Chapter 16: SQL Server Integration Services

In this chapter:

- The Import and Export Wizard
- Creating a Package
- Working with Connection Managers
- Building Data Flows
- Building Control Flows
- Creating Event Handlers
- Saving and Running Packages

Files needed:

- ISProject1.zip
- ISProject2.zip

Microsoft says that SQL Server Integration Services (SSIS) "is a platform for building high performance data integration solutions, including extraction, transformation, and load (ETL) packages for data warehousing." A simpler way to think of SSIS is that it's the solution for automating SQL Server. SSIS provides a way to build packages made up of tasks that can move data around from place to place and alter it on the way. There are visual designers (hosted within Business Intelligence Development Studio) to help you build these packages as well as an API for programming SSIS objects from other applications.

In this chapter, you'll see how to build and use SSIS packages. First, though, we'll look at a simpler facet of SSIS: The SQL Server Import and Export Wizard.

 If you choose to use the supplied solution files rather than building your own, you'll need to edit the properties of the OLE DB Connection Managers within the projects to point to your own test server. You'll learn more about Connection Managers in the "Working with Connection Managers" section later in this chapter.

The Import and Export Wizard

Though SSIS is almost infinitely customizable, Microsoft has produced a simple wizard to handle some of the most common ETL tasks: importing data to or exporting data from a SQL Server database. The Import and Export Wizard protects you from the complexity of SSIS while allowing you to move data between any of these data sources:

- SQL Server databases
- Flat files
- Microsoft Access databases
- Microsoft Excel worksheets
- Other OLE DB providers

You can launch the Import and Export wizard from the Tasks entry on the shortcut menu of any database in the Object Explorer window of SQL Server Management Studio.

Try It!

To import some data using the Import and Export Wizard, follow these steps:

1. Launch SQL Server Management Studio and log in to your test server.

2. Open a new query window.
3. Select the master database from the Available Databases combo box on the toolbar.
4. Enter this text into the query window:

```
CREATE DATABASE Chapter16
```

5. Click the Execute toolbar button to create a new database.
6. Expand the Databases node in Object Explorer
7. Right-click on the Chapter16 database and select Tasks ▶ Import Data.
8. Read the first page of the Import and Export Wizard and click Next.
9. Select SQL Native Client for the data source and provide login information for your test server.
10. Select the AdventureWorks database as the source of the data to import.
11. Click Next.
12. Because you're importing data, the next page of the wizard will default to connection information for the Chapter16 database. Click Next.
13. Select Copy Data From One or More Tables or Views and click Next. Note that if you only want to import part of a table you can use a query as the data source instead.
14. Select the HumanResources. Department, HumanResources. Employee, HumanResources. EmployeeAddress, HumanResources. EmployeeDepartmentHistory, and HumanResources. EmployeePayHistory tables, as show in Figure 16-1. As you select tables, the wizard will automatically assign names for the target tables.
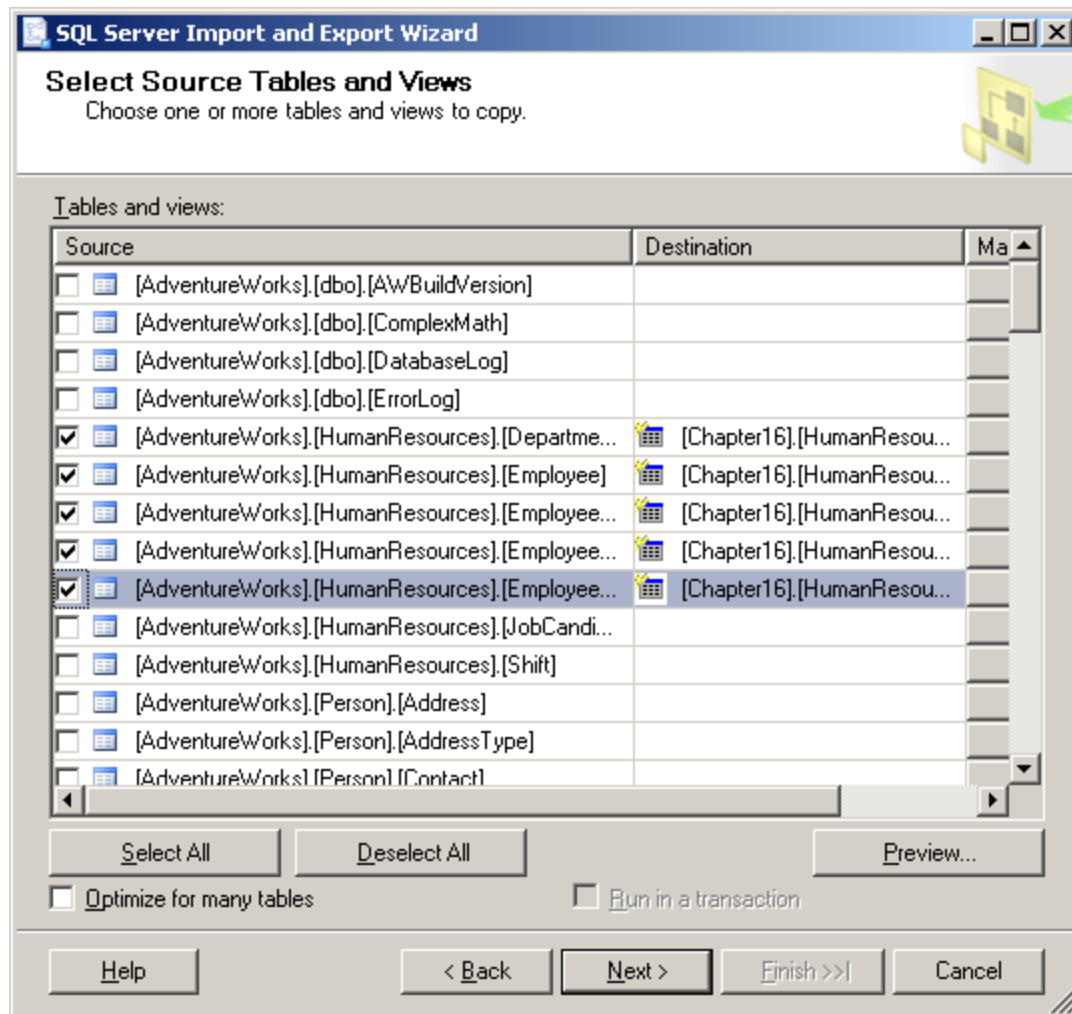
Figure 16-1: Selecting tables to import

15. Click the Edit button in the Mapping column for the
    `HumanResources.Department` table.
16. The Column Mappings dialog box lets you change the name, data type, and
    other properties of the destination table columns. You can also set other options
    here, such as whether to overwrite or append data when importing data to an
    existing table. Click Cancel when you're done inspecting the options.
17. Click Next.
18. Check Execute Immediately and click Next.
19. Click Finish to perform the import. SQL Server will display progress as it
    performs the import, as shown in Figure 16-2.

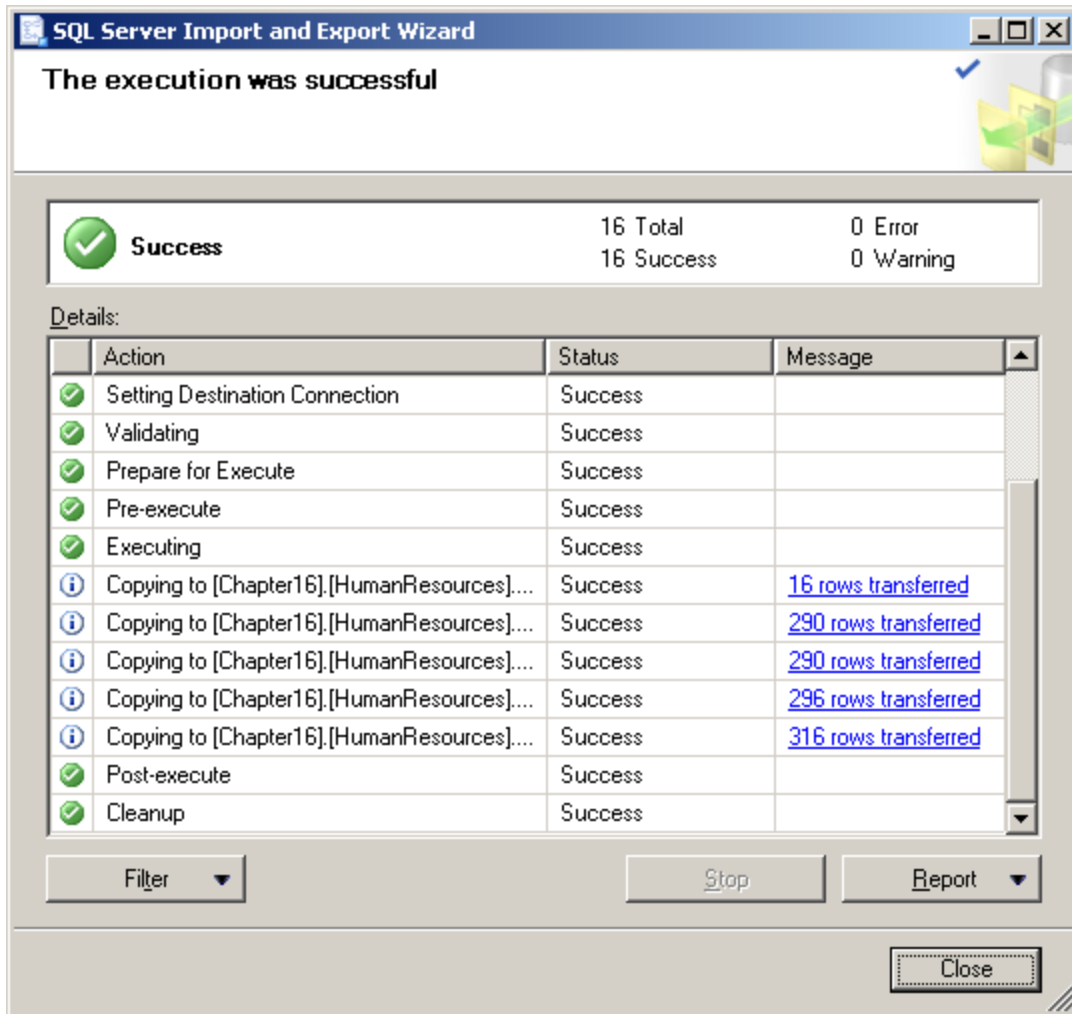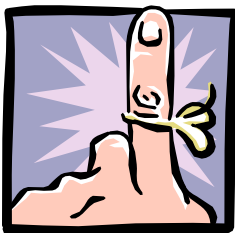Introduction to SQL Server 2005

Figure 16-2: Import Wizard results

20. Click Close to dismiss the report.
21. Expand the Tables node of the Chapter16 database to verify that the import succeeded.

