



MICROSOFT®
SQL SERVER™ 2005
INTEGRATION
SERVICES

Step by Step

*Paul Turley, Joe Kasprzak, Scott Cameron, Satoshi Iizuka,
and Pablo Guzman of Hitachi Consulting*

PUBLISHED BY

Microsoft Press
A Division of Microsoft Corporation
One Microsoft Way
Redmond, Washington 98052-6399

Copyright © 2007 by Hitachi Consulting

All rights reserved. No part of the contents of this book may be reproduced or transmitted in any form or by any means without the written permission of the publisher.

Library of Congress Control Number: 2007928769

Printed and bound in the United States of America.

1 2 3 4 5 6 7 8 9 QWT 2 1 0 9 8 7

Distributed in Canada by H.B. Fenn and Company Ltd.

A CIP catalogue record for this book is available from the British Library.

Microsoft Press books are available through booksellers and distributors worldwide. For further information about international editions, contact your local Microsoft Corporation office or contact Microsoft Press International directly at fax (425) 936-7329. Visit our Web site at www.microsoft.com/mspress. Send comments to mspinput@microsoft.com.

Microsoft, Microsoft Press, Active Directory, ActiveX, Excel, IntelliSense, MSDN, SQL Server, Visual Basic, Visual C#, Visual Studio, Windows, and Windows Server are either registered trademarks or trademarks of Microsoft Corporation in the United States and/or other countries. Other product and company names mentioned herein may be the trademarks of their respective owners.

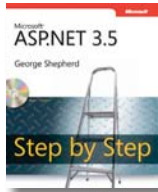
The example companies, organizations, products, domain names, e-mail addresses, logos, people, places, and events depicted herein are fictitious. No association with any real company, organization, product, domain name, e-mail address, logo, person, place, or event is intended or should be inferred.

This book expresses the author's views and opinions. The information contained in this book is provided without any express, statutory, or implied warranties. Neither the authors, Microsoft Corporation, nor its resellers, or distributors will be held liable for any damages caused or alleged to be caused either directly or indirectly by this book.

Acquisitions Editor: Ken Jones
Developmental Editor: Devon Musgrave
Project Editor: Denise Bankaitis
Technical Reviewer: Bob Hogan

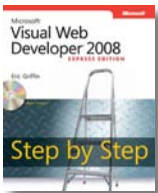
Body Part No. X13-76794

For Web Developers



**Microsoft® ASP.NET 3.5
Step by Step**
George Shepherd
ISBN 9780735624269

Teach yourself ASP.NET 3.5—one step at a time. Ideal for developers with fundamental programming skills but new to ASP.NET, this practical tutorial delivers hands-on guidance for developing Web applications in the Microsoft Visual Studio® 2008 environment.



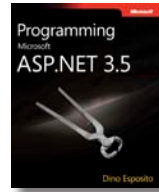
**Microsoft Visual Web
Developer 2008
Express Edition
Step by Step**
Eric Griffin
ISBN 9780735626065

Your hands guide to learning fundamental Web-development skills. This tutorial steps you through an end-to-end example, helping build essential skills logically and sequentially. By the end of the book, you'll have a working Web site, plus the fundamental skills needed for the next level—ASP.NET.



**Introducing Microsoft
Silverlight™ 2,
Second Edition**
Laurence Moroney
ISBN 9780735625280

Get a head start with Silverlight 2—the cross-platform, cross-browser plug-in for rich interactive applications and the next-generation user experience. Featuring advance insights from inside the Silverlight team, this book delivers the practical, approachable guidance and code to inspire your next solutions.



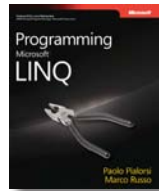
**Programming Microsoft
ASP.NET 3.5**
Dino Esposito
ISBN 9780735625273

The definitive guide to ASP.NET 3.5. Led by well-known ASP.NET expert Dino Esposito, you'll delve into the core topics for creating innovative Web applications, including Dynamic Data; LINQ; state, application, and session management; Web forms and requests; security strategies; AJAX; Silverlight; and more.



**JavaScript
Step by Step**
Steve Suehring
ISBN 9780735624498

Build on your fundamental programming skills, and get hands-on guidance for creating Web applications with JavaScript. Learn to work with the six JavaScript data types, the Document Object Model, Web forms, CSS styles, AJAX, and other essentials—one step at a time.



**Programming Microsoft
LINQ**
Paolo Pialorsi and Marco Russo
ISBN 9780735624009

With LINQ, you can query data—no matter what the source—directly from Microsoft Visual Basic® or C#. Guided by two data-access experts who've worked with LINQ in depth, you'll learn how Microsoft .NET Framework 3.5 implements LINQ, and how to exploit it. Study and adapt the book's examples for faster, leaner code.

ALSO SEE

**Developing Service-Oriented AJAX
Applications on the Microsoft Platform**
ISBN 9780735625914

Microsoft ASP.NET 2.0 Step by Step
ISBN 9780735622012

Programming Microsoft ASP.NET 2.0
ISBN 9780735625273

**Programming Microsoft ASP.NET 2.0
Applications: Advanced Topics**
ISBN 9780735621770

**Microsoft®
Press**

Contents at a Glance

Part I	Getting Started with Integration Services	
1	Introduction to SQL Server Integration Services	3
2	Building Your First Package	13
Part II	Designing Packages	
3	Extracting and Loading Data	35
4	Using Data Flow Transformations	67
5	Managing Control Flow	113
6	Scripting Tasks	139
7	Debugging Packages	175
8	Managing Package Execution	201
Part III	Managing Packages	
9	Detecting and Handling Processing Errors	237
10	Securing and Deploying SSIS Packages	287
11	Optimizing SSIS Packages	309
Part IV	Applying SSIS to Data Warehousing	
12	Data Warehouse Concepts	341
13	Populating Data Warehouse Structures	367
14	SSIS General Principles	413

Table of Contents

<i>Introduction</i>	xv
<i>Finding Your Best Starting Point</i>	xv
<i>About the Companion CD-ROM</i>	xvii
<i>System Requirements</i>	xvii
<i>Installing and Using the Sample Files</i>	xvii
<i>Conventions and Features in This Book</i>	xviii

Part I **Getting Started with Integration Services**

1 Introduction to SQL Server Integration Services	3
Common SSIS Applications	4
SSIS Objects and Process Control Components	4
SSIS Process Control	5
SSIS Control Flow	6
SSIS Data Flow	6
SSIS Data Pipeline	6
SSIS Event Handler	7
SSIS Components	8
SSIS Development Studio	8
SSIS Runtime Services	9
SSIS Package Deployment	9
SQL Server 2000 DTS Migration	10
Chapter 1 Quick Reference	10
2 Building Your First Package	13
Exploring Business Intelligence Development Studio	13
Solution Explorer	15
Docking Utility Windows	16

 **What do you think of this book? We want to hear from you!**

Microsoft is interested in hearing your feedback so we can continually improve our books and learning resources for you. To participate in a brief online survey, please visit:

www.microsoft.com/learning/booksurvey/

- Exploring an SSIS Project in BIDS 17
- Using the SSIS Import and Export Wizard 26
 - Creating Tables in a New Database 26
- Reviewing Package Elements 29
 - Reviewing a Package Created Using the Import and Export Wizard 30
- Testing a Package 31
 - Executing the Package in the Designer 31
- Chapter 2 Quick Reference 32

Part II Designing Packages

3 Extracting and Loading Data 35

- Connection Managers 35
 - Connection Manager Types 35
 - Creating a New Integration Services Project 36
 - Adding Connection Managers 39
 - Creating a Data Flow 41
 - Adding Data Adapters 43
 - Executing the Package 49
- Using Data Sources and Data Source Views 50
 - Creating a Data Source 50
 - Creating a Data Source View 52
 - Creating a New Named Query 54
 - Copying Data from a Named Query to a Flat File 57
 - Executing the Package 63
- Chapter 3 Quick Reference 65

4 Using Data Flow Transformations 67

- Creating Data Flow in a Package 67
 - Data Flow Sources 67
 - Data Flow Transformations 68
 - Data Flow Destinations 68
 - Data Source Connections 68
- SSIS Transformations 69
 - Row Transformations 70
 - Rowset Transformations 70
 - Split and Join Transformations 71

Data Quality Transformations	71
Data-Mining Transformations	72
Other Transformations	72
Synchronous and Asynchronous Transformations	73
Using Expressions in Packages	73
Expression Usage in SSIS	73
Expressions Elements	74
Building Expressions	75
Using Data Flow Transformations	75
Opening and Exploring the SSIS Project	75
Creating the Data Flow Task	78
Using a Flat File Source	80
Adding a Connection Manager	82
Adding a Conditional Split Transformation	85
Adding a Derived Column Transformation	88
Viewing the Properties of the Derived Column Transformation	91
Adding a Flat File Destination Data Adapter and Executing the Package	92
Sending Output to Different Destinations	95
Configuring Error Output	102
Types of Errors	102
Error Options	102
Exploring the LookupGeography Package	102
Creating a Task	104
Creating and Naming a Flat File Source	104
Adding a Data Conversion Transformation	105
Adding a Lookup Transformation	106
Adding a Flat File Destination for Lookup Errors	108
Adding a Flat File Destination for Successful Lookups	109
Executing the Package and Checking the Results	110
Chapter 4 Quick Reference	112
5 Managing Control Flow	113
Control Flow Elements	113
Control Flow Components	114
Using Containers	119
Adding a Fuzzy Lookup Transformation	125
Adding a Foreach Loop Container	130

Applying Precedence Constraints	132
Chapter 5 Quick Reference	137
6 Scripting Tasks	139
Understanding Scripting Tasks	139
Implementing a Script Task	141
Creating a New Script Task and Initiating Code	141
Handling Errors	145
Providing a Message to the Progress Tab	148
Providing Verbose Information to the Log File	152
Using Variables	156
Modifying a Variable at Run Time	159
Understanding the Script Component	161
Implementing the Script Component	162
Reviewing a Sample Project	162
Understanding an ActiveX Script Task	169
Implementing an ActiveX Script Task	169
Chapter 6 Quick Reference	173
7 Debugging Packages	175
Debugging Control Flow	175
Understanding Breakpoints	176
Reviewing Debug Windows	181
Understanding Progress Messages	183
Executing a Package Partially	185
Debugging Data Flow	186
Browsing Data By Using Data Viewers	186
Understanding Other Options	190
Debugging Script Task	190
Walk Through Code by Using Breakpoints	190
Reviewing State by Using VSA Features	194
Chapter 7 Quick Reference	198
8 Managing Package Execution	201
Understanding Package Configurations	201
Configuration Benefits	201
Configuration Types	202
Understanding the XML Configuration File	202

Specifying a New XML Configuration File Location	202
Creating and Editing an XML Configuration File	203
Opening the SSIS Project and Executing the Package	203
Creating an XML Configuration File	204
Editing the XML Configuration File	207
Testing the Package with the New Configuration	208
Multiple Configuration Files	208
Environment Variable	208
Registry Entry	209
Parent Package Variables	209
SQL Server Tables	209
Direct and Indirect Configurations	209
Using Configuration Files	209
Determining Configuration Order	210
Evaluating Configuration Failure	210
Using Multiple Configurations	210
Creating Multiple Configuration Files	210
Creating the Database and the OLE DB Connection Manager	210
Creating the Environment Variable	212
Creating the Environment Variable Configuration	213
Creating the SQL Server Configuration	215
Testing the Package with the New Configuration	217
Exploring the Parent Package	218
Creating the Parent Package Variable Configuration	219
Exploring Package Execution Options	221
Using the SQL Server Import and Export Wizard to Execute Packages	221
Using DTEXecUI to Execute Packages	222
Using DTEXec to Execute Packages	222
Using SQL Server Management Studio to Execute a Package	223
Extending Package Execution Options	224
Using SQL Server Agent	224
Using the Execute Package Utility	226
Understanding Package Logging	229
Implementing Package Logging	230
Configuring Package Logging	230
Executing the Package and Viewing the Logs	232
Chapter 8 Quick Reference	234

Part III **Managing Packages**

9 Detecting and Handling Processing Errors 237

- Basic Error Detection and Handling 237
 - Understanding Metadata Lineage 238
 - Understanding Validation 238
 - Understanding Precedence Constraints 238
 - Understanding Data Flow Transformations 239
- Understanding Event Handlers 239
 - Using Event Handlers to Perform Tasks 240
 - Triggering an Event Handler 240
 - Using the Event Handlers Provided by SSIS 241
- Creating Event Handlers 241
 - Accessing the SSIS Design Environment 242
 - Creating an *OnPreExecute* Event Handler 243
 - Adding a Task to an Event Handler 244
 - Configuring the Task 245
 - Mapping SSIS Variables to SQL Statement Parameters 247
 - Creating a Log Finish Event Handler 248
 - Creating a Log Error Event Handler 250
 - Executing the Package 251
 - Testing the Package with Invalid Data 253
 - Creating an Event Handler to Fix the Problem 255
 - Creating a Task to Move the File with Invalid Data 256
 - Setting Connection Manager Settings 257
 - Preventing Events from Escalating to Containers and Packages 259
 - Changing Error Count Properties 260
 - Executing the Package 261
- Maintaining Data Consistency with Transactions 263
 - Configuring Transactions 263
- Using Checkpoint Restarts 263
 - Understanding the Benefits of Checkpoints 263
 - Configuring Packages for Checkpoints 264
- Using Checkpoints and Transactions 264
 - Preparing to Use Checkpoints and Transactions to Fix the Error 264
 - Becoming Familiar with the LoadDimProd Package 267

Fixing the Error	278
Implementing Checkpoints	281
Chapter 9 Quick Reference	285
10 Securing and Deploying SSIS Packages	287
Creating a Deployment Utility.	288
Using the Package Installation Wizard.	288
Securing a Package	290
Package Encryption.	290
Password Protection	291
<i>ProtectionLevel</i> Property.	291
Role-Based Security	291
Applying Security	292
Deployment Options	294
Push Deployment	294
Pull Deployment	295
Managing Packages on the SSIS Server.	295
Creating and Applying a Configuration	297
Adding a Configuration to the Project	297
Executing a Deployed Package.	298
Monitoring Package Execution and Event Logs.	300
Applying a Configuration	301
Chapter 10 Quick Reference	307
11 Optimizing SSIS Packages	309
SSIS Engine Overview	310
Runtime Engine	310
Data Pipeline Engine	310
Memory Buffer Architecture	310
Buffer Usage.	311
Execution Trees	312
Synchronous and Asynchronous Processing.	312
Data Blocking.	313
Blocking Transformations.	313
Partially Blocking Transformations	313
Row Transformations.	314
Non-blocking Transformations	314
Sources	315

Buffer Settings	315
Managing Parallelism	316
Data Source Tuning	316
Performance Management	317
Loops	318
Flat File Sources	319
Filters and Variables	320
Data Destination Management	321
Performance-Tuning Exercises	322
Working with Buffer Properties	323
Working with a SQL Server Destination	324
Design Considerations	326
Performance Management	329
Execution Trees	330
Execution Plans	331
Iterative Design Optimization	333
Logging an Execution Plan	334
SSIS Log Reports	337
Chapter 11 Quick Reference	337

Part IV **Applying SSIS to Data Warehousing**

12 Data Warehouse Concepts	341
Data Warehouse Objectives	341
Data Warehouse Characteristics	342
Providing Data for Business Analysis Processes	343
Integrating Data from Heterogeneous Source Systems	344
Combining Validated Source Data	344
Organizing Data into Nonvolatile, Subject-Specific Groups	345
Storing Data in Structures Optimized for Extraction and Queries	345
Data Warehouse Fundamentals	346
Business Intelligence Solution Goals	346
Combining Relevant Data from Multiple Sources	347
Providing Fast and Easy Access	347
Focus on Decisions	348
Data Granularity	349

Supporting Business Decisions	350
Update Frequency and Persistence	350
Historical Data	351
Changing Dimensions.	352
Surrogate Keys.	352
Additive Measures.	353
Reviewing an Operational and Database Schema	354
Creating a Database Diagram	355
Data Warehouse System Components.	359
Fact and Dimension Tables.	359
Dimension Table Characteristics	360
Reviewing and Comparing a Data Warehouse Database Schema.	362
Creating a Database Diagram	363
Data Warehouse in Summary	365
Chapter 12 Quick Reference	366
13 Populating Data Warehouse Structures.	367
Data Warehouse Characteristics	368
Implementing Staging Tables	369
Types of Staging Schemes	370
Staging Data from Multiple Sources.	370
Staggered Staging	371
Persisted Staging	371
Accumulated Staging	372
Chunked Accumulated Staging	373
Other Destination Considerations.	373
Managing Dimension Tables Part 1	374
Loading Dimension Tables by Using a Left Outer Join	375
Managing Dimension Tables Part 2	380
Loading Dimension Tables Part 2	380
Slowly Changing Dimensions	384
Managing Slowly Changing Dimensions.	385
Managing Fact Tables.	397
Aggregating Data in Fact Tables	398
Loading Fact Tables.	398
Chapter 13 Quick Reference	411

14	SSIS General Principles	413
	Designing SSIS Packages	414
	OVAL Principles of SSIS Package Design	414
	Using SSIS Components in Your Design	417
	Create a Master–Child Package	424
	Organizing Package Components	430
	Managing SSIS Application Deployment	434
	Chapter 14 Quick Reference	437
	Index	439



What do you think of this book? We want to hear from you!

Microsoft is interested in hearing your feedback so we can continually improve our books and learning resources for you. To participate in a brief online survey, please visit:

www.microsoft.com/learning/booksurvey/

Introduction

A complete database solution requires data to be integrated from a variety of sources. One of the greatest challenges facing business today is that important business information exists in multiple locations and in different formats. As an industry, we have empowered business leaders and information workers with access to corporate data and powerful analysis tools. With access to so much information, decision makers need one indisputable version of data. A reliable ETL (extract, transform, and load) process is the backbone of business-critical data consolidation and business intelligence (BI) services used to lead and support business direction. Microsoft SQL Server 2005 Integration Services (SSIS) provides a foundation to design and perform effective ETL processes.

The goal of this book is to help you design and implement ETL solutions as quickly as possible, both in concept and in practice. You will learn and understand the core principles and concepts of effective data transformation. Through simple hands-on exercises, you will quickly learn to design Integration Services packages used to transform data between files and relational databases; handle conditional logic; and to alter, split, match, merge, combine, and join data in a data flow. After completing these exercises, you will know how to use the appropriate tasks, transformations, connection managers, and data source and destination adapters in concert to form SSIS packages. You will learn to deploy, configure, and optimize packages to run on production servers.

This book is written to address the requirements of professionals with different needs. Database administrators and application developers need to transform data to support specific applications. BI system architects require data to be consolidated from multiple source systems to a central data warehouse or data mart. A scheduled ETL package must be flexible enough to handle errors and data anomalies. Whether you need to run a package to perform a quick, one-time data import or you need a scheduled process to populate the corporate data warehouse every night, you will learn to design an Integration Services solution to meet that need.

Finding Your Best Starting Point

Although the range of topics addressed in this book is comprehensive, this book also caters to readers with varying skills who are involved in one or more stages of the data transformation life cycle. Accordingly, you can choose to read only the chapters that apply to the activities for which you are responsible and skip the remaining chapters. If you choose to take this approach, we recommend that you at least review the chapters that apply to other roles in order to obtain a broad exposure to the product. To find the best place to start, use the following table.

If you are	Follow these steps
An information worker who needs to import or export data	<ol style="list-style-type: none"> 1. Install the sample files as described in "Installing and Using the Sample Files." 2. Read Chapter 1, "Introduction to Integration Services," to learn the concepts of data transformation. 3. Work through Chapters 2, "Building Your First Package," 3, "Extracting and Loading Data," 4, "Using Data Flow Transformations," and 5, "Managing Control Flow," to learn the core components of package design. 4. Review Chapter 14, "Applying Best Practices to SSIS Package Design," to understand package design best practices.
An information worker or application developer who needs to build an ETL solution	<ol style="list-style-type: none"> 1. Install the sample files as described in "Installing and Using the Sample Files." 2. Read Chapter 1 to learn the concepts of data transformation. 3. Work through or review Chapters 2, 3, 4, and 5 to learn the core components of package design. 4. Work through Chapters 6, "Scripting Tasks," 7, "Debugging Packages," 8, "Managing Package Execution," and 9, "Detecting and Handling Processing Errors," to learn how to design advanced packages with error handling. 5. Work through or review Chapter 11, "Optimizing SSIS Packages," to learn how to optimize package design and execution. 6. Work through Chapter 14 to understand package design best practices.
A BI architect or solution designer	<ol style="list-style-type: none"> 1. Install the sample files as described in "Installing and Using the Sample Files." 2. Read Chapter 1 to learn the concepts of data transformation. 3. Work through or review Chapters 2, 3, 4, and 5 to learn the core components of package design. 4. Work through Chapters 6, 7, 8, and 9 to learn how to design advanced packages with error handling. 5. Work through or review Chapter 11 to learn how to optimize package design and execution. 6. Work through Chapters 12, "Data Warehouse Concepts," and 13, "Populating Data Warehouse Structures," to learn data warehouse concepts and how to populate data warehouse structures. 7. Work through Chapter 14 to understand package design best practices.
A system or database administrator who needs to configure and optimize a solution	<ol style="list-style-type: none"> 1. Install the sample files as described in "Installing and Using the Sample Files." 2. Work through Chapters 10, "Securing and Deploying SSIS Packages," and 11 to learn how to secure, deploy, and optimize packages. 3. Review other chapters as needed to understand design elements and best practices.

About the Companion CD-ROM

The CD that accompanies this book contains the sample files that you need to follow the step-by-step exercises throughout the book. For each chapter, use the Microsoft Visual Studio solution files that have projects or packages created for you as starting points in preparation for adding other features to the projects or packages. These sample files allow you to build on what you've learned rather than spend time setting up the prerequisites for an exercise. The exercises for each chapter are separate and may be used independently.

System Requirements

To install Integration Services and to use the samples provided on the companion CD, your computer configuration will need to meet the following requirements:

- Microsoft Windows 2000, Windows XP Professional, or Windows Server 2003 with the latest service pack applied.
- Microsoft SQL Server 2005, Developer or Enterprise Edition, with any available service packs applied and using Windows or Mixed Mode authentication. Refer to “Hardware and Software Requirements for Installing SQL Server 2005,” listed at <http://msdn2.microsoft.com/en-us/library/ms143506.aspx> to determine which edition is compatible with your operating system.

The step-by-step exercises in this book and the accompanying practice files were tested using Windows XP Professional, Service Pack 2, and Microsoft SQL Server 2005 Developer and Enterprise Editions with Service Pack 1. If you are using another version of the operating system or a different edition of either application, you might notice some slight differences.

Installing and Using the Sample Files

The sample solution and database files require approximately 300 MB of disk space on your computer. To install and prepare the sample files for use with the exercises in this book, follow these steps:

1. Insert the companion CD into your CD-ROM drive.

Note If the presence of the CD-ROM is automatically detected and a Start window is displayed, you can skip to step 3.

2. Click the Start button, click Run, and then type **D:\started** in the Open box, replacing the drive letter with the correct letter for your CD-ROM drive if necessary.
3. Click Install Sample Files to launch the Setup program, and then follow the directions on the screen.

The sample files will be copied from the CD-ROM to your local hard drive. The default installation folder is C:\Documents and Settings\\My Documents\Microsoft Press\is2005sbs, where <username> is the logon name you use to operate your computer. You can change this installation folder to a different location and reference the new location when working through the exercises. For each chapter that uses sample files, you will find a corresponding folder in the is2005sbs folder. You'll be instructed where to find the appropriate sample files when an exercise requires the use of an existing file.

Tip In the My Documents\Microsoft Press\is2005sbs\Answers folder, you will find a separate folder for each chapter in which you make changes to the sample files. The files in these folders are sample projects in their completed state. You can refer to these files if you want to preview the results of an exercise after the steps have been completed. Because the project files are modified as you work through the chapter exercises, if you ever wish to begin an exercise over again, you will need to restore a backup of the project folder or manually copy and replace these files from the CD.

4. Remove the CD-ROM from the drive when installation is complete.
5. Use Windows Explorer to open My Documents\Microsoft Press\is2005sbs\Setup\Query and double-click to launch the attach_databases.bat file. This will attach three SQL Server 2005 databases used throughout the book.

This step attaches the SQL Server databases that are the data sources used in packages you will create and use throughout this book.

Note The attach_databases.bat script will work only if these databases are not previously attached, SQL Server 2005 is running as a local default instance, and the user account you're logged in with has administrative rights on your database server. The student files must also be installed to the default My Documents path in order for this script to run correctly. If any of these conditions don't apply to your environment, you should use SQL Server Management Studio to manually attach all three databases located in the \Setup\Database folder.

You're now ready to get started!

Conventions and Features in This Book

To use your time effectively, be sure that you understand the stylistic conventions that are used throughout this book. The following list explains these conventions:

- Hands-on exercises for you to follow are presented as lists of numbered steps (1, 2, and so on).
- Text that you are to type appears in bold type.

- Properties that you need to set in Visual Studio are sometimes displayed in a table as you work through steps.
- Pressing two keys at the same time is indicated by a plus sign between the two key names, such as Alt + Tab when you need to hold down the Alt key while pressing the Tab key.
- A note that is labeled NOTE gives you more information about a specific topic.
- A note that is labeled IMPORTANT points out information that can help you avoid a problem.
- A note that is labeled TIP conveys advice that you might find useful when using Integration Services.

Introduction to SQL Server Integration Services

After completing this chapter, you will be able to:

- Understand the purpose of SSIS with data integration applications.
- Understand SSIS objects used to create SSIS applications.
- Understand SSIS performance processing architecture.
- Understand SSIS development, administration, and run-time components.

Microsoft SQL Server 2005 Integration Services (SSIS) is the toolset used to help you implement data integration process applications among your business application system's files and databases. SSIS is much more than a simple extract, transform, and load (ETL) process. SSIS enables database administrators and application developers to design, implement, and manage complex, high-performance ETL applications. Using SSIS, you can select data from one or more sources and standardize, join, merge, cleanse, augment, derive, calculate, and perform just about any other function and operation required for your data integration applications. SSIS also provides procedures to automate many of the administrative functions for SQL Server databases, tables, On-Line Analytical Processing (OLAP) Cubes, and many other functions for components of SQL Server 2005.

The ETL phase of data warehousing, data migration, application integration, and business intelligence projects are commonly from 60 percent to as much as 80 percent of the work effort. Effective deployment of technology such as SQL Server 2005 Integration Services can significantly reduce the time, effort, and cost for this phase. This book is designed to show you how to use the features of SSIS and how best to implement these SSIS features and capabilities with data integration projects for your own application systems environments. Through a series of step-by-step demonstrations and exercises, you will work with common, practical, real-world examples to build SSIS applications. These exercises will show how to work with relational and non-relational data sources, manage referential integrity, handle slowly changing dimensions and other data warehousing and business intelligence challenges, and implement complex transformations. You will also learn how to use the debugging and error-handling features in SSIS to detect, troubleshoot, and recover from errors that might occur during data integration process execution. This book will also show you how to manage SSIS applications as well as provide you with best practices and disciplines for building and maintaining SSIS applications within your business application systems environments.

Common SSIS Applications

One common use for SSIS is to move data from one data source to another. The reasons for moving data are too numerous to count. Some common business reasons for using SSIS include migrating business data from one application to another, extracting data for distribution to external entities, integrating data from external entities, creating sample test data sources for development environments, and extracting and loading data into business intelligence (BI) application systems.

SSIS works extremely well in SQL Server environments, but it can also be used with many non-SQL Server database file types and many of the other database management systems deployed within your Information Technology (IT) environment. SSIS has the ability to read data from other Microsoft products, such as Microsoft Office Excel spreadsheets, as well as text, Extensible Markup Language (XML), and other flat-file types.

One common IT demand in the past few decades has been the need to provide business information to a wider audience within an organization. *Business intelligence* is a relatively new term, but it is certainly not a new concept. The idea is simply to use information already available in your company to help decision makers across the company make decisions better and faster. BI systems can be custom developed or deployed through a variety of packaged reporting and analytic tools. The common component among the various BI systems is the underlying data that drives the information and analysis.

When you need to provide fast-response BI applications for many purposes throughout a large organization, the data that drives such systems most often comes from multiple sources. SSIS provides you with the ability to design and execute data integration operations as simple as moving data between application databases or as complex as consolidating large volumes of data from multiple data sources in different formats, while at the same time applying rules to standardize, modify, and cleanse data content prior to loading into BI data warehouses designed for reporting and analytical applications. You will learn more about data warehouse application characteristics and the role of SSIS within BI and data warehouse applications in Chapter 12, “Data Warehouse Concepts,” and Chapter 13, “Populating Data Warehouse Structures,” later in this book.

Even if you’re not responsible for creating and maintaining a data warehouse, a reporting operational data store, OLAP cubes, or other BI applications, you’ll find the features in SSIS quite useful for routine database administrative tasks and many other activities in which you need to move, transform, and load data in any form.

SSIS Objects and Process Control Components

Before you begin learning how to create SSIS applications, it is important to familiarize yourself first with the SSIS process control components and the objects used to create SSIS applications. The first object to note within SSIS is the *package*.

An SSIS package is the highest-level object within an SSIS application. A package is a discrete unit of work that you define for ETL operations or SQL Server Services administration operations or both. It is a collection of SSIS process control components and their objects that define the operations, process dependencies, and sequence flow of activities and operations required for a data integration application. Package objects include *containers*, *tasks*, *precedence constraints*, *variables*, *data sources*, *data destinations*, *SQL Server administration functions*, and *custom tasks* that you can create to address unique requirements for your applications. Package objects are applied to package process control components that include the *control flow*, *data flow*, and *event handler*.

To control the sequence of activities and operations within a package, you apply the precedence constraint object. Precedence constraints are defined between your package objects and are used to specify the order sequence of operations processing and to control processing branching among optional process flows, dependent data values, and conditions or error conditions.

Another useful object of a package is the container. A container is the package object used to group other objects and other containers. Common uses of containers are for performing iterative processing such as looping through a dataset or processing a set of data files within a directory. Although the container object is within a package, you can consider the SSIS package itself as a special high-level container.

SSIS objects also include a comprehensive set of transformation tasks that are important for data integration and BI solutions. These tasks are designed for merging or aggregating data and for converting and transforming data formats and types. Some new tasks have been provided for handling specialized BI operations such as managing slowly changing dimension data. You can also extend SSIS with your own custom tasks and transformations to handle unique requirements within your business application systems environment.

Perhaps best of all, you will find that with all the SSIS objects available to you for package creation, you can create robust, high-performance ETL and data integration applications with no programming code required. By simply dragging and dropping containers, sources, destinations, transformations, and other objects, the SSIS designer automatically creates all the package executable code for you. Throughout the next several chapters, you will learn more about package objects and control components and practice with many of the objects available to design and develop SSIS packages.

SSIS Process Control

A significant advancement to SSIS is the package architecture design for its process control management. You've already learned that the SSIS process control architecture includes the control flow, data flow, and event handler components. Each of these process control components includes common and unique sets of objects for you to use when designing and creating your packages.

SSIS Control Flow

SSIS package objects (containers, data flow tasks, administration tasks, precedence constraints, and variables) are elements of the control flow component of the process control architecture. The control flow is the highest-level control process. It allows you to orchestrate and manage the run-time process activities of data flow and other processes within a package. In fact, you can design a control flow by using an Execute Package task to manage the sequence of processing for a set of existing packages in a Master Package concept. This capability allows you to combine individual packages into a highly manageable workflow process. Use precedence constraints to set the process rules and to specify sequence within the control flow. An SSIS package consists of a control flow and one or more objects. Data flow and event handler process control components are optional.

SSIS Data Flow

When you want to extract, transform, and load data within a package, you add an SSIS data flow task to the package control flow. Each data flow task creates its own data flow process control component for processing at run time. You configure each data flow to manage data sources, data destinations, and optional data transformations for any kind of data manipulation your packages might require. You can have as many data flow components within a package as you need to handle all the kinds of data sources and destinations you might have.

The SSIS data flow component provides a comprehensive set of pre-defined data sources and destination objects to enable you to design and develop packages easily for most of the databases and data source files you might have within your IT environment. You can add custom data sources if you need them. Data destinations allow you to deliver data from a data flow process in a variety of formats. An SSIS package can even provide data directly to an application by storing it in an ASP.NET *DataReader* destination object. Using this destination-type object, you don't have to place the data in a persistent data store, and you can design application integrations, enabling near real-time data delivery.

A set of data transformation task objects is provided within SSIS data flow. These transformation tasks have been designed to meet most, if not all, of the kinds of data conversion, manipulation, standardization, merging, splitting, fuzzy matching, and other types of transformations without having to write complicated programming code. You will learn about many of these transformation tasks, data sources, and destination objects later, in Part II of this book, "Designing Packages."

SSIS Data Pipeline

The SSIS data flow process control component and its tasks are processed by the *data flow engine* within SSIS. A key feature of the SSIS data flow engine is the *data pipeline*, shown in Figure 1-1, which uses memory buffers to improve processing performance. The data pipeline enables parallel data processing options and reduces or eliminates multiple passes of

reading and writing of the data during package execution and processing. This level of efficiency means you can process significantly more data in shorter periods of time than is possible if you rely simply on stored procedures for your ETL processes.

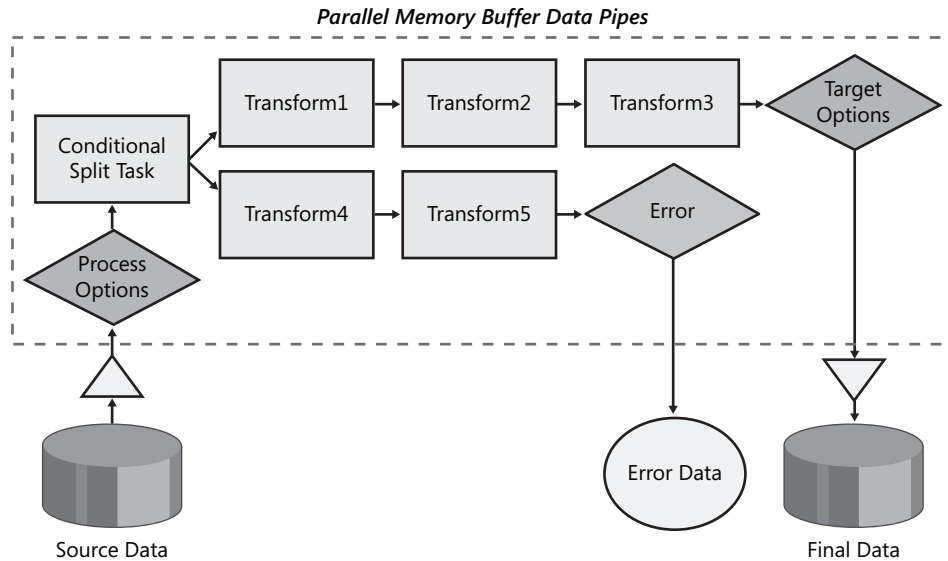


Figure 1-1 The SSIS data flow data pipeline

Maximum data processing performance for SSIS packages is achieved because the data pipeline uses buffers to manipulate data in memory. Source data, whether it's relational, structured as XML data, or stored in flat files like spreadsheets or comma-delimited text files, is converted into table-like structures containing columns and rows and loaded directly into memory buffers without the need of staging the data first in temporary tables. Transformations within a data flow operate on the in-memory buffered data as well as on sorting, merging, modifying, and enhancing the data before sending it to the next transformation or on to its final destination. By avoiding the overhead of re-reading from and writing to disk, the processes required to move and manipulate data can operate at optimal speed.

SSIS Event Handler

The *event handler* process control, unlike the data flow process control, is not managed by the control flow. When you want to control processing at specific occurrences of events during package execution, you use the SSIS event handler process control component. An event handler runs in response to an event raised by the package or by a task or container within the package. Typically, event handlers are created in a package to perform special processing as a result of data anomalies, to trigger other programs, or to launch other packages based upon the event state within the running package. For example, you can create an event handler to send an e-mail alert notification in the event of a task or package for either a success or a failure or simply for a completion state.

You will learn more about SSIS package architecture and its objects and process control components later, in Part II of this book.

SSIS Components

So far, you've learned about SSIS objects and process control architecture. Now you will learn about the SSIS components that you use to design, test, deploy, manage, schedule, and execute SSIS packages. Some of the SSIS components reside on the SSIS server, whereas other components reside on your desktop workstation. A sample configuration scenario is shown in Figure 1-2.

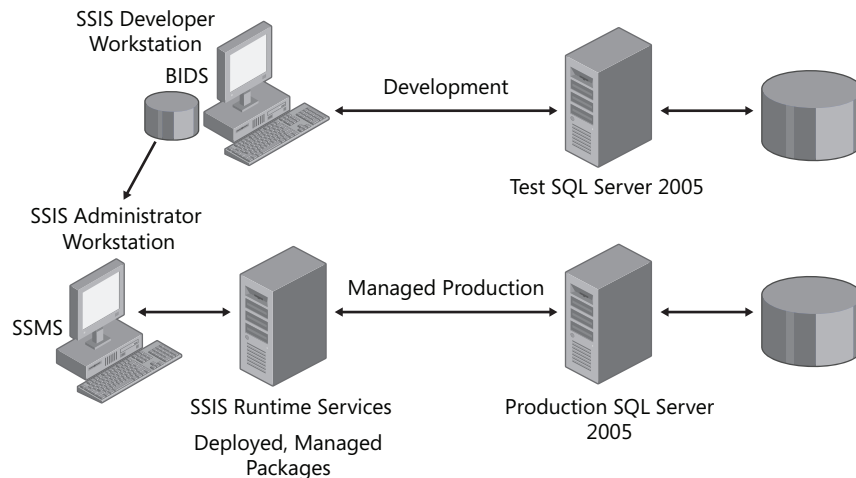


Figure 1-2 Sample SSIS components configuration scenario

SSIS Development Studio

The Business Intelligence Development Studio (BIDS) is the desktop workstation component you use to design, develop, and test SSIS packages. BIDS provides you with a totally graphical-oriented development environment, allowing you to copy, maintain, and create new packages by using a menu and toolbox drag-and-drop method for development. BIDS is a comprehensive development platform that supports collaboration with source code management and version control; provides debugging tools such as breakpoints, variable watches, and data viewers; and includes the SQL Server Import and Export Wizard to jump-start package development.

Within BIDS, the SQL Server Import and Export Wizard allows you to generate SSIS packages to copy data from one location to another quickly and easily. The Import and Export Wizard guides you through a series of configuration editor pages that allow you to select the source

data, select your target destination, and map source to target data elements. You might find this wizard helpful for creating a starting point for a package. Once a package is generated by the wizard, you can then further enhance the package by using BIDS. You will learn how to use BIDS in Chapter 2, “Building Your First Package.”

SSIS Runtime Services

SSIS Runtime Services manages storage of packages in .dtsx (SSIS package system file format) files or in the MSDB database and manages and monitors their execution. SSIS Runtime Services saves your package layout, applies configurations, executes packages, manages data source and destination connection strings and security, and supports logging for tracking and debugging. SSIS Runtime Services executables include the package and all its containers, tasks, custom tasks, and event handlers.

After you design, develop, and complete your testing of SSIS packages from your desktop BIDS, you will want to deploy and implement the packages for scheduled or on-demand processing to the SSIS Runtime Services server. In some companies, the deployment of finished packages is oftentimes performed by a production administrator or other authorized group. At other times, packages can be deployed by the developer. Either way, you can use the graphical interface or a command-line utility to configure and complete the package deployment.

SSIS Package Deployment

The SQL Server Management Studio (SSMS) is a desktop workstation component for the deployment and management of packages into production environments. SSMS connects directly to SSIS Runtime Services and provides access to the Execute Package utility, is used to import and export packages to and from available storage modes (MSDB database or SSIS Package Store), and allows you to view and monitor running packages.

There are also two command-line utilities that you can use to manage, deploy, and execute SSIS packages. Use Dtexec.exe to run a package at the command prompt. An alternative to SSMS, Dtutil.exe, provides package management functionality at the command prompt to copy, move, or delete packages or to confirm that a package exists. You will learn all about the roles of these services and other SSIS application deployment procedures later, in Part III of this book, “Managing Packages.”

Finally, a more advanced feature is the Integration Services Object Model that includes application programming interfaces (APIs) for customizing run-time and data flow operations and automating package maintenance and execution by loading, modifying, or executing new or existing packages programmatically from within your business applications.

SQL Server 2000 DTS Migration

SSIS is the next generation of the former Microsoft Data Transformation Services (DTS) that is included within the previous versions of SQL Server. SSIS has been designed with a new, high-performance, and advanced underlying architecture. The good news is that if you already have an inventory of SQL Server Data Transformation Services (DTS) packages, all of these packages will continue to run in SSIS environments without any changes. In addition, SSIS provides the *Package Migration Wizard* that you can use to convert SQL Server 2000 DTS packages to SSIS packages. Because of some of the significant improvements, such as the SSIS control flow and data pipeline architectures, as well as many of the new and enhanced tasks and transformations, DTS package conversion might not always be complete and could require some final manual enhancements. You might also want to redesign some of your existing DTS packages to take advantage of the performance improvements and additional task functionality that is now available within SSIS.

Chapter 1 Quick Reference

This term	Means this
SSIS package	A discrete executable unit of work composed of a collection of control flow and other objects, including data sources, transformations, process sequence, and rules, error and event handling, and data destinations.
Containers	Package objects that provide structure to packages and special services to tasks. Containers are used to support repeating control flows in packages and to group tasks. Containers can include other containers in addition to tasks.
Tasks	Package elements that define activities and processes, including data sources, destinations, transformations, and others.
Precedence constraints	Constraints that link executables, containers, and tasks within the package control flow and specify conditions that determine the sequence and conditions for determining whether executables run.
Variables	Storage for values that an SSIS package and its containers, tasks, and event handlers can use at run time. The scripts in the Script task and the Script component can also use variables.
Control flow	An SSIS package process control component used to control flow elements: the containers that provide structure in packages and services to tasks, tasks that provide functionality in packages, and precedence constraints that connect containers and tasks.
Data flow	An SSIS package data process control component defined from within package control flow that loads data from sources, transforms and routes it through transformations, and saves it to destinations.
Event handler	An SSIS package process control component used to define the process activities to be performed at the time of a specific event state for the package or for any of its tasks or containers.

This term	Means this
Data Pipeline	The memory-based, multithreaded, buffered transformation process flow of data through an SSIS data flow task during package execution.
BIDS	SQL Server Business Intelligence Development Studio. Provides the Integration Services project in which you create packages, their data sources, and data source views.
SSMS	SQL Server Management Studio. Provides the Integration Services service that you use to manage packages and monitor running packages.

Extracting and Loading Data

After completing this chapter, you will be able to:

- Understand and create connection managers.
- Extract and load data from different data sources to different destinations.
- Use data sources and data source views to extend the functionality of a regular connection manager.

In Chapter 2, “Building Your First Package,” you learned about how to build your first package, and you explored SQL Server Business Intelligence Development Studio (BIDS) and its basic components. In this chapter, you’ll learn how to set up a new Microsoft SQL Server Integration Services (SSIS) project, add a data flow task to extract data from a source, and load the results into a destination. Specifically, you’ll learn how to create and configure a connection manager for Microsoft Office Excel, SQL DB, and flat files. You will also learn how to use BIDS data sources and data source views to extend the functionality of a regular connection manager.

Connection Managers

A connection manager is an SSIS object that contains the information required to create a physical connection to data stores as well as the metadata describing the structure of the data. In the case of a flat file, a connection manager contains the file path, file name, and metadata identifying rows and columns. A connection manager for a relational data source contains the name of the server, the name of the database, and the credentials for authenticating access to the data. Connection managers are the bridge between package objects and physical data structures. They are used by tasks that require a connection (such as the Execute SQL task), by data adapters that define sources and destinations, and by transformations that perform lookups to a reference table.

Connection Manager Types

A connection manager is a logical representation of a connection. At design time, the properties of a connection manager describe the physical connection that Integration Services creates when the package runs. For example, a connection manager includes the *ConnectionString* property that is set at design time; at run time, a physical connection is created, using the value in the *ConnectionString* property.

Many tasks use connections. For example, an Execute SQL task (that runs SQL statements) requires a connection to a relational database. The sources and destinations in package data flows use connections to extract and load data. Some transformations also require connections to do their work. For example, the Lookup transformation uses a connection to access a reference table to look up and retrieve values. The following is the list of connection managers available in SSIS:

ADO	HTTP	ODBC
ADO.NET	MSMQ	SMO Server
Excel	MSOLAP90	SMTP
File	MultiFile	SQL Mobile
FlatFile	MultiFlatFile	WMI
FTP	OleDb	

This list represents the typical connection managers. However, SSIS gives developers the ability to write source components that can connect to custom data sources and supply data from those sources to other components in a data flow task.

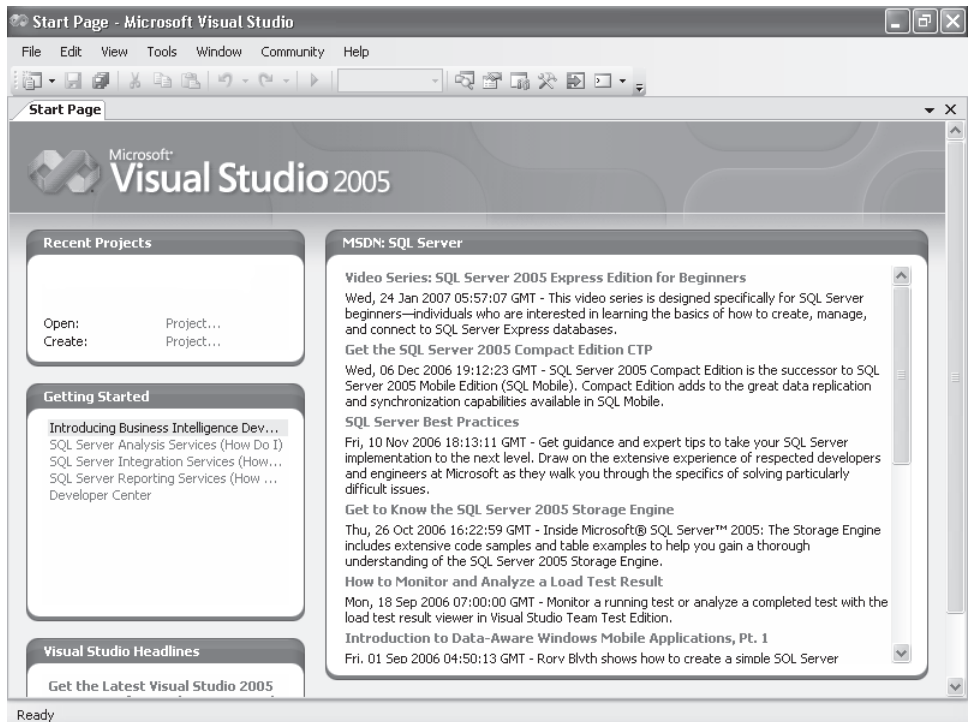
Creating a New Integration Services Project

The process of creating a SQL Server Integration Services project consists of several steps. The first step is to define a name and location for your project and solution. You can also define a new name for the default package that SSIS creates as part of this initial step. The second step in building an SSIS project is to create connection managers for data source and data destinations. You need to know where your data is stored, what the server name that hosts the data is, and which database or file stores the data. Verify that you have all the required credentials to retrieve that data and store the new data in a destination database or file. The third step in creating your new SSIS project is the creation of at least one data flow, for instance, to extract and load data. To create a data flow task to extract and load data, you will need to specify data adapters linked to the source and destination connection managers you define. You can create more than one data flow in a control flow and, indeed, you can connect them in a logical sequence. You will learn more about how to manage a set of data flows in Chapter 5, “Managing Control Flow.”

Now you will create a new Integration Services project to which you will add a data flow task. You will create a new package to extract data from a source table and load the data to an Office Excel file. These transformation processes simulate data-delivering routines that you might perform when working in a data warehouse or enterprise environment.

Create a new Integration Services project

1. Start SQL Server Business Intelligence Development Studio. Your screen should look similar to this:

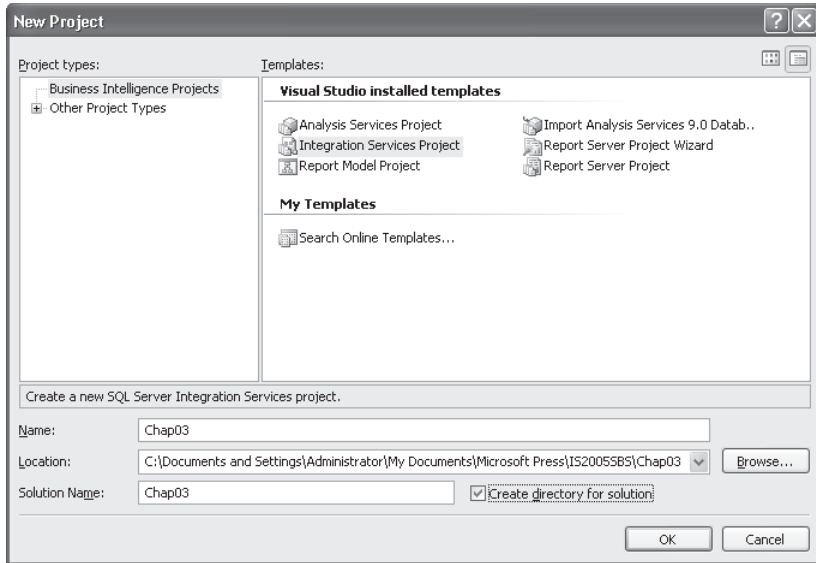


2. On the File menu, point to New, and then click Project.
3. Make sure that the Project Type is set to Business Intelligence Projects, and then click the Integration Services Project template.
4. Type a name for the project: **Chap03**

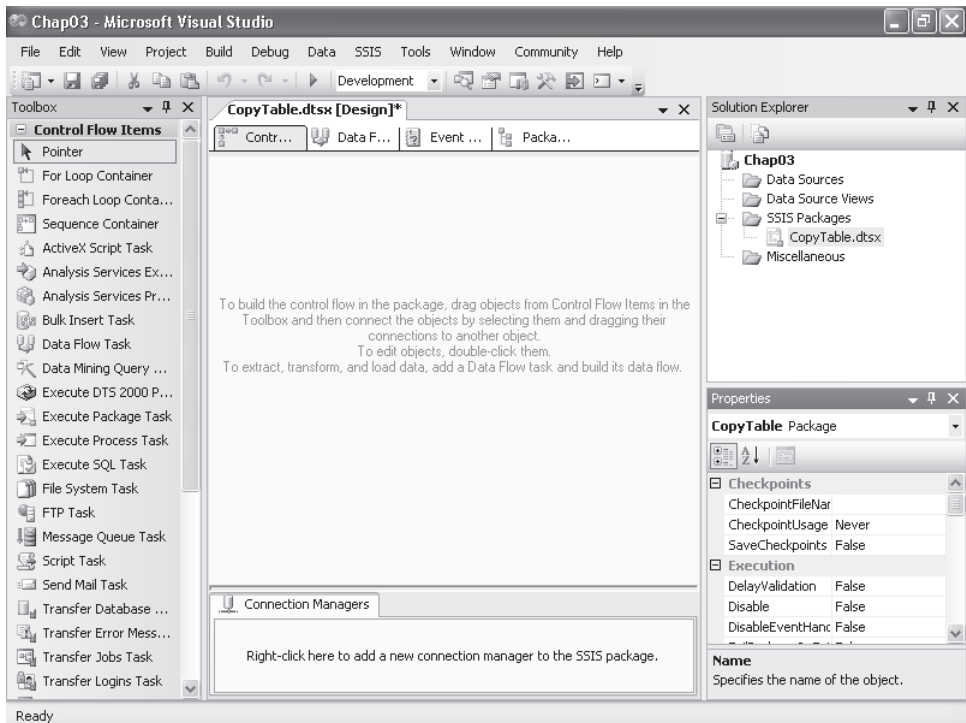


Note Notice that the text in the Solution Name box changes automatically to match the project name. You can change the name of the solution, especially when you have a solution with several projects. For now, leave it as Chap03.

5. Change the location for the project to **C:\Documents and Settings\\My Documents\Microsoft Press\is2005sbs\Chap03** and confirm that the Create Directory For Solution check box is selected. The New Project dialog box should look like this:



6. Click OK to continue.
7. In Solution Explorer, right-click the package and choose Rename to change the package name to **CopyTable.dtsx**. Click Yes when prompted.
8. Click Yes to rename the package object as well. Now you should see this:



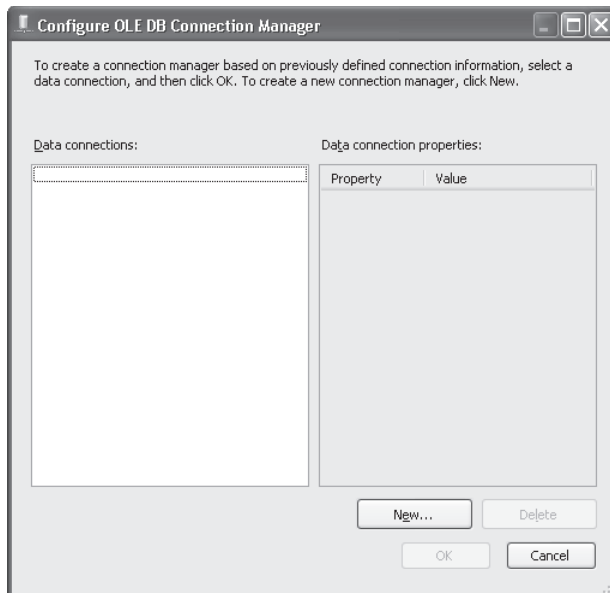
Adding Connection Managers

The second step in building an SSIS project is to create connection managers for data source and data destinations. As described before, connection managers are logical representations of a connection. Connection managers you can add include connections to Oracle, FTP, and HTTP sites; Analysis Services databases; flat files; and more. Each connection manager has its own configuration, depending on the type of connection you want to set.

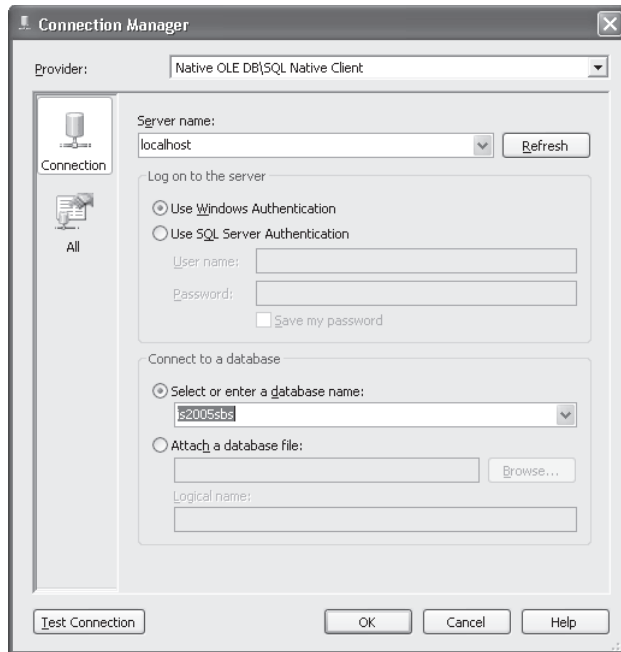
In the next two procedures, you'll add a connection manager for a SQL Server 2005 database and another connection manager for Office Excel.

Add an OLE DB connection manager for the *is2005sbs* database

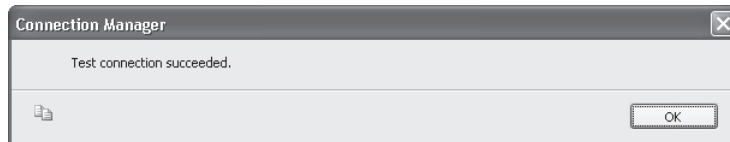
1. Right-click anywhere in the Connection Managers pane at the bottom of the Control Flow tab and click New OLE DB Connection.



2. Click New to define a new connection, click the Provider drop-down list to review available providers, and then click Cancel to keep default: Native OLE DB\SQL Native Client.
3. Type **localhost** for the Server Name.
4. Select Use Windows Authentication.
5. Choose *is2005sbs* as the database.



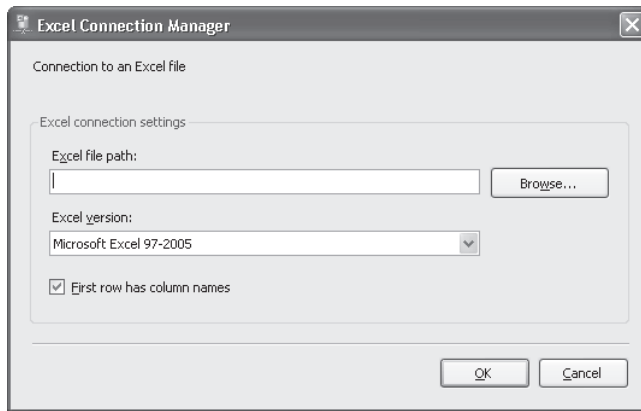
6. Click the Test Connection button. The following window will appear:



7. Click OK twice.

Add an Office Excel connection manager to the Employee.xls file

1. Create a new folder called **Data** in C:\Documents and Settings\\My Documents\Microsoft Press\is2005sbs\Chap03.
2. Right-click anywhere in the Connection Managers pane at the bottom of the Control Flow tab and click New Connection.
3. In the Add SSIS Connection Manager dialog box, click EXCEL (connection manager for Excel files) and click Add.



4. Browse to C:\Documents and Settings\\My Documents\Microsoft Press\is2005sbs\Chap03\Data\.
5. Type **Employee** in the File Name box, click Open, and then click OK.
6. Right-click Excel connection manager, select Rename, and rename the connection **Employee**.

Creating a Data Flow

An SSIS package needs at least one component in a control flow. This component could be a data flow task or any component from Control Flow Items or Maintenance Plan Tasks in the Microsoft Visual Studio Toolbox. Basically, you build a control flow by adding tasks or control flow components to the Control Flow tab.

The third step in creating your new SSIS project is the creation of at least one data flow, for instance, to extract and load data. To create a data flow task to extract and load data, you will need to specify data adapters linked to the source and destination connection managers you define. There are different ways to create data flows in a control flow. In this procedure, you'll create a data flow task.

Create a data flow task

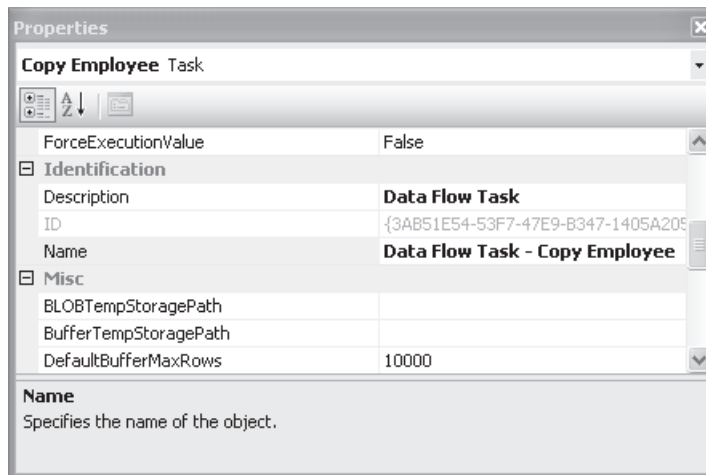
1. Click the Data Flow tab.

Click the message link in the center of the page to add a task.



Tip If you go to the Data Flow tab right after creating a package, you see a message that no data flow tasks have been added to the package. Clicking the message link adds a new task that you can also access from the Control Flow page.

2. In the Properties pane, change the *Name* property to **Data Flow Task – Copy Employee**.



Tip If the Property panel is not active, press F4 to activate it.

Adding Data Adapters

Now you are ready to add data adapters to your data flow task. The term *data adapter* refers to a set of objects that provide the ability to connect to, and interact with, databases, files, and other resources that provide data storage. Data adapters are used to read, insert, modify, and delete data from these various data storage devices. Within a data flow task, data sources and data destinations are specific implementation types of data adapters.

A data adapter is an object that can be used only in the data flow task and requires a connection manager to be established.

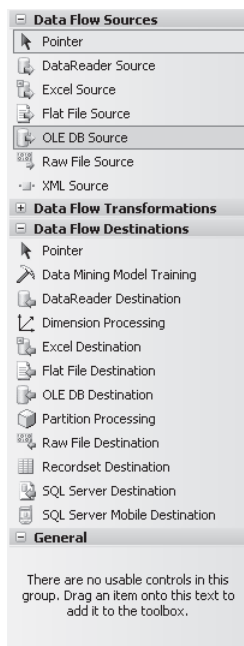
In this procedure, you'll add and map source and destination data adapters.

Add an OLE DB source data adapter

1. Open the Toolbox and review the available objects.



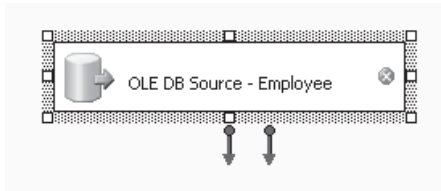
Note Note that the Toolbox changes. Objects are organized into three main groups in the Toolbox when you are designing a Data Flow: Data Flow Sources, Transformations, and Destinations.



2. Drag OLE DB Source from the Toolbox to the grid.
3. In the Properties pane, change the *Name* property to **OLE DB Source - Employee**.



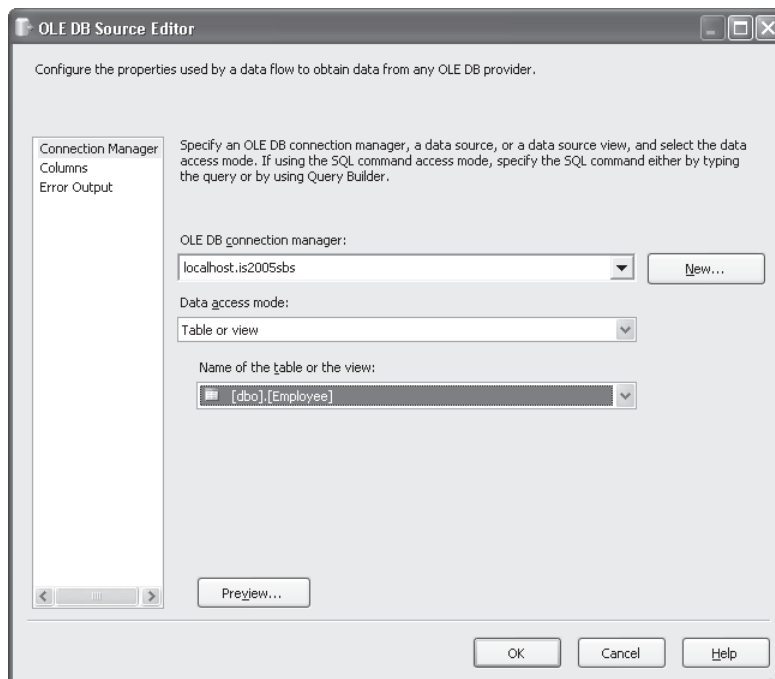
Note Notice the small red circle with an x inside of it. Integration Services adds an indicator to the object to let you know that it needs a connection manager, which allows tasks to connect to external data sources.



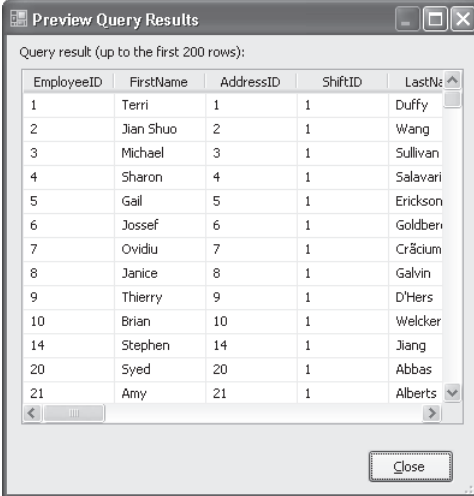
4. On this step, you'll add the connection manager to the source adapter.

Add the localhost.is2005sbs Connection Manager to the OLE DB Source data adapter

1. Double-click the OLE DB Source – Employee data adapter to open the OLE DB Source Editor and click the OLE DB Connection Manager drop-down list.
2. In the drop-down list, select localhost.is2005sbs, and then click OK.
3. Click the Data Access Mode drop-down list to see the different access mode.
4. Select Table Or View.
5. In the Name Of The Table Or The View drop-down list, select the [dbo].[Employee] table.



- Click the Preview button to see sample data of employees, and then click Close.



Preview Query Results

Query result (up to the first 200 rows):

EmployeeID	FirstName	AddressID	ShiftID	LastNm
1	Terri	1	1	Duffy
2	Jian Shuo	2	1	Wang
3	Michael	3	1	Sullivan
4	Sharon	4	1	Salavari
5	Gail	5	1	Erickson
6	Jossef	6	1	Goldber
7	Ovidiu	7	1	Crăciun
8	Janice	8	1	Galvin
9	Thierry	9	1	D'Hers
10	Brian	10	1	Welcker
14	Stephen	14	1	Jiang
20	Syed	20	1	Abbas
21	Amy	21	1	Alberts

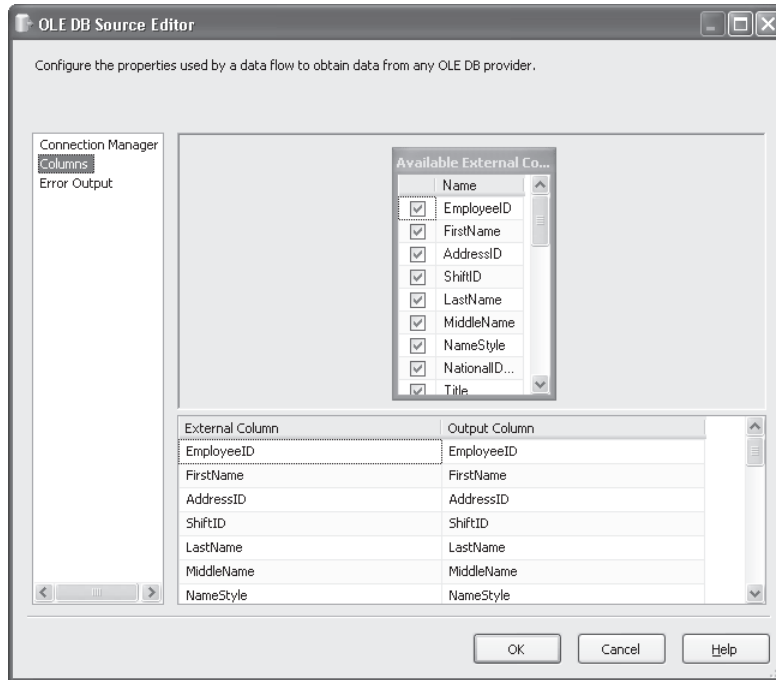
Close

Map the connection manager to the data adapter

- Click Columns from the left panel of the Editor. This action maps columns from the connection manager to output columns of the adapter.



Note Mapping between the external column (from the connection manager) and the output column (from the data adapter) is generated automatically when you open this page.



Now you have a data adapter that has been associated with a connection manager and is now ready to be used in a transformation.

2. Click OK.



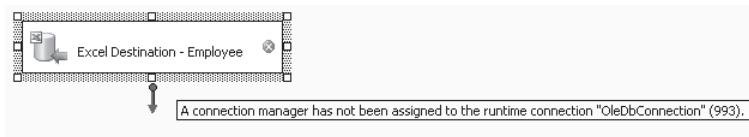
Note Notice that the small red circle on this data adapter has disappeared.

Add an Excel Destination data adapter

1. Open the Toolbox and expand the Data Flow destinations.
2. Drag Excel Destination from the Toolbox to the grid.
3. In the Properties pane, change the *Name* property to **Excel Destination – Employee**.



Note Notice the small red circle with an x inside of it on this data adapter. Integration Services adds an indicator to the object to let you know that it needs a connection manager.



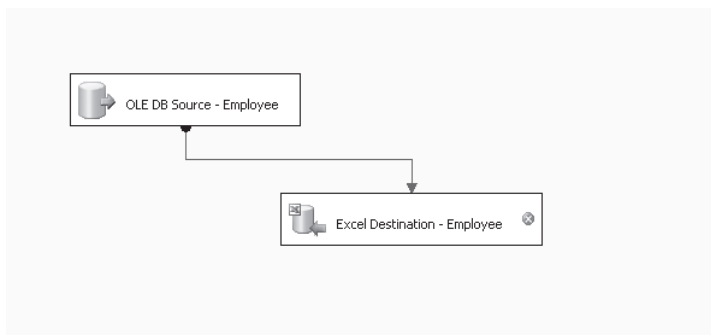
Add the Employee connection manager to the Excel Destination data adapter

1. Double-click the Excel Destination – Employee data adapter.



Important Note that a warning is displayed. This component has no available input columns. You need to connect the source and the destination.

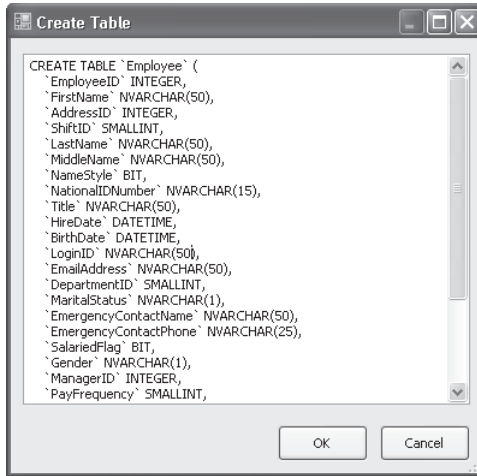
2. Click No.
3. Click the OLE DB Source – Employee adapter and connect it to the Excel Destination adapter by dragging the green arrow from OLE DB Source – Employee to Excel Destination – Employee.



4. Double-click the Excel Destination – Employee data adapter to open the Excel Destination Editor and verify that Employee is selected in the OLE DB Connection Manager drop-down list.
5. In the Name Of The Excel Sheet drop-down list, click New.
6. Change the name of the sheet to **Employee** by replacing the current name, Excel Destination, next to the CREATE TABLE statement. Keep the quotation marks and change the size of the LoginID column to NVARCHAR(50).



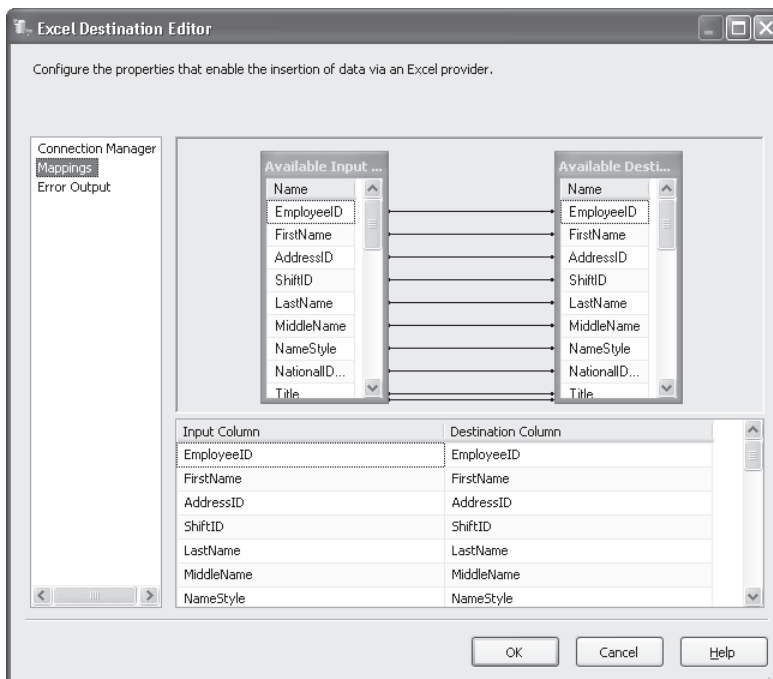
Note The Excel connection manager will not allow creation of long columns.



7. Click OK.
8. Click Preview and see that the new table is empty, and then click Close.
9. Click Mappings in the left panel of the Editor.



Note Mapping between the input column and the destination column (from the Excel data adapter) is generated automatically when you open this page.



10. Click OK.



Note Notice the warning icon on the Excel Destination – Employee adapter. Integration Services warns that a Truncation might occur in the LoginID column because the length of the source LoginID column is 256. In this case, it is not a problem because that column has no data larger than 50 characters.

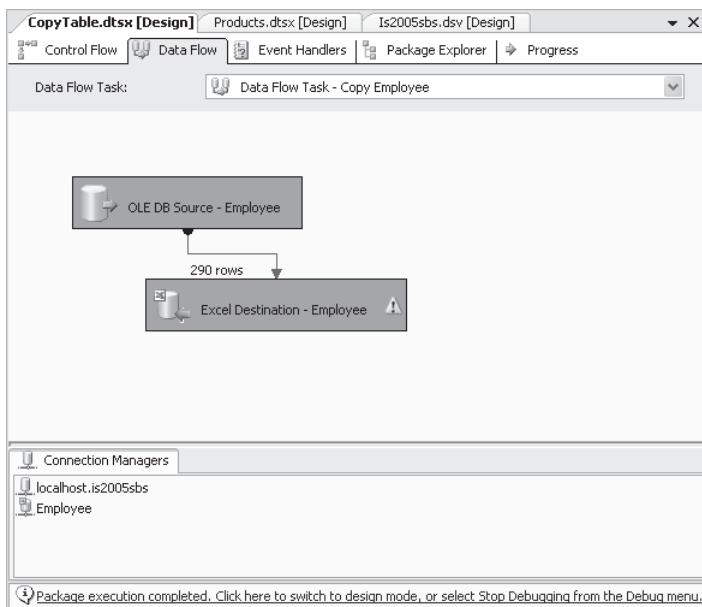
Executing the Package

Once you have created a new SSIS project with connection managers for sources and destinations, created a data flow task with source and destination data adapters, and mapped the columns that you want to transfer from your source table to your destination Office Excel file, you are ready to run this package.

When you execute a package, Integration Services validates the package and executes the tasks defined in the control flow. You can change certain properties to optimize the processing time. You can learn more about optimization in Chapter 11, “Optimizing SSIS Packages.” In this procedure, you’ll execute the package you have built.

Execute the package

1. Right-click the CopyTable.dtsx package and choose Execute Package.



2. Click the Stop Debugging button on the Debug toolbar.

3. Using Windows Explorer, navigate to the C:\Documents and Settings\\My Documents\Microsoft Press\IS2005sbs\Chap03\Data\ folder.
4. Open the Employee.xls file to confirm that data appears in the file.
5. Click the Employee tab, and data should appear.

Using Data Sources and Data Source Views

A data source is a connection that represents a simple connection to a data store; it includes all tables and views in the data store. A data source has project scope, which means that a data source created in an Integration Services project is available to all the packages in the project. A data source can be defined and then referenced by connection managers in multiple packages. This makes it easy to update all connection managers that use that data source. A project can have multiple data sources, just as it can have multiple connection managers.

Although a data source includes all tables and views, a data source view selects specific database objects (such as tables and views) or adds new relationships between objects. You can extend a data source view by adding calculated columns that are populated by custom expressions, adding new relationships between tables, replacing tables in the data source view with queries, and adding related tables. You can also apply a filter to a data source view to specify a subset of the data selected.

The objective of the next exercise is to load data from a new table, Products, to a flat file. You will create the product's table by defining a named query in a data source view. In addition, you will create a new data source, as source for the data source view, in the connection manager.



Note Use the previous project as the source.

Creating a Data Source

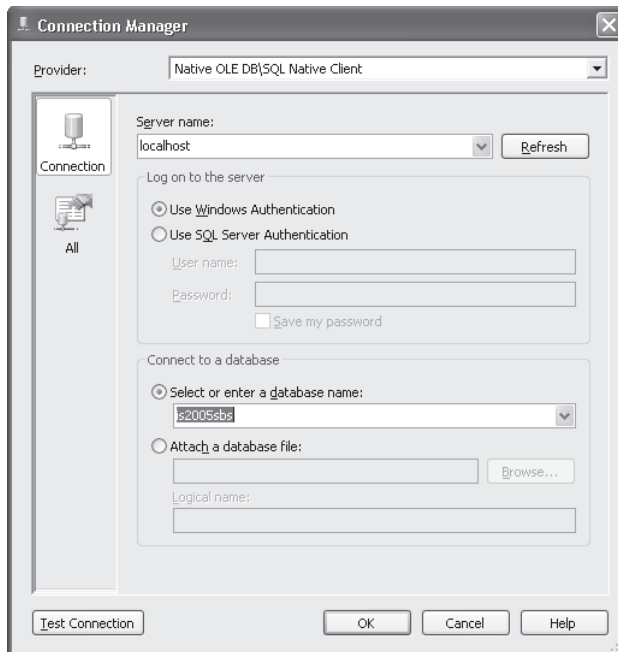
In this step, you make your decision about how to define the connection string for your data source. You can create a new connection, a data source based on an existing connection, a data source based on another object, such as an existing data source in your solution, or an Analysis Services project.

In this procedure, you'll create a data source based on a new connection.

Create a data source

1. In Solution Explorer, right-click the Data Sources folder, and then click New Data Source.
2. On the Welcome To The Data Source Wizard page, click Next.

3. On the Select How To Define the Connection page, verify that Create A Data Source Based On An Existing Or New Connection is selected, and then click New.
4. The connection manager dialog box appears with Native OLE DB\SQL Native Client selected in the Provider drop-down list.
5. Leave the Native OLE DB\SQL Native Client provider selected.
6. Type **localhost** in the Server Name box.
7. Select Use Windows Authentication.
8. Select is2005sbs as the database from the drop-down list. Your screen looks like this:

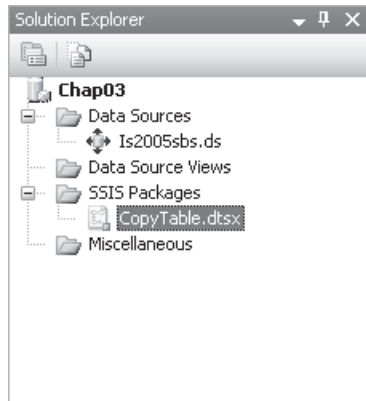


9. Click the Test Connection button and verify that it is successful. Then click OK twice.



Note The New Connection localhost.is2005sbs should now appear in the Data Connections pane.

10. Click Next. The Completing The Wizard page will appear, and a default data source name is displayed in the Data Source Name box.
11. Click Finish. The new data source will appear in the Data Sources folder in Solution Explorer.



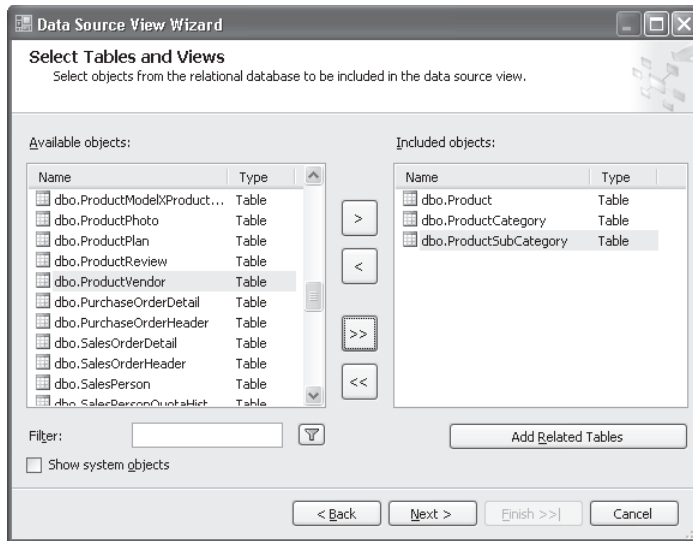
Creating a Data Source View

In this step, you select objects from the relational database to be included in the data source view. You can also include system objects or select one table and automatically add related tables to that one.

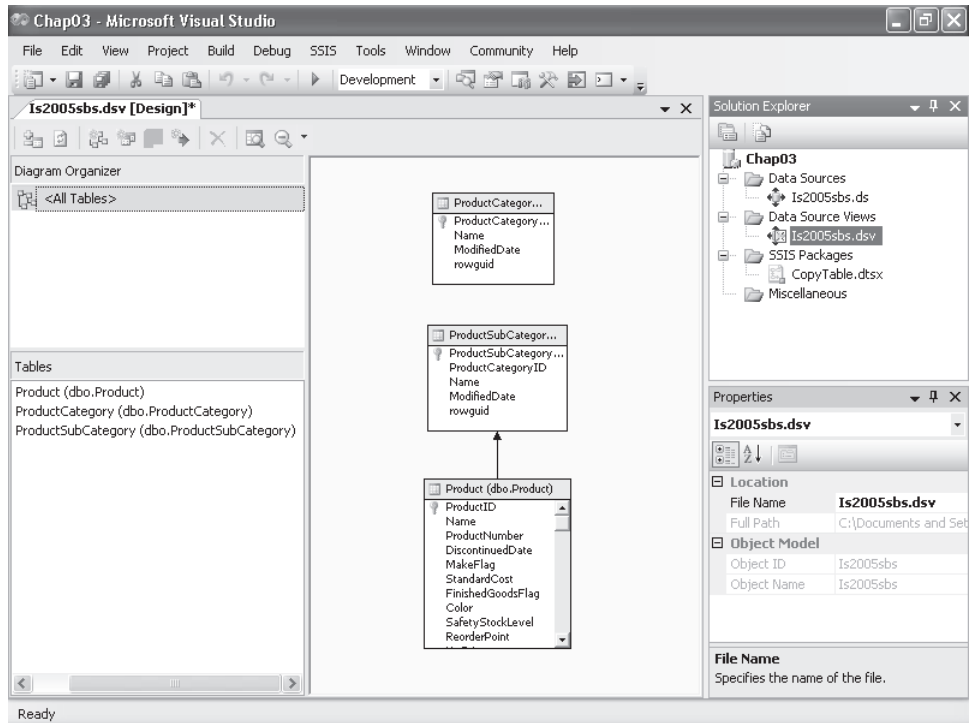
In this procedure, you'll specify a data source and select tables to define a new data source view.

Create a data source view

1. In Solution Explorer, right-click the Data Source Views folder, and then click New Data Source View.
2. On the Welcome To The Data Source View Wizard page, click Next.
3. On the Select A Data Source page, in the Relational Data Sources list, click the existing data source Is2005sbs as the primary data source for the data source view. The properties of the selected data source appear in the Data Source Properties pane.
4. Click Next.
5. On the Select Tables And Views page, select:
 - dbo.Product.
 - dbo.ProductCategory.
 - dbo.ProductSubCategory.
6. Click the right arrow to include them in the Included Objects.



7. Click Next. Leave Is2005sbs as a name for this data source view. This is the default data source view name, which is the name of the data source for which you are creating the data source view. The Preview pane displays a tree view of the objects in your new data source view.
8. Click Finish. The new data source view will appear in the Data Source Views folder in Solution Explorer.



Creating a New Named Query

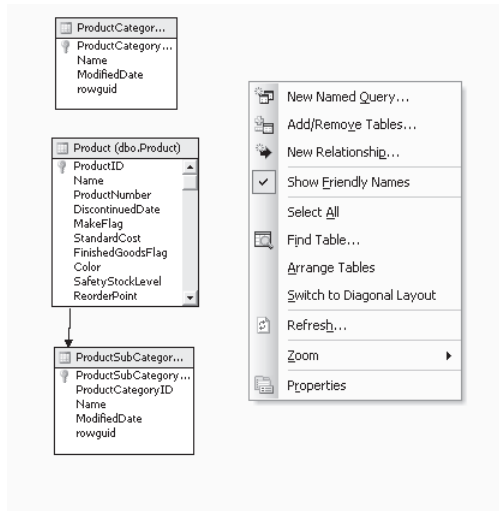
A named query is a table based on a SQL Expression. In this SQL Expression, you can specify columns and rows from more than one table even from different data sources. You can expand a relational schema by using named queries without modifying the original data source. You can split tables or join tables into a single data source views table.



Note You cannot base a named query on a table that contains a named calculation.

Create a named query

1. In Solution Explorer, expand the Data Source Views folder, and then open the .dsv file in Data Source View Designer by doing one of the following:
 - a. Double-click the .dsv file.
 - b. Right-click the .dsv file and click Open.
 - c. Select the .dsv file, and then, on the View menu, click Open.
2. In the Tables pane, right-click an open area, and then click New Named Query.

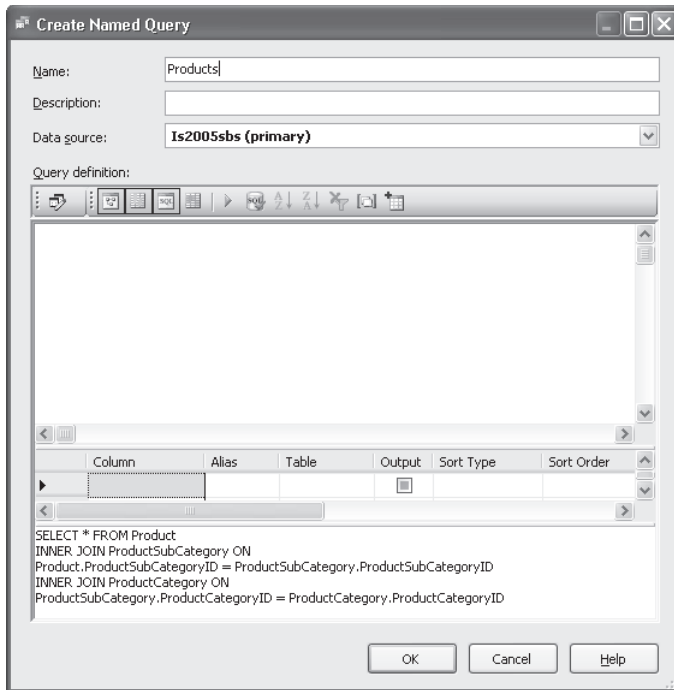


3. In the Create Named Query dialog box, do the following:
 - a. In the Name text box, type **Products**.
 - b. In the Data Source drop-down list, verify that Is2005sbs (primary) is selected.
 - c. Type or copy the next query in the bottom pane. Replace the current statement.

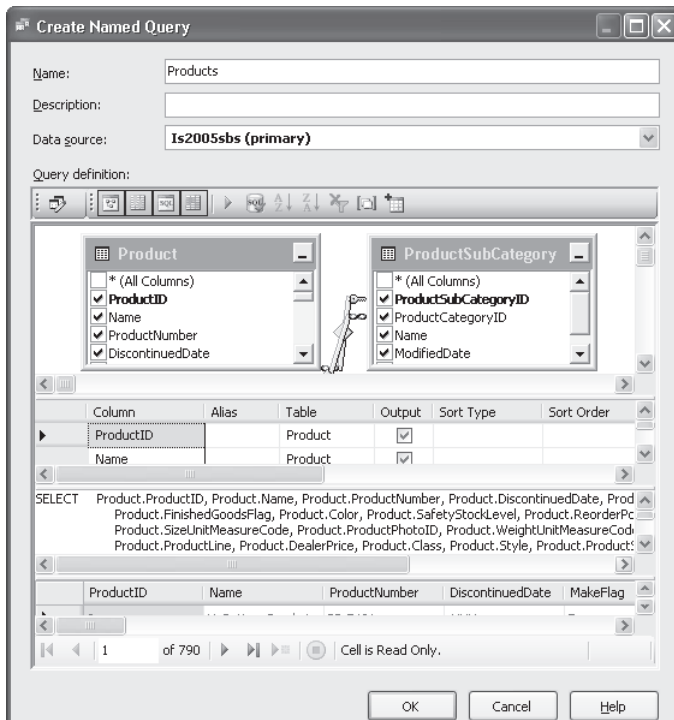
```

SELECT * FROM Product
INNER JOIN ProductSubCategory ON
Product.ProductSubCategoryID =
ProductSubCategory.ProductSubCategoryID
INNER JOIN ProductCategory ON
ProductSubCategory.ProductCategoryID = ProductCategory.ProductCategoryID

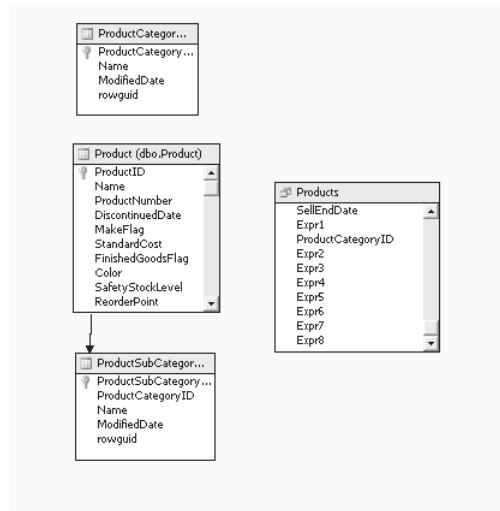
```



- Under Query Definition, click the Run icon.



5. Click OK.
6. Click OK. A new table will appear in the design pane with the name Products.



Copying Data from a Named Query to a Flat File

Once you have created a new table by defining a named query, you are ready to use it in a data flow. Then, the next steps are to create a new data flow, create source and destination data adapters, and map the flow of data.

In this next procedure, you'll create a new data flow task, create and configure an OLE DB Source adapter using the named query created in the previous step, and create and configure a destination flat file data adapter.

Copy data from a named query products table to a flat file

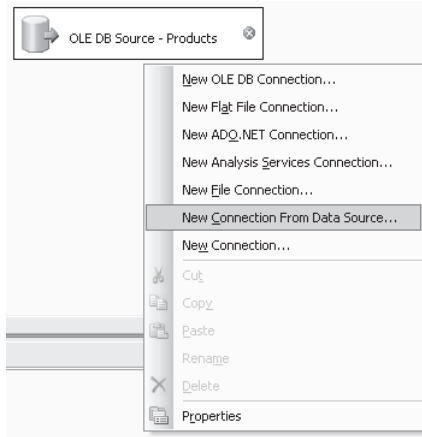
1. In Solution Explorer, right-click SSIS Packages, and then select New SSIS Package.
2. Right-click Package1.dtsx, select Rename, and name the new package **Products**.
3. Click Yes to also rename the package object.
4. In the designer, drag a Data Flow Task from the Control Flow Items group of the Tool-box to the Control Flow design area.
5. In the Properties pane, change the *Name* property to **Data Flow Task – Copy Products**.
6. In the designer, double-click in the Data Flow Task component to open the Data Flow design area.
7. In the designer, drag an OLE DB Source from the Data Flow Sources group of the Tool-box to the Data Flow design area.

8. In the Properties pane, change the *Name* property to **OLE DB Source – Products**.

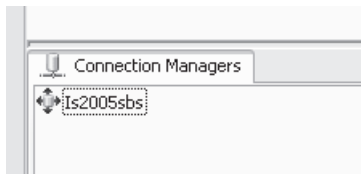


Tip Note the warning icon that appears in the OLE DB Source. You can hover your mouse over it to read the text of the warning.

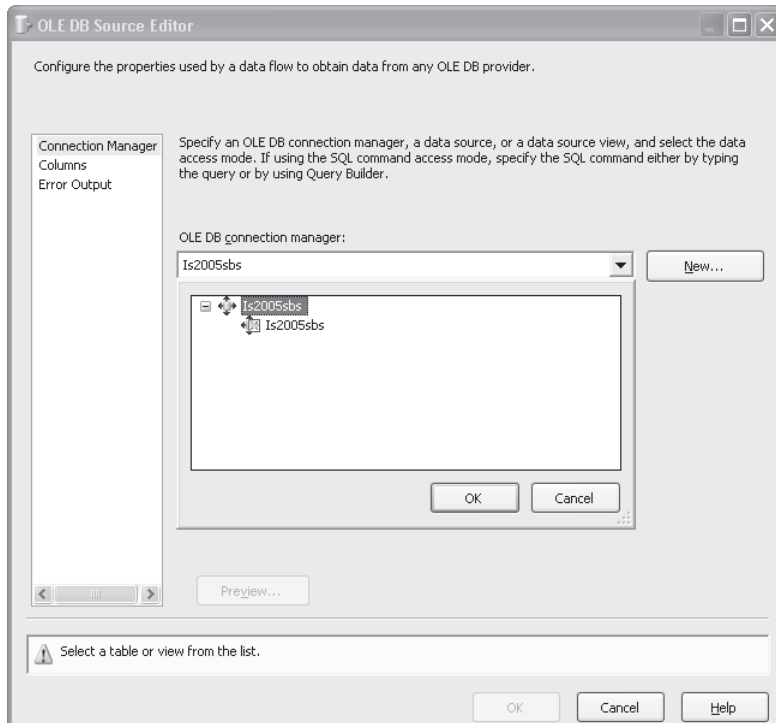
9. In the Connection Managers pane, right-click an open area, and then click New Connection From Data Source.



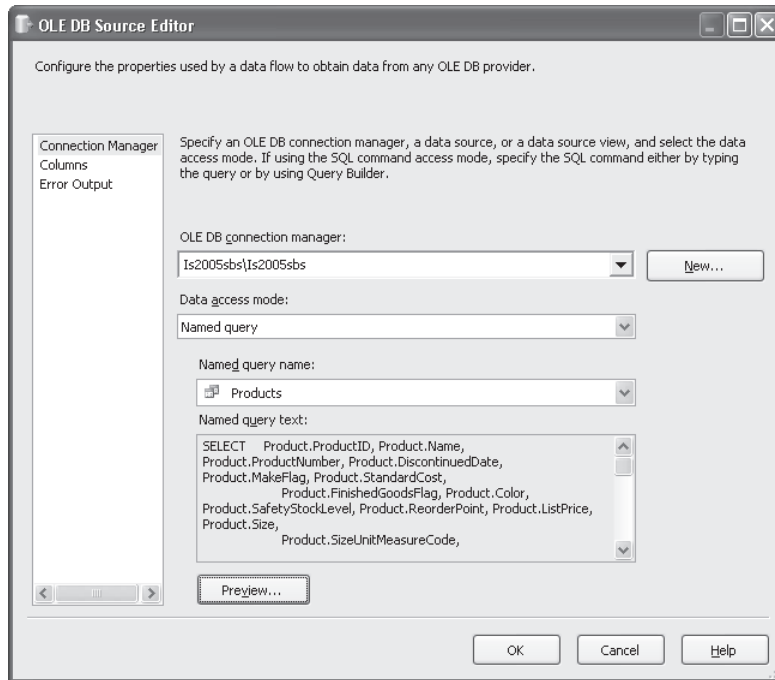
10. In the Select Data Source dialog box, ensure that Is2005sbs is selected, and then click OK.
11. Note that a new connection manager icon appears in the Connection Managers pane.



12. Double-click the OLE DB Source – Products component. Select Connection Manager, and then expand Is2005sbs from the OLE DB Connection Manager drop-down list. Select Is2005sbs Data Source View from the tree and click OK.



13. Now, in the Data Access Mode drop-down list, select Named Query. The named query products will be displayed. Click the Preview button to check the data.
14. Click the Close button, and then click OK to finish.



Connect to a flat file destination

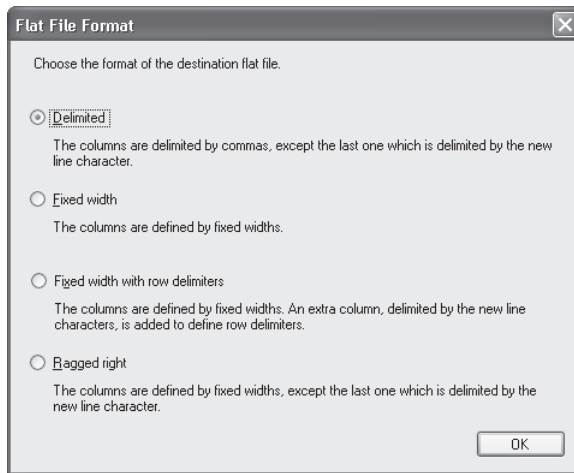
1. In the designer, drag a Flat File Destination from the Data Flow Destinations group of the Toolbox to the Data Flow design area.
2. In the Properties pane, change the *Name* property to **Flat File Destination – Products**.



Note Note the warning icon that appears in the Flat File Destination. You can hover your mouse over it to read the text of the warning.

3. Link OLE DB Source – Products and Flat File Destination – Products by dragging the green arrow from OLE DB Source – Products to Flat File Destination – Products.

4. Double-click Flat File Destination – Products to open the Flat File Destination Editor.
5. Ensure that Connection Manager is selected. Click the New button in the Flat File Connection Manager to open the Flat File Format window. Select Delimited and click OK.



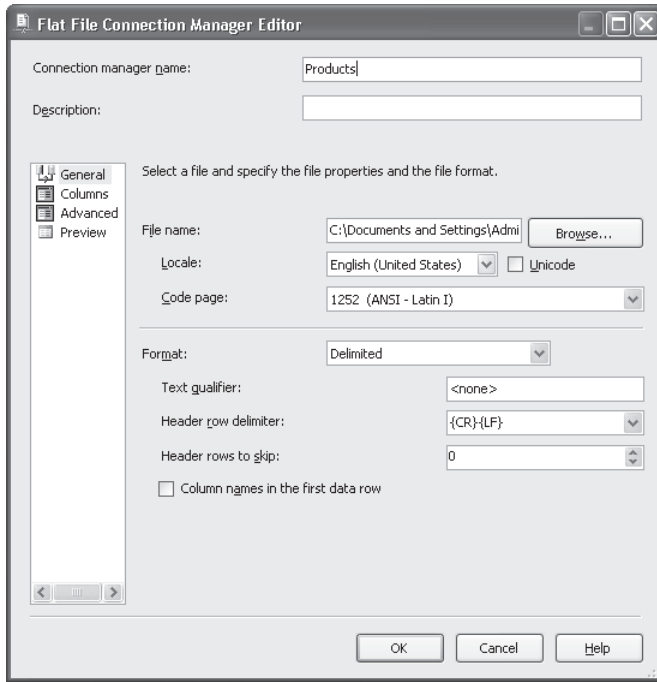
6. In the Connection Manager Name, change the *Name* property to **Products**.
7. In the Flat File Connection Manager Editor, click the Browse button and type **Products** in the File Name text box. Click Open.

Be sure that the folder is C:\Documents and Settings\\My Documents \Microsoft Press\is2005sbs\Chap03\Data.

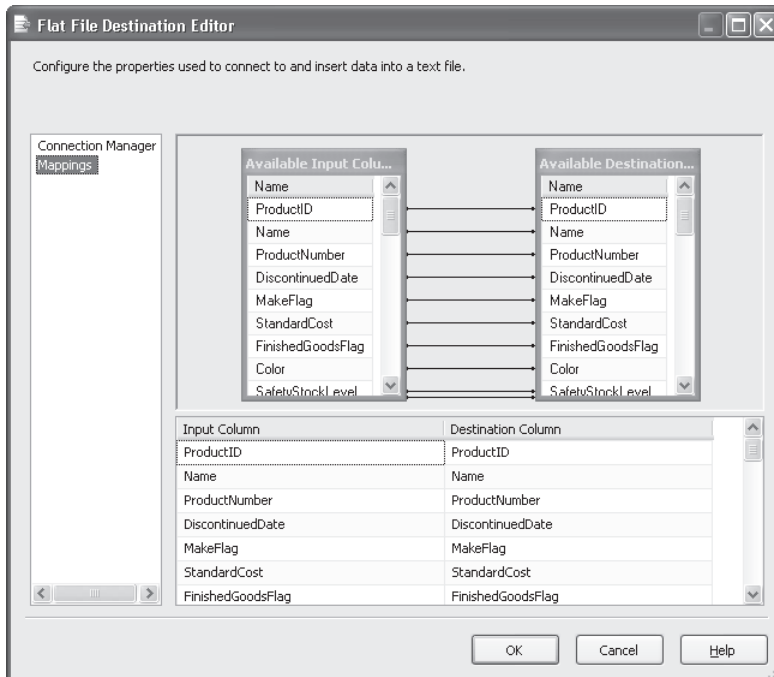
8. In the Flat File Connection Manager Editor, click OK.



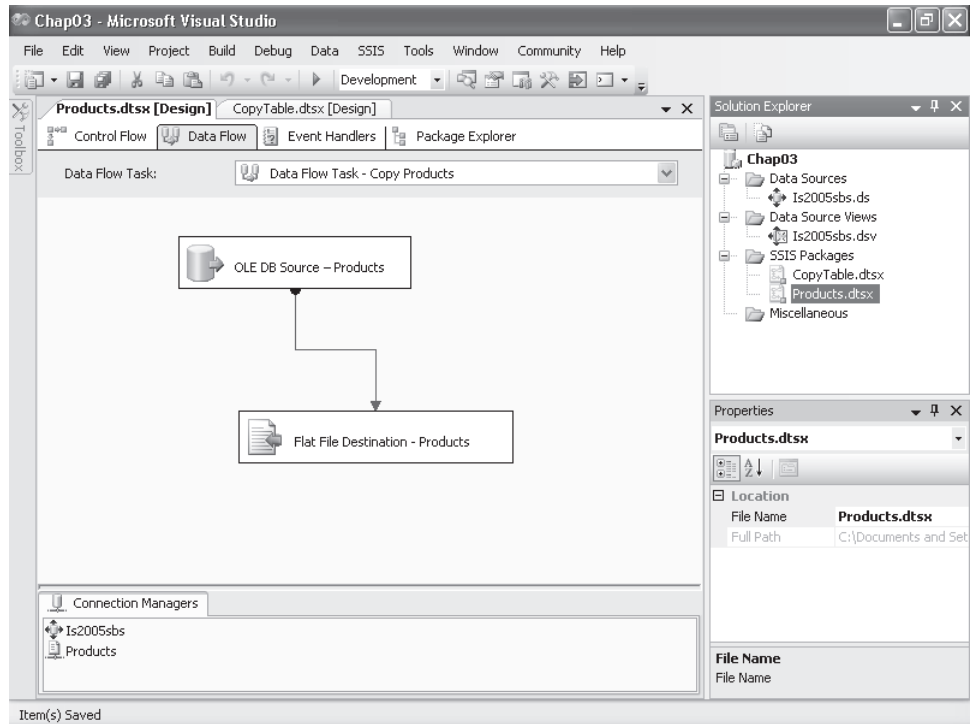
Note Note that the OK button is disabled in the Flat File Destination Editor. It is because mappings columns have not yet been set.



9. In the Flat File Destination Editor, click Mappings in the left pane. Verify that the columns are mapped correctly and click OK.



- Now you are ready to execute your package (Products.dtsx). Your package should look like this:

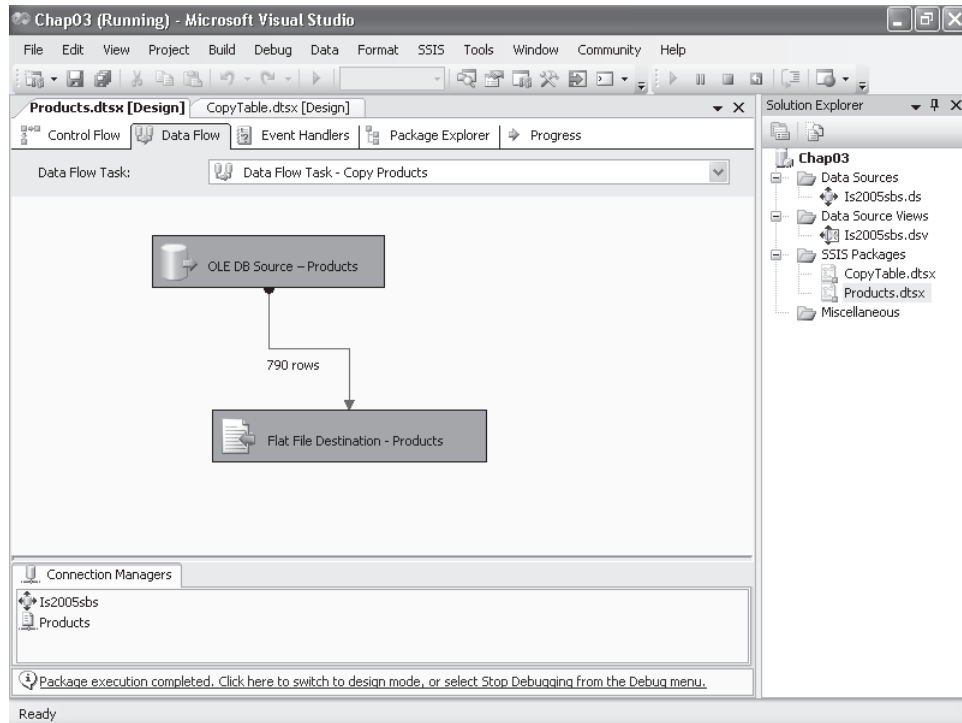


Executing the Package

This package is a very simple one that includes only one data flow. You have configured an OLE DB source based in a named query created in a data source view. When this package is executing, the data flow reads a buffer of data from the data source view and loads the data defined to the Named Query Products to a Products.txt file.

To execute this package, you can go to the Debug menu and select the Start Debugging button, press the F5 key, or right-click the package and choose Execute Package.

When the Data Flow is complete, all the components in the Data Flow change color from yellow to green. It means that they have all completed successfully. The last view will look like this:



Stop Debugging

1. Click the Stop Debugging button on the Debug toolbar.
2. Using Windows Explorer, navigate to the C:\Documents and Settings\\My Documents\Microsoft Press\IS2005SBS\Chap03\Data\ folder.
3. Open the Products.txt file to confirm data appears in the file.
4. Save the solution.

Chapter 3 Quick Reference

To	Do this
Create an Integration Services package	Start SQL Server Business Intelligence Development Studio. On the File menu, point to New, and then click Project. Make sure that the Project Type is set to Business Intelligence Projects, and then click the Integration Services Project template. Type a name for the project. Specify the location folder for the project and confirm that the Create Directory For Solution check box is selected. Click OK.
Add an OLE DB connection manager	Right-click anywhere in the Connection Managers pane at the bottom of the Control Flow tab and click New OLE DB Connection. Click New to define a new connection. Keep the default: Native OLE DB\SQL Server Native Client. Type localhost for the Server Name. Select Use Windows Authentication and choose the desired database.
Review available connection manager types	Right-click anywhere in the Connection Managers pane at the bottom of the Control Flow tab and explore the list of connections available: Flat File Connection, ADO.NET Connection, Analysis Services Connection, and so on.
Add an Excel connection manager	Right-click anywhere in the Connection Managers pane at the bottom of the Control Flow tab and click New Connection. In the SSIS connection manager, click EXCEL (connection manager for Excel files) and click Add. Type a name for the Excel file and specify an Office Excel file path.
Create a data flow task	Click the Data Flow tab. If you go to the Data Flow page right after creating a package, you will see a message stating that no data flow tasks have been added to the package. Click the message link to add a new task. You can also access it from the Control Flow page.
Add an OLE DB Source data adapter	Drag OLE DB Source from the Toolbox to the grid. The small red circle on this data adapter means that it needs a connection manager.
Add a connection manager to the OLE DB Source data adapter	Double-click the OLE DB Source data adapter to open the Editor and click OLE DB Connection Manager. In OLE DB Connection Manager, select a connection manager. In the Data Access Mode, select Table Or View and choose the desired table.
Map the connection manager to the data adapter	Click columns from the left panel of the OLE DB Source Editor. This action maps columns from the connection manager to output columns of the adapter.
Add an Excel Destination data adapter	Open the Toolbox and expand Data Flow Destinations. Drag Excel Destination from the Toolbox to the grid. The small red circle on this data adapter means that it needs a connection manager.

To	Do this
Add an Excel connection manager to the Excel Destination data adapter	Double-click the Excel Destination data adapter. This is a destination component that needs to be connected to the input source component. Click the OLE DB Source adapter and connect it to the Excel Destination adapter by dragging the green arrow from OLE DB Source to Excel Destination. Double-click the Excel Destination data adapter to open the Editor and verify that the connection manager is selected. In the Name text box of the Excel sheet, click New. Change the name of the sheet and change the size of long columns. The Excel connection manager will not allow creation of long columns. Finally, click Mapping in the left panel of the Editor.
Execute the package	Right-click the desired package and choose Execute Package.
Create a data source	In Solution Explorer, right-click the Data Sources folder, and then click New Data Source. On the Welcome To The Data Source Wizard page, click Next. On the Select How To Define The Connection page, verify that Create A Data Source Based On An Existing Or New Connection is selected, and then click New. Leave the Native OLE DB\SQL Native Client provider selected. Type localhost in the Server Name text box, select Use Windows Authentication, and select a database. Click OK, and then click Finish twice.
Create a data source view	In Solution Explorer, right-click the Data Source Views folder, and then click New Data Source View. Click Next on the Welcome To The Data Source View Wizard page. Select a data source and click Next. Select the objects you want to include in your data source view. Click next, and then click Finish.
Create a new Named Query	In Solution Explorer, expand the Data Source Views folder, and then open the data source view. In the Tables pane, right-click an open area, and then click New Named Query. Type a name for the new named query and specify a SQL statement in the bottom pane to define your named query. Click OK.
Add a connection manager from Data Source	In the Connection Managers pane, right-click an open area, and then click New Connection from Data Source. In Select Data Source, choose a data source that you created.
Set an OLE DB Source from a Named Query	In the designer, drag a Data Flow task from the Control Flow, open the Data Flow, and drag an OLE DB Source from the Data Flow Sources tab. Double-click the OLE DB Source component. Then, select and expand the data source you created from the OLE DB connection manager list. Select the data source view from the tree and click OK. In data access mode, select Named query and click OK.

Index

Symbols

.dtsx (DST package) files, 9

A

accessing package variables, 174
accessing SSIS design environment, 242–243
accumulated staging, 372
ActiveX Script Task
 creating a new package with single Active X Script task, 170–172
 implementing, 169–172
 implementing custom code using, 174
 Script Task and, 139
adding a new package, 380–381
adding a Sequence Container control flow item, 137
adding conditional split to find new customers, 378–379
adding configurations to a project, 297–298
adding connection manager for Fuzzy Lookup input, 138
adding data adapters, 43–49
 OLE DB source data adapters, 43–44
adding error-handling code in Script Task, 146–148
adding Excel connection manager, 65
adding Execute SQL task, 129, 138
adding flat file destination, 379–380, 383–384
adding Foreach Loop containers, 130–132, 138
 Sequence Container – File Exists, 130–132
adding Fuzzy Lookup transformation, 138
adding Lookup task process, 381–382
adding Script Tasks, 137
adding sequence containers, 121
adding SQL Server destinations, 127–129, 138
adding tables to find new dimension members, 376–377
adding variables, 137
additive measures, 353–354
Aggregate task
 adding, 399–400
applying a configuration, 301–306
 adding configuration to the ImportCustomers.dtsx, 302–303
 deleting existing package from SSMS, 302
 deploying a package, 304
 executing package, 306
 executing package using configuration files, 306
 inspecting alternate configuration files, 304–305
 running deployed packages, 305
 starting Package Installation Wizard, 304
applying precedence constraints, 132–136, 138
 constraint values, 132
applying security, 292–294

assigning a value to variables, 122–123
assigning reader/writer roles to a package, 293–294
asynchronous tasks
 defined, 437
asynchronous transformations, 73, 312
attach_databases.bat script, 4
Audit transformation, 72
Autos window, 194

B

basic error detection and handling, 237–239
 configuring transformation to fail when an error occurs, 239
 configuring transformation to ignore errors, 239
 configuring transformation to re-route error-causing records, 239
 data flow transformations, 239
 metadata lineage, 238
 precedence constraints, 238
 validation, 238
BI architects
 getting started, 2
BI solution goals, 346–348
 combining data from multiple sources, 347
 providing fast and easy access, 347–348
BI solutions
 choosing, 348
 data granularity, 349–350
BIDS, 13–26
 adding a Conditional Split transformation, 112
 adding a connection manager, 112
 adding a Derived Column transformation, 112
 adding a Destination data adapter, 112
 adding a lookup transformation, 112
 adding a Multicast transformation, 112
 adding conditional split transformation, 86–88
 adding connections manager, 82–85
 adding derived column transformation, 89–90
 adding flat file destination data adapter, 92–93
 adding multicast transformation, 95–96
 adding SQL server destination data adapter, 97–99
 creating a data adapter, 112
 creating a data flow task, 112
 creating a Flat File, 112
 creating a task, 112
 creating data adapter, 81–82
 creating Data Flow task, 78–80
 Data Flow designer, 69
 Data Flow Task – Employee List, 19–20

- defined, 32
 - docking utility windows, 16–17
 - editing an object, 112
 - executing CreateLists package, 24
 - executing package/checking results, 101, 110
 - executing the package, 94–95
 - exploring connection managers, 20–21
 - exploring SSIS projects in, 17–26
 - ImportCustomers package, 25–27
 - interface, 13
 - opening and exploring SSIS project, 76–77
 - opening the Derived Column – FullName task, 22
 - opening the Flat File Destination – Employees CSV task, 22–23
 - opening the OLE DB Source – Employee Query task, 21
 - opening the Sort – Department Shift Employee task, 22
 - opening the SSIS project, 112
 - opening/exploring package, 103
 - reviewing ImportCustomers package, 25
 - ScrapReason data source view in the Data Source view designer, 18–19
 - Solution Explorer, 15–16
 - viewing CreateLