SQL Server Integration Services Using Visual Studio 2005: A Beginner's Guide

by Jayaram Krishnaswamy

Packt Publishing. (c) 2007. Copying Prohibited.

Reprinted for SATHYANARRAYANAN SHANMUGAM, Cognizant Technology Solutions

SathyaNarrayanan.Shanmugam@cognizant.com

Reprinted with permission as a subscription benefit of **Skillport**,
http://skillport.books24x7.com/

All rights reserved. Reproduction and/or distribution in whole or in part in electronic,paper or other forms without written permission is prohibited.

Chapter 8: Using a Conditional Split Data Transformation

This chapter shows you how to create a package that can split data based on a given set of criteria into separate flows, a conditional split of data. You will also learn how to use the in-memory data flow destination, the Recordset Destination.

Often, you want to separate out data meeting certain established criteria and direct them to different destinations. The data splitting may be needed for any number of reasons, to focus on a business objective: weed out data that are suspicious; data that does not follow some standard pattern; identify missing data, etc. The data flow transformation, Conditional Split Transformation is ideally suited for such an operation.

Hands-On Exercise: Splitting Data Retrieved from a SQL Server

In order to follow the steps as indicated, you will need a data flow task that connects to a data source, and destinations to which the separated flows can be directed into, which includes a destination for the default flow that does not meet the criteria established for data splitting. You will also need to add and configure an in-memory ADO recordset destination, the Recordset Destination component to visualize the data.

The following are the major steps of this exercise:

- Create a BI Project and add a Data Flow Task. Add and configure the DataReader Source to pull data from the Local SQL Server.
- Add a Conditional Split Transformation.
- Establish a path to Connect DataReader Source with the Conditional Split Data Transformation.
- Configure the Conditional Split Data Transformation.
- Add Recordset Destination(s).
- Configure the Recordset Destination(s).
- Build, execute the package, and review results.

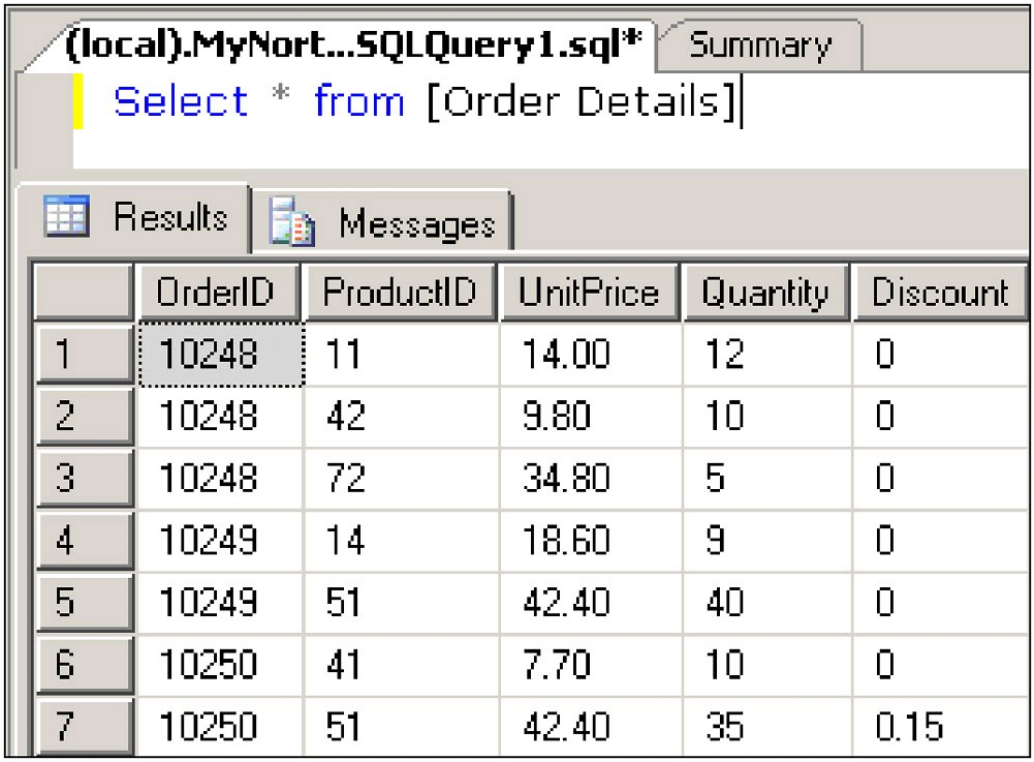
Step 1: Create a BI Project and Add a Data Flow Task. Add and Configure the DataReader Source to Pull Data from the Local SQL Server

Creating a BI project, changing the name of the default Package, adding a DataReader Source, and configuring it to provide at its output data selected from the database has been described in the previous chapters.

