

# Use DTS Packages in SQL Server 2000

Automate the database reporting process.

By Alok Mehta and Steven T. Martino

**A**s a database administrator (DBA) or an application developer, you may want to send reports to yourself or users whenever a new record is added in an important table within your database. In this article, we show you how to harness the power of SQL Server 2000 to easily automate the reporting process. By creating VBScript packages in SQL Server's Data Transformation Services (DTS), you can automate the creation and distribution of these reports and schedule this to happen whenever you want.

In this article, we demonstrate how a VBScript package can query and send XML data via e-mail to the DBA using VBScript DTS packages, or send e-mail based on the on-success or on-failure workflow task.

## What is DTS?

Companies have to centralize and consolidate data because their data is usually stored in a variety of formats and in different locations. DTS addresses this issue by providing a set of tools that lets you retrieve and transform data from different sources into a channeled destination. DTS tools provide a GUI and an object model you can easily program to develop powerful solutions. You can use DTS in many contexts, from performing a

one-time data transfer to developing complex workflow-driven packages.

## What are DTS packages and tasks?

A DTS package is a combination of connections, DTS tasks, DTS transformations, and workflow constraints. You can easily assemble the DTS package with a graphical DTS tool. When executed, the package connects to the correct data sources, copies data and database objects, transforms data, and notifies other users or processes of events. Usually, a DTS package includes one or more DTS tasks. Each task defines a work item to be performed as part of the data movement and data transformation process. VBScript is a type of task within the DTS package.

## Why use VBScript as a task?

A VBScript task can perform functions that aren't available in the other tasks in DTS. You can extend the capabilities of your DTS package by using VBScript; for example you can:

- Create and use ActiveX Data Objects (ADO) connections, commands, recordsets, and other objects to access and manipulate data.
- Format and transform the data using functions, subroutines, and COM objects.

### TECHNOLOGY

- SQL Server 2000
- VBScript
- Data Transformation Services
- ADO

Alok Mehta is the senior vice-president and chief technology officer of American Financial Systems, Inc. (AFS). AFS is a software development firm in Weston, Massachusetts. Alok has more than 12 years experience as a software developer. Alok has a B.S. degree in mathematics and physics. He holds an MBA degree from Plymouth State College and an M.S. in computer science from Northeastern University. He is also pursuing a Ph.D. at WPI.

Steven T. Martino is a vice-president at American Financial Systems, Inc. (AFS). He has been developing financial and insurance-based software since 1989. Steve graduated from the University of Connecticut with a B.A. degree in applied mathematics and computational sciences. He holds an MBA degree in finance from the University of Hartford.

- Create, use, and modify the values stored in DTS global variables and ActiveX script constants.
- Integrate other DTS tasks and workflows.

You can schedule VBScript to run as tasks. The script will run once each time the package calls the task.

## Application scenario

This article's example is a Web-based application that lets users create accounts. When a user account is created, the Web application will add a record to the tblSystem\_Users table. At the end of the day, we use a VBScript task in a DTS to create an XML data file of new users that will be sent via e-mail to the manager of this Web site. In addition, we will format this list as an XML document. Although the database we're using is basic, the technique is generic and you can apply it to any data model.

Here are the four steps in the example (figure 1):

1. The Web-based application updates certain key tables with the database application.
2. VBScript DTS package executes automatically daily.
3. VBScript DTS package creates an XML file containing records added today.
4. At the successful completion of an ActiveX script task, the on-success workflow task within the DTS package automatically calls the e-mail task. The e-mail task is pre-configured to send the e-mail with an XML file as an attachment.

If this process fails, the on-failure workflow task within the DTS package automatically calls an e-mail task. This e-mail task is pre-configured to send an e-mail message notifying the recipient of this failure.

## Implementing the sample application

When your Web site generates a new user account, your Web application (or any other process) adds a new record to the tblSystem\_Users table. The Web application fills in the fields that correspond to the user's name and the creation date for the account. Here are the field names and their definitions:

User_First_Name	User's first name
User_Middle_Initial	User's middle initial
User_Last_Name	User's last name
Account_Create_Date	Account creation date

## Understanding the source code

Listing 1 shows the source code for the example application. A description of the code follows.

