

# 5

## Transferring Data to a Microsoft Excel File

This chapter shows you how to create a package that can transfer data from a table in an SQL Server 2005 Server database to a Microsoft Excel Spreadsheet. You will also learn how to use a Character Map Data Transformation. In the hands-on exercise, you will be transferring data retrieved from an SQL 2005 table to MS Excel 2003 spreadsheet file. You will be using a Data Flow Task consisting of a source connected to a SQL 2005 Server-based connection manager, as in the previous chapter, and an Excel Destination connected to an Excel connection manager.

### Hands-On Exercise: Transferring Data to an Excel File

This exercise consists of the following major steps:

- Creating a BI project and adding a Data Flow Task.
- Configuring the DataReader's Connection Manager.
- Configuring the DataReader Source.
- Adding a Character Map Transformation.
- Adding an Excel Destination and establishing a path to it from the Character Map Data Flow Component.
- Configuring the Excel Destination Component.
- Testing the package.

In order to follow the steps as indicated, you will need a source and a destination: the source data is extracted from the *MyNorthwind* database (just a renamed version of the *Northwind* database) on the SQL Server 2005 Server, and the destination is loading this to an MS Excel 2003 spreadsheet file on your hard drive. You also need to establish a path connecting them. In addition, you will also interpose a Character Map Data Flow Task that will convert the text in one of the data fields, so that all characters in that column are capitalized after the transformation.

## Step 1: Creating a BI Project and Adding a Data Flow Task

In this section, you will be creating a Business Intelligence project and changing the name of the default package object. Since it is data-related, you will be adding a Data Flow Task. You will also be adding a DataReader component to the data flow.

1. Create a business intelligence project **Ch 5** as described in Chapter 2 or Chapter 3.
2. Change the default name of package from `Package.dtsx` to `TableToXls.dtsx`.
3. Drag and drop a **Data Flow Task** from the **Toolbox** onto the **Control Flow** page.
4. Click open the **Data Flow** tab, which displays the **Data Flow** page.  
Now, you will be able to access the Data Flow Items of the **Toolbox** consisting of **Data Flow Sources**, **Data Flow Destinations**, and **Data Flow Transformations** (refer to Chapter 1).
5. Drag and drop a **DataReader Source** from the **Data Flow Sources** group onto the **Data Flow** page.

## Step 2: Configuring the DataReader's Connection Manager

Configuring the **DataReader** source that connects to the local SQL Server 2005 has been described in earlier chapters. Here, only a couple of the images relevant to this chapter will be shown.

1. Right-click inside the **Connection Managers** page below the Canvas, and from the drop-down choose **New ADO.Net Connection....**

---

If you are continuing with this chapter after Chapter 4, you will see the **Configure ADO.NET Connection Manager** screen displaying the previously configured connection manager. If you need to create a new one, follow the steps shown in the previous chapter.

2. Click on the **OK** button in the **Configure ADO.NET Connection Manager** window.

A connection manager, **localhost.MyNorthwind.sa**, will be added to the **Connection Manager**'s page.

### Step 3: Configuring the DataReader Source

1. Right-click the **DataReader Source**, in the drop-down menu.
2. Choose the **Edit...** menu item in the drop-down menu.

This opens the **Advanced Editor for DataReader Source**. At first, you need to indicate a connection manager that the DataReader can use.

3. In the **Advanced Editor for DataReader Source** that gets displayed click on the **Connection Managers** tab.
4. Click on an empty area (in grey) below the list heading, **Connection Manager**. Here, you will see the connection manager that you added in step 1.
5. Choose this **Connection Manager**.

6. Next, click on the **Component Properties** tab to open the properties of the DataReader component.

Here, you will notice that this requires an **SqlCommand** (the only empty field now).