1. Create a BI project **Ch 8** and change the default name of the package to `splitter.dtsx`. Add a DataReader Source and provide it with a connection manager [LocalHost.MyNorthwind.sa] that retrieves data from the Local SQL Server.
2. Configure the DataReader Source with the following SQL statement for retrieving the data. **Select * from [Order Details]**.
3. The following screenshot shows the result of running this query directly in the query window of the SQL Server Management Studio.

Reprinted for CTS/272008, Cognizant Technology Solutions

Packt Publishing (c) 2007, Copying Prohibited.



	OrderID	ProductID	UnitPrice	Quantity	Discount
1	10248	11	14.00	12	0
2	10248	42	9.80	10	0
3	10248	72	34.80	5	0
4	10249	14	18.60	9	0
5	10249	51	42.40	40	0
6	10250	41	7.70	10	0
7	10250	51	42.40	35	0.15

Step 2: Add a Conditional Split Transformation

The data output from the DataReader Source will be the input to the conditional split transformation. In a **Conditional Split Data Flow Transformation,** the data flow is read and depending on the conditions and their order of implementation you define, the data is split into components that satisfy those conditions. Those that do not satisfy are separated out into another data flow. It acts very much like a sieve sizing the grains, or in basic computer terms it is based on the **case** statement. This was explained in Chapter 1.

In this present example, our condition for splitting will be based on the value in the **Discount** column shown in the previous screenshot. Specifically, we separate out the data in the table based on those that have a discount > 0 and those that have discount <=0. These are the Boolean conditions that will be implemented.

1. Drag and drop a **Conditional Split** Data Flow Transformation from the **Data Flow Transformations** group in the **Toolbox** to the **Data Flow** page of the canvas, 'Canvas'.

Step 3: Establish a Path to Connect DataReader Source with the Conditional Split Data Transformation

1. Right click the **Data Reader Source** and from the drop-down click on **Add Path**.

   This displays the **Data Flow** window with the "**From:**" showing **Data Source Reader**. The process of establishing a path is same as in the previous chapters.

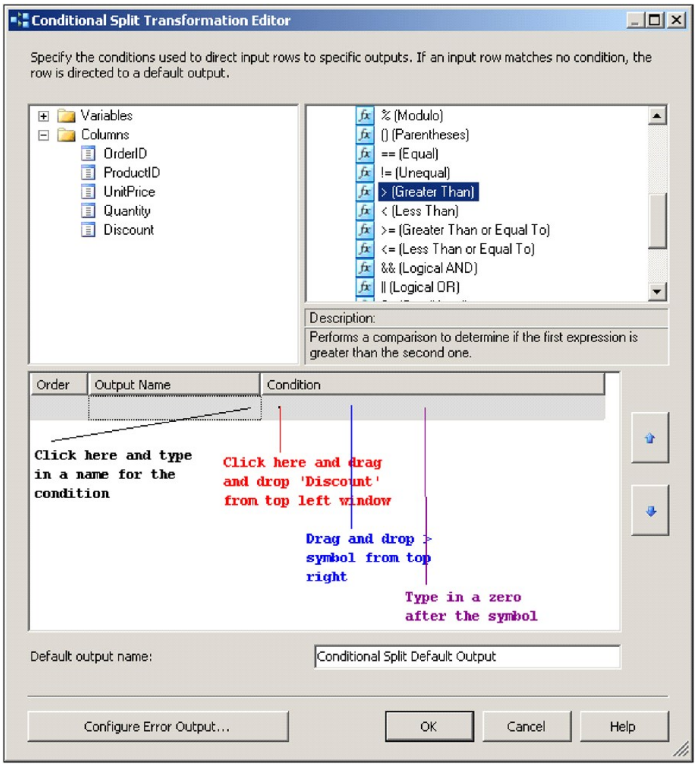
When establishing the path is completed, you should be able to see a thin green line from the **Data Reader Source** to the **Conditional Split** transformation in the canvas.