	In addition to executing its operations immediately, the Import and Export Wizard can also save a package for later execution. You'll learn more about packages in the remainder of this chapter.

Creating a Package

The Import and Export Wizard is easy to use, but it only taps a small part of the functionality of SSIS. To really appreciate the full power of SSIS, you'll need to use BIDS to build an SSIS *package*. A package is a collection of SSIS objects including:

- *Connections* to data sources.
- *Data flows*, which include the sources and destinations that extract and load data, the transformations that modify and extend data, and the paths that link sources, transformations, and destinations.
- *Control flows*, which include tasks and containers that execute when the package runs. You can organize tasks in sequences and in loops.
- *Event handlers*, which are workflows that runs in response to the events raised by a package, task, or container.

You'll see how to build each of these components of a package in later sections of the chapter, but first, let's fire up BIDS and create a new SSIS package.

Try It!

To create a new SSIS package, follow these steps:

1. Launch Business Intelligence Development Studio
2. Select File ▶ New ▶ Project.
3. Select the Business Intelligence Projects project type.
4. Select the Integration Services Project template.
5. Select a convenient location.
6. Name the new project ISProject1 and click OK.

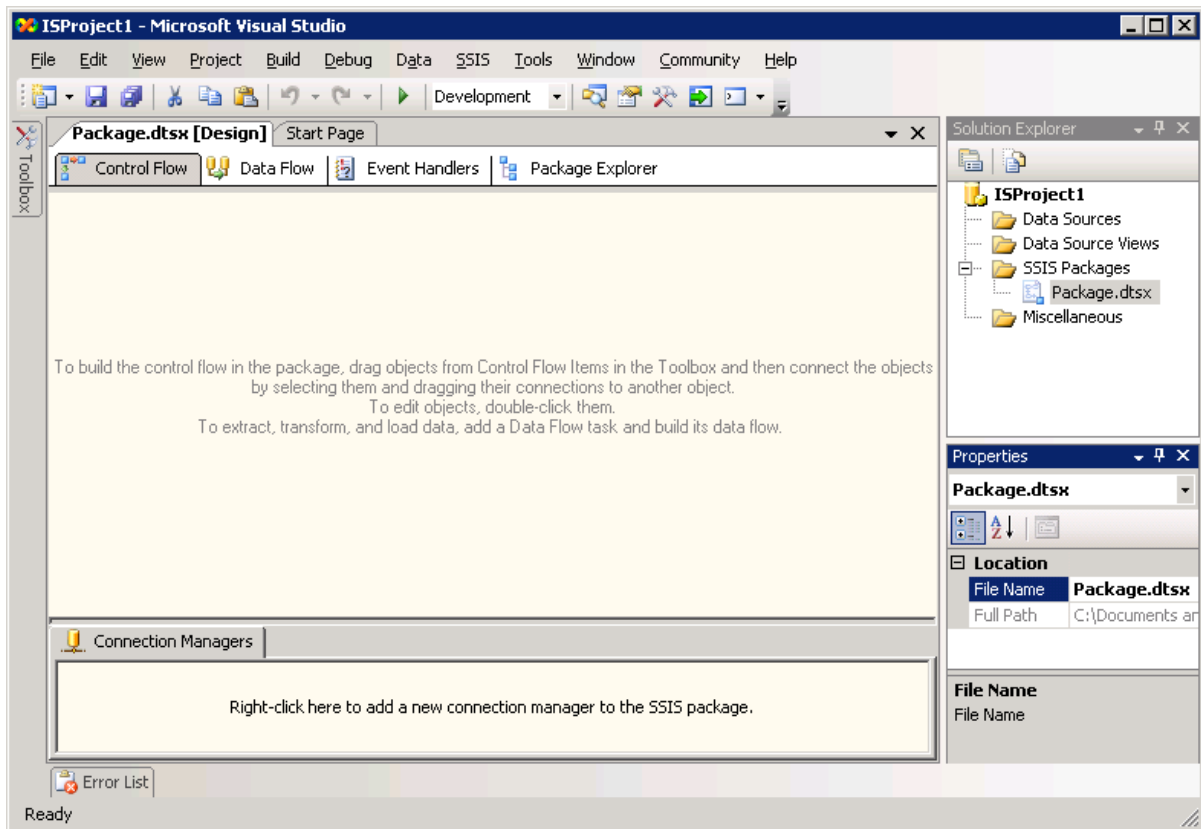Figure 16-3 shows the new, empty package.

Figure 16-3: Empty SSIS package

Working with Connection Managers

SSIS uses connection managers to integrate different data sources into packages. SSIS includes a wide variety of different connection managers that allow you to move data around from place to place. Table 16-1 lists the available connection managers.

Connection Manager	Handles
ADO Connection Manager	Connecting to ADO objects such as a `Recordset`.
ADO.NET Connection Manager	Connecting to data sources through an ADO.NET provider.
Analysis Services Connection Manager	Connecting to an Analysis Services database or cube.
Excel Connection Manager	Connecting to an Excel worksheet.
File Connection Manager	Connecting to a file or folder.
Flat File Connection Manager	Connecting to delimited or fixed width flat files.
FTP Connection Manager	Connecting to an FTP data source.
HTTP Connection Manager	Connecting to an HTTP data source.
MSMQ Connection Manager	Connecting to a Microsoft Message Queue.

Multiple Files Connection Manager	Connecting to a set of files, such as all text files on a particular hard drive.
Multiple Flat Files Connection Manager	Connecting to a set of flat files.
ODBC Connection Manager	Connecting to an ODBC data source.
OLE DB Connection Manager	Connecting to an OLE DB data source.
SMO Connection Manager	Connecting to a server via SMO.
SMTP Connection Manager	Connecting to a Simple Mail Transfer Protocol server.
SQL Server Mobile Connection Manager	Connecting to a SQL Server Mobile database.
WMI Connection Manager	Connecting to Windows Management Instrumentation data.

Table 16-1: Available Connection Managers

To create a Connection Manager, you right-click anywhere in the Connection Managers area of a package in BIDS and choose the appropriate shortcut from the shortcut menu. Each Connection Manager has its own custom configuration dialog box with specific options that you need to fill out.

Try It!

To add some connection managers to your package, follow these steps:

1. Right-click in the Connection Managers area of your new package and select New OLE DB Connection.
2. Note that the configuration dialog box will show the data connections that you created in Chapter 15; data connections are shared across Analysis Services and Integration Services projects. Click New to create a new data connection.
3. In the Connection Manager dialog box, select the SQL Native Client provider.
4. Select your test server and provide login information.
5. Select the Chapter16 database.
6. Click OK.
7. In the Configure OLE DB Connection Manager dialog box, click OK.
8. Right-click in the Connection Managers area of your new package and select New Flat File Connection.
9. Enter DepartmentList as the Connection Manager Name.
10. Enter C:\Departments.txt as the File Name (this file will be supplied by your instructor).
11. Check the Column Names in the First Data Row checkbox. Figure 16-4 shows the completed General page of the dialog box.
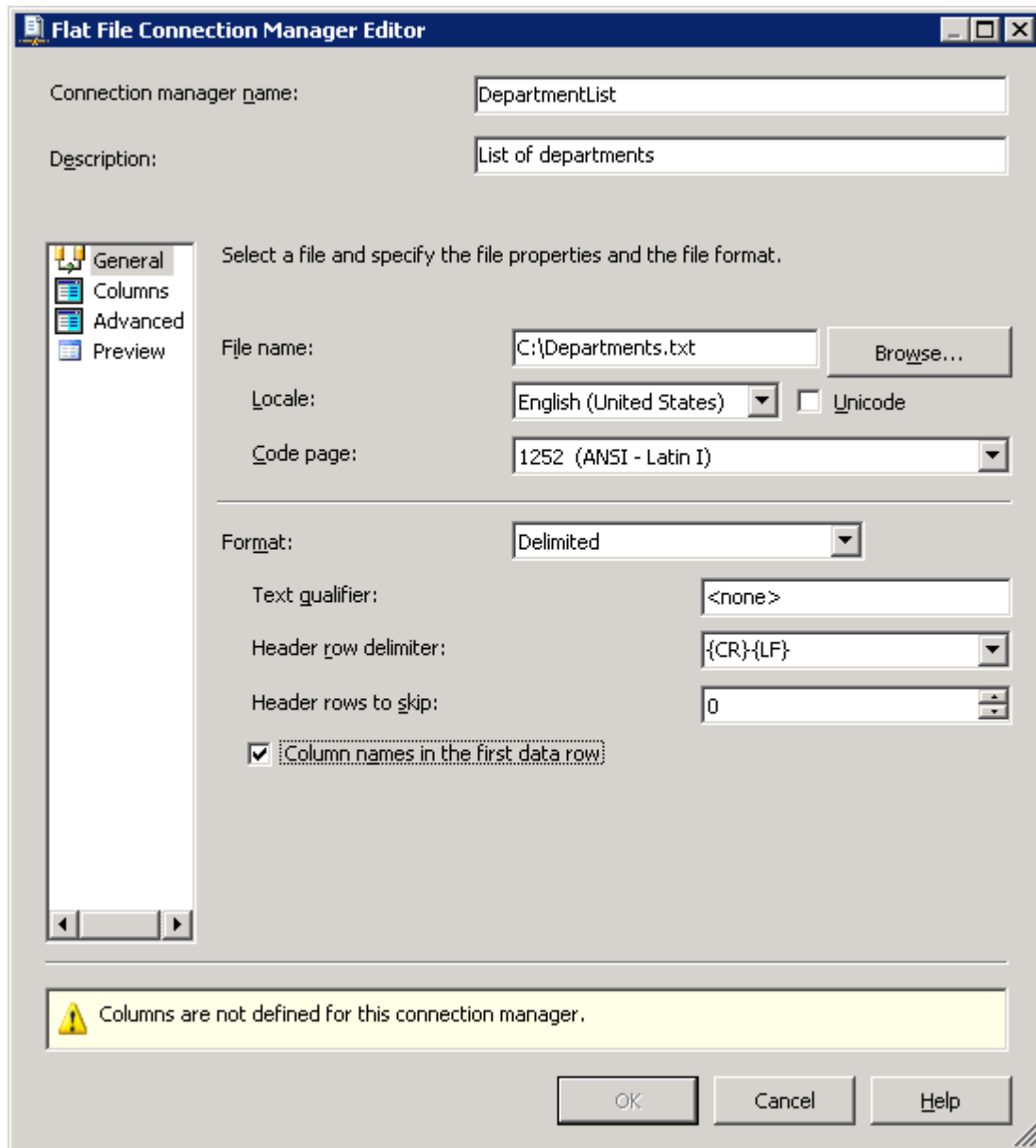
Figure 16-4: Defining a Flat File Connection Manager

12. Click the Advanced icon to move to the Advanced page of the dialog box.
13. Click the New button.
14. Change the Name of the new column to `DepartmentName`.
15. Click OK.
16. Right-click the DepartmentList Connection Manager and select Copy.
17. Right-click in the Connection Managers area and select Paste.
18. Click on the new DepartmentList 1 connection to select it.
19. Use the Properties Window to change properties of the new connection.
    Change the Name property to `DepartmentListBackup`. Change the
    ConnectionString property to `C:\DepartmentsBackup.txt`.