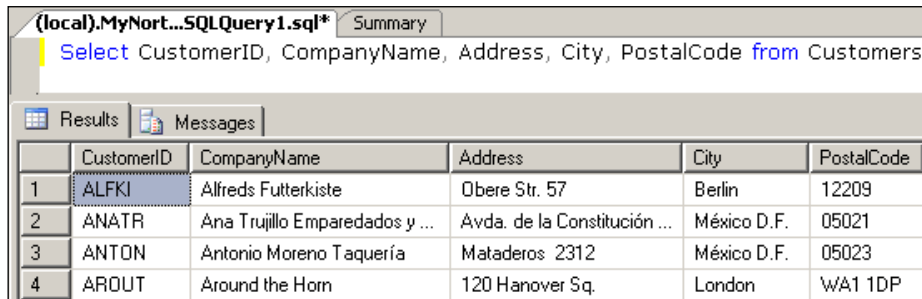
7. Click on the ellipsis button along its side to display a text editor where you can type in your **SqlCommand**.

You may also directly type-in the **SQL Command**:

```
SELECT CustomerID, CompanyName, Address, City, PostalCode  
FROM Customers
```

8. Click on the **Refresh** button.

This query will allow the DataReader to read the data from the five columns. A sample of the table data is shown in the following screenshot, taken from SQL Server 2005 Management Studio. If you recall, these were the same columns that were used in the previous chapter as well.



	CustomerID	CompanyName	Address	City	PostalCode
1	ALFKI	Alfreds Futterkiste	Obere Str. 57	Berlin	12209
2	ANATR	Ana Trujillo Emparedados y ...	Avda. de la Constitución ...	México D.F.	05021
3	ANTON	Antonio Moreno Taquería	Mataderos 2312	México D.F.	05023
4	AROUT	Around the Horn	120 Hanover Sq.	London	WA1 1DP

9. Click on the **Column Mappings** tab.

This will open the **Column Mappings** page showing the columns that are the output of the DataReader.

In the last tab on this editor, **Input and Output properties**, you can add/remove items from the **External Columns**, the **Output Columns** and the **DataReader Error output**. For this tutorial, no modifications are made.

10. Click on the **OK** button in the above window.

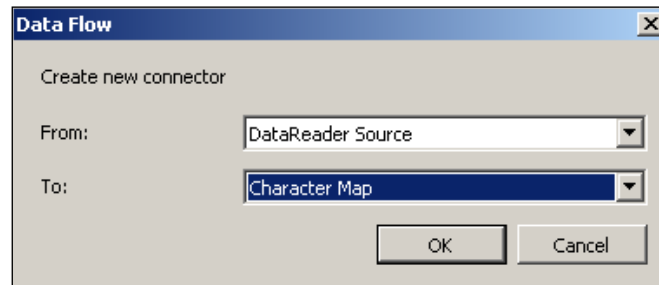
This completes the configuration of the DataReader which brings five columns from the SQL 2005 Server.

## Step 4: Adding a Character Map Transformation

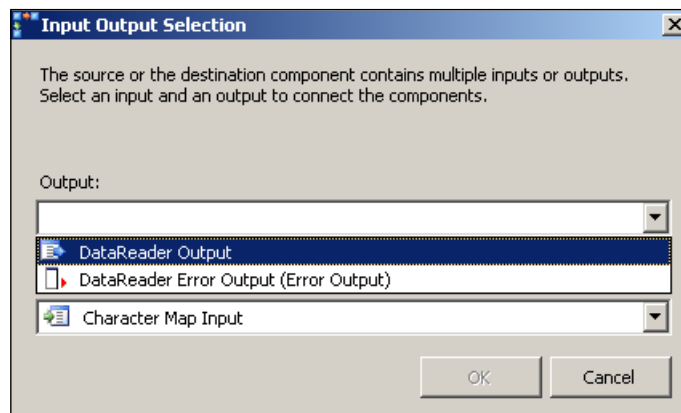
The *Character Map* transformation is described in Chapter 1, but here you will be experimenting with this transformation. The transformation manipulates the text string that is coming to it and outputs the manipulated string. For example, in the screenshot we have just seen above, the **CompanyName** has mixed case. Using this transformation, we will capitalize all the characters that appear in the **CompanyName** column before it is written to an Excel File – Alfreds Futterkiste will become ALFREDS FUTTERKISTE, etc.

1. Drag and drop a **Character Map** data flow item from the **Data Flow Transformations Group** in the **Toolbox** onto the **Data Flow** page of the Canvas.

- Right-click on the **DataReader Source** and from the drop-down click on the **Add Path** menu item.
- From the displayed window, **Data Flow**, choose **Character Map** for the **To:** field as shown.



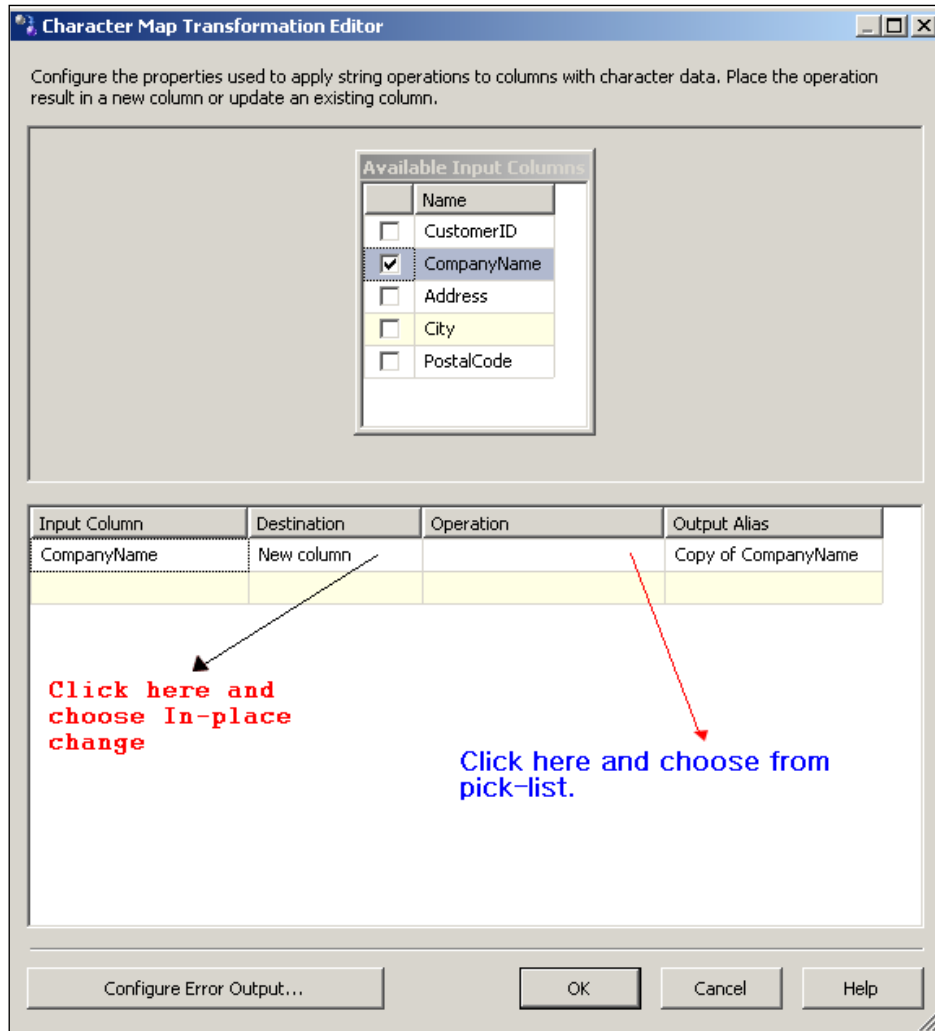
- Click **OK** in the above window, and the following window is displayed.



Here, you need to indicate the output from this component, from the drop-down shown.

- Choose **DataReader Output Source** from the drop-down.  
At present, you will not be dealing with any errors in this tutorial. When you choose the above option, the **OK** button gets enabled.
- Click on the **OK** button in the above window.  
This establishes the path from the **Data Reader Source** to the **Character Map** data flow component. The path is established but it still needs configuration.
- Right-click the **Character Map** component and from the drop-down menu choose, **Edit**.

This opens the **Character Map Editor**, as shown in the following screenshot. Place a check mark for the **Company Name** column.

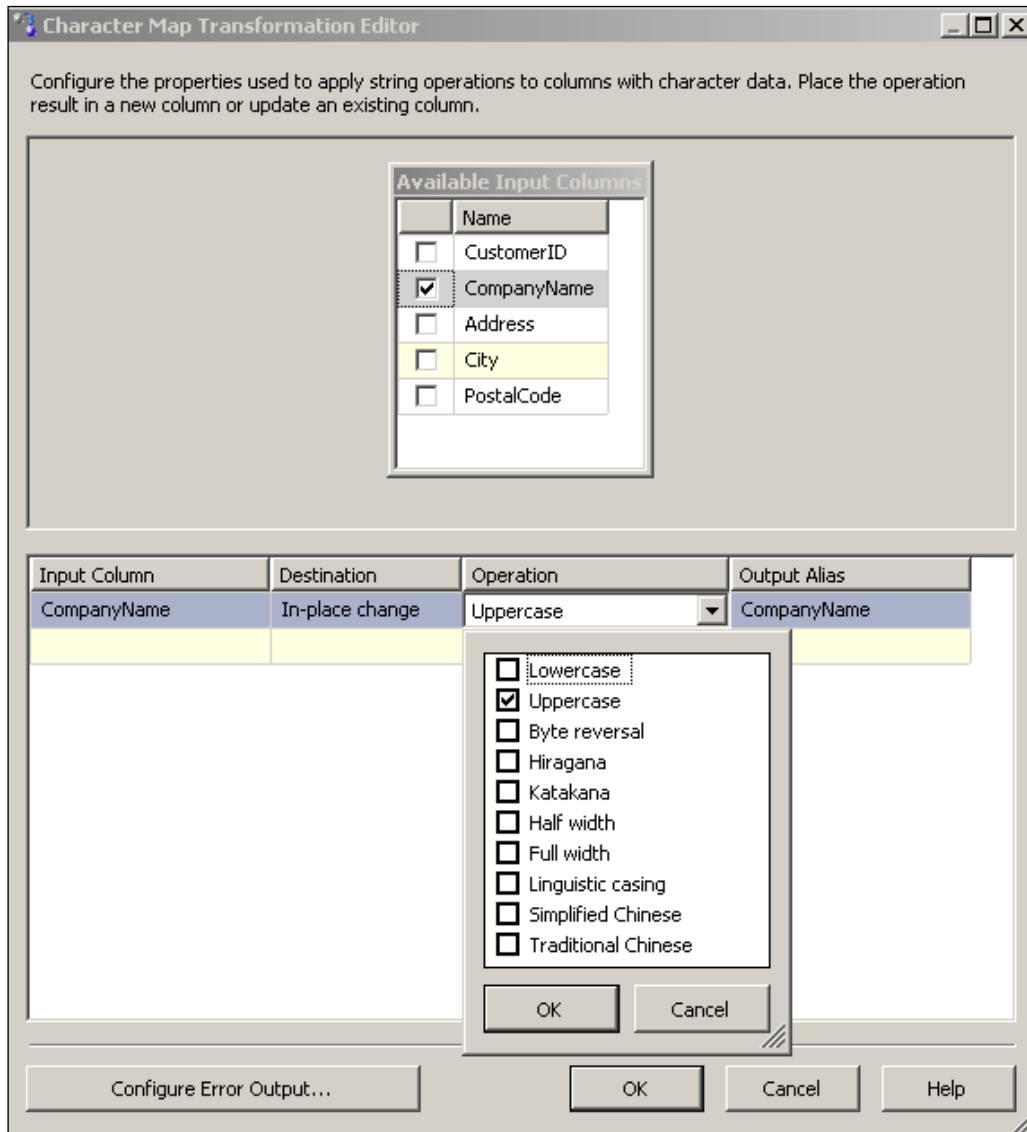


If the default as shown in the above were to be chosen, then an extra column will be added to the output. We choose the option **In-place change** from the drop-down.

8. Click on the cell, **New Column**, under **Destination** in the above window. From the drop-down choose, **In-place change**.

9. Click just below the **Operation** list-header and from the drop-down list choose **Upper Case** as shown in the following screenshot.

The output alias remains the same as it is an in-line change.



10. Click on the button **OK** in the pick-up list and to the **OK** button in the **Character Map Transformation Editor**.