### Listing 1: VBScript Source Code—Caption needed.

```

*****
'Visual Basic ActiveX Script
*****
'Author:      Alok Mehta and Steven Martino
'Date:       9/1/2002
'New User Report
'You should have SQL Server 2000 and Win2K Server
'installed. Change the following three constants to make
'this example work for you. Or see your DBA or network
'administrator for changing these constants
*****

```

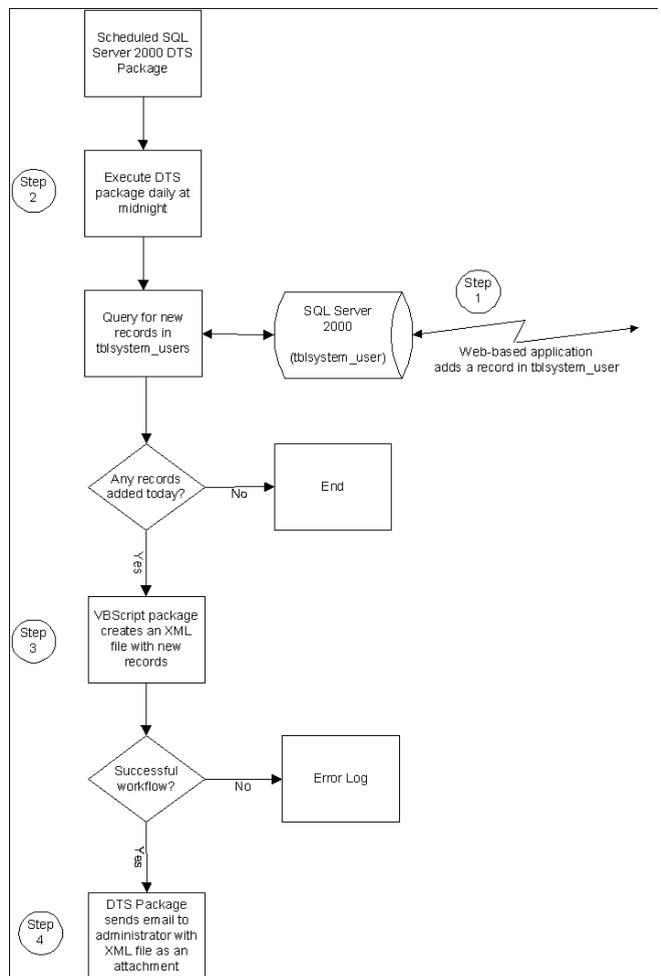


Figure 1: Overview—Here's an overview of this article's example project.

```

'Location of the XML File
Const XML_File = "C:\NewUsers.XML"
Const adPersistXML = 1
'ADO Connection String to the WEB database. Most likely
'you'll have to change your password here. See your DBA.
Const DB_CONNECT_STRING = "Provider=SQLOLEDB.1;Data
Source=(local); Initial Catalog=Pubs;user id = _
'sa';password=''"
.....
'''
'Name:      Main
'Author:    Alok Mehta and Steven Martino
'Date:     1/1/2002
'Purpose:  Calls Process_New_Sales
'Return    DTSTaskExecResult_Success. It is a DTS Task
'Constant can be used by other process or steps.
.....
'''
Function Main()
  If Process_New_Users then
    Main = DTSTaskExecResult_Success
  Else
    Main = DTSTaskExecResult_Failure
  End If
End Function
.....
'''
'Name:      Function Process_New_Users
'Author:    Alok Mehta and Steven Martino
'Date:     1/1/2002
'Purpose:  Based upon todays date, this function picks up
'Author's book sales in various stores
'Return    DTSTaskExecResult_Success to TRUE or FALSE