Figure 16-5 shows the SSIS package with the three Connection Managers defined.
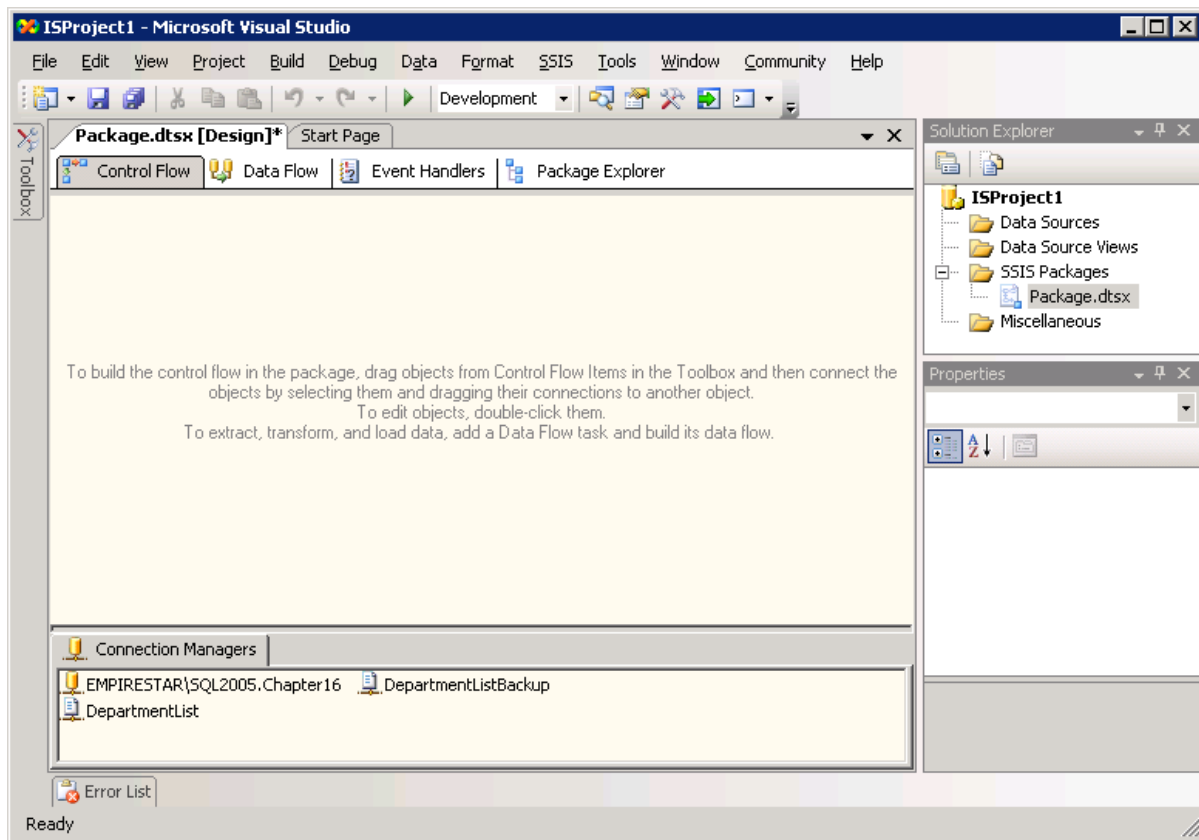

Figure 16-5: An SSIS package with two Connection Managers

Building Control Flows

The Control Flow tab of the Package Designer is where you tell SSIS what the package will do. You create your control flow by dragging and dropping items from the toolbox to the surface, and then dragging and dropping connections between the objects. The objects you can drop here break up into four different groups:

- *Tasks* are things that SSIS can do, such as execute SQL statements or transfer objects from one SQL Server to another. Table 16-2 lists the available tasks.
- *Maintenance Plan tasks* are a special group of tasks that handle jobs such as checking database integrity and rebuilding indexes. Table 16-3 lists the maintenance plan tasks.
- The *Data Flow Task* is a general purpose task for ETL (extract, transform, and load) operations on data. There's a separate design tab for building the details of a Data Flow Task.
- *Containers* are objects that can hold a group of tasks. Table 16-4 lists the available containers.

Task	Purpose
ActiveX Script	Execute an ActiveX Script
Analysis Services Execute DDL	Execute DDL query statements against an Analysis Services server
Analysis Services Processing	Process an Analysis Services cube
Bulk Insert	Insert data from a file into a database
Data Mining Query	Execute a data mining query
Execute DTS 2000 Package	Execute a Data Transformation Services Package (DTS was the SQL Server 2000 version of SSIS)
Execute Package	Execute an SSIS package
Execute Process	Shell out to a Windows application
Execute SQL	Run a SQL query
File System	Perform file system operations such as copy or delete
FTP	Perform FTP operations
Message Queue	Send or receive messages via MSMQ
Script	Execute a custom task
Send Mail	Send e-mail
Transfer Database	Transfer an entire database between two SQL Servers
Transfer Error Messages	Transfer custom error messages between two SQL Servers
Transfer Jobs	Transfer jobs between two SQL Servers
Transfer Logins	Transfer logins between two SQL Servers
Transfer Master Stored Procedures	Transfer stored procedures from the master database on one SQL Server to the master database on another SQL Server
Transfer SQL Server Objects	Transfer objects between two SQL Servers
Web Service	Execute a SOAP Web method
WMI Data Reader	Read data via WMI
WMI Event Watcher	Wait for a WMI event
XML	Perform operations on XML data

Table 16-2: SSIS control flow tasks

Task	Purpose
Back Up Database	Back up an entire database to file or tape
Check Database Integrity	Perform database consistency checks
Execute SQL Server Agent Job	Run a job
Execute T-SQL Statement	Run any T-SQL script
History Cleanup	Clean out history tables for other maintenance tasks
Maintenance Cleanup	Clean up files left by other maintenance tasks
Notify Operator	Send e-mail to SQL Server operators
Rebuild Index	Rebuild a SQL Server index
Reorganize Index	Compacts and defragments an index
Shrink Database	Shrinks a database

Update Statistics	Update statistics used to calculate query plans

Table 16-3: SSIS maintenance plan tasks

Container	Purpose
For Loop	Repeat a task a fixed number of times
Foreach	Repeat a task by enumerating over a group of objects
Sequence	Group multiple tasks into a single unit for easier management

Table 16-4: SSIS containers

Try It!

To add control flows to the package you've been building, follow these steps:

1. If the Toolbox isn't visible already, hover your mouse over the Toolbox tab until it slides out from the side of the BIDS window. Use the pushpin button in the Toolbox title bar to keep the Toolbox visible.
2. Make sure the Control Flow tab is selected in the Package Designer.
3. Drag a File System Task from the Toolbox and drop it on the Package Designer.
4. Drag a Data Flow Task from the Toolbox and drop it on the Package Designer, somewhere below the File System task.
5. Click on the File System Task on the Package Designer to select it.
6. Drag the green arrow from the bottom of the File System Task and drop it on top of the Data Flow Task. This tells SSIS the order of tasks when the File System Task succeeds.
7. Double-click the connection between the two tasks to open the Precedence Constraint Editor.
8. Change the Value from `Success` to `Completion`, because you want the Data Flow Task to execute whether the File System Task succeeds or not.
9. Click OK.
10. Select the File System task in the designer. Use the Properties Window to set properties of the File System Task. Set the Source property to `DepartmentList`. Set the Destination property to `DepartmentListBackup`. Set the OverwriteDestinationFile property to `True`.
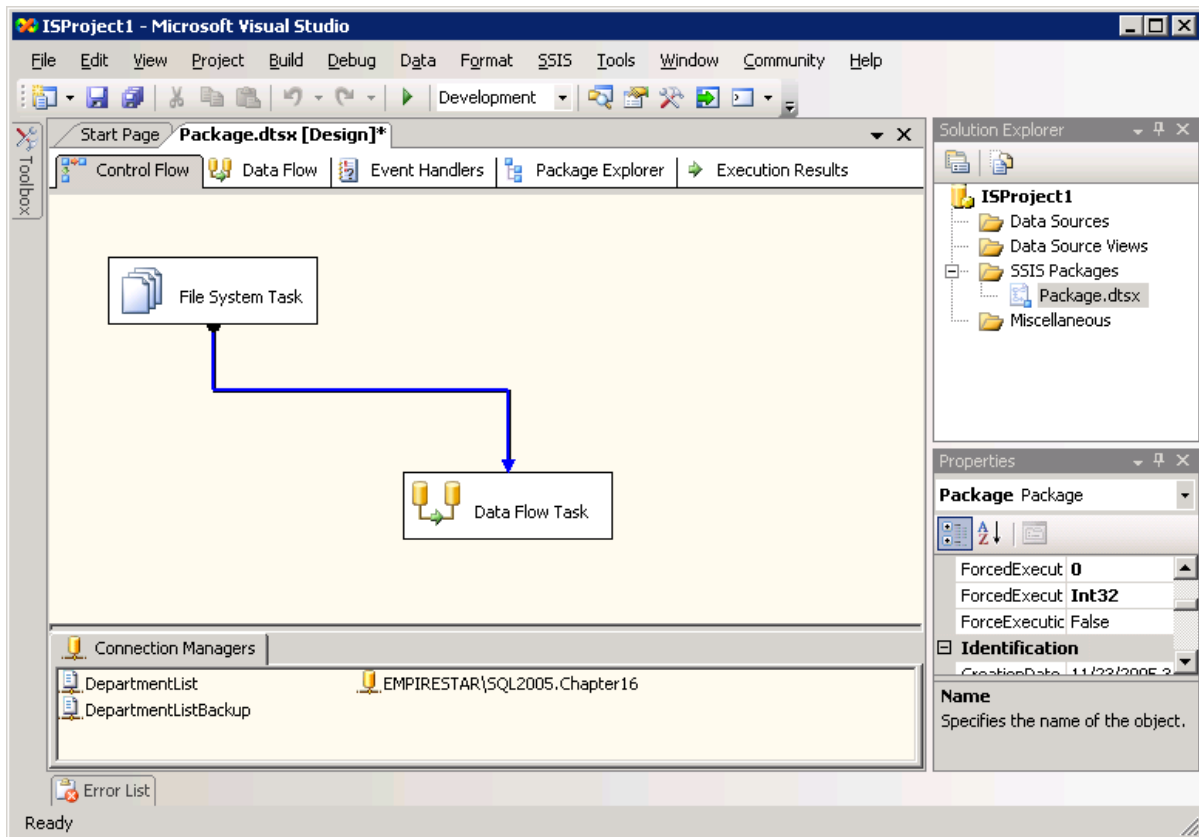
Figure 16-6 shows the completed set of control flows.

