http://www.sqlservercentral.com/articles/DTS/overviewofdtspackages/2127/
Printed 2007/12/21 06:24AM

Overview of DTS Packages

By Rama Nageshwara, 2005/10/13

DTS Introduction

Many organizations need to centralize data to improve corporate decision-making. However, their data may be stored in a variety of formats and in different locations. Data Transformation Services (DTS) addresses this vital business need by providing a set of tools that lets you extract, transform, and consolidate data from disparate sources into single or multiple destinations supported by DTS connectivity. By using DTS tools to graphically build DTS packages or by programming a package with the DTS object model, you can create custom data movement solutions tailored to the specialized business needs of your organization.

DTS Basics

Data Transformation Services (DTS) provides a set of tools that lets you extract, transform, and consolidate data from disparate sources into single or multiple destinations. You create a DTS solution as one or more *packages*. Each package may contain an organized set of tasks that define work to be performed, transformations on data and objects, workflow constraints that define task execution, and connections to data sources and destinations. DTS packages also provide services, such as logging package execution details, controlling transactions, and handling global variables.

DTS supplies a number of tasks that are part of the DTS object model that can be accessed graphically, through DTS Designer, or programmatically. These tasks, which can be configured individually, cover a wide variety of data copying, data transformation, and notification situations. For example:

- **Importing and exporting data**
  DTS can import data from a text file or an OLE DB data source (for example, a Microsoft Access 2000 database) into SQL Server. Alternatively, data can be exported from SQL Server to an OLE DB data destination (for example, a Microsoft Excel 2000 spreadsheet). DTS also allows high-speed data loading from text files into SQL Server tables.
- **Transforming data**
  DTS Designer includes a Transform Data task that allows you to select data from a data source connection, map the columns of data to a set of transformations, and send the transformed data to a destination connection. DTS Designer also includes a Data Driven Query task that allows you to map data to parameterized queries.
- **Copying database objects**
  With DTS, you can transfer indexes, views, logins, stored procedures, triggers, rules, defaults, constraints, and user-defined data types in addition to the data. In addition, you can generate the scripts to copy the database objects.
- **Sending and receiving messages to and from other users and packages**
  DTS includes a Send Mail task that allows you to send an e-mail if a package step succeeds or fails. DTS also includes an Execute Package task that allows one package to run another as a package step, and a Message Queue task that allows you to use Message Queuing to send and receive messages between packages.

- **Executing a set of Transact-SQL statements or Microsoft ActiveX® scripts against a data source**
  The Execute SQL and ActiveX Script tasks allow you to write your own SQL statements and scripting code and execute them as a step in a package workflow.

Using DTS Designer

The DTS Designer interface consists of a work area for building packages, toolbars containing package elements that you can drag onto the design sheet, and menus containing workflows and package management commands.

Connections: Accessing and Moving Data

To successfully execute DTS tasks that copy and transform data, a DTS package must establish valid connection(s) to its source and destination data and to any additional data sources, such as lookup tables. When creating a package, you can configure connections by selecting a connection type from a list of available OLE DB providers and ODBC drivers. Table 1 shows the connection details.

Table 1: Connections

	Connection	Description
	Microsoft OLE DB Provider for SQL Server	

Tasks: Defining Steps in a Package

A DTS package usually includes one or more tasks. Each task defines a work item that may be performed during package execution. Table 2 shows some of the tasks generally used.

Table 2: Tasks generally used

	Task	Description
	Transform Data task	Use to move data between a source and destination and to optionally apply column-level transformations to the data.
	ActiveX Script task	Use to write code to perform functions that are not available in the other DTS tasks.
	Send Mail task	Use to send an e-mail message.

Workflows: Setting Task Precedence

When you define a group of tasks, there is usually an order in which the tasks should be performed. When tasks have an order, each task becomes a step of a process. In DTS Designer, you manipulate tasks on the DTS Designer design sheet and use precedence constraints to control the sequence in which the tasks execute. Table 3 shows the workflow precedence and description.