Step 4: Configure the Conditional Split Data Transformation

1. Right click the **Conditional Split** component on the canvas and from the drop-down menu choose **Edit…**.

   This opens the **Conditional Split Transformation Editor**. Read the instructions in this window. It has three main areas with **Variables** and **Columns** in the left window and some standard core items in the right window. The bottom portion of the window is where the data splitting conditions are developed by drag and drop operations. The variables used in splitting, and the operation needed to fashion the split can both be dragged and dropped. The way the splitting is configured is shown in annotation. In this part of the window, the name for the split is provided in a drop-down list. There is also a provision to order the split by using the arrows on the right, and a button that will take you to configure the default of a conditional split transformation.

Reprinted for CTS/272008, Cognizant Technology Solutions
Packt Publishing (c) 2007, Copying Prohibited.



2. Click under **Output Name** and type in the name "**Case 1**" for this example, the **Order** field gets populated with 1.
3. Now following the schematic (red, blue, magenta instructions) in the above screenshot, format **Discount > 0** in the **Condition** column.
4. Using the same procedure, create Case 2 below Case 1 with **Discount <= 0**.

   By default, Case 1 will be the first and Case 2 is the second in order. This can be changed by using the up/down arrows at the far right of the window.
5. Provide the name, "SplitDefault", for the **Default Output** name.

   In this particular example, we will not configure an error output. The completed window appears as shown in the following screenshot.
6. Click on the **OK** button to complete the Conditional Split Transformation.

The Conditional Split Transformation will be producing three outputs, one each for Discount > 0, Discount <= 0, and the default that is neither of the two.

Step 5: Add Recordset Destination(s)

Although the three streams of data can be channeled to any of the different **Data Flow Destination** types that we have already used in the previous chapters, we will do something a little different here.

When a Recordset Destination component is used in the SSIS designer, an in-memory ADO (ActiveX Data Objects) recordset is created and populated. A **Recordset Destination** requires a variable to be defined where the ADO recordset is going to be stored. The variable name is a string you specify at design time. You may use this variable outside the data flow to be consumed by other elements or used in scripts. It has one input and no error output.

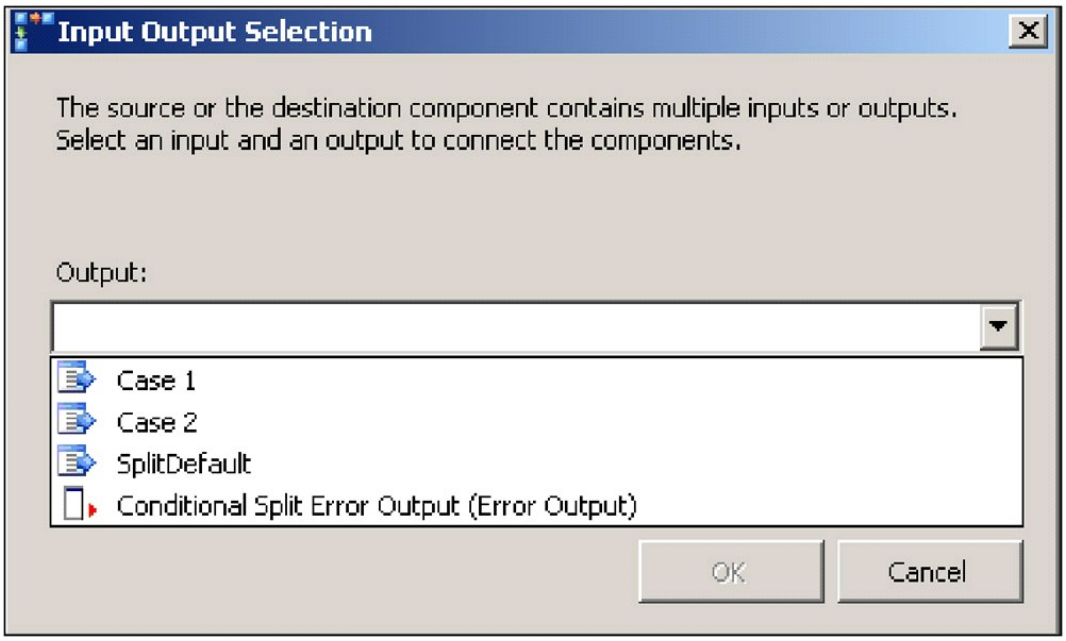
1. Drag and drop (or alternately double-click in the **Toolbox**) a **Recordset Destination** component into the **Data Flow** page of the canvas. "Canvas".
2. Right-click the **Conditional Split** component and choose **Add Path**.

