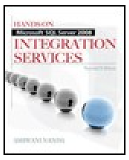


Hands-On Microsoft SQL Server 2008 Integration Services

by Ashwani Nanda
McGraw-Hill/Osborne. (c) 2011. Copying Prohibited.

Chapter 2: Getting Started with Wizards

Overview

Businesses of today are dealing with customers via a variety of channels—postal mail, telephone, e-mail, text message, web portal, agent, dealer, direct marketing, through business partners, and the list goes on. All these channels have their own systems to support business operations and result into data stored in a variety of data storage systems using diverse methodologies, and in different geographical locations. Along with storage requirement considerations, the sheer volume of data can be overwhelming.

Data must be integrated from all disparate sources rapidly within the organization to help it better understand its customers. It is the job of information analysts to integrate data stored in disparate sources. The toolset used to implement data consolidation are commonly known as extraction, transformation, and loading (ETL) tools. SQL Server Integration Services (SSIS) makes the job easy by providing various components that work with most of the data sources and the data stores, and transform data to meet even the most complex requirements.

However, before you jump-start the data integration process, it would be wise to understand the underlying issues with your data. Information is critical but only if accurate. To merge data from different sources, you should make sure that the data is accurate and consistent. Integration Services 2008 provides a Data Profiling Task to profile and analyze such data quality issues. This task generates statistical information in XML format that can be reviewed using Data Profile Viewer. After reviewing the statistical information and taking the corrective actions, you are ready to integrate your data streams.

The SQL Server Import and Export Wizard is the easiest utility to help you move data around. Its interactive and simple graphical user interface (GUI) allows even beginners to use the wizard for learning purposes and simple package development. Seasoned developers also use it for creating basic packages quickly, to be later extended using Business Intelligence Development Studio (BIDS) to reduce development time. The wizard is designed to perform simple tasks quickly and easily; hence it has limited transformation capabilities.

In this chapter you will study

- The SQL Server Import and Export Wizard
- The Integration Services Connections Project Wizard
- The Data Profiling Task and Data Profile Viewer

As the Data Profiling Task works with the data saved in SQL Server 2000 and later, you will be using SQL Server Import and Export Wizard to import data first in SQL Server and then will use Data Profiling Task to analyze the quality of data. This should be your standard practice even in real life while working with Data Profiling Task.

Starting SQL Server Import and Export Wizard

While you can use the SQL Server Import and Export Wizard to import and export data from any SQL Server database, the wizard can also move and transform data that may or may not involve SQL Server.

SQL Server Import and Export Wizard is handily available at locations where you would need it. Following are the locations from where you can start the wizard:

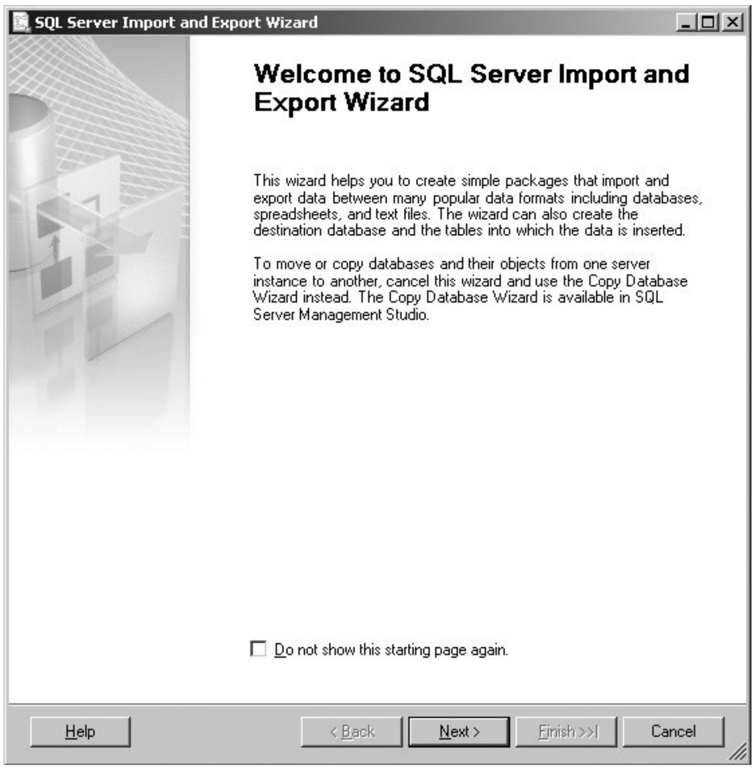
- DTSWizard.exe file
- Microsoft SQL Server 2008 program group
- SQL Server Management Studio
- Business Intelligence Development Studio (BIDS)

The behavior of SQL Server Import and Export Wizard varies slightly based on the location from where you start it. You will use different methods to start SSIS Import and Export Wizard while working in the following exercises. Before we progress further, install the software provided for the exercises in the book. Please refer to the Appendix for further details on this.

Hands-On: Importing a Flat File into SQL Server 2008

In your first exercise, you will import a simple text file into SQL Server 2008 and create a new database while importing this file.

1. Choose Start | Run. In the Run dialog box's Open field, type **DTSWizard**; then click OK to open the SQL Server Import and Export Wizard's Welcome page, as shown in Figure 2-1. (You can also run this file from the command prompt or by double-clicking the DTSWizard.exe file in the default location of C:\Program Files\Microsoft SQL Server\100\DTS\Binn.)


SQL Server Import and Export Wizard
Welcome to SQL Server Import and Export Wizard
Do not show this starting page again.
Help
< Back
Next >
Finish >>|
Cancel

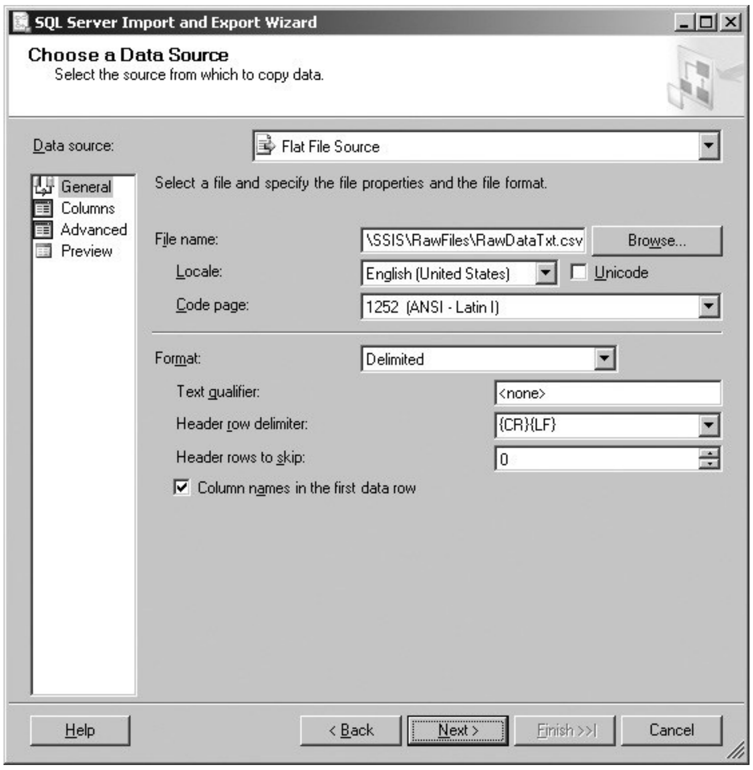
This wizard helps you to create simple packages that import and export data between many popular data formats including databases, spreadsheets, and text files. The wizard can also create the destination database and the tables into which the data is inserted.

To move or copy databases and their objects from one server instance to another, cancel this wizard and use the Copy Database Wizard instead. The Copy Database Wizard is available in SQL Server Management Studio.

Figure 2-1: The Import and Export Wizard's Welcome page

2. Click Next to open the Choose A Data Source window. By default, the wizard is set to point to a (local) server using SQL Server Native Client 10.0. Click the down arrow next to the Data Source field to see the other available data sources to which this wizard can connect. The drop-down list shows providers for several data sources that are installed on the local machine, including the following:
   - Flat Files
   - Microsoft Excel
   - Microsoft Access
   - .NET Framework Data Provider for Oracle
   - Microsoft OLE DB Provider for SQL Server
   - SQL Server Native Client 10.0
3. The Import and Export Wizard can connect to and extract data from any of these data sources. Select Flat File Source from the Data Source drop-down list. The options on screen change to match the Flat File Source requirements.
4. Click the Browse button next to the File Name field, and in the Open dialog box navigate to the C:\SSIS\RawFiles folder. Select the CSV files (*.csv) as the file type to display .csv files, and then choose the RawDataTxt.csv file and click Open. You'll see that most of the options in the flat files fields will be automatically filled in for you as the wizard reads the file format.
5. Click the Column names in the first data row check box option, as shown in Figure 2-2, and then click Columns in the

pane on the left.


SQL Server Import and Export Wizard

Choose a Data Source
Select the source from which to copy data.