Table 3: Workflow Precedence

Precedence constraint	Description

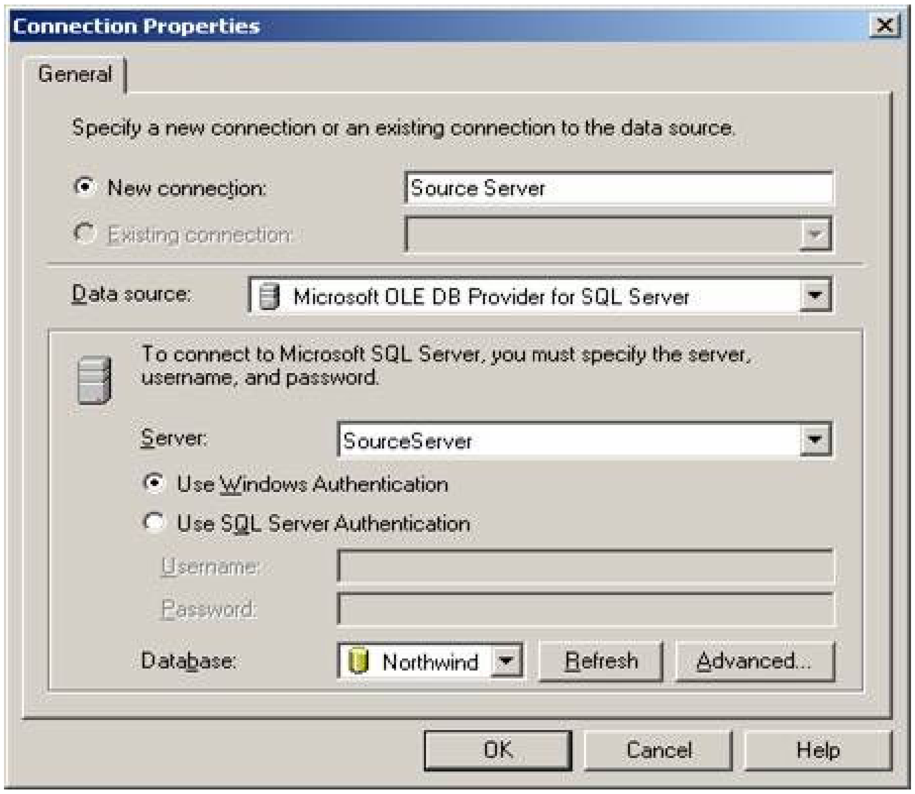
On Completion (blue arrow)	If you want Task 2 to wait until Task 1 completes, regardless of the outcome, link Task 1 to Task 2 with "On Completion" precedence constraint.
On Success (green arrow)	If you want Task 2 to wait until Task 1 has successfully completed, link Task 1 to Task 2 with "On Success" precedence constraint.
On Failure (red arrow)	If you want Task 2 to begin execution only if Task 1 fails to execute, link Task 1 to Task 2 with an "On Failure" precedence constraint.

Scenario

Assume that you want to transfer data from one table to another table using some parameters (here date is considered as a parameter). One can pass the date values to the query by declaring "Global Variables". Before transferring the data we can check the date values using ActiveX Script. After the data transfer is complete we can send a mail to a person to convey the status of the task as "Success" or "Failure".

Creating a Connection object

To establish a connection, click on "Microsoft OLE DB Provider for SQL Server" from Connection tab. Here you can mention the connection details of the database server. For source and destination database servers you need to create two connection objects. Figure 1 illustrates the steps.



Specify a new connection or an existing connection to the data source.

New connection: Source Server

Existing connection:

Data source: Microsoft OLE DB Provider for SQL Server

To connect to Microsoft SQL Server, you must specify the server, username, and password.