   This opens the Data Flow window displaying the "**From:**" as Conditional Split.

Reprinted for CTS/272008, Cognizant Technology Solutions
Packt Publishing (c) 2007, Copying Prohibited.

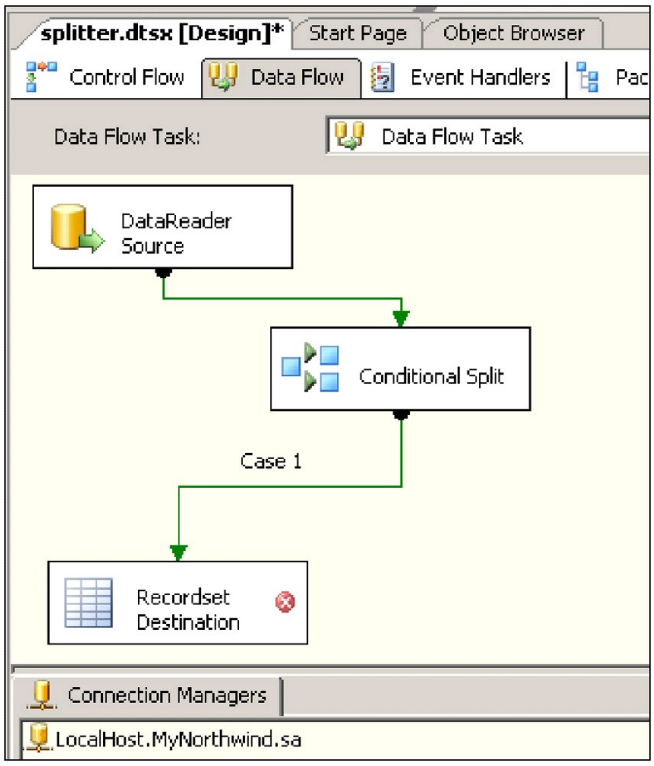
3. Click on the drop-down arrow head in the "**To:**" window. In the drop-down, you will see three options. Choose **Recordset Destination**. Click on the **OK** button in the **Data Flow** window.

   This opens the **Input Output Selection** window with the input displaying "Recordset Destination" and the output from the conditional shows all the previously configured splits as shown in the following screenshot.



4. Click on **Case 1** and click on the **OK** button in the **Input Output Selection** window.

5. At this point, the canvas appears as shown in the following screenshot. The Recordset Destination is not completely configured as yet.

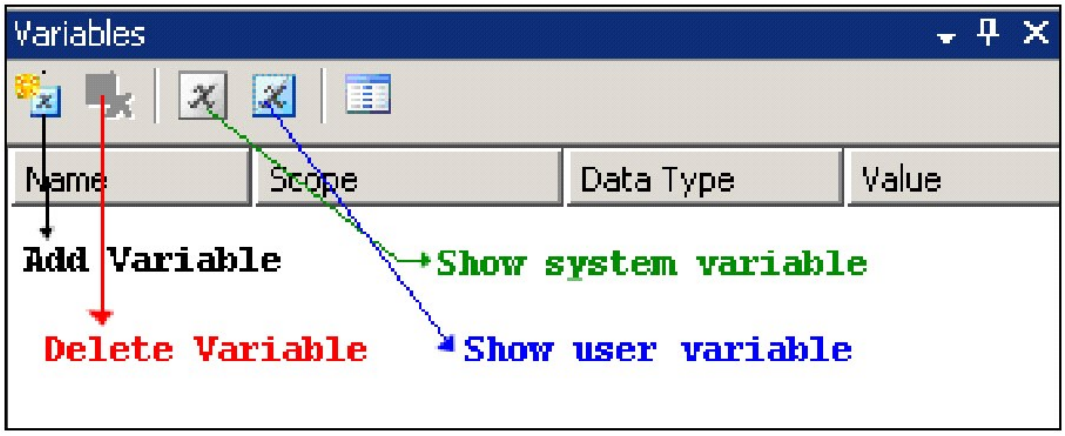


6. Similar to the above, add two more Recordset Destinations, one each for "Case2" and the "SplitDefault" data.

Reprinted for CTS/272008, Cognizant Technology Solutions
Packt Publishing (c) 2007, Copying Prohibited.

Step 6: Configure the Recordset Destination(s)

1. Right-click the **Recordset Destination** (the first one added) component in the canvas and choose the drop-down menu item **Variables**.
2. This opens the **Variables** window as shown in the next screenshot where we need to associate this recordset with a variable. A variable carries the information in the in-memory recordset and that is the reason we need to associate a variable with the recordset destination.

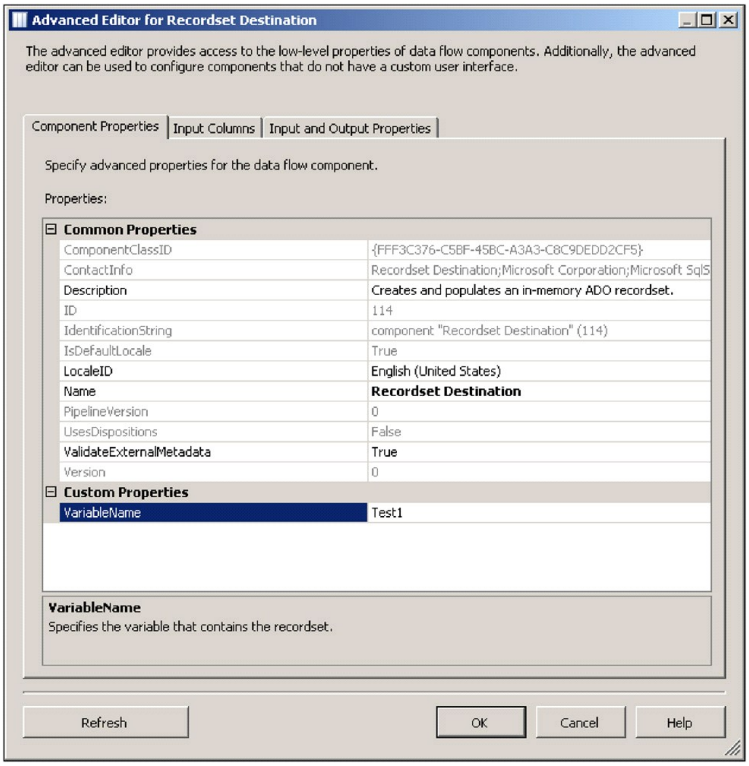


3. Click on the **Add** symbol in the **Variables** window.
4. In the **Variables** window, change the default values as follows: **variable** in **Name** to **Test1** [no space between Test and 1] by typing it in; **int32** to **Object** (choose from drop-down menu) since recordset's data type is object in **Scope**.

   The **Data Type** will display **Object**, and **Value** will display **System.object**. A completed view of this window is presented later.
5. Now right-click the **Recordset Destination** and choose **Edit…**.

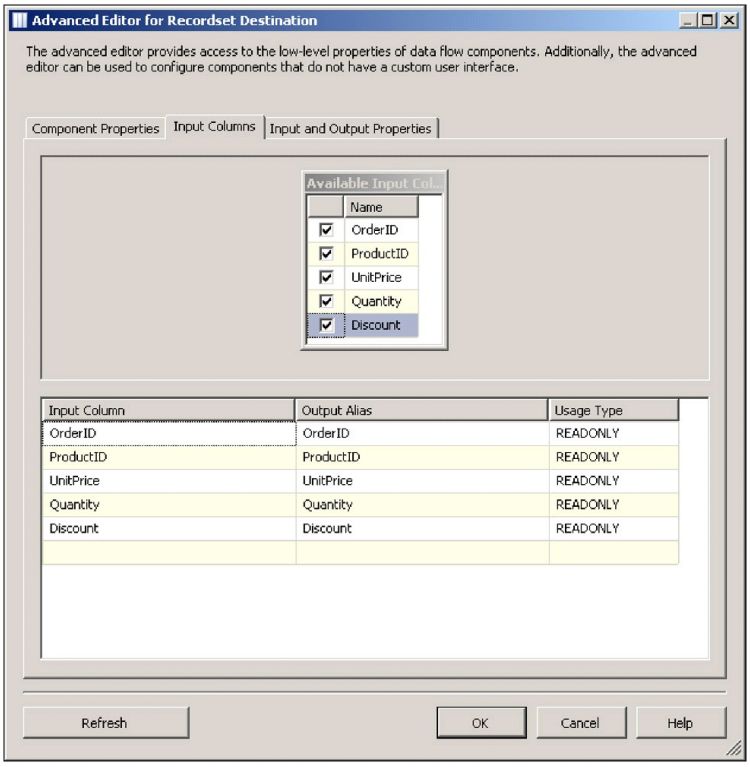
   This opens the **Advanced Editor for Recordset Destination** as shown in the following screenshot. For **VariableName** line item type in **Test1** by its side, as shown. If you did not configure a variable and try to proceed to the next tab you will be faced with an error.

