```
Function Main()

      Dim fso
      Set fso = CreateObject("Scripting.FileSystemObject")
      IF NOT(fso.FileExists("C:\MyFile.txt")) THEN
         DTSGlobalVariables("retries").Value  =
               DTSGlobalVariables("retries").Value  + 1
         IF DTSGlobalVariables("retries").Value > 5 THEN
```

```
            Main = DTSStepScriptResult_DontExecuteTask
         ELSE
            MsgBox "Retry #" &  DTSGlobalVariables("retries").Value
            Main = DTSStepScriptResult_RetryLater
         END IF
      ELSE
         Main = DTSStepScriptResult_ExecuteTask
      END IF

End Function
```

Implementing Loop Conditions

The following example of ActiveX script step code, written in VBScript, is assigned to the second step in a two-step workflow connected with a precedence constraint (Step 1 -> On Completion precedence constraint -> Step 2).

The ActiveX script associated with Step 2 serves as a loop. The script operates as follows:

1. Creates the global variable *counter* (initialized to 0 by default), which is incremented each time Step 2 is executed.

2. For the first four times Step 2 is executed, a message box with the value of *counter* is displayed, and the execution status of the previous step is set to waiting, which causes the task associated with Step 1 to restart. The task associated with Step 2 is not executed.

3. When the value of *counter* reaches five, the task associated with Step 2 is run, and the package completes execution.

```
Function Main()

   Dim oPkg
      DTSGlobalVariables("counter").Value = _
      DTSGlobalVariables("counter").Value + 1

      If DTSGlobalVariables("counter").Value < 5 THEN
         Msgbox DTSGlobalVariables("counter").Value
         Set oPkg = DTSGlobalVariables.Parent

         'Set previous step status to waiting.
         oPkg.Steps("DTSStep_DTSActiveScriptTask_1").ExecutionStatus = _
            DTSStepExecStat_Waiting

         'Do not execute task 2, step 1 will restart.
         Main = DTSStepScriptResult_DontExecuteTask

      Else
         'Execute task 2, do not restart step 1.
         Main = DTSStepScriptResult_ExecuteTask
      END IF

End Function
```