Server: SourceServer

Use Windows Authentication

Use SQL Server Authentication

Username:

Password:

Database: Northwind

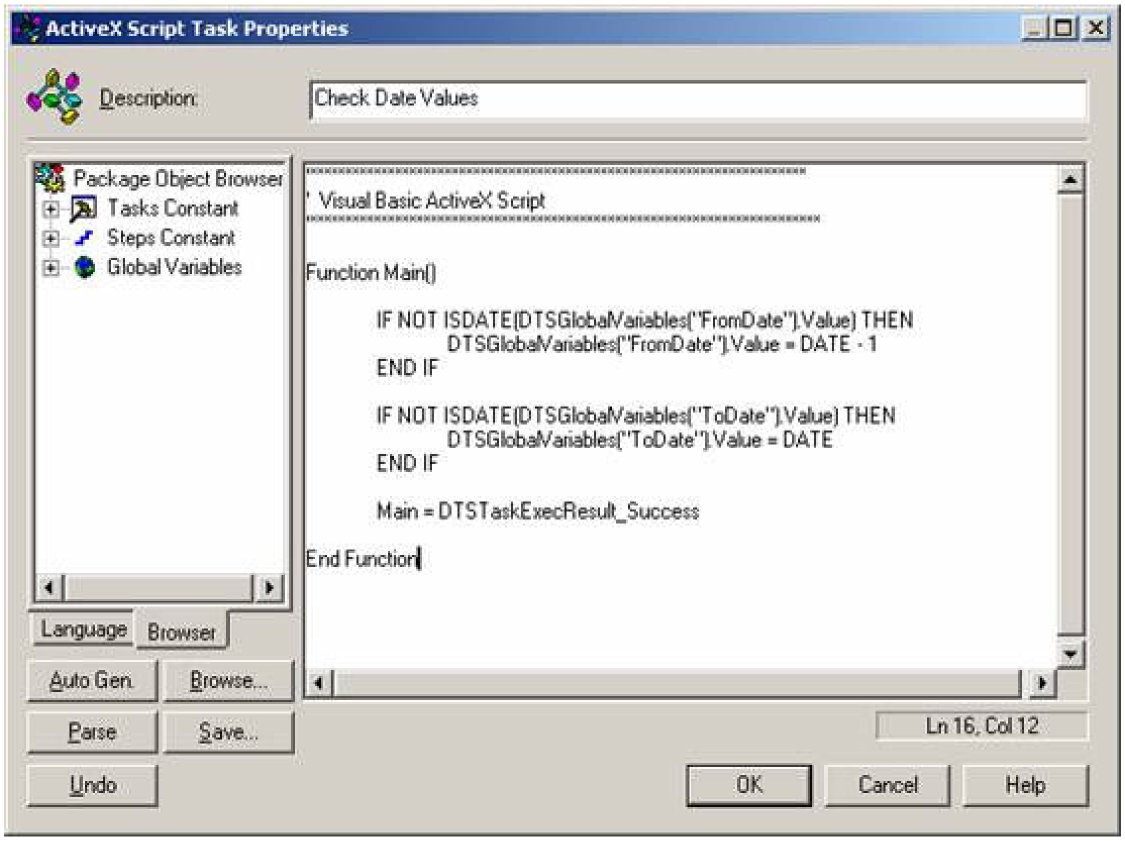
Figure 1: Connection Properties

Creating Global Variables

To pass the variable(s) to a query we need to create global variable(s). These variable(s) can also be used in ActiveX Script also. To create a global variable right click on the DTS Package and select "Package Properties" and select "Global Variables" tab. Click on the "New" button to create a new global variable. Note that global variables won't reset the values to the null after execution of the package i.e., it contains the values before execution of the package. In our scenario we create two global variables: "FromDate" and "ToDate".

Creating ActiveX Script

You can write your own ActiveX Script in VB language. You can run these scripts before or after the required tasks. Here the script is used to validate the date stored in global variables. If the date given in global variables are correct then those dates are considered, if those dates are not in valid format then for FromDate we will consider yesterday's date and for ToDate we will consider today's date. Figure 2 shows the ActiveX Script for this scenario.



Description: Check Date Values

```
'**********************************************************************
'  Visual Basic ActiveX Script
'************************************************************************

Function Main()

        IF NOT ISDATE(DTSGlobalVariables("FromDate").Value) THEN
                DTSGlobalVariables("FromDate").Value = DATE - 1
        END IF

        IF NOT ISDATE(DTSGlobalVariables("ToDate").Value) THEN
                DTSGlobalVariables("ToDate").Value = DATE
        END IF

        Main = DTSTaskExecResult_Success

End Function
```

Figure 2: ActiveX Script

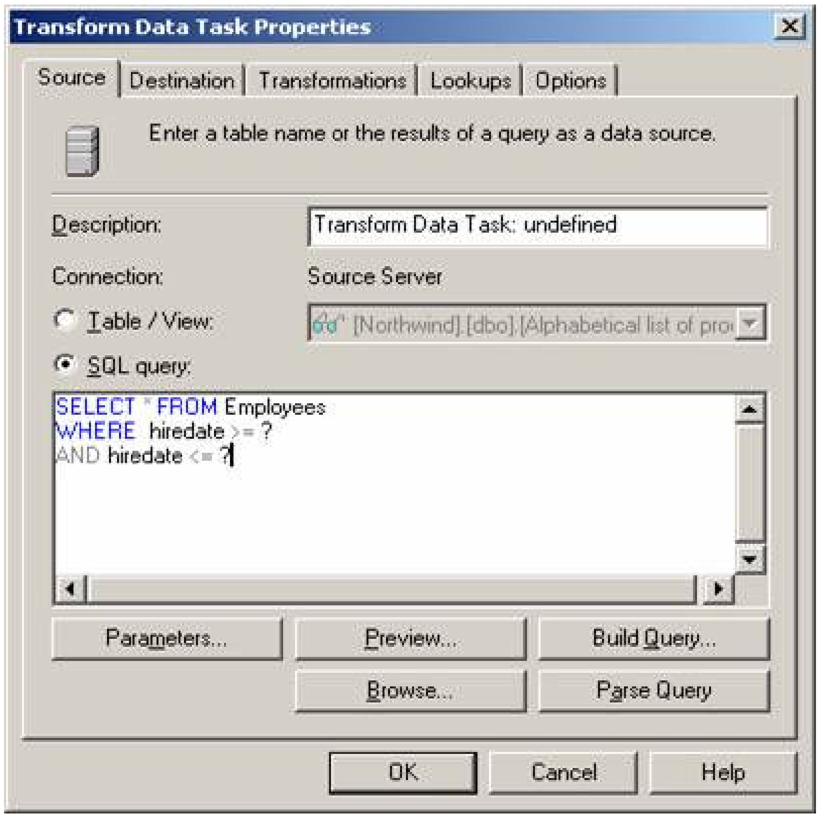
Creating a Transform Data Task

You use the Transform Data task to copy data between a source and destination and to optionally apply column-level transformations to the data. The Transform Data task is the most basic implementation of the data pump engine in Data Transformation Services (DTS). Click on "Transform Data Task" from Task tab. Using this task we can mention the table from which the data needs to be transferred or we can go for "SQL Query". In this scenario we will use "SQL Query".

We can configure the properties associated with this task by right-clicking on the relevant Transform

Data Task and selecting properties. To log an error select “Options” tab from the “Transform Data Task Properties” and enter the exception file name.

Figure 3 shows the SQL query used in “Transform Data Task Properties”.



Enter a table name or the results of a query as a data source.

Description: Transform Data Task: undefined

Connection: Source Server

Table / View: [Northwind].[dbo].[Alphabetical list of pro

[✓ SQL query:]

```
SELECT * FROM Employees
WHERE  hiredate >= ?
AND hiredate <= ?
```

Figure 3: Transform Data Task Properties (Source Tab)

In the SQL query “?” represents the parameters. To map the parameters click on “Parameters” button.

Clicking on Parameters button will show a window with “Input Global Variables” and “Parameters”. Here you can map the global variables to the input parameters.