Reprinted for CTS/272008, Cognizant Technology Solutions
Packt Publishing (c) 2007, Copying Prohibited.



6. Click on the **Input Columns** tab to open the **Input Columns** window. Place a check mark for the columns that you want into the recordset (all are chosen for this exercise) as shown in the following screenshot. Click on the **Refresh** button to refresh the data.

7. If you want, you may choose fewer columns as well as their output aliases.



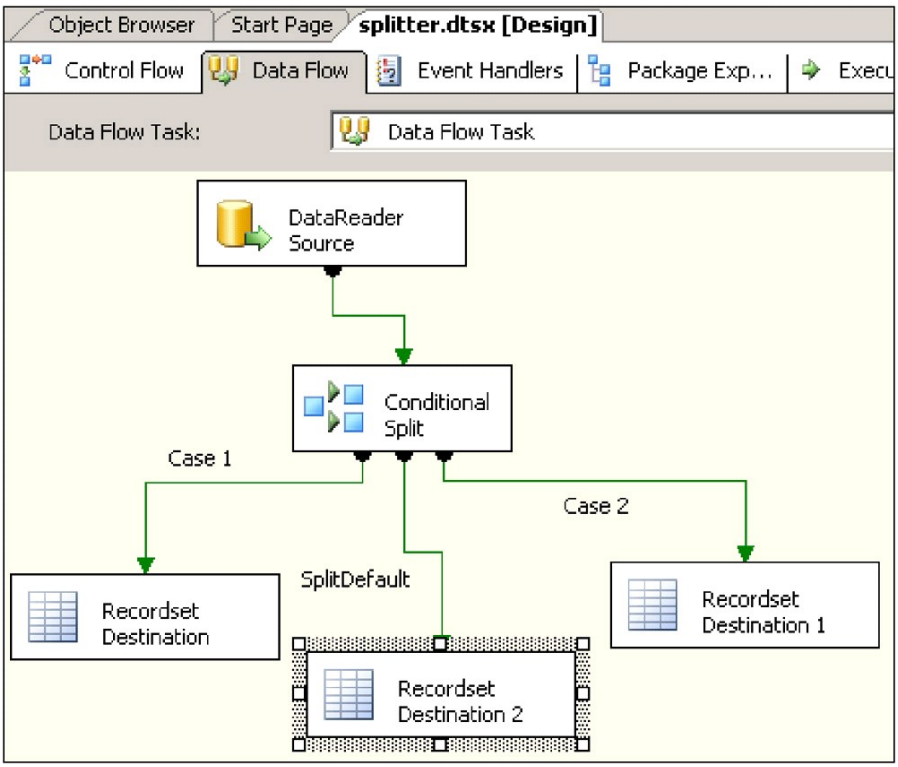
8. Click on the **OK** button because this type of destination does not have an output and no need to go to the next tab, **Input and Output Properties**.

Reprinted for CTS/272008, Cognizant Technology Solutions
Packt Publishing (c) 2007, Copying Prohibited.

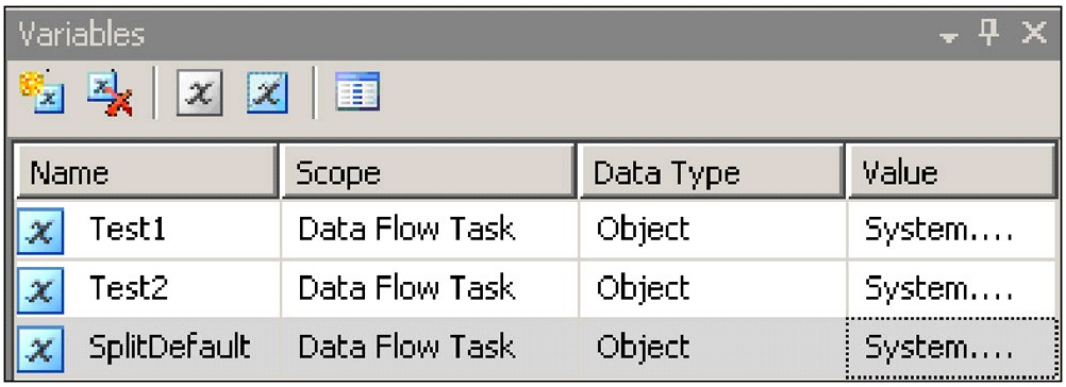
This completes the configuration of the Recordset Destination that will have the data channelled by Case 1.