Data source: Flat File Source

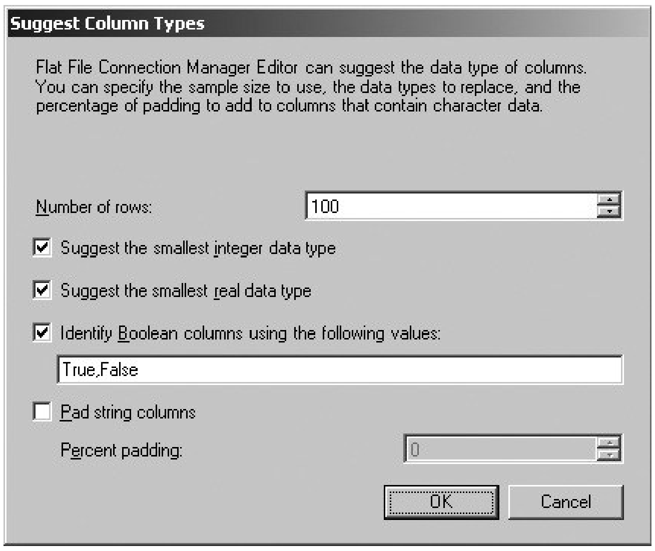
Select a file and specify the file properties and the file format.

File name: \SSIS\RawFiles\RawDataTxt.csv
Locale: English (United States)
☐ Unicode
Code page: 1252 (ANSI - Latin I)

Format: Delimited
Text qualifier: <none>
Header row delimiter: {CR}{LF}
Header rows to skip: 0
☑ Column names in the first data row

Figure 2-2: Selecting a data source in the SQL Server Import and Export Wizard

On this page, data is listed in the bottom half of the window. Note that the Column delimiter is Comma by default and that the headings of columns are displayed properly and are actually picked up from the first row of the text file. Go to the Advanced page by clicking it in the left pane. You'll see the list of columns in the middle pane and the column properties for the selected column in the right pane. You can change column properties such as Data Type or OutputColumnWidth here. To help you quickly identify the data types and length of the columns in the flat file a Suggest Types button has been provided. Click Suggest Types to see the options available in the Suggest Column Types dialog box shown in Figure 2-3. If you feel that the data types suggested by the flat file adapter are not correct, the most likely reason is that the flat file adapter didn't sample enough rows to guess it right. You can change this in Number Of Rows box by increasing the number of rows to sample up to a maximum of 9999. Note that if you choose a very large number here, the flat file adapter will take more time to suggest data types and you still must check that the appropriate column lengths and data types have been selected. Review the other options and click Cancel to come back.


Suggest Column Types

Flat File Connection Manager Editor can suggest the data type of columns. You can specify the sample size to use, the data types to replace, and the percentage of padding to add to columns that contain character data.

Number of rows: 100
☑ Suggest the smallest integer data type
☑ Suggest the smallest real data type
☑ Identify Boolean columns using the following values:
True,False
☐ Pad string columns
Percent padding: 0

Figure 2-3: Specifying a number of rows to sample appropriately

Finally, if you click Preview, you will be able to set a value in the Data Rows to Skip field. Click Next to move on.

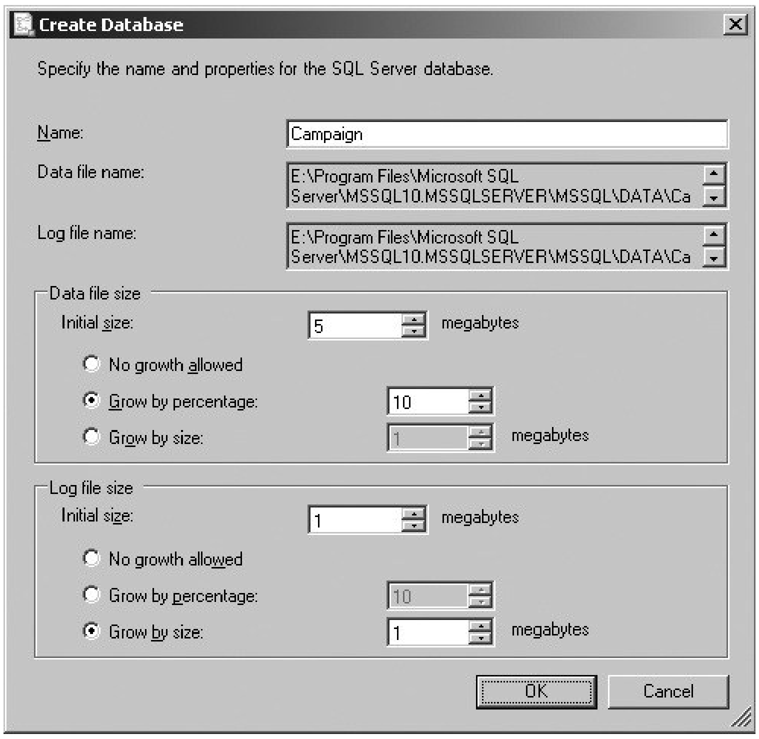
6. In the Choose a Destination window, leave the default SQL Server Native Client 10.0 selected in the Destination field. Microsoft SQL Server Native Client 10.0 provides an OLE DB interface to Microsoft SQL Server databases and supports several new features added to SQL Server 2008 such as Table-Valued Parameters, FILESTREAM, Large CLR UDTs, Sparse Columns, extended Service Principal Names (SPN) support, and Date and Time improvements.

   Just to remind you once again, you don't need to have SQL Server to run the SQL Server Import and Export Wizard or BIDS for developing Integration Services packages. You can actually move or transform your data from any source to any destination without involving SQL Server. So, for example, you can choose Destination as a text file and remove some of the columns from the destination file (so that you generate a modified text file from a text file), one operation that isn't easy to perform on text files otherwise.

   You can specify your server name in the Server Name field or select a name from the drop-down list by clicking the down arrow. Leave (local) specified for this exercise and leave the radio button for Use Windows Authentication selected.

   In the Database field, either you can specify a database name or you can opt to create a new database. The software provided for the book contains files for the Campaign database that you will use to perform various exercises. Please refer to the Appendix for further details on this. If you haven't attached the provided Campaign database yet to your SQL Server database engine, you can create it here by clicking the New button. Click New to create a new database.

7. If your login for SQL Server is correct, a new Create Database dialog box will open. Note that even if you can connect and open this dialog box, you still are required to have appropriate permissions on the SQL Server to create a new database; otherwise you'll a get create database permission denied error when you click OK to create a new database. Type **Campaign** in the Name field. The wizard will complete the rest of the details using the default database options similar to those shown in Figure 2-4. Leave the default options as is, and click OK to create the Campaign database. Note that if you've already attached the Campaign database provided with the software for this book, then you do not need to create the Campaign database here. Click Next to proceed.



Specify the name and properties for the SQL Server database.

Name: Campaign

Data file name: E:\Program Files\Microsoft SQL Server\MSSQL10.MSSQLSERVER\MSSQL\DATA\Ca

Log file name: E:\Program Files\Microsoft SQL Server\MSSQL10.MSSQLSERVER\MSSQL\DATA\Ca

Data file size
Initial size: 5 megabytes
No growth allowed
[✓ Grow by percentage: 10]
Grow by size: 1 megabytes

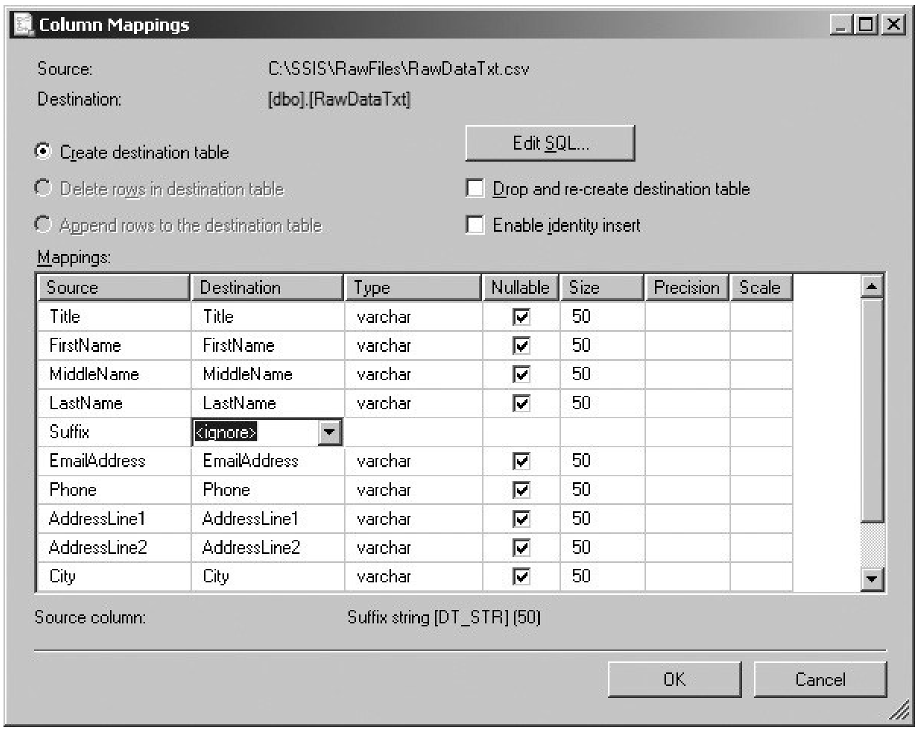
Log file size
Initial size: 1 megabytes
No growth allowed
Grow by percentage: 10
[✓ Grow by size: 1 megabytes]