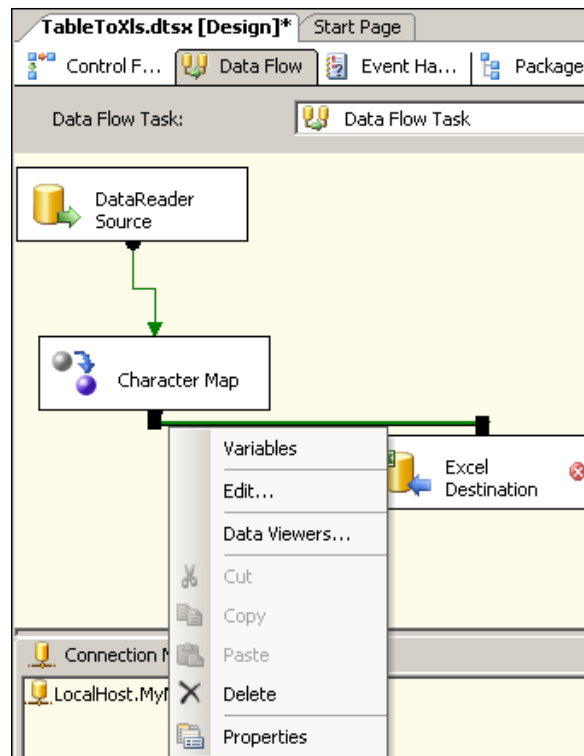
This completes the Character Map configuration.

## **Step 5: Adding an Excel Destination and Establishing a Path to It from the Character Map Data Flow Component**

In this step, we will add an *Excel Destination*. We will then establish a path from the *Character Map* to *Excel Destination*.

1. Add an **Excel Destination** component from the **Data Flow Destinations** group in the toolbox to the **Data Flow** page.  
This can be accomplished either by double-clicking the component in the **Toolbox** or a drag-and-drop operation.
2. Right-click **Character Map** and from the drop-down menu choose **Add Path**.  
This opens up the window, **Data Flow**, which allows you to establish a data flow path, and displays the "From:" location as **Character Map**.
3. Click on the drop-down along "To:", which shows both the **Excel Destination** as well as the **DataReader Source**.
4. Choose the **Excel Destination** and click **OK** to the screen.  
This opens up the **Input Output Selection** window that shows the available **output** and **input** windows. The Output window is empty whereas the input shows **Excel Destination Input**. The path should connect from the **Character Map** to the **Excel Destination Input**.
5. Choose the above options and click on the **OK** button on this screen.  
You will see a green line connecting the **Character Map** Data Flow Component to the **Excel Destination**, as shown in the next screenshot. Alternately, the process of establishing the path can be simplified by just picking the green dangling line from **Character Map** and dropping it onto the **Excel Destination** object on the **Data Flow** page. As seen in the next screenshot, you may also edit the path after it is created by right-clicking on this green line and choosing the **Edit...** drop-down menu item.





## Step 6: Configuring the Excel Destination Component

The data is on its way through the path, represented by the green line in the previous step. The Excel Destination also requires a connection manager.

The **Excel Destination** connects to an MS Excel on your hard drive using the connection properties defined in a connection manager.

1. Right-click the **Excel Destination** and from the drop-down menu choose **Edit**. This displays the **Excel Destination Editor**. Excel requires an OLE DB connection manager and if there are no configured connection managers (by you or a previous user) the drop-down will be empty.
2. Click on the **New...** button.

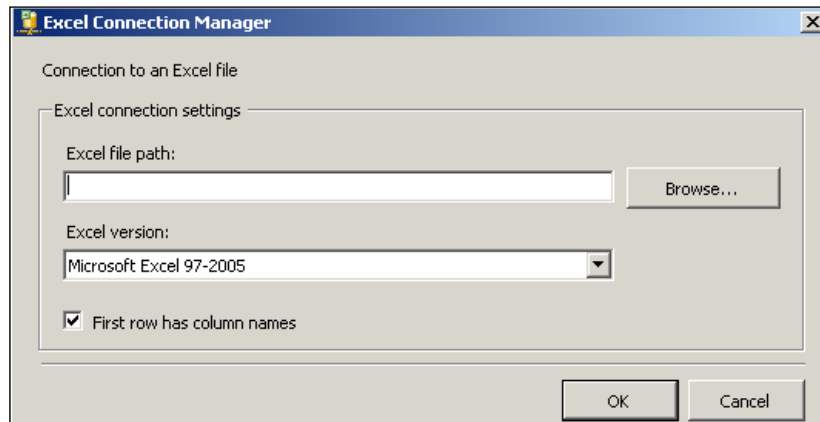
The **Excel Connection Manager** window gets displayed as shown in the next screenshot. Here, you need to use the **Browse** button and pick the Excel file as the destination. The data will be written to the destination when the package is run.