9. Repeat the same procedure for the other two outputs from the Conditional Split, **Case 2** and **SplitDefault**.

   The completed design in the canvas should appear as shown in the following screenshot.



The associated **Variables** window with this task should appear as in the following screenshot.

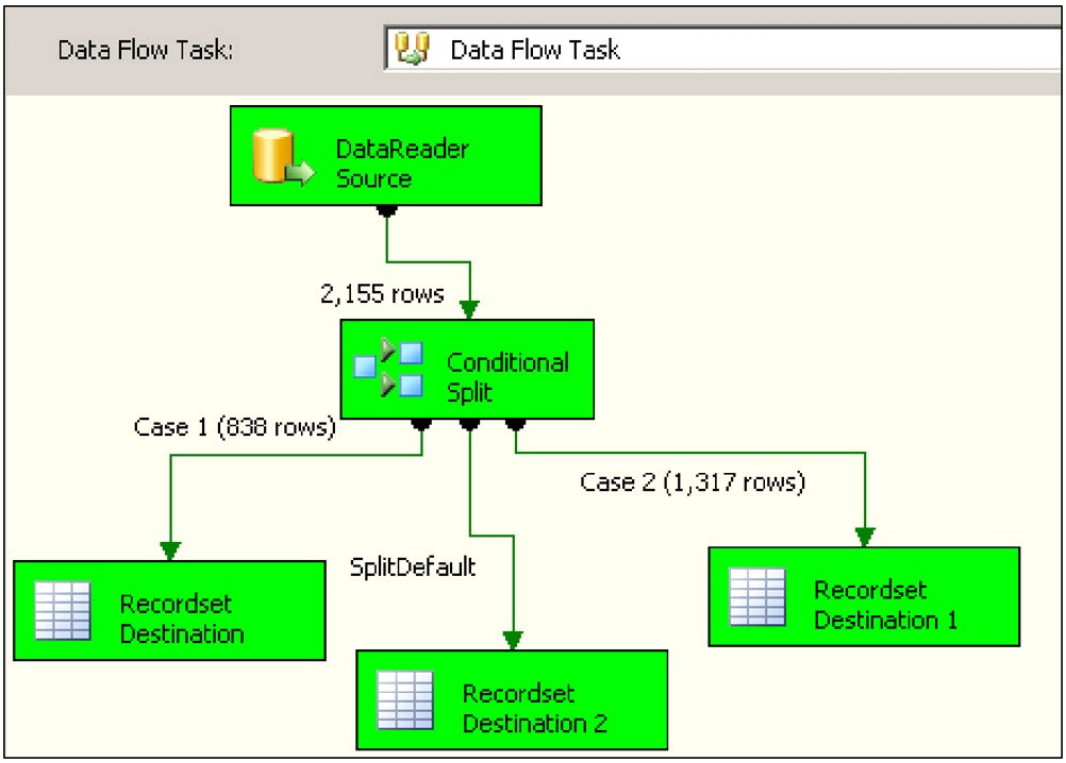


Step 7: Build, Execute the Package and Review

Build the project and execute the package, similarly to hie you did in other chapters. The program runs, and after a while you will see that all the objects turn green, indicating success as shown in the following screenshot. You will also see that 828 records went to the Recordset Destination for Case 1, and 1317 records went over to Case 2 making a total of 2155 rows. None went to the default. You may verify whether this is correct by running suitable select queries [for example, select count(*)from [Order Details] where Discount > 0] in the SQL Server.

Reprinted for CTS/272008, Cognizant Technology Solutions

Packt Publishing (c) 2007, Copying Prohibited.



Summary

This chapter described in detail the usage of a Conditional Split data transformation. You should use this generic task wherever you want to branch out based on some business logic. This logic can be something that is a part of the data itself, or some external conditions. Although the split data was channeled to three Recordset Destinations, one may very well use other types of destinations as well. In further chapters, we will see how the variable used in the Recordset Destination transformation can be used (at present it is showing the number of rows that are held therein).

Reprinted for CTS/272008, Cognizant Technology Solutions

Packt Publishing (c) 2007, Copying Prohibited.