Figure 2-4: Creating a new database in the SQL Server Import and Export Wizard

8. In the Select Source Tables And Views window, verify that the check box next to the filename listed under Source is selected. The Destination will automatically be filled in with a table name. Click Edit Mappings. In the Column Mappings dialog box as shown in Figure 2-5, you have various options to configure your destination table and the

way you want to insert data into it. Following is the description of each of these options:

- **Create destination table** This option will be selected in your case. Choose this option whenever you want to create a destination table. When this option is highlighted and selected, you can click Edit SQL to generate an SQL statement that will be used to create this new table. You can customize this SQL statement; however, if you modify the default SQL statement, you will have to manually modify the column mappings.
- **Delete rows in destination table** This option will not be available to you, as the destination table is yet to be created. However, if you select an existing table to import data into, you can then choose to delete rows from the existing table.
- **Append rows to the destination table** If you're importing into an existing table, you can choose to append the rows to the existing table. However, in your case, this option will be grayed out.
- **Drop and re-create destination table** This check box option allows you to delete an existing table and then re-create and load data into it.
- **Enable identity insert** You will be using this option if you want to keep the identity in the input data and want to insert it into the destination table.


Column Mappings

Source: C:\SSIS\RawFiles\RawDataTxt.csv
Destination: [dbo].[RawDataTxt]

[✓ Create destination table]
Delete rows in destination table
Append rows to the destination table
Drop and re-create destination table
Enable identity insert

Mappings:

Source	Destination	Type	Nullable	Size	Precision	Scale
Title	Title	varchar	✓	50		
FirstName	FirstName	varchar	✓	50		
MiddleName	MiddleName	varchar	✓	50		
LastName	LastName	varchar	✓	50		
Suffix	<ignore>					
EmailAddress	EmailAddress	varchar	✓	50		
Phone	Phone	varchar	✓	50		
AddressLine1	AddressLine1	varchar	✓	50		
AddressLine2	AddressLine2	varchar	✓	50		
City	City	varchar	✓	50		

Source column: Suffix string [DT_STR] (50)

Figure 2-5: Making changes in the mappings between Source and Destination

9. Let's change a mapping in the Mappings section to see how it works. Click in the Suffix destination column and then the drop-down box to select <Ignore>, as shown in Figure 2-5.

   Now click Edit SQL, and note that the SQL statement for creating the destination table has picked up the change you just made. You can make changes to the SQL statement or use already existing scripts to create a table. However, once you have modified the SQL statement, any subsequent changes made from the Mappings section will not be automatically picked up by the SQL, but must be manually edited. First click Cancel and then click OK to return to the Select Source Tables And Views window. Click Next to move on to the Save And Run Package dialog box.

10. By default, the Run Immediately option will be selected. Check the Save SSIS Package option to save this package, and then select the File System radio button. Click the drop-down arrow in the Package Protection Level dialog box to see the available options. Integration Services protects your package according to the protection level you select for your package. You will study these options in detail in Chapter 7; however, brief descriptions of these options are provided here for your quick review:

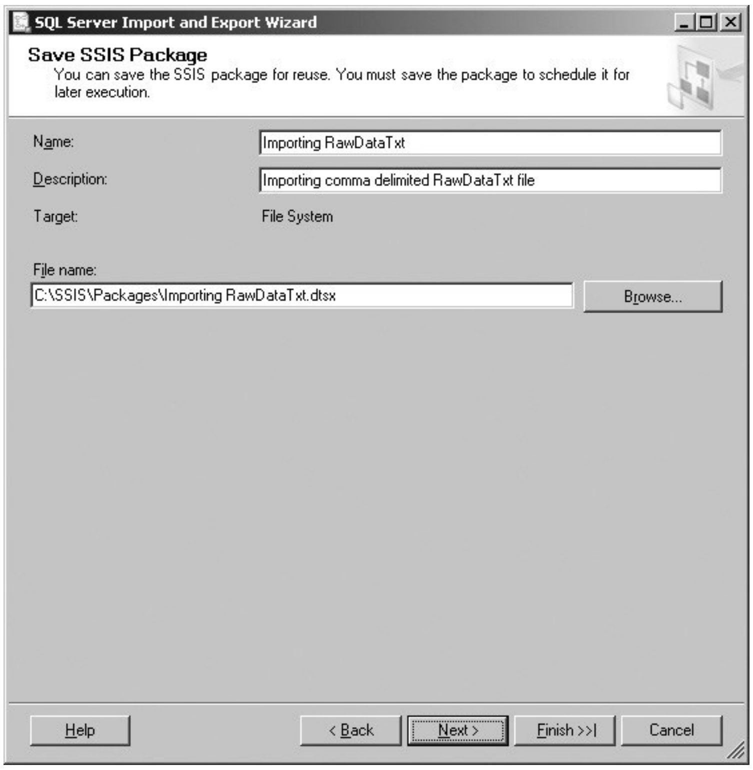
    - **Do not save sensitive data** Using this option, you choose not to save sensitive information such as connection strings, passwords, and so on within the package.

- **Encrypt sensitive data with user key** This is the default option that can save sensitive information within the package after encrypting it using a key that is based on the user who creates and saves the package. With this option selected, only the user who creates the package will be able to run the package without providing the sensitive information. If other users try to load the package, the sensitive information is replaced with blank data and the package will fail to run unless they provide the sensitive information manually.
- **Encrypt sensitive data with password** With this option, you choose to save the sensitive information in the package after encrypting it with a password. Once the package is encrypted with this option, any user will be able to open and execute the package by providing the password. If the user is unable to provide the password, he or she will have to provide the sensitive information manually to run the package.
- **Encrypt all data with user key** Selecting this option means that all the information and metadata in the package will be encrypted using a key that is based on the user who is creating the package. Once the package has been encrypted using this option, only the user who created the package can open and run it.
- **Encrypt all data with password** With this option, you can encrypt all the information and the metadata in the package by providing a password. Once the package has been encrypted with this option, it can be opened and run only after providing the password.

If you choose to save the SSIS package into SQL Server, you will see one additional option in the Package protection level:

- **Rely on server storage and roles for access control** When you save your packages to the MSDB database of an SQL Server, Integration Services provides additional security features that allow you to secure packages based on the three fixed database-level roles: db_ssisadmin, db_ssisltduser, and db_ssisoperator. You will study more about these roles and the ways to use them to control access to your packages in Chapter 7.

11. Select the Encrypt sensitive data with password option and type **12wsXZaq** in the Password and Retype Password fields and click OK.
12. Type **Importing RawDataTxt** in the Name field and **Importing comma delimited RawDataTxt file** in the Description field. Type **C:\SSIS\Packages\Importing RawDataTxt** in the File Name field, as shown in Figure 2-6.



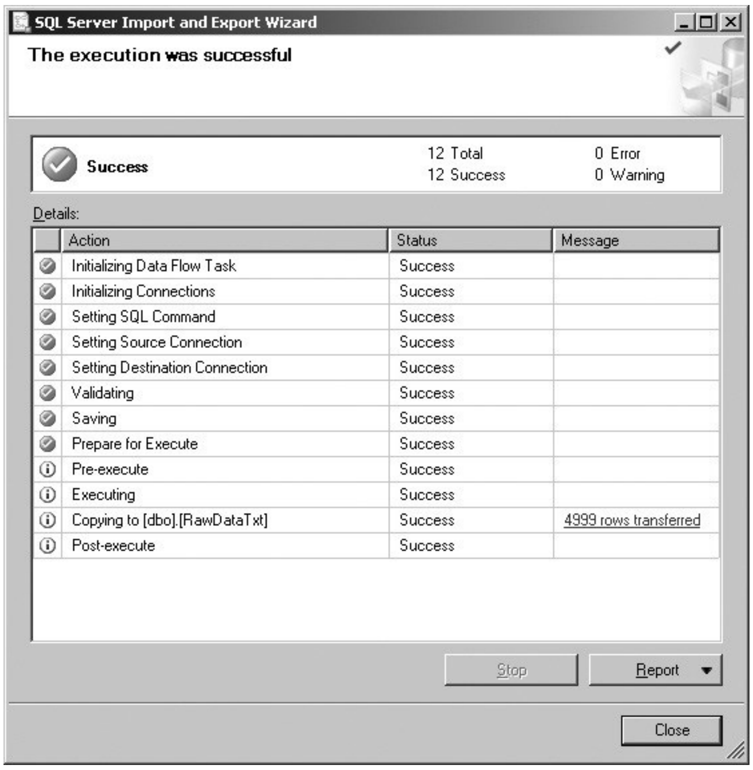
You can save the SSIS package for reuse. You must save the package to schedule it for later execution.