```

```

.....
...

Function Process_New_Users()

Dim Todays_Date
Dim strNewUsers
Dim rstNewUsers

'Note that not all databases are populated sales for
'today, so by playing with following two lines you should
'be able to generate a report

Todays_Date = "" & Date() & "" 'Not all databases have
data for today

'Get list of authors with sales
strNewUsers = "Select * from tblSystem_Users Where _
account_create_date = " & Todays_Date
'Get the ADO Recordset
Set rstNewUsers = ExecuteSQL(strNewUsers)
'If the Recordset is not empty then proceed
If Not (rstNewUsers.Eof and rstNewUsers.Bof ) Then
call DeleteAFile(XML_FILE)
call rstNewUsers.Save(XML_File,adPersistXML)
Process_New_Users = TRUE
Else
Process_New_Users = FALSE
End If

'Clean up variables
Set rstNewUsers = Nothing

End Function

.....
'Name: Function ExecuteSQL
'Author: Alok Mehta and Steven Martino
'Date: 9/1/2002
'Purpose: Executes a SQL statement
'Return ADO Recordset
.....

Function ExecuteSQL(mySQLCmdText)
'Instantiate the ADO objects
Dim myConn
Dim myRecordset
set myConn = CreateObject("ADODB.Connection")
set myRecordset = CreateObject("ADODB.Recordset" )

'Set the connection properties to point to the database
'using the constant
myConn.Open = DB_CONNECT_STRING
myRecordset.Open mySQLCmdText, myConn
Set ExecuteSQL = myRecordset

End Function

.....
'Name: Private Sub Delete File
'Author: Alok Mehta and Steve Martino
'Date: 9/1/2002
'Purpose: Returns a recordset in a formatted XML style
'so that it can be written as a file
'Return True if the XML file was written to the disk
'otherwise False
.....
Sub DeleteAFile(filespec)
Dim fso
Set fso = CreateObject("Scripting.FileSystemObject")
If (fso.FileExists(filespec)) Then
fso.DeleteFile(filespec)
Set fso = nothing
End Sub

```

**Constants**

You can adjust the following three constants according to your system and network settings. If you aren't sure about changing them, see your DBA and/or network administrator.

```
Const XML_File = "c:\NewUsers.XML"
```

is the XML report file generated by the VBScript.

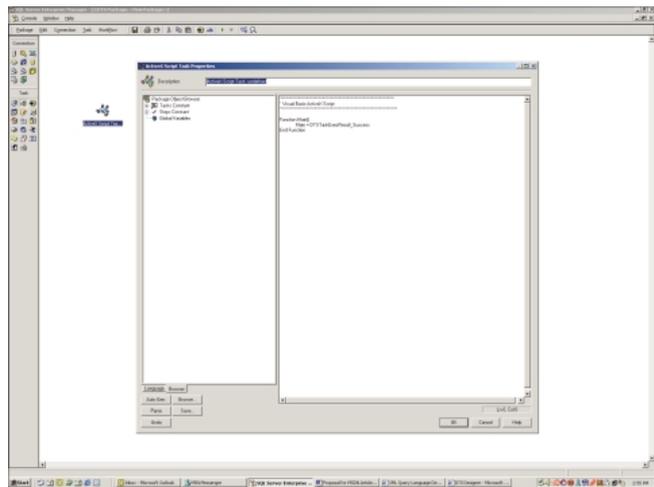
```
Const AdPersistXML = 1
```

sets the format in which to save the recordset in XML format. You will see this constant used in the recordset Save command. This is the ADO connection string:

```
Const DB_CONNECT_STRING = _
"Provider=SQLOLEDB.1;Data
Source=(local); Initial Catalog=Pubs;user id = _
'sa';password=''"
```

**Functions**

By default, the entry function to the ActiveX Task is named Main. However, you can change this based on your application. The Main function consists of only one conditional statement and one statement each for the success or failure of the conditional. The conditional line calls the Process\_New\_Users() function and, based on the return value, will set the return value of the main function to the ActiveX Script Constant of DTSTaskExecResult\_Success if Process\_New\_Users is successful or DTSTaskExecResult\_Failure if Process\_New\_Users is unsuccessful. The Browser button (figure 2) lists all the Constants and Global Variables that various Tasks might set and use. You can use these Constants and Global Variables to control execution of steps within a DTS Package. For the sake of simplicity, we won't describe these Constants and Global Variables in much detail. It's important that Main returns ActiveX Script Constant as DTSTaskExecResult\_Success or DTSTaskExecResult\_Failure because the DTS package will act on this return value.



**Figure 2: DTS Designer (Drag-Drop ActiveX Script Task)**—This Browser button lists the Constants and Global Variables Tasks might set and use.

The function Process\_New\_Users is the brain of the application (see function Process\_New\_Users in listing 1). It gets a recordset containing all the new users created today. You can easily modify this routine to get all the users created since a particular day or all the users created between any two dates. It then creates an XML file containing all the users created today. The function then tells the calling routine whether the creation of the XML file was successful.