Figure 16-6: Adding control flows

As it stands, this package uses the file system task to copy the file specified by the `DepartmentList` connection to the file specified by the `DepartmentListBackup` connection, overwriting any target file that already exists. It then executes the data flow task. In the next section, you'll see how to configure the data flow task.

Building Data Flows

The Data Flow tab of the Package Designer is where you specify the details of any Data Flow tasks that you've added on the Control Flow tab. Data Flows are made up of various objects that you drag and drop from the Toolbox:

- *Data Flow Sources* are ways that data gets into the system. Table 16-5 lists the available data flow sources.
- *Data Flow Transformations* let you alter and manipulate the data in various ways. Table 16-6 lists the available data flow transformations.
- *Data Flow Destinations* are the places that you can send the transformed data. Table 16-7 lists the available data flow destinations.

SQL Server Integration Services

Source	Use
DataReader	Extracts data from a database using a .NET `DataReader`
Excel	Extracts data from an Excel workbook
Flat File	Extracts data from a flat file
OLE DB	Extracts data from a database using an OLE DB provider
Raw File	Extracts data from a raw file
XML	Extracts data from an XML file

Table 16-5: Data flow sources

Transformation	Effect
Aggregate	Aggregates and groups values in a dataset
Audit	Adds audit information to a dataset
Character Map	Applies string operations to character data
Conditional Split	Evaluates and splits up rows in a dataset
Copy Column	Copies a column of data
Data Conversion	Converts data to a different datatype
Data Mining Query	Runs a data mining query
Derived Column	Calculates a new column from existing data
Export Column	Exports data from a column to a file
Fuzzy Grouping	Groups rows that contain similar values
Fuzzy Lookup	Looks up values using fuzzy matching
Import Column	Imports data from a file to a column
Lookup	Looks up values in a reference dataset
Merge	Merges two sorted datasets
Merge Join	Merges data from two datasets by using a join
Multicast	Creates copies of a dataset
OLE DB Command	Executes a SQL command on each row in a dataset
Percentage Sampling	Extracts a subset of rows from a dataset
Pivot	Builds a pivot table from a dataset
Row Count	Counts the rows of a dataset
Row Sampling	Extracts a sample of rows from a dataset
Script Component	Executes a custom script
Slowly Changing Dimension	Updates a slowly changing dimension in a cube
Sort	Sorts data
Term Extraction	Extracts data from a column
Term Lookup	Looks up the frequency of a term in a column
Union All	Merges multiple datasets
Unpivot	Normalizes a pivot table

Table 16-6: Data Flow Transformations

Destination	Use

Data Mining Model Training	Sends data to an Analysis Services data mining model
DataReader	Sends data to an in-memory ADO.NET DataReader
Dimension Processing	Processes a cube dimension
Excel	Sends data to an Excel worksheet
Flat File	Sends data to a flat file
OLE DB	Sends data to an OLE DB database
Partition Processing	Processes an Analysis Services partition
Raw File	Sends data to a raw file
Recordset	Sends data to an in-memory ADO Recordset
SQL Server	Sends data to a SQL Server database
SQL Server Mobile	Sends data to a SQL Server Mobile database

Table 16-7: Data Flow Destinations

Try It!

To customize the data flow task in the package you're building, follow these steps:

1. Select the Data Flow tab in the Package Designer. The single Data Flow Task in the package will automatically be selected in the combo box.
2. Drag an OLE DB Source from the Toolbox and drop it on the Package Designer.
3. Drag a Character Map Transformation from the Toolbox and drop it on the Package Designer.
4. Drag a Flat File Destination from the Toolbox and drop it on the Package Designer.
5. Click on the OLE DB Source on the Package Designer to select it.
6. Drag the green arrow from the bottom of the OLE DB Source and drop it on top of the Character Map Transformation.
7. Click on the Character Map Transformation on the Package Designer to select it.
8. Drag the green arrow from the bottom of the Character Map Transformation and drop it on top of the Flat File Destination.
9. Double-click the OLE DB Source to open the OLE DB Source Editor.
10. Select the HumanResources.Department view. Figure 16-7 shows the completed OLE DB Source Editor.
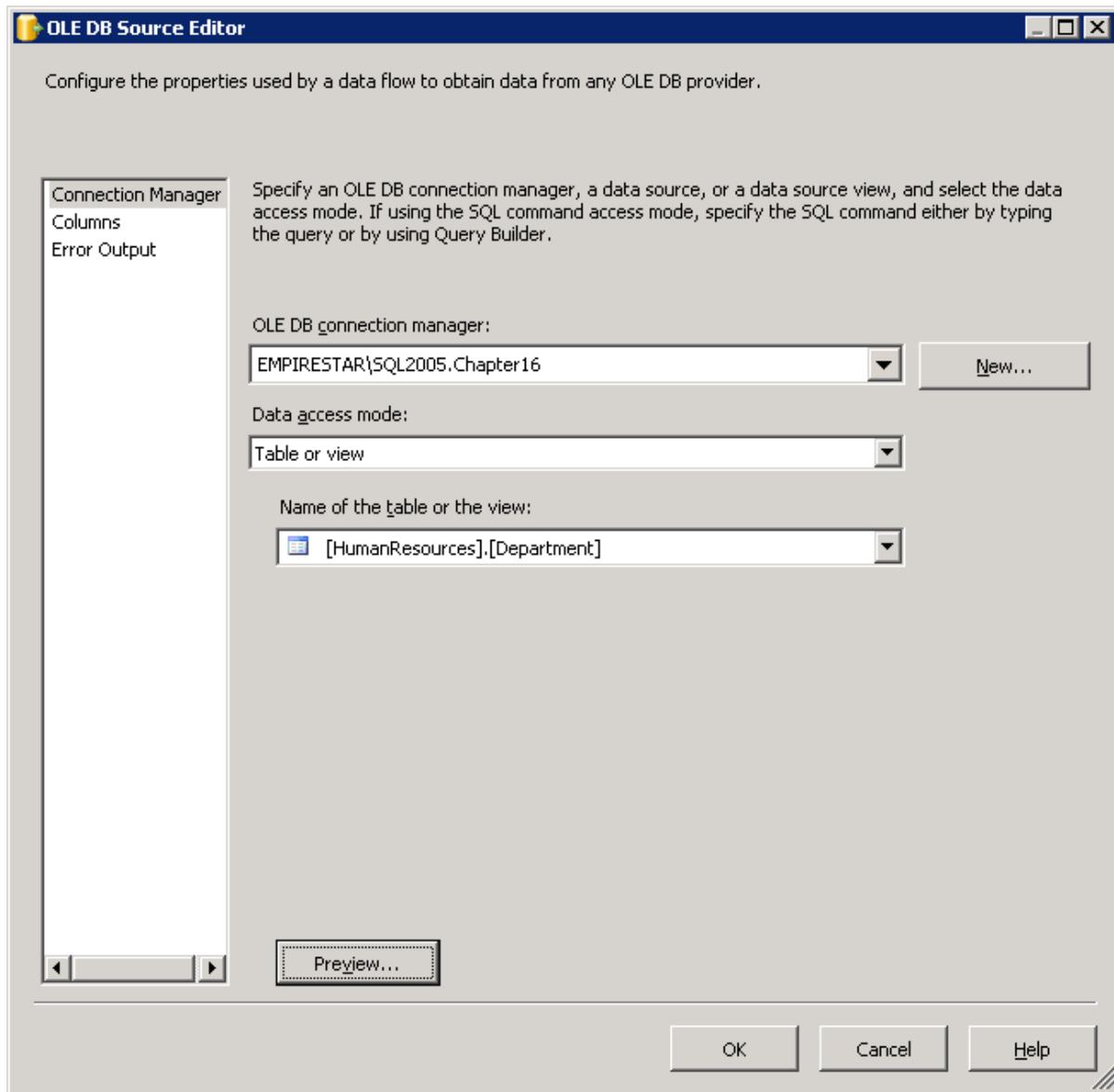
Figure 16-7: Setting up the OLE DB Source

11. Click OK.
12. Double-click the Character Map Transformation.
13. Check the Name column.
14. Select In-Place Change in the Destination column.
15. Select the Uppercase operation. Figure 16-8 shows the completed Character Map Transformation Editor.
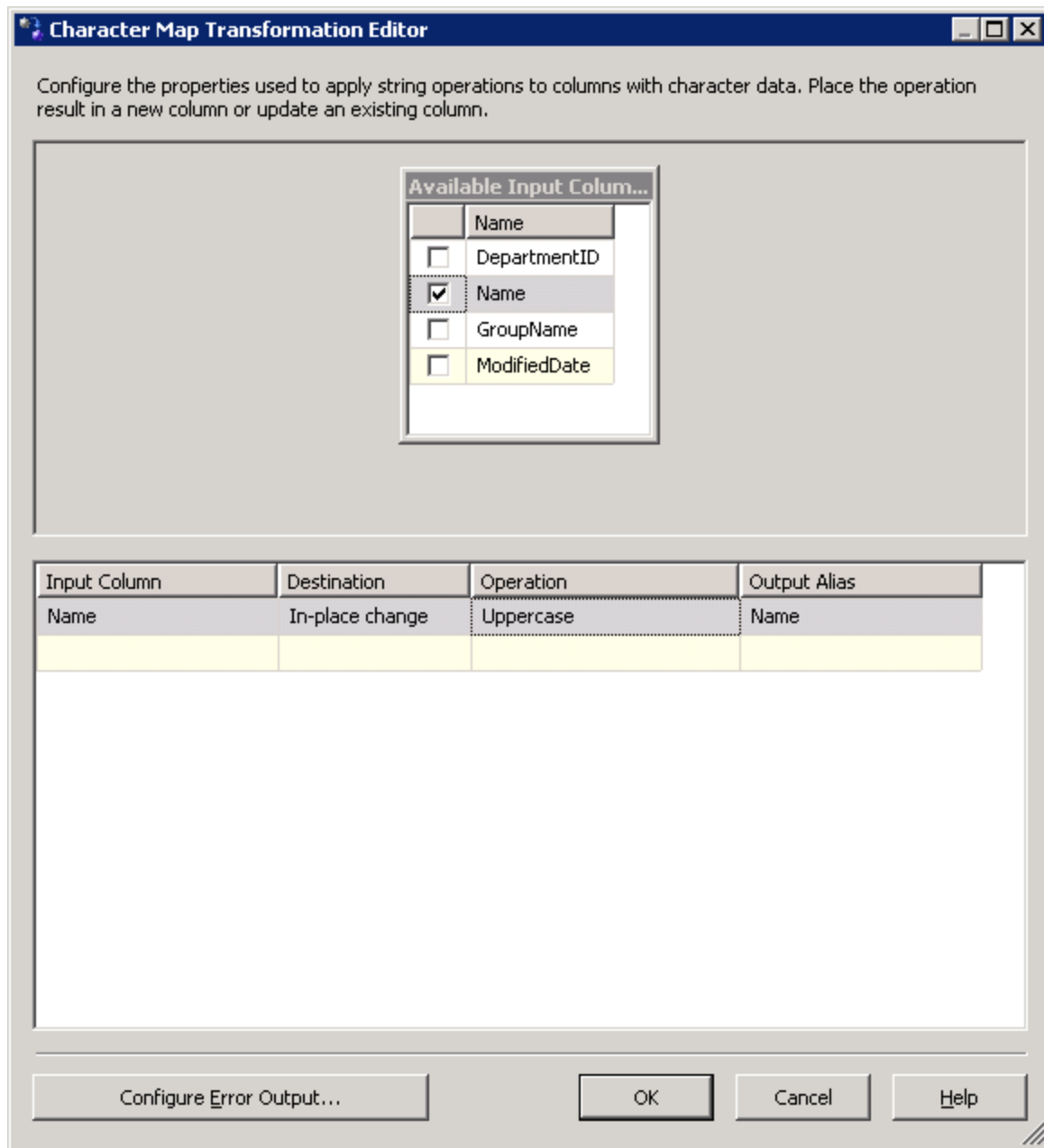
Figure 16-8: Setting up the Character Map Transformation

16. Click OK.
17. Double-click the Flat File Destination.
18. Select the DepartmentList Flat File Connection Manager.
19. Select the Mappings page of the dialog box.
20. Drag the Name column from the Available Input Columns list and drop it on top of the DepartmentName column in the Available Destination Columns list. Figure 16-9 shows the completed Mappings page.
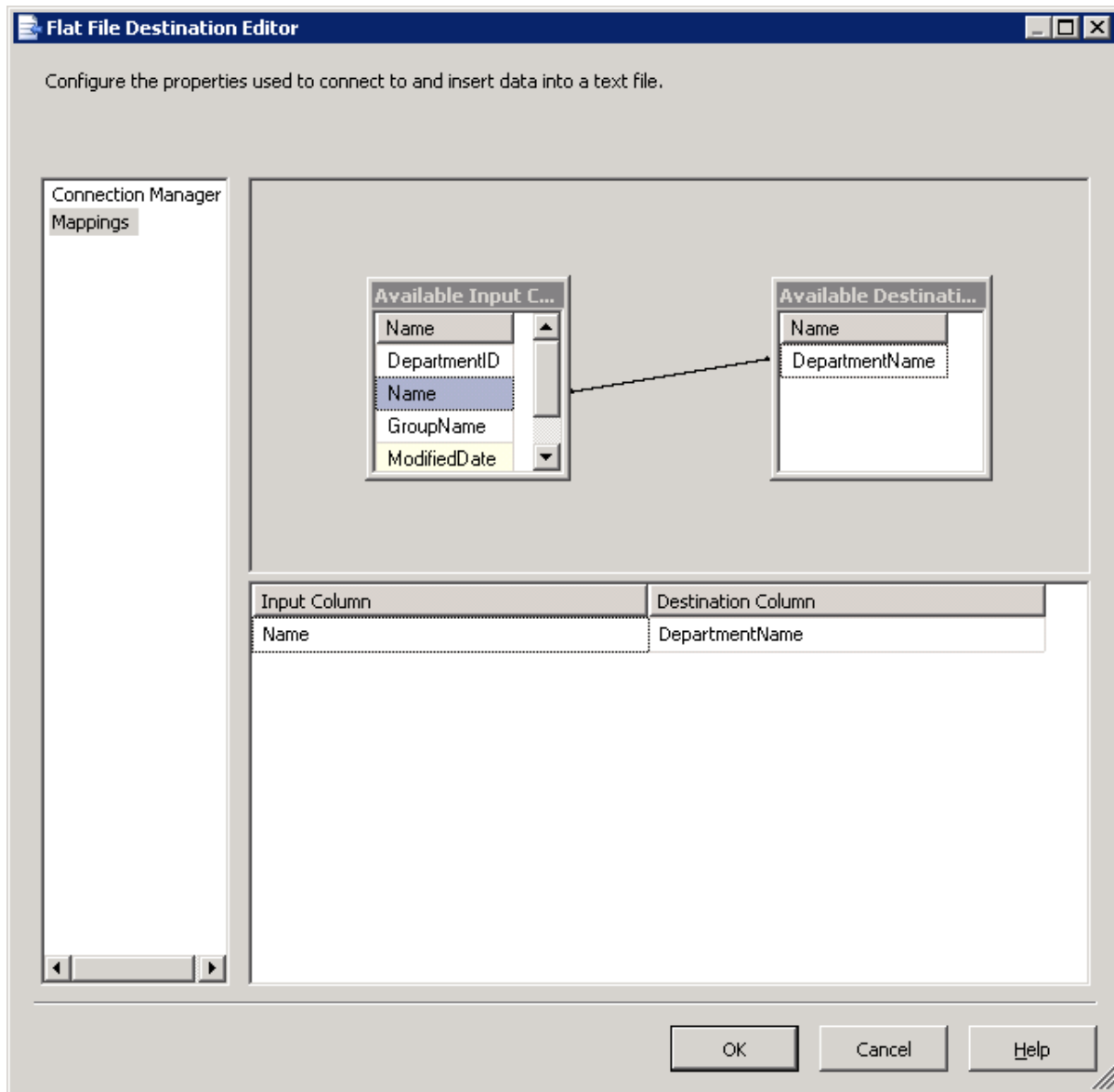
Figure 16-9: Configuring the Flat File Destination

21. Click OK.

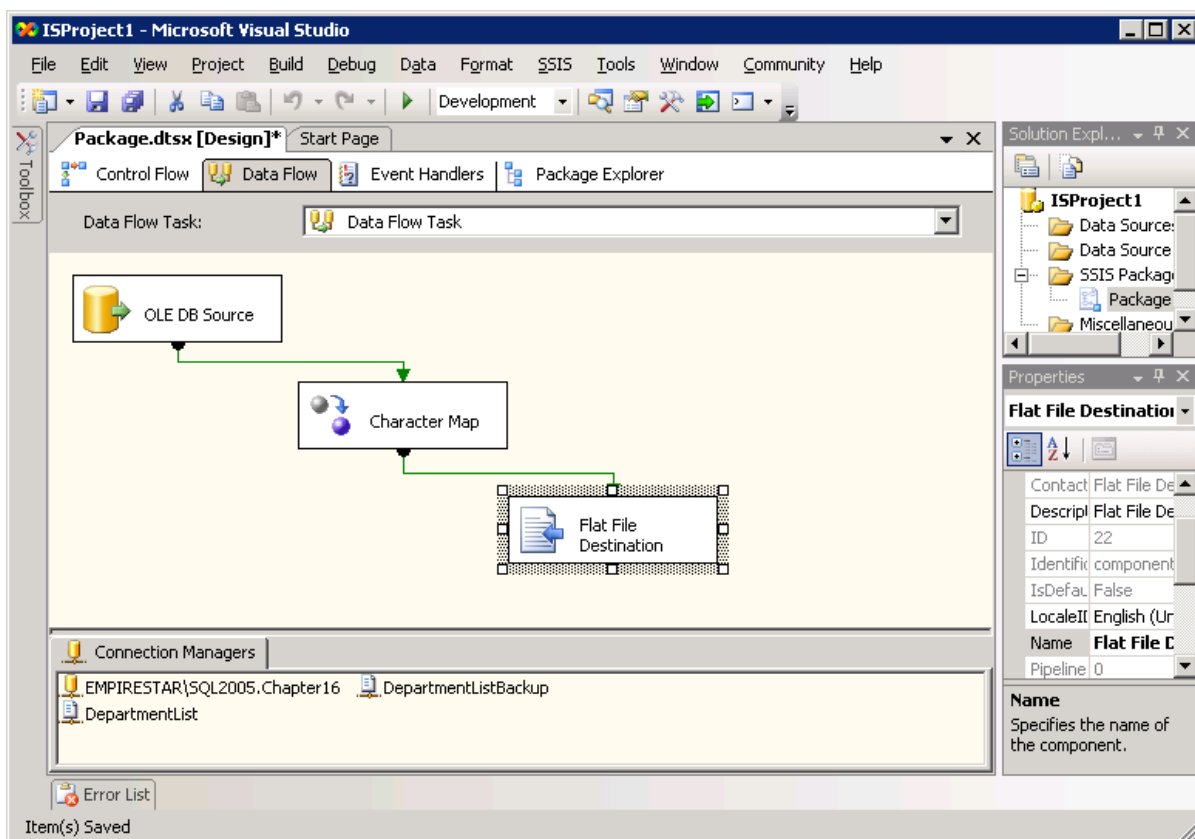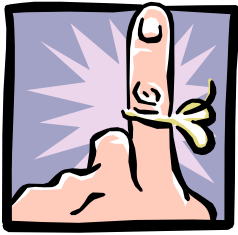Figure 16-10 shows the completed set of data flows.

Figure 16-10: Adding data flows

The data flows in this package take a table from the Chapter16 database, transform one of the columns in that table to all uppercase characters, and then write that transformed column out to a flat file.

Creating Event Handlers

SSIS packages also support a complete event system. You can attach event handlers to a variety of events for the package itself or for the individual tasks within a package. Events within a package "bubble up." That is, suppose an error occurs within a task inside of a package. If you've defined an OnError event handler for the task, then that event handler is called. Otherwise, an OnError event handler for the package itself is called. If no event handler is defined for the package either, the event is ignored.

Event handlers are defined on the Event Handlers tab of the Package Designer. When you create an event handler, you handle the event by building an entire secondary SSIS package, and you have access to the full complement of data flows, control flows, and event handlers to deal with the original event.

	By adding event handlers to the `OnError` event that call the Send Mail task, you can notify operators by e-mail if anything goes wrong in the course of running an SSIS package.

Try It!

To add an event handler to the package we've been building, follow these steps:

1. Open SQL Server Management Studio and connect to your test server.
2. Create a new query and select the Chapter16 database in the available databases list on the toolbar.
3. Enter this text into a query window:

```
CREATE TABLE DepartmentExports(
    ExportID int IDENTITY(1,1) NOT NULL,
    ExportTime datetime NOT NULL
    CONSTRAINT DF_DepartmentExports_ExportTime DEFAULT
(GETDATE()),
    CONSTRAINT PK_DepartmentExports PRIMARY KEY CLUSTERED
    (
    ExportID ASC
    )
)
```

4. Click the Execute toolbar button to create the table.
5. Switch back to the Package Designer in BIDS.
6. Select the Event Handlers tab.
7. Expand the Package and then the Executables node.
8. Select the Data Flow Task in the Executable dropdown list.
9. Select the OnPostExecute event handler.
10. Click the hyperlink on the design surface to create the event handler.
11. Drag an Execute SQL task from the Toolbox and drop it on the Package Designer.
12. Double-click the Execute SQL task to open the Execute SQL Task Editor.
13. Select the OLE DB connection manager as the task's connection.
14. Set the SQL Statement property to INSERT INTO DepartmentExports (ExportTime) VALUES (GETDATE()).
15. Click OK to create the event handler.

This event handler will be called when the Data Flow Task finishes executing, and will insert one new row into the tracking table when it is called.

Saving and Running Packages

Now that you've created an entire SSIS package, you're probably ready to run it and see what it does. But first, let's look at the options for saving SSIS packages. When you work in BIDS, your SSIS package is saved as an XML file (with the extension dtsx) directly in the normal Windows file system. But that's not the only option. Packages can also be saved in the msdb database in SQL Server itself, or in a special area of the file system called the Package Store.

Storing SSIS packages in the Package Store or the msdb database makes it easier to access and manage them from SQL Server's administrative and command-line tools without needing to have any knowledge of the physical layout of the server's hard drive.

Saving Packages to Alternate Locations

To save a package to the msdb database or the Package Store, you use the File ▶ Save Package As menu item within BIDS.

Try It!

To store copies of the package you've developed, follow these steps.

1. Select File ▶ Save Copy of Package.dtsx As from the BIDS menus.
2. Select SSIS Package Store as the Package Location.
3. Select the name of your test server.
4. Enter `/File System/ExportDepartments` as the package path.
5. Click OK.
6. Select File ▶ Save Copy of Package.dtsx As from the BIDS menus.
7. Select SQL Server as the Package Location.
8. Select the name of your test server and fill in your authentication information.
9. Enter `ExportDepartments` as the package path.
10. Click OK.

Running a Package

You can run the final package from either BIDS or SQL Server Management Studio. When you're developing a package, it's convenient to run it directly from BIDS. When the package has been deployed to a production server (and saved to the msdb database or the Package Store) you'll probably want to run it from SQL Server Management Studio.

	SQL Server also includes a command-line utility, `dtsexec`, that lets you run packages from batch files.

Running a Package from BIDS

With the package open in BIDS, you can run it using the standard Visual Studio tools for running a project. Choose any of these options:

- Right-click the package in Solution Explorer and select Execute Package.
- Click the Start Debugging toolbar button.
- Press F5.

Try It!

To run the package that you have loaded in BIDS, follow these steps:

1. Click the Start Debugging toolbar button. SSIS will execute the package, highlighting the steps in the package as they are completed. You can select any tab to watch what's going on. For example, if you select the Control Flow tab, you'll see tasks highlighted, as shown in Figure 16-11.
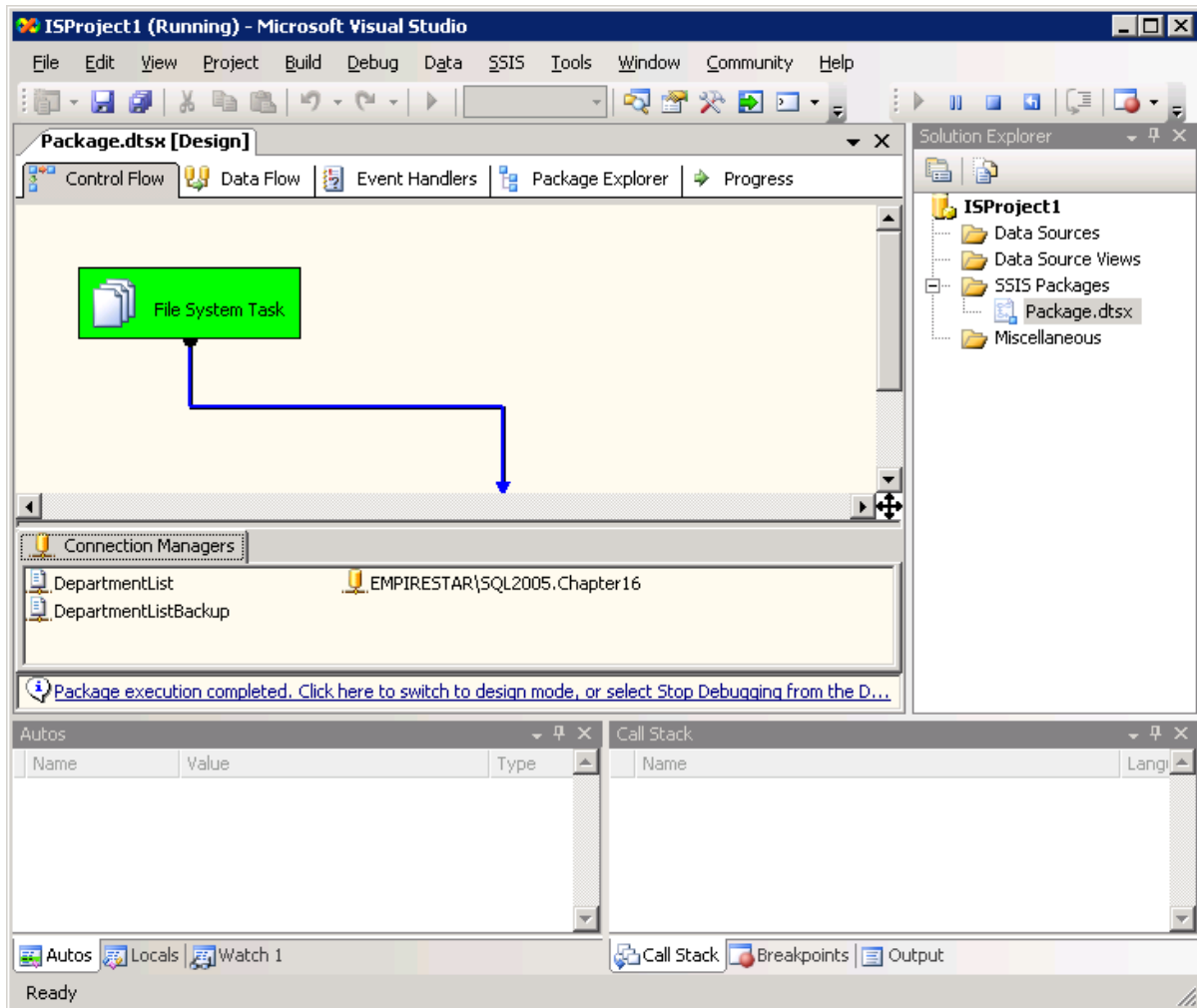
Figure 16-11: Executing a package in the debugger

2. When the package finishes executing, click the hyperlink underneath the Connection Managers pane to stop the debugger.
3. Click the Execution Results tab to see detailed information on the package, as shown in Figure 16-12.
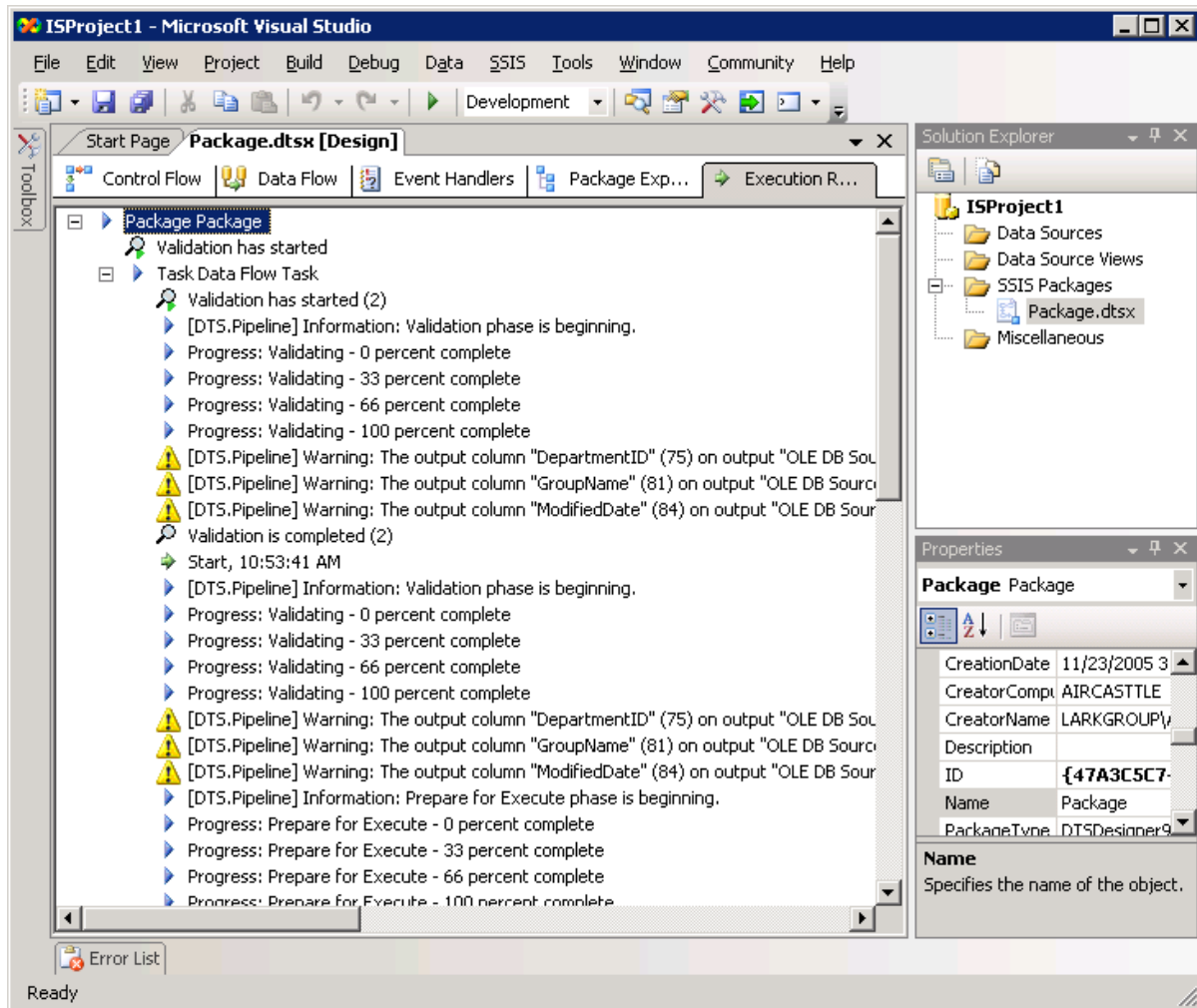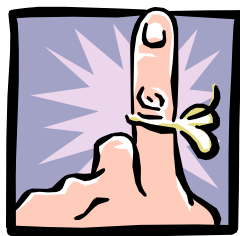
Figure 16-12: Information on package execution

All of the events you see in the Execution Results pane are things that you can create event handlers to react to within the package. As you can see, DTS issues a quite a number of events, from progress events to warnings about extra columns of data that we retrieved but never used.

Running a Package from SQL Server Management Studio

To run a package from SQL Server Management Studio, you need to connect Object Browser to SSIS.

Try It!

1. In SQL Server Management Studio, click the Connect button at the top of the Object Explorer window.
2. Select Integration Services.
3. Choose the server with Integration Services installed and click Connect. This will add an Integration Services node at the bottom of Object Explorer.
4. Expand the Stored Packages node. You'll see that you can drill down into the File System node to find packages in the Package Store, or the MSDB node to find packages stored in the msdb database.
5. Expand the File System node.
6. Right-click on the ExportDepartments package and select Run Package. This will open the Execute Package utility, shown in Figure 16-13.
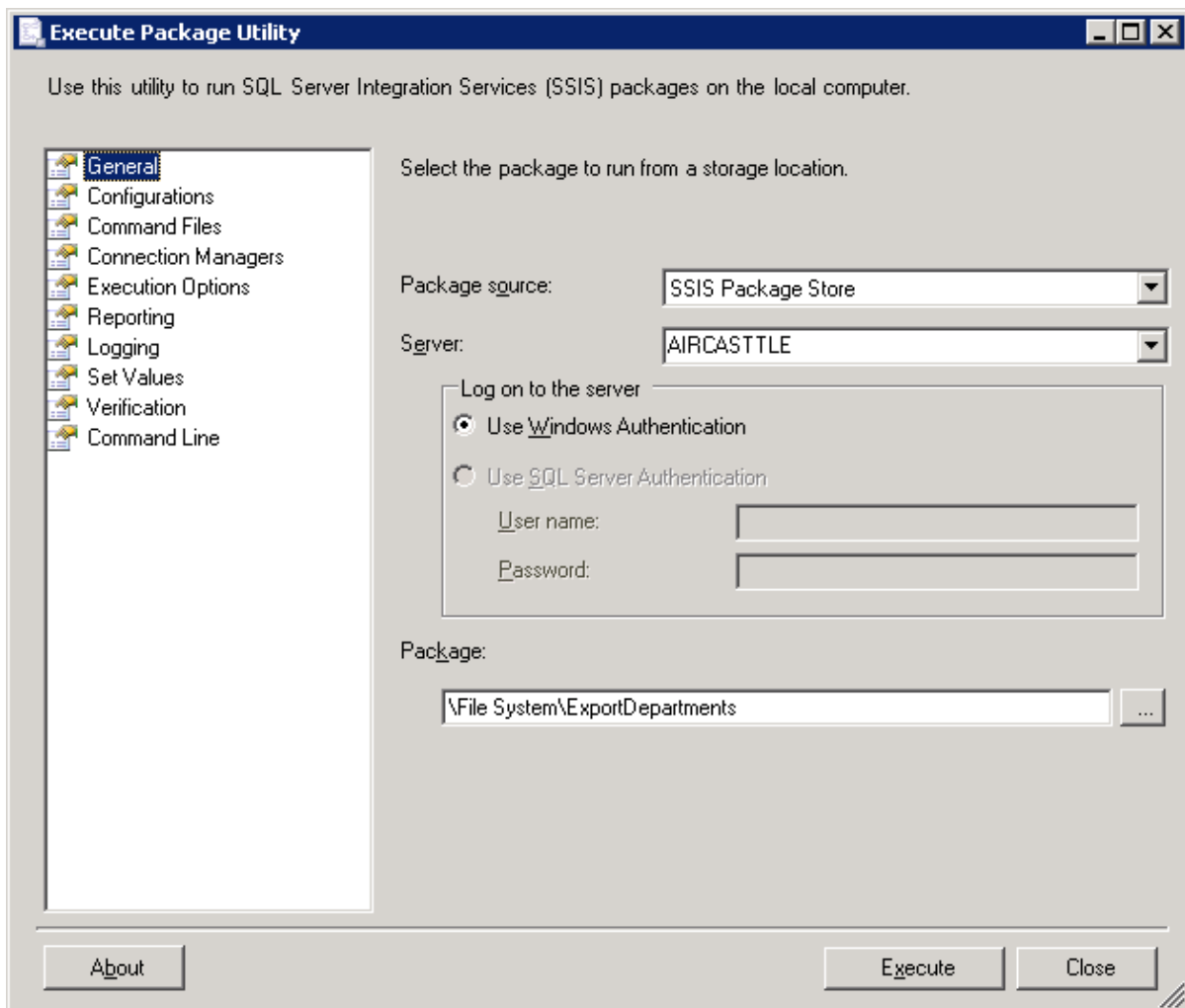
Figure 16-13: Executing a package from SQL Server Management Studio

7. Click Execute.

8. Click Close twice to dismiss the progress dialog box and the Execute Package Utility.

9. Enter this text into a query window with the Chapter16 database selected:

```
SELECT * FROM DepartmentExports
```

10. Click the Execute toolbar button to verify that the package was run. You should see one entry for when the package was run from BIDS and one from when you ran it from SQL Server Management Studio.

Exercises

One common use of SSIS is in data warehousing - collecting data from a variety of different sources into a single database that can be used for unified reporting. In this exercise you'll use SSIS to perform a simple data warehousing task.

Use SSIS to create a text file, c:\EmployeeList.txt, containing the last names and network logins of the AdventureWorks employees. Retrieve the last names from the Person.Contact table in the AdventureWorks database. Retrieve the logins from the HumanResources.Employee table in the Chapter16 database.

You can use the Merge Join data flow transformation to join data from two sources. One tip: the inputs to this transformation need to be sorted on the joining column.

Solutions to Exercises

1. Launch Business Intelligence Development Studio
2. Select File ▶ New ▶ Project.
3. Select the Business Intelligence Projects project type.
4. Select the Integration Services Project template.
5. Select a convenient location.
6. Name the new project ISProject2 and click OK.
7. Right-click in the Connection Managers area of your new package and select New OLE DB Connection.
8. Click New to create a new data connection.
9. In the Connection Manager dialog box, select the SQL Native Client provider.
10. Select your test server and provide login information.
11. Select the AdventureWorks database.
12. Click OK.
13. Right-click in the Connection Managers area of your new package and select New OLE DB Connection.
14. Select the existing connection to the Chapter16 database and click OK.
15. Right-click in the Connection Managers area of your new package and select New Flat File Connection.
16. Enter EmployeeList as the Connection Manager Name.
17. Enter C:\Employees.txt as the File Name.
18. Check the Column Names in the First Data Row checkbox.
19. Click the Advanced icon to move to the Advanced page of the dialog box.
20. Click the New button.
21. Change the Name of the new column to `LastName`.
22. Click the New button.
23. Change the Name of the new column to `Login`.
24. Click OK.
25. Select the Control Flow tab in the Package Designer.
26. Drag a Data Flow Task from the Toolbox and drop it on the Package Designer.
27. Select the Data Flow tab in the Package Designer. The single Data Flow Task in the package will automatically be selected in the combo box.
28. Drag an OLE DB Source from the Toolbox and drop it on the Package Designer.
29. Drag a second OLE DB Source from the Toolbox and drop it on the Package Designer.
30. Drag a Sort Transformation from the Toolbox and drop it on the Package Designer.
31. Drag a second Sort Transformation from the Toolbox and drop it on the Package Designer.
32. Drag a Merge Join Transformation from the Toolbox and drop it on the Package Designer.

33. Drag a Flat File Destination from the Toolbox and drop it on the Package Designer.
34. Click on the first OLE DB Source on the Package Designer to select it.
35. Drag the green arrow from the bottom of the first OLE DB Source and drop it on top of the first Sort Transformation.
36. Click on the second OLE DB Source on the Package Designer to select it.
37. Drag the green arrow from the bottom of the second OLE DB Source and drop it on top of the second Sort Transformation.
38. Click on the first Sort Transformation on the Package Designer to select it.
39. Drag the green arrow from the bottom of the first Sort Transformation and drop it on top of the Merge Join Transformation.
40. In the Input Output Selection dialog box, select the Merge Join Left Input.
41. Click OK.
42. Click on the second Sort Transformation on the Package Designer to select it.
43. Drag the green arrow from the bottom of the second Sort Transformation and drop it on top of the Merge Join Transformation.
44. Click on the Merge Join Transformation on the Package Designer to select it.
45. Drag the green arrow from the bottom of the Merge Join Transformation and drop it on top of the Flat File Destination. Figure 16-14 shows the Data Flow tab with the connections between tasks.
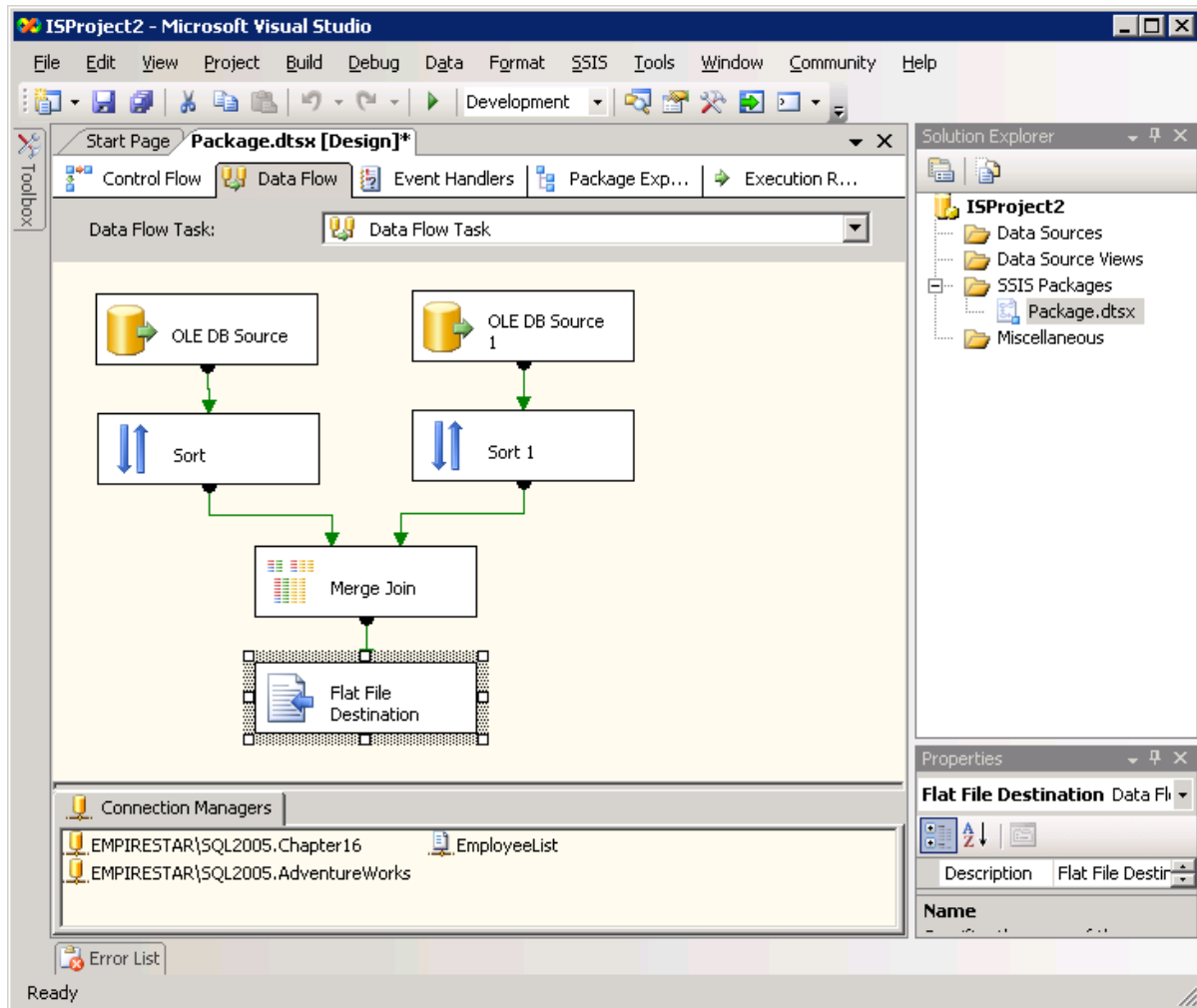
Figure 16-14: Data flows to merge two sources

46. Double-click the first OLE DB Source to open the OLE DB Source Editor.
47. Select the connection to the AdventureWorks database.
48. Select the Person.Contact view.
49. Click OK.
50. Double-click the second OLE DB Source to open the OLE DB Source Editor.
51. Select the connection to the Chapter16 database.
52. Select the HumanResources.Employee view.
53. Click OK.
54. Double-click the first Sort Transformation.
55. Check the ContactID column.
56. Click OK
57. Double-click the second Sort Transformation.
58. Check the ContactID column.
59. Click OK
60. Double-click the Merge Join Transformation.

Introduction to SQL Server 2005

61. Check the Join Key checkbox for the ContactID column in both tables.
62. Check the selection checkbox for the LastName column in the left-hand table and the ContactID and LoginID columns in the right-hand table. Figure 16-15 shows the completed Merge Join Transformation Editor.
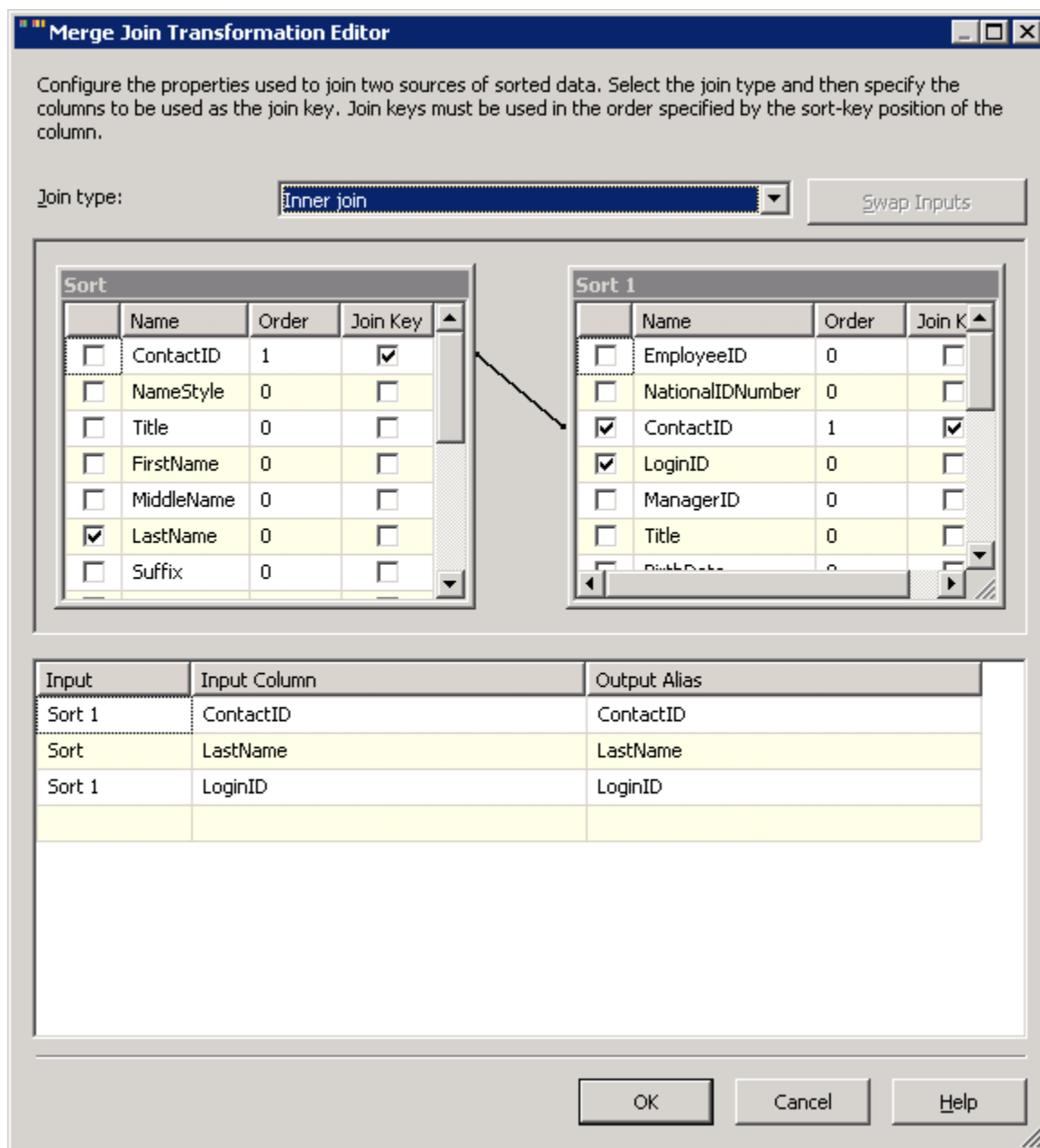

Figure 16-15: Setting up a Merge Join

63. Click OK.
64. Double-click the Flat File Destination.
65. Select the EmployeeList Flat File Connection Manager.
66. Select the Mappings page of the dialog box.

67. The LastName columns will be automatically mapped. Drag the LoginID column from the Available Input Columns list and drop it on top of the Login column in the Available Destination Columns list.
68. Click OK.
69. Right-click the package in Solution Explorer and select Execute Package.
70. Stop debugging when the package is finished executing.
71. Open the c:\Employees.txt file to inspect the results.