Name: Importing RawDataTxt
Description: Importing comma delimited RawDataTxt file
Target: File System
File name: C:\SSIS\Packages\Importing RawDataTxt.dtsx

Figure 2-6: Saving SSIS package to the file system

13. Click Next to open the Complete The Wizard screen. Here you have the option of reviewing the choices you have made. Once you are happy with them, click Finish.

    The wizard now runs the package created, imports all the records, and shows a comprehensive report for all the intermediary steps it processed, as shown in Figure 2-7. This window not only shows any messages or errors appearing at run time but also suggests some of the performance enhancements that can be applied to your SSIS packages.



Action	Status	Message
Initializing Data Flow Task	Success	
Initializing Connections	Success	
Setting SQL Command	Success	
Setting Source Connection	Success	
Setting Destination Connection	Success	
Validating	Success	
Saving	Success	
Prepare for Execute	Success	
Pre-execute	Success	
Executing	Success	
Copying to [dbo].[RawDataTxt]	Success	4999 rows transferred
Post-execute	Success	

Figure 2-7: Checking the execution results

You can see the detailed report by clicking Report and selecting View Report. You can also save the report to a file, copy it to the Clipboard, or send it as an e-mail using the Report button.

14. Click Report and choose View Report. Scroll down to the Validating section to read that SQL Server Import and Export Wizard suggests removing the Suffix column from the source selections to improve the performance of execution, as this is not used in the Data Flow task. The data is handled in the memory buffers as it flows through the data flow or the pipeline. More columns mean that fewer rows can be accommodated in a memory buffer. So, removing unused columns will save memory space and allows more rows to be accommodated in a buffer, making the data flow task more efficient. It is always worth taking a quick look at this report, which may offer valuable comments about the package performance. Close all the windows.

15. Start Windows Explorer and navigate to the C:\SSIS\Package folder. You will see a file called Importing RawDataTxt.dtsx saved in this folder. This is the Integration Services package saved in the XML format with an extension of *.dtsx*. Right-click this file, choose Open With, and then choose Notepad from the list of programs. You will see that the package settings listed in this file are in XML format. Close Notepad without making any changes.

16. Choose Start | All Programs | Microsoft SQL Server 2008, and click SQL Server Management Studio to start the program.

17. When the SQL Server Management Studio starts, you will be asked to provide authentication and connection details in the Connect To Server dialog box. In the Server Type field, select Database Engine. You can connect to various services such as Analysis Services, Integration Services, Reporting Services, and SQL Server Compact Edition by selecting an option from the drop-down menu. Specify the server name or simply type **localhost** to connect to the local server in the Server name field. In the Authentication field, leave Windows Authentication selected and click Connect to connect to the SQL Server 2008 database engine.

18. When the SQL Server Management Studio loads, in the Object Explorer on the left, expand the Databases node. Note that the Campaign Database is also listed in the list of databases. Expand the Campaign Database node and then Tables node by clicking the plus sign next to them. You will see the dbo.RawDataTxt table listed there.

19. Click New Query on the menu bar of Management Studio to open a query pane. Type the following in the query pane, highlight it, and press F5 to run it.

    ```
    SELECT * FROM Campaign.dbo.RawDataTxt
    ```

    You will see all the records imported into this table in the lower half of the pane.

Review

Now that you've used the SQL Server Import and Export Wizard to import data from a .CSV text file, try your hand at exporting the data from Campaign database to an Excel file by starting the SQL Server Import and Export Wizard from within the SQL Server Management Studio. To give you a hint, right-click the Campaign database and choose Export to start. Also, don't forget to save the package, as you will use it in the next Hands-On.

Using Business Intelligence Development Studio

One of the features of SSIS Import and Export Wizard is that the packages can be saved for later use and can be extended with BIDS. However, note that the packages cannot be saved while working with SQL Server Import and Export Wizard in the SQL Server Express Edition.

You have read about BIDS in Chapter 1 and have also created a blank Integration Services project. In the following Hands-On exercise, you will open this blank project and then add the packages you just created in the last Hands-On. Don't worry if you don't understand exactly how Integration Services works; the purpose of the following exercise is to give you a preview, a feel, of how an Integration Services package looks. All the features shown here are covered in detail in Chapter 3.

Hands-On: Exploring an SQL Server Import and Export Wizard Package Using BIDS

In this Hands-On exercise, you will open the various components of the Integration Services package and explore their properties and configurations. Though you will be learning about various attributes and properties of the components, the focus will be on learning how these components work together rather than understanding each component in detail.

Method

This Hands-On exercise is broken down into the following parts:

- You will add an existing package to an Integration Services blank project.
- You will then explore the components in the Control Flow and Data Flow Designer surfaces.

Exercise (Adding SSIS Import and Export Wizard Package in BIDS)

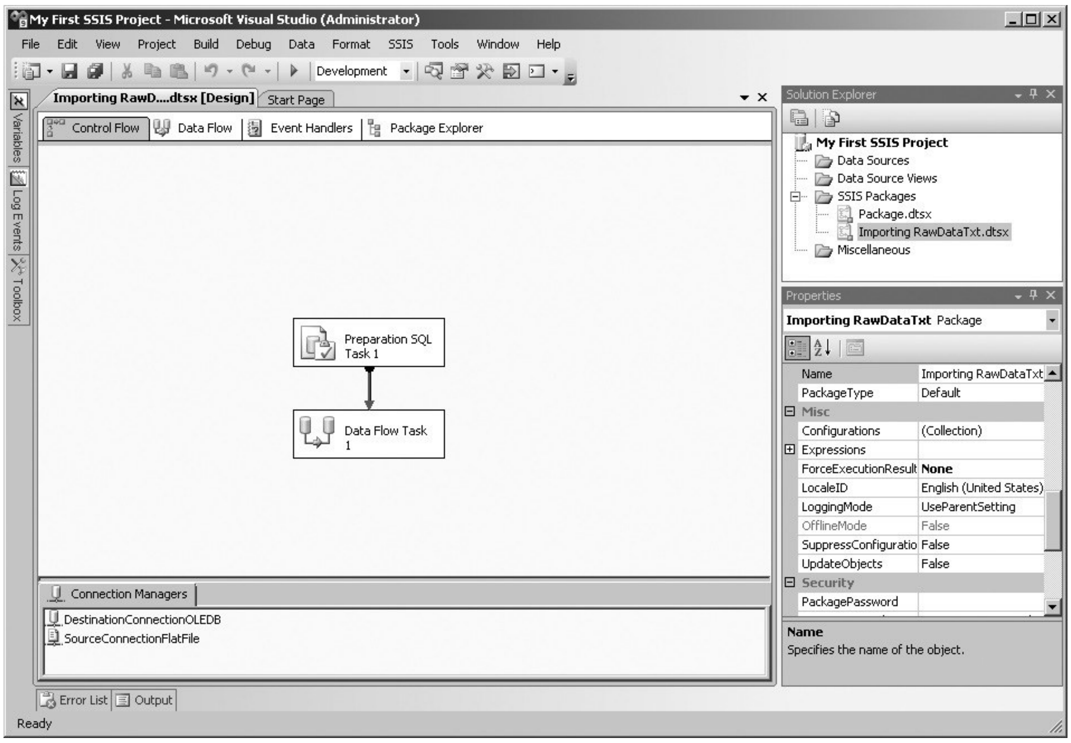
In this part, you will add the package Importing RawDataTxt.dtsx that you've created earlier using the SQL Server Import and Export Wizard, in My First SSIS Project that you created in Chapter 1.

1. Start the SQL Server Business Intelligent Development Studio.

2. Open My First SSIS Project from the Start Page | Recent Projects section. Alternatively, you can also open this package from the File | Open | Project/Solution option and navigating to the C:\SSIS\ Projects\My First SSIS Project folder to select My First SSIS Project.sln. The blank solution you created in Chapter 1 will open up.

3. In Solution Explorer, right-click the SSIS Packages node and select Add Existing Package from the context menu. This will open the Add Copy Of Existing Package dialog box. Select the File System in the Package Location field. In the Package Path field, type **C:\SSIS\Packages\Importing RawDataTxt.dtsx**. Alternatively, you can use the ellipsis button (. . .) provided opposite to the Package Path field to locate and select this package. Click OK to add this package to the SSIS Packages node.

4. Though the package has been added to the SSIS Package node, it has not yet been loaded into the designer environment. Double-click the Importing RawDataTxt .dtsx package to load it into the designer environment. You will see the package loaded in BIDS, as shown in Figure 2-8.



Name	Importing RawDataTxt
PackageType	Default
Misc	
Configurations	(Collection)
Expressions	
ForceExecutionResult	None
LocaleID	English (United States)
LoggingMode	UseParentSetting
OfflineMode	False
SuppressConfiguratio	False
UpdateObjects	False
Security	
PackagePassword	

Name
Specifies the name of the object.

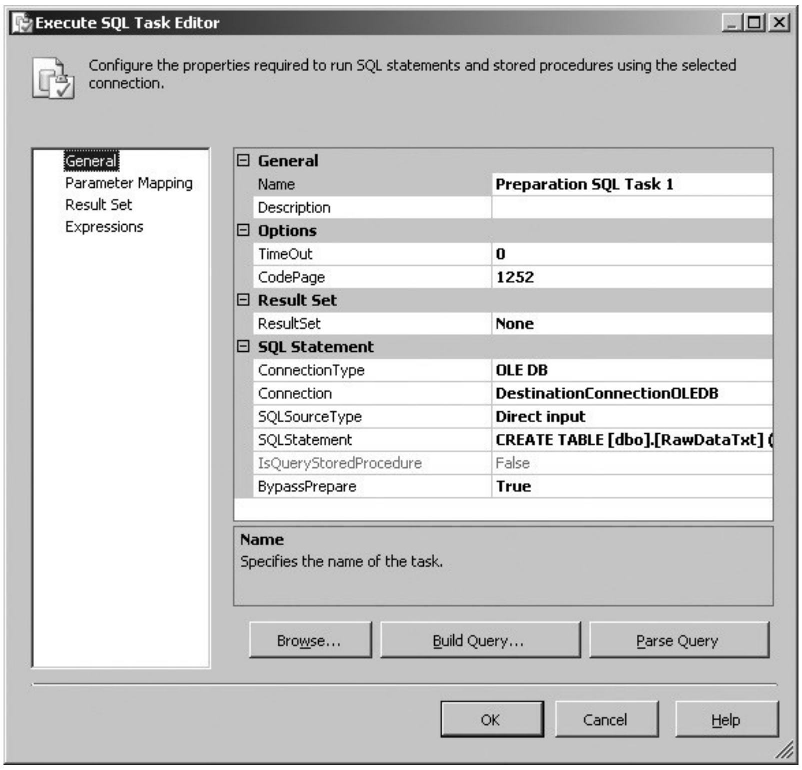
Figure 2-8: Loading the Importing RawDataTxt.dtsx package in BIDS

5. Note that the Connection Managers tab, on the bottom half of the SSIS designer, shows two connection managers—DestinationConnectionOLEDB for choosing the Campaign database as a destination and SourceConnectionFlatFile for the source RawDataTxt.csv text file. Right-click SourceConnectionFlatFile and choose Edit from the context menu to edit this connection. You will see the Flat File Connection Manager Editor dialog box, which is similar to the window you filled in for the flat file while running the SSIS Import and Export Wizard. Explore the various tabs and click Cancel when you're done.
6. Right-click DestinationConnectionOLEDB and click Edit. In the Connection Manager dialog box, note that the layout is different, but the settings are exactly the same as those you chose in the Choose A Destination window while running the SQL Server Import and Export Wizard. Click Cancel.

Exercise (Exploring Control Flow Components)

As you know, Integration Services has separate engines—control flow for managing workflow and data flow to manage the pipeline activities. The package Importing RawDataTxt.dtsx creates a table called RawDataTxt in the Campaign database while running in the control flow engine's scope. It uses Execute SQL task to perform this function. After it has created the table, it passes control over to the data flow engine to perform the data-related activities.

7. In the Control Flow tab, double-click Preparation SQL Task 1 to open the Execute SQL Task Editor dialog box, shown in Figure 2-9. You'll see various configuration options that have been set for this task.



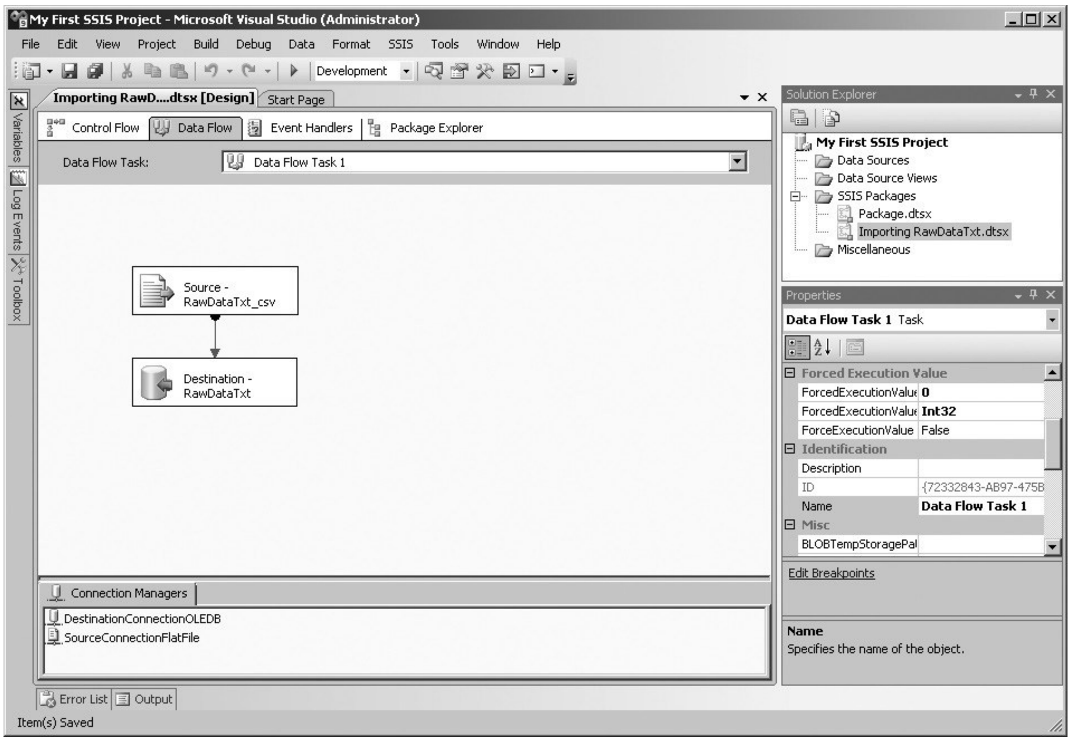
General	
Name	Preparation SQL Task 1
Description	
Options	
TimeOut	0
CodePage	1252
Result Set	
ResultSet	None
SQL Statement	
ConnectionType	OLE DB
Connection	DestinationConnectionOLEDB
SQLSourceType	Direct input
SQLStatement	CREATE TABLE [dbo].[RawDataTxt] (
IsQueryStoredProcedure	False
BypassPrepare	True

Name
Specifies the name of the task.

Figure 2-9: Preparation SQL Task Settings

Note that under the SQL Statement section, the Connection field is pointing to the Campaign database using DestinationConnectionOLEDB Connection Manager. SQLSourceType is currently set to Direct Input, which gives you the option of typing an SQL statement. If you hover your mouse over the SQLStatement field, the SQL query will pop up. Clicking in this field enables an ellipsis button, which you can click to open the Enter SQL Query window. In the Enter SQL Query window, you can type in a long query or paste in an existing one. Click Cancel to close the Enter SQL Query window. You can also configure this task to read an SQL statement from a file, which is available only when you've selected File Connection in the SQLSourceType field. You can also click Build Query to build simple queries and Parse Query to check the syntax of your queries. Click Cancel to undo any changes you have made and return to the SSIS Designer. The most important thing to understand at this time is that using this task, Integration Services package creates a table RawDataTxt in the Campaign database.

8. Double-click Data Flow Task 1 to see the Data Flow tab of the BIDS. The components that constitute data flow in a package reside here. The Data Flow tab will look similar to Figure 2-10.



Data Flow Task: Data Flow Task 1

Forced Execution Value	
ForcedExecutionValu	0
ForcedExecutionValu	Int32
ForceExecutionValue	False
Identification	
Description	
ID	{72332843-AB97-475B
Name	Data Flow Task 1
Misc	
BLOBTempStoragePat	

Figure 2-10: Data Flow consists of a Source and a Destination

The data flow engine is responsible for data extraction, manipulation and transformation, and then loading data into the destination. In this package, the source and destination are:

- **Source** RawDataTxt_csv, a Data Flow Source
- **Destination** RawDataTxt, a Data Flow Destination

This is a simple data loading package that doesn't have any transformation component. The Data Flow Source, Source—RawDataTxt_csv, extracts data from the flat file RawDataTxt.csv. It then passes on the data to the Destination—RawDataTxt that loads the extracted data into an SQL Server table, which was created earlier by an Execute SQL task in the Control Flow.

9. The first component, Source—RawDataTxt_csv in the Data Flow, is called a *Flat File Source*, which reads the data rows from the RawDataTxt.csv file row by row and forwards them to the downstream destination component.

   If you hover your mouse on the Toolbox, which is shown as a tabbed window on the left side of the screen in Figure 2-10, it will slide out to show you all the components that can be used in the data flow. Note that Flat File Source appears in the Data Flow Sources section and the OLE DB Destination appears in the Data Flow Destinations section. There are several transformations components listed in the Data Flow Transformations section, though none is used here. This simple package shows a typical example of a data flow that contains a Data Flow Source and a Data Flow Destination.