Begin by dimensioning three local variables, one ADO recordsets, one string, and one date:

```
Dim rstNewUsers
```

```
Dim strNewUsers
Dim Todays_Date
```

Because not all databases will be populated with new user information for today, you can change this assignment to match the data in your database. We assumed you have data for today, but if you don't please comment out the first line below, uncomment the second line, and fill in an appropriate date:

```
Todays_Date = "" & Date() & ""
'Todays_Date = "8/10/2000"
```

Next, obtain the recordset with a list of all the new users that have been added today. By using a simple SQL query, you can select all users who have a creation date of today. This query is:

```
strNewUsers = "Select * from tblSystem_Users Where _
account_create_date = " & Todays_Date
```

Now convert the SQL statement into an ADO recordset with the following call:

```
Set rstNewUsers = ExecuteSQL(strNewUsers)
```

Next make sure that rstNewUsers is in fact not empty before you create the XML file:

```
If Not (rstNewUsers.EOF and rstNewUsers.BOF ) Then
```

Create an XML file based on the information contained in the recordset rstNewUsers. This may sound like a difficult or tedious task, but it couldn't be simpler.

```
Call rstNewUsers.Save(XML_File,adPersistXML)
```

You can take advantage of ADO's ability to automatically create an XML file from a recordset. All you have to do is provide the XML file name and tell ADO that you want to save this recordset as an XML file by setting the PersistFormat parameter as adPersistXML.

Next, you indicate to the calling program that this operation has completed successfully:

```
Process_New_Users = TRUE
```

If the recordset of new users rstNewUsers had been empty, you would indicate that this operation had failed with the following instruction:

```
Process_New_Users = FALSE
```

Finally, clean up by clearing out our recordset object rstNewUsers:

```
Set rstNewUsers = Nothing
```

The first function we'll describe is the Function ExecuteSQL(strSQL As String) because this function is only function to connect directly to the database. Inside this function we access the database, execute the SQL string provided by the caller, and return the results, if there are any. We create an ADO connection:

```
Set myConn = CreateObject("ADODB.Connection")
```

Then we create an ADO recordset:

```
set myRecordset = CreateObject("ADODB.Recordset" )
```

Next, we open the connection using the DB\_CONNECT\_STRING constant:

```
myConn.Open = DB_CONNECT_STRING
```

After the connection is open, we open the recordset using the connection and the SQL:

```
myRecordset.Open mySQLCmdText, myConn
```

Finally, we return the results of opening the recordset:

```
Set ExecuteSQL = myRecordset
```

If you have spent any time working with ADO, you'll see this is straightforward.

## Creating a new DTS package

Here are the steps to create a new DTS package.

Go to Start > Programs > Microsoft SQL Server > Enterprise Manager.

Right-click on the Data Transformation Services (DTS) of the console tree, as shown in (figure 2). This step will bring you into DTS Designer.

## Creating a new VBScript Task within the DTS Package (a.k.a., DTS Designer)

Using DTS Designer's graphical user interface, you can build and configure packages by using drag-and-drop methods and completing property sheets on the various DTS objects composing the package. To create a VBScript Task, simply drag and drop the ActiveX Script Task from the task toolbar to the main window pane (figure 2). Change the name of the Task from "ActiveX Script Task: undefined" to "Create New User Report."

## Copy the functions

Within the ActiveX Script Task properties window, make sure the language is set to "VBScript Language." You can set the language by clicking on the language button (figure 2). To copy the code provided in listing 1, follow these steps:

1. Remove the function Main that's already in the DTS Designer by highlighting/selecting and deleting.
2. Either type in the source code or cut and paste from an electronic copy.

You can use all the VBScript keywords from the left panel. Using the browse button, you can also open another VBScript file.

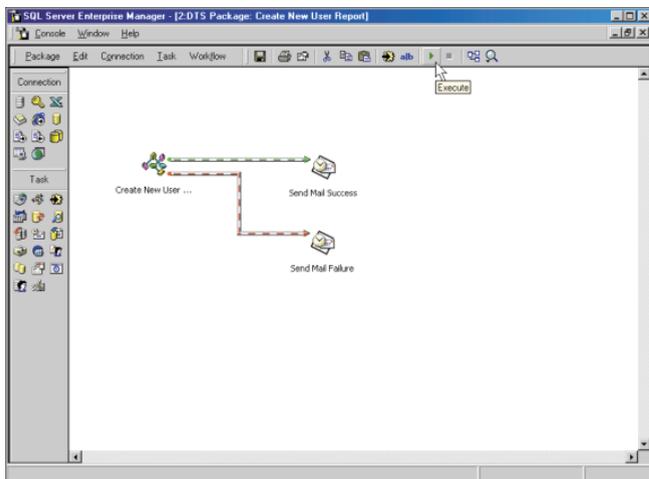
## Change constants to make the sample application work for you

You must make sure the database connection string is appropriately configured. Please refer to the source code section to get more details on these constants. See your DBA and/or network administrator if you aren't sure about changing the constants. Be sure to save the task when done (figure 3). We name the task "Create New User Report."