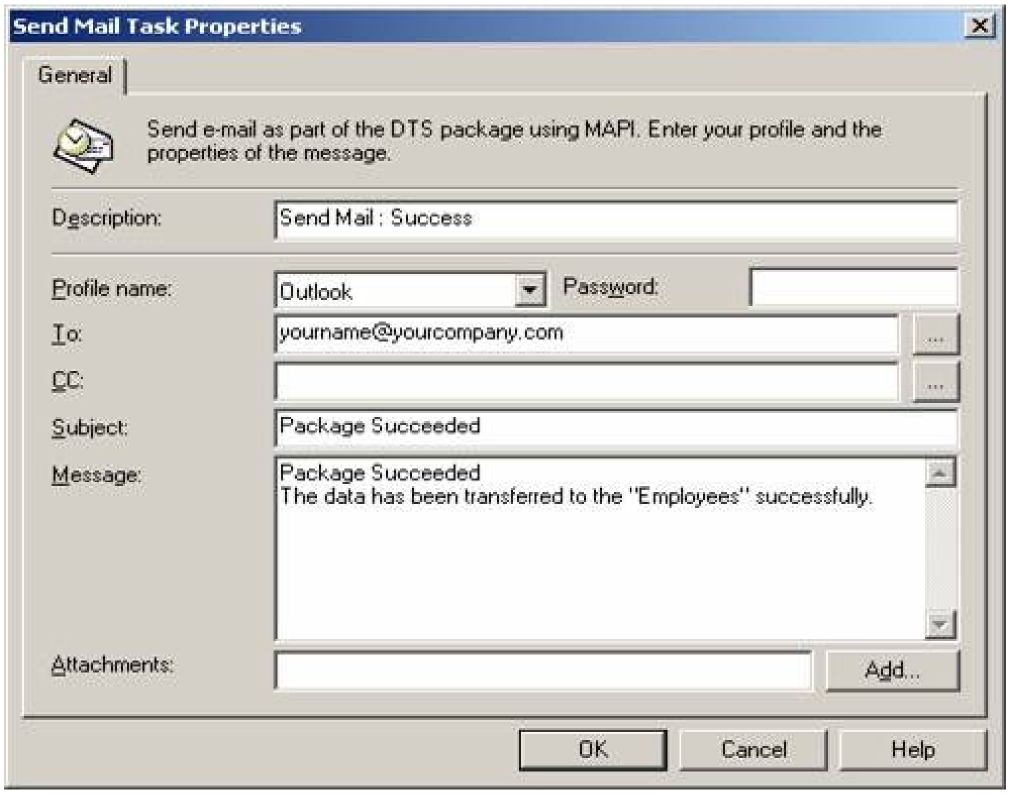
Managing Transactions between two or more tasks

Transactions play a very important role in database. We can set transactions between two tasks by Right clicking the Transform Data Task and select “Workflow Properties” then select “Options” tab. Here you can select “**Join transaction if present**” option or you can choose “Commit transaction on successful completion of this step” option.

You can also set the transaction at package level by right clicking the package and select “Package Properties” then select “Advanced” tab, check the “Use Transactions” checkbox to enable the transactions at package level. By default it is enabled.

Send Mail Task

The **SendMailTask** object lets you send an e-mail as a task. For example, if you want to notify a database administrator about the success or failure of a particular task (such as a backup), you can link a **SendMailTask** object with a precedence constraint to the previous task. To use a **SendMailTask**, the computer must have the Microsoft messaging API installed with a valid user profile. In this scenario we have created two send mail tasks, one for success and other for failure.



Send e-mail as part of the DTS package using MAPI. Enter your profile and the properties of the message.

Description: Send Mail : Success
Profile name: Outlook
Password:
To: yourname@yourcompany.com
CC:
Subject: Package Succeeded
Message: Package Succeeded
The data has been transferred to the "Employees" successfully.
Attachments:

Figure 4: Send Mail Task Properties

Creating Workflow Properties

Using workflow task we can say prioritize which task needs to be executed after task i.e., we can set the precedence of the various tasks. Right click on the task on which you need to define the workflow properties and select "**Workflow**" and then select "**Workflow Properties**". Now click on "New" button to add the precedence. Various precedence available are: Success, Failure and Completion. Select the task as "Source Step" and set the precedence. In this scenario we have created two workflow properties, one for success and other for failure of data transfer task.