   Finally, the Data Flow Destination, Destination—RawDataTxt, is an OLE DB destination that loads the converted records to the RawDataTxt table in the Campaign database.

10. The data flow components of an Integration Services package expose their properties in a custom user interface that is built for most of the components, or in the Advanced Editor that is common to all the components. Some of the components do not have a custom user interface, so they use only the Advanced Editor to expose their properties and attributes. Sometimes you may have to use the Advanced Editor even though the component has a custom UI, as some components do not expose all the properties in the custom UI. You can open the custom user interface by choosing the Edit command from the component's context menu and the Advanced Editor using the Show Advanced Editor command. Right-click the Source—RawDataTxt_csv object and choose Show Advanced Editor from the context menu.

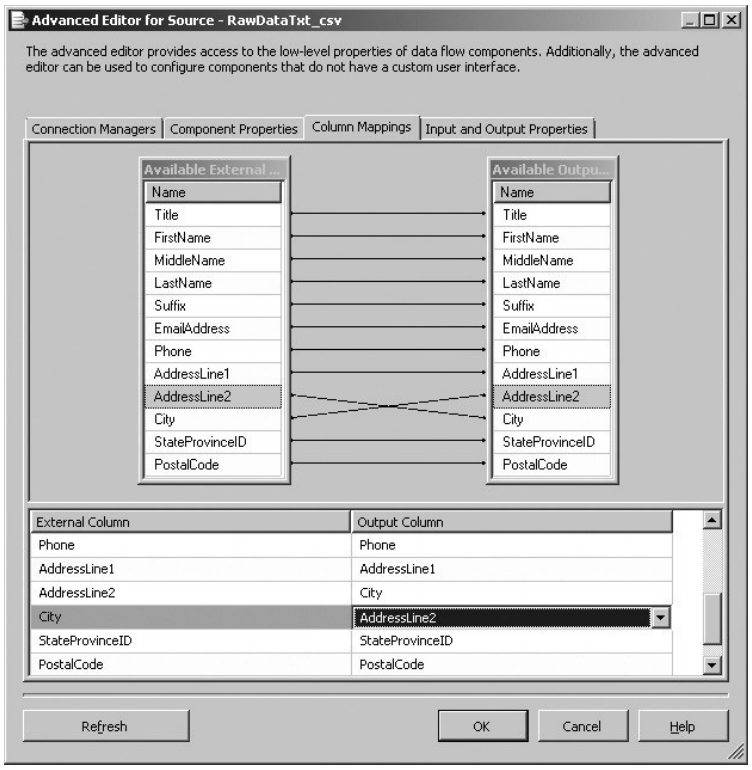
11. You will see four tabs in the Advanced Editor. The Connection Managers tab specifies SourceConnectionFlatFile connection manager that this component uses to connect to the RawDataTxt.csv flat file. The Connection Manager

field here displays all the connection managers defined in the package.

12. Move on to the Component Properties tab. Here you will see the Common Properties that specify properties such as Name and Description, and Custom Properties sections.

13. Click the Column Mappings tab. In the upper half of this tab, you can see the columns mapped by the mapping lines and the lower half lists these mapped external columns with the output columns. External columns reference the data columns read from the source text file and the output columns are the columns this adapter passes on to the downstream data flow component. These output columns will become input columns for the next component in the data flow.

14. You can change these mappings if you want an External Column to be redirected to a different Output Column. Click the mapping line joining the AddressLine2 columns of Available External Columns and Available Output Columns and press the DELETE key on your keyboard. Similarly, delete the mapping line joining City columns. Now click and hold the mouse on the AddressLine2 column in the Available External Columns list and drag and drop it on the City column in the Available Output Columns list. You've created a mapping line to map AddressLine2 column to City column, which means the data in the AddressLine2 column will be sent to City column. This can also be done in the lower half of the window. Click the column that shows <Ignore>, just below City, in the Output Column. The column is converted into a drop-down list box. Click the down arrow to see the list of available columns and choose AddressLine2 from the list. As you do that, a mapping line corresponding to the affected columns will be added in the upper section. Your mappings should look as shown in Figure 2-11.



The advanced editor provides access to the low-level properties of data flow components. Additionally, the advanced editor can be used to configure components that do not have a custom user interface.

External Column	Output Column
Phone	Phone
AddressLine1	AddressLine1
AddressLine2	City
City	AddressLine2
StateProvinceID	StateProvinceID
PostalCode	PostalCode

Figure 2-11: Working with column mappings

Now right-click anywhere on the blank surface in the upper half and choose Select All Mappings to select all the mapping lines. Again right-click and choose Delete Selected Mapping. This will remove all the mappings, and the Output Column in the lower half of the window shows <Ignore> in all the columns. Again, right-click anywhere in the upper section and choose Map Items By Matching Names. This will map all the corresponding columns together.

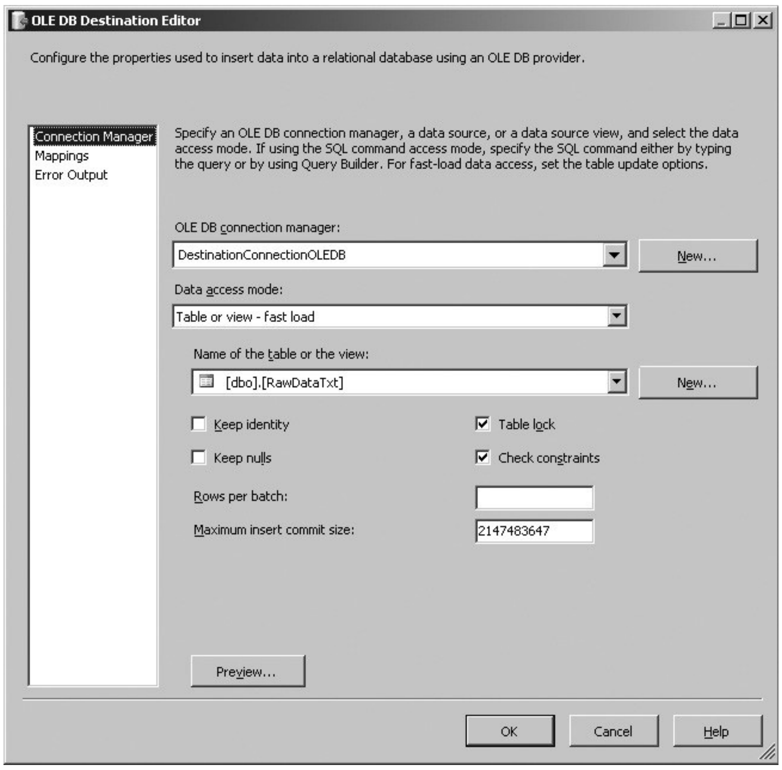
15. Open the Input And Output Properties tab, and you can see Flat File Source Output and Flat File Source Error Output under Inputs and Outputs. Expand the Flat File Source Output node to see External Columns and Output Columns. As mentioned earlier, External Columns are the reference columns of the source text file and Output Columns are the columns that Flat File Source Adapter passes on to the downstream component in the data flow path. Expand External Columns and click any column to see column properties such as CodePage, DataType, and Length in the right pane.

Now expand Output Columns and click any of the columns; you will see the Output Column properties such as CodePage, DataType, Length, FastParse, SortKeyPosition, and so on. Note that the Data Type of External Columns and Output Columns is [DT_STR] by default. The FastParse option can be set to either True or False. To load data between heterogeneous data sources, the source adapters parse the data of each column to convert it to SSIS data type, and when the data is to be loaded into a data store, the destination adapter parses the data and converts it to the type destination requires.

The two parsing techniques, Fast parse (when FastParse option is True) and Standard parse (when FastParse option is False), are available in the Flat File source and Flat File destination adapters and the Data Conversion and Derived Column transformations. This is because only these data flow components convert data from a string to a binary data type, or vice versa. The FastParse option allows use of simpler (commonly used date and time formats), quicker, but locale-insensitive, fast parsing routines. You can set FastParse to True on the columns that are not locale-sensitive to speed up the parsing process. By default, FastParse is set to False, indicating Standard parse is used, which supports all the data type conversions. For more information on parsing techniques, refer to Microsoft SQL Server 2008 Books Online.

Click Cancel to return to the SSIS Designer.

16. An OLE DB Destination loads input records into an OLE DB–compliant data store. To explore its custom user interface, double-click the Destination—RawDataTxt component. You will see the Connection Manager page shown in Figure 2-12.


OLE DB Destination Editor

Configure the properties used to insert data into a relational database using an OLE DB provider.

Specify an OLE DB connection manager, a data source, or a data source view, and select the data access mode. If using the SQL command access mode, specify the SQL command either by typing the query or by using Query Builder. For fast-load data access, set the table update options.