## Test the application

You can test the application two ways:

Click on the parse button to make sure you don't have any syntax errors. Note this is available within the ActiveX Script Task properties window.



**Figure 3: Save the task**—Execute the create new user package.

Click on the green triangle in the toolbar of the DTS Designer. You can also right-click on the VBScript Task and select Execute Step. Lastly, you can also execute the Create New User Report package (figure 3) within the Enterprise Manager.

## E-mail package

If the "Create New User Report" VBScript task completes successfully, you want to send an e-mail to the administrator indicating new users have been created and send the XML document containing the new user information. If this task fails, we'll send an e-mail stating there was an error creating this XML document. We can provide this functionality by creating two send mail tasks. We'll call one Send Mail Success and the other Send Mail Failure.

Follow these steps to create the Send Mail Success task:

1. From the Task menu select Send Mail Task
2. Fill in the To: field with the administrator's e-mail address or the address of the proper recipient.

In the Subject field, we've added the text "New Users Added." You can add any relevant message in the Message field.

3. Add the XML document (C:\NewUsers.XML) as an attachment.

Now create the Send Mail Failure task:

1. In the Description field, add Send Mail Failure.
2. From the Task menu, select Send Mail Task.
3. Fill in the To: field with the administrator's e-mail address or the address of the proper recipient.

In the Subject field, we've added the text "Error Generating New User Report." You can add any relevant message in the Message field.

Now that we've created our two send mail tasks, we need to tie the execution of these tasks to the success or failure of the "Create New User Report" VBScript tasks:

1. Hold down the CTRL key and select the Create New User Report and the Send Mail Success tasks.
2. On the Workflow menu, select On Success.

You'll see a green striped line between the Create New User Report and the Send Mail Success tasks. This indicates the Send Mail Success task will be executed if Create New User Report completes successfully.

3. Hold down the CTRL key and select the Create New User Report and the Send Mail Failure tasks.

4. On the Workflow menu item, select On Failure.

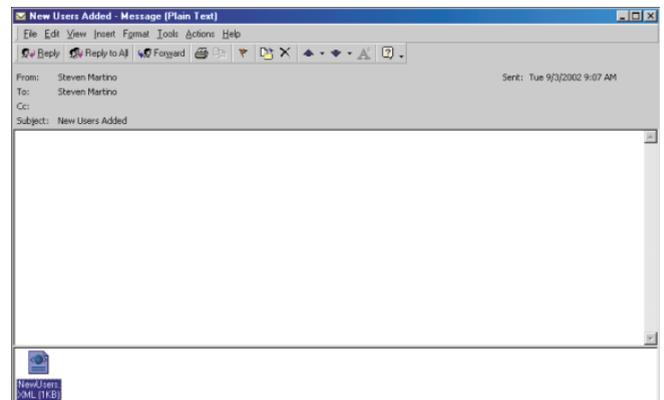
You'll see a green striped line between the Create New User Report and the Send Mail Failure tasks. This indicates the Send Mail Failure task will execute if the Create New User Report completes unsuccessfully.

## Workflow Connector

Click on the green triangle in the toolbar of the DTS Designer. You can also right-click on the VBScript Task and select Execute Step.

## Scheduling the VBScript task

You can automatically schedule the DTS package to run by selecting the Schedule Package menu item. Enter the parameters (). Using the parameters, the DTS package will run daily at 11:00 p.m. starting from 1/1/2002. Note that SQL Server Agent must be loaded and configured for you to schedule DTS packages. You should see the DTS package as a job within Enterprise Manager > Management > SQL Server Agent > Jobs. Figure 4 shows the e-mail message and attached XML file that are sent when this package runs successfully.



**Figure 4: Success**—If the package runs successfully, the e-mail message that contains the XML file is sent.

## Conclusion

In this article, we have shown you how to push information using SQL Server 2000, VBScript, ADO, and DTS Packages. We have used a simple SQL statement to query the data model. You can certainly use a more complex data model and external reporting components to send fancier reports to either yourself or to your users. Furthermore, you can also integrate other DTS tasks in the VBScript task. This technique can be used to push content to your users in many applications, such as providing timely financial statements, sending order status of goods, or sending a copy of clients' data monthly. **ADVISOR**