Error Logging

Finding if a DTS Package is running

I was stuck at finding a solution to if a DTS Package is executing or not. It could have been started with the scheduler, or started manually. The only solution to this problem which I could come up with without any code changes to the package itself, was to enable Logging to SQL Server, and then checking the state of the package in the **sysdtspackagelog** table by the following query

The various details of the scheduled DTS packages are available in *sysdtspackagelog*, *sysdtssteplog* and *sysdtstasklog* tables in the msdb database. The *sysdtspackagelog* contains the details at package level with columns such as name, lineagefull, start time, end time, computer, operator, logdate, error code, error description etc. The *sysdtssteplog* table contains the details at step level with columns such as lineagefull, step name, start time, end time, error code, error description, etc. The *stsdtstasklog* table contains details at task level. It gives description about the errors occurred in a particular task.

Sample query to get the details of the package from sysdtspackagelog table.

SELECT TOP 1 endtime FROM msdb.sysdtspackagelog WHERE name = 'Your Package Name' ORDER BY starttime DESC

If it returns a date, then the package is not running, and if it returns NULL, then the package is currently in executing stage.

Error Logging in an external file

Using error logging it becomes very easy to track the errors. To achieve this right click "Package Properties" and select "Logging" tab, where we can enter the name of the file in "**Error File**" textbox.

Error log stores the details at package level and at step level also, such as package name, start time, end time, error description, step name, step details, etc.

Logging in Event Viewer

To log the completion of package in the "**Event Viewer**", "Package Properties" and select "Logging" tab, check the "**Write completion status to event log**". To view the event viewer, on the Start menu, point to "Programs/Administrative Tools" and then click on "Event Viewer".

Executing the Package

To execute the package click "Execute" from the Package Menu. The Executing DTS Package dialog box appears, providing step and status information for the two steps. The Package Execution Results dialog box then appears.

Saving DTS Package

To save a DTS package four options are available:

1. Meta Data Services: With this save option, you can maintain historical information about the data manipulated by the package. However, Meta Data Services and the repository database must be installed and operational on your server. You can track the columns and tables that are used by the package as a source or destination. You also can use the data lineage feature to track which version of a package created a particular row. You can use these types of information for decision-support applications.

2. SQL Server: With this default save option, you can store a package as a SQL Server **msdb** table, allowing you to: store packages on any instances of SQL Server on your network; keep a convenient inventory of saved packages in SQL Server Enterprise Manager; and create, delete, and branch multiple package versions during the package development process.

3. Structured Storage File: With this save option, you can copy, move, and send a package across the network without having to store the file in a SQL Server database. The structured storage format allows

you to maintain multiple packages and multiple package versions in a single file.

4. Visual Basic File: With this save option, you can programmatically customize a package created in DTS Designer or the DTS Import/Export Wizard. The option scripts out the package as Visual Basic code, and you can later open the Visual Basic file and modify the package definition in your development environment.

DTS Package Passwords

When you save a package to Microsoft® SQL Server™ or as a structured storage file, you can use DTS package passwords. You use DTS passwords in addition to the Windows Authentication or SQL Server Authentication passwords you use to connect to an instance of SQL Server. The following types of DTS package passwords are available:

1. If you set an **owner** password, the package user needs the password to edit or run the package.

2. If you set a **user** password, you also must set an owner password. Package users with access only to the user password can run the package. However, they can neither open nor edit the package unless they have access to the owner password. If global variables are used in the package then they cannot open the package in design mode also.

Loading a DTS Package

You can open an existing package from a .dts file by right clicking "Data Transformation Services" from Enterprise Manager and select "Open Package" option and mention the filename of the DTS to open. Then you can save this file to your server by using "Save As" option from File Menu.

Download the Code

Author

Rama Nageshwara Rao

Email id - rama.nageshwara@wipro.com

Profile

I am working as a Senior Project Engineer since 2004, primarily focused on Web and Windows applications using ASP.Net , C# and SQL Server 2000. I am an MCAD certified professional and currently working for WIPRO TECHNOLOGIES, INDIA. Worked extensively on Microsoft.NET technologies which include ASP.NET with C# and SQL Server 2000.