OLE DB connection manager: DestinationConnectionOLEDB

Data access mode: Table or view - fast load

Name of the table or the view: [dbo].[RawDataTxt]

☐ Keep identity ☑ Table lock

☐ Keep nulls ☑ Check constraints

Rows per batch:

Maximum insert commit size: 2147483647

Figure 2-12: Connection Manager page of the OLE DB Destination Editor

17. As the name suggests, an OLE DB Destination uses an OLE DB connection manager to connect to the destination table. In this package, DestinationConnectionOLEDB is used, which is specified in the OLE DB Connection Manager field.

18. Click in the Data Access Mode field, and you'll see the available five options for data access in the drop-down list:

    - **Table or view** When you select this option, the data is loaded into a table or view in the database specified by the OLE DB connection manager, and you specify the name of the table or the view in the Name of the table or the view field.

    - **Table or view – fast load** Using this data access mode, you can load data into a table or view as in the previous option, but using the fast load options such as acquiring table lock and specifying maximum insert commit size.

- **Table name or view name variable** Using this option, you still load data into a table or a view, but instead of specifying the table or view name directly, you specify a variable that contains the table or view name.
- **Table name or view name variable – fast load** This data access mode works like Table or view – fast load access mode, except in this access mode you supply the variable that contains the table or the view name.
- **SQL command** You can load the result set of an SQL statement using this option.

19. Go to the Mappings page and note that the Available Input Columns are mapped to Available Destination Columns. Note that the Suffix column in the input columns is not mapped to any destination column. You may have to scroll up and down to see the mappings properly. Click Cancel to close the editor.

Review

This Hands-On exercise presented a simple package that contains various types of components for you to see how Integration Services packages are organized. You've seen how Control Flow manages workflow in a package and makes the required objects available when they are required by Data Flow components. We haven't tried to execute this package because the package is not designed from a multiuse perspective. The Execute SQL task creates a table in a Campaign database for the first time the package is run, but what do you expect will happen if you again try to run the same package? The package will not succeed, as the table that Execute SQL task tries to create in the subsequent runs already exists, and the Execute SQL task attempt will fail, resulting in failure of the package. If you want to run the package more than once, you could either drop the table already created before trying to create it again or use TRUNCATE TABLE command with the existing table instead of creating a new table.

It will be worthwhile to review the package you have created during the last Hands-On when you exported data to an Excel file. I would encourage you to add it to this project and explore its various components to get a feel for them. Don't worry if they don't make much sense to you now, as each of the preconfigured components that SSIS provides will be covered in detail in the chapters to come.

Last, if you still want to play a little bit more with BIDS and SQL Server Import and Export Wizard, you can perform another Hands-On exercise using C:\SSIS\ RawFiles\RawDataAccess.mdb file to build an Integration Services package directly in BIDS. To give you a hint, start the SQL Server Import and Export Wizard from Project menu command and note that this time the wizard doesn't give you an option to save the package as it has done in the previous exercises.

Integration Services Connections Project Wizard

Just as you can use the SQL Server Import and Export Wizard to create a basic package in BIDS that you can enhance later, Integration Services provides you another wizard to quickly create a package with all the required connection managers. It allows you to choose data providers and configure them to create connection managers. Once configured, you can them select to use a connection manager as a source or a destination or both. Finally, it creates a project with configured connection managers and a data flow task containing defined sources and destinations. This can be very helpful as creating connection managers is usually the first task when you're creating a package. You can invoke this wizard from File | New | Project and then choosing the Integration Services Connections Project Wizard from the Visual Studio project templates.

Analyzing Data Quality with the Data Profiling Task

During various stages of a data warehouse project, you'll need to make sure that the values in certain columns stay within the defined perimeters. To verify and implement this requirement, you may be required to run ad hoc queries such as distinct values, lengths of various values in the column, or percentage of null values against the data warehouse. And if you find deviations in data, you'll need to fix that either (optimally) in the ETL or using ad hoc queries. You might choose to apply constraints in the data warehouse to prevent deviations from happening; however, constraints bring their own problems, such as failures of ETL, increased loading time, and complex delete operations. It will be easier if you can quality control these issues at the loading stage and hence the data warehouse always receives the correct data. This will enable the data warehouse to perform better avoiding unnecessary ad hoc queries and changes. To explain it further, consider if business reports are using a two-digit country code column extensively, you'll need to make sure that this column always has the correct country code and doesn't include any stray values. You may check the nulls in the column, the length of country code values, or distinct values in the column as part of your resolution. If you implement these checks and their relative corrections while loading the data warehouse, you will have solved most of your data specifications–related problems upfront. Integration Services now includes a Data Profiling Task to facilitate the process of finding anomalies in data.

The Data Profiling Task connects to an SQL Server database table or view and creates various aggregate statistics to help you discover the problems in data. The Data Profiling Task enables you to compute statistics either on a single column or on multiple columns or both. The column analysis brings out the true metadata of a column, as it is based on the data itself and helps you to understand column data in detail. The multiple column statistics give you an insight on how the values in one column depend upon the values in another. These configurations are called Profile Requests; five of them are available for individual column statistics, and three are available to analyze multiple columns or relationships between columns.

Single-Column Profiles

Single-column profiles enable you to analyze single column independently for Null values, column statistics, pattern profile, length distribution, and value distribution within the column.

- **Column Length Distribution Profile** You will perform this computation on a column containing text strings to identify any outliers. For example, if the column you are profiling contains fixed-length codes, any variation in length will indicate a problem in the data. This profile type computes all the distinct lengths of string values in the selected column and the percentage of rows in the table that each length represents.
- **Column Null Ratio Profile** You will perform this computation to find out missing data in a column with any data type. For example, an unexpectedly high ratio of null values in a column indicates the absence of data. This profile computes the percentage of null values in the selected column.
- **Column Pattern Profile** This profile request generates a set of regular expressions and the percentage of related string values. You will be using this profile to determine invalid strings in data. This profile can also suggest regular expressions that can be used in the future to validate new values.
- **Column Statistics Profile** This profile request works with numeric and datetime columns and can compute statistics for minimum and maximum values. Additionally, you can also generate statistics for average and standard deviation values for numeric columns. This profile can help you to identify values that lie outside the range you expect in a column or have a higher standard deviation than expected.
- **Column Value Distribution Profile** This profile will be of the most interest to you in case you want to know the distinct values and their percentage of rows in the column. This can help you understand your data a bit more, or if you already know the number of values, you can figure out the problems in data. This profile request works with most data types, such as numeric, string, and datetime formats.

Multiple-Column Profiles

Using multiple-column profile, you can profile a column based on the values existing in other columns such as candidate key profile, functional dependency profile, and the value inclusion profile.

- **Candidate Key Profile** This profile request can identify the uniqueness of a column or set of columns and hence can help you to determine whether the column or set of columns is appropriate to serve as a key for the selected table. You can also use this profile request to find duplicates in the potential key column.
- **Functional Dependency Profile** This profile request finds out the extent to which the values in one column are dependent on the values in another column or set of columns. Using this profile, you can validate the data in a column based on the other column.
- **Value Inclusion Profile** This profile request checks whether the values in a column also exist in another column. Using this profile, you can identify the dependency and can determine whether a column or set of columns is appropriate to serve as a foreign key between the selected tables.

You can choose one or more of these profile requests to create data profiles. Based on the profile requests, the Data Profiling Task first runs metadata queries against INFORMATION_SCHEMA.COLUMNS to find out the column names and their attributes, such as data type, character length, numeric precision and scale, null-ability, and collation name. Then it runs several queries to compute values such as SUM, COUNT, DISTINCT, and LEN. While computing all this, it keeps the calculations and the information in the temporary tables in the TEMPDB database and drops them later once it's done with all the computations.

As you can imagine from this, you need read/write and create table permissions on the TEMPDB database to be able to run the Data Profiling Task as it performs various activities. In the end, all that information is written in an XML format in a variable or an output file. You can review the data statistics using the Data Profile Viewer, a stand-alone utility provided with SQL Server 2008 for viewing and analyzing data profiles. Though you can review and analyze the profiles manually by inspecting the output file and decide whether to import the data from the profiled table, you can actually automate this decision making in the workflow of your package by checking the data statistics in the xml variable.

Hands-On: Using Data Profiling Task

In this Hands-On you will use Data Profiling Task to profile the data imported into RawDataTxt table and will use Data Profile Viewer utility to review the statistics generated.

1. Start BIDS and open My First SSIS Project. Double-click Package.dtsx to open this blank package if it is not open already.