3. Open Windows Explorer and create an Excel file in the C:\ drive.

For this tutorial, `TableToXls.xls` is chosen.

Besides connecting to an existing file, the Excel Connection Manager supports creating a file on the folder of your choice in the machine using the **Browse...** button.

4. Now browse to the newly created file using the **Browse...** button and choose this file.
5. Click on the **OK** button in the **Excel Connection Manager** window.

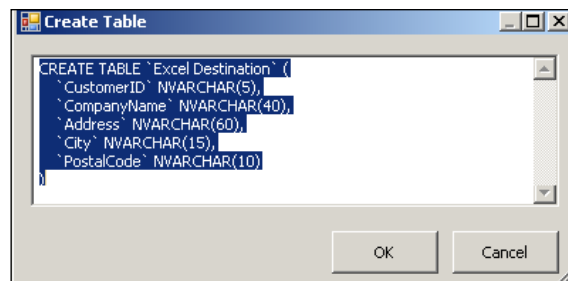


6. For the **Data Access Mode**, accept the default, **Table or View**.

You have to indicate the name of the Excel sheet that will be used. (Do not click on the drop-down for locating the sheet. The drop-down will show the three Excel sheets that are found in a newly created Excel worksheet file; all of them having just one column each.)

7. Click on the **New...** button.

You are creating a new Excel sheet. This pops-up a **Create Table** window showing the columns that are being piped into the component, as shown in the following screenshot.

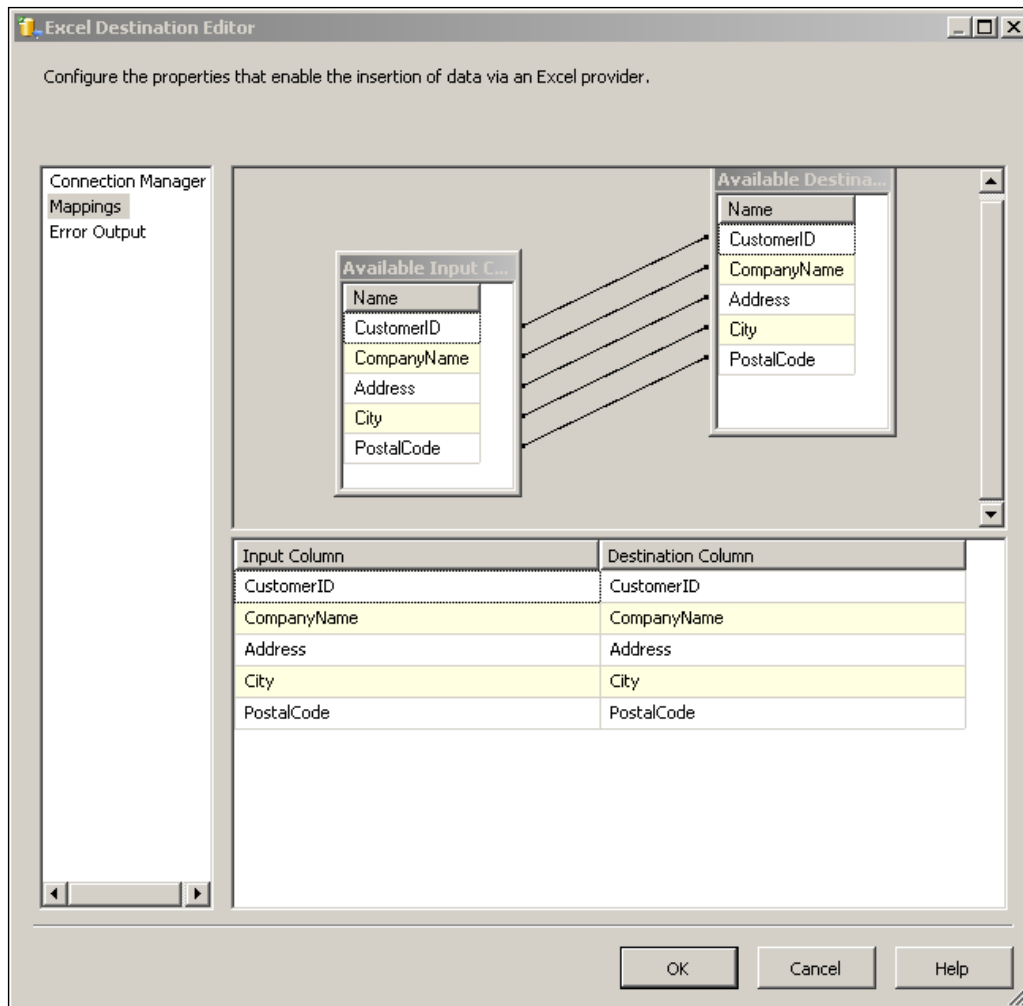


- Click on the **OK** button, shown in the previous screenshot.

A new Excel Sheet, **Excel Destination**, will be added to the `TableToXLS.xls` file. If you now open and review this file (`TableToXLS.xls`) you will see the column headers are added to this sheet.

- Click on the **Mappings** in the left-hand-side pane of the **Excel Destination Editor**, which shows the mappings from the input to the output.

This shows all the columns from the **Character Map** Data Flow Component being written to the destination file, as shown in the following screenshot.



- Click on the **OK** button in this window. The package is now completely configured and ready for execution.

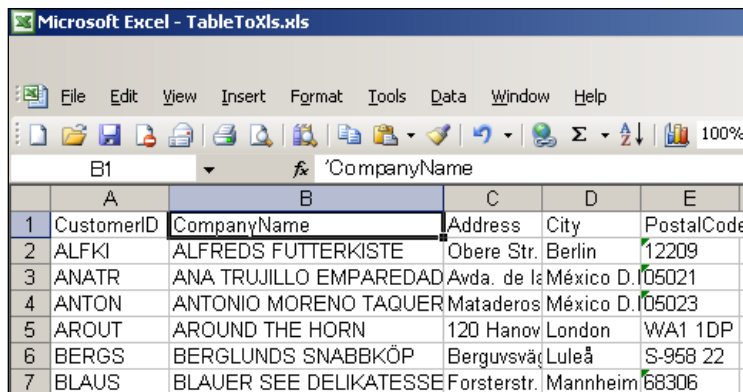
## Step 7: Testing the Package

1. Right-click the `TableToXls.dtsx` in the **Solution Explorer** and from the drop-down choose, **Execute Package**.

All three components on the Data Flow page turn green, indicating that the package executed successfully without errors. You may review the **Progress** tab in the 'Canvas', which shows all the details of the execution of the package.

2. Now open up the `TableToXls.xls` file and review.

A few rows of data are shown in the next screenshot. Notice that the *Character Map Data Transformation* component has capitalized all the characters in the **CompanyName** column.



	A	B	C	D	E
1	CustomerID	CompanyName	Address	City	PostalCode
2	ALFKI	ALFREDS FUTTERKISTE	Obere Str.	Berlin	12209
3	ANATR	ANA TRUJILLO EMPAREDAD	Avda. de la	México D.	05021
4	ANTON	ANTONIO MORENO TAQUER	Mataderos	México D.	05023
5	AROUT	AROUND THE HORN	120 Hanov	London	WA1 1DP
6	BERGS	BERGLUNDS SNABBKÖP	Berguvsväg	Luleå	S-958 22
7	BLAUS	BLAUER SEE DELIKATESSE	Forsterstr.	Mannheim	68306

## Summary

This chapter described the following:

- Configuring data flow components that are necessary for transferring data from single table on the SQL Server 2005 to an Excel file.
- The usage of a Character Map Data Transformation component.

The reverse of this transfer can be accomplished by choosing an Excel Source and an SQL Server Destination.