2. From the Toolbox, drag and drop the Data Profiling Task on to the Control Flow surface.

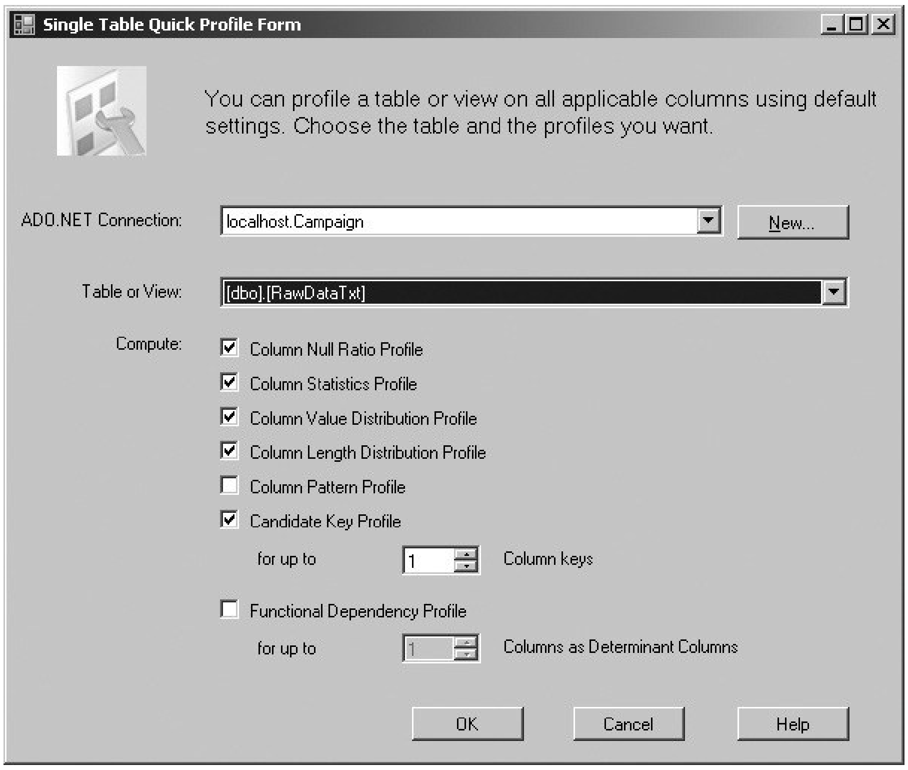
3. Double-click the icon to open the Data Profiling Task Editor dialog box.

4. In the General Page, click in the DestinationType field and then expand the list by clicking the drop-down arrow. Note that you have two options to choose from, File or a variable, where you would like the output of this task to be written. Leave the FileConnection selected. Click in the Destination field and select <New File Connection. . .> from the drop-down list to open File Connection Manager Editor.

5. Choose Create File in the Usage type field and type **C:\SSIS\RawFiles\ DataProfileFile.xml** in the File field.

6. Click Quick Profile to open the Single Table Quick Profile Form. Click New shown opposite to ADO.NET Connection to open the Connection Manager dialog box. Note that it limits you to using the SqlClient Data Provider, indicating that the Data Profile task can profile only SQL Server 2000 and above databases. Type your server name or type **localhost** in the Server Name field. Select the Campaign database in the Select box or enter a database name field. Click Test Connection to test the configuration. Click OK twice to come back to Single Table Quick Profile Form.

7. Select [dbo].[RawDataTxt] in the Table Or View field as shown in Figure 2-13. Click OK to create profile requests.


Single Table Quick Profile Form

You can profile a table or view on all applicable columns using default settings. Choose the table and the profiles you want.

ADO.NET Connection: localhost.Campaign

Table or View: [dbo].[RawDataTxt]

Compute:
- [✓] Column Null Ratio Profile
- [✓] Column Statistics Profile
- [✓] Column Value Distribution Profile
- [✓] Column Length Distribution Profile
- [ ] Column Pattern Profile
- [✓] Candidate Key Profile
  - for up to 1 Column keys
- [ ] Functional Dependency Profile
  - for up to 1 Columns as Determinant Columns

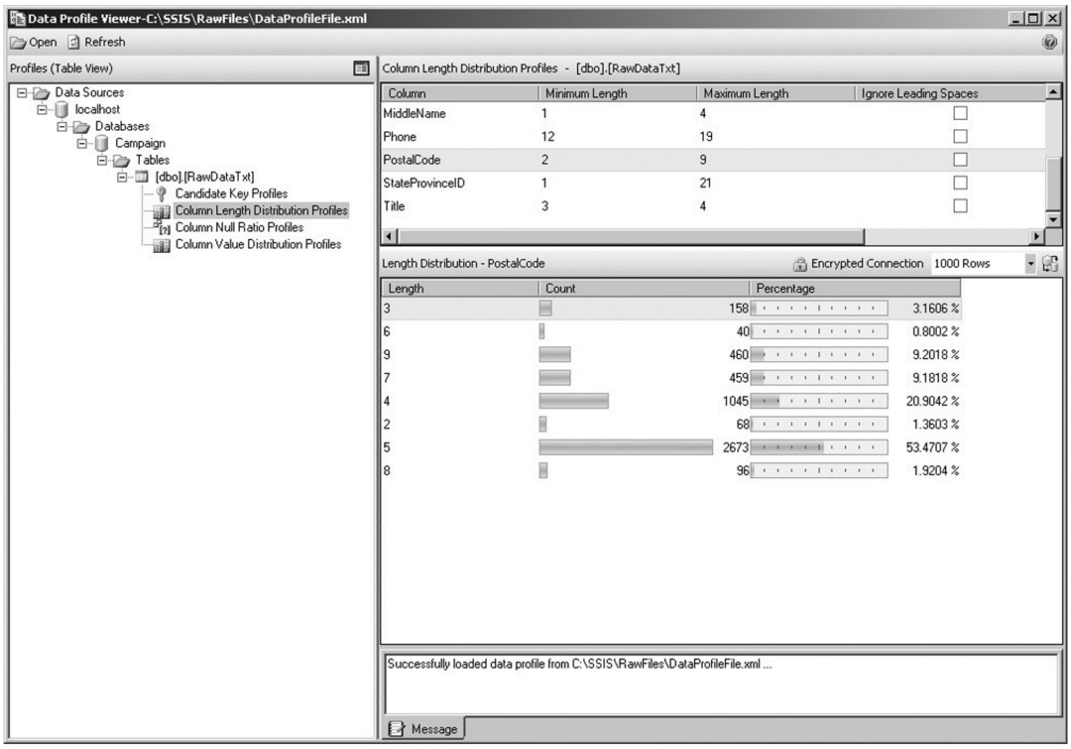
Figure 2-13: Using the Quick Profile Form

8. Go to the Profile Requests page and review the various profile requests and their options. Click OK to complete the Data Profiling Task configurations.

9. From the Debug menu, select Start Debugging or press F5 to run the package, or else press the respective button on the toolbar. Once the package completes execution, stop the package by pressing SHIFT-F5.

10. Navigate to C:\SSIS\RawFiles and verify that the DataProfileFile.xml file has been created. Click Start | All Programs | Microsoft SQL Server 2008 | Integration Services | Data Profile Viewer to start the viewer utility.

11. In the Data Profile Viewer dialog box, click Open and navigate to C:\SSIS\RawFiles and open DataProfileFile.xml. Review the different profiles to understand the way Data Profiling Task creates the profiles. For example, if you review the Candidate Key Profiles, you will see that EmailAddress column has been selected as the Key column with a Key Strength of 100%. Similarly, Figure 2-14 shows the Column Length Distribution Profiles for the PostalCode column.



Column Length Distribution Profiles - [dbo].[RawDataTxt]

Column	Minimum Length	Maximum Length	Ignore Leading Spaces
MiddleName	1	4	☐
Phone	12	19	☐
PostalCode	2	9	☐
StateProvinceID	1	21	☐
Title	3	4	☐

Length Distribution - PostalCode

Length	Count	Percentage
3	158	3.1606 %
6	40	0.8002 %
9	460	9.2018 %
7	459	9.1818 %
4	1045	20.9042 %
2	68	1.3603 %
5	2673	53.4707 %
8	96	1.9204 %

Successfully loaded data profile from C:\SSIS\RawFiles\DataProfileFile.xml ...

Figure 2-14: Column Length Distribution Profiles

Review

In the preceding Hands-On you've worked with the Data Profiling Task and used the Quick Profiles option to create profile requests quite easily. However, you can configure the requests manually the way you want by clicking in the new row in Profile Requests page. Though the review of the profiles was manual, yet it provided you greater understanding of the profile structure created by the Data Profiling Task. If you really want to monitor data quality going forward, you will need to build business rules around data quality to create a scorecard matrix. To begin with, you can create a baseline of scorecards. When the data profiling activity is repeated over time, you can compare the scorecards generated each time against the baseline and deduce whether the data quality has improved or deteriorated over time.

Summary

You created an Integration Services blank project in Chapter 1. In this chapter, you created packages using the SQL Server Import and Export Wizard and then added those packages into your blank project. You also created a package directly in the BIDS again using the SQL Server Import and Export Wizard. But above all, you explored those packages by opening component properties and configurations, and now hopefully you better understand the constitution of an Integration Services package. Last, you worked with the Data Profiling Task to identify quality issues with your data. In the next chapter, you will learn about the basic components, the nuts and bolts of Integration Services packages, before jumping in to make complex packages in Chapter 4 using various preconfigured components provided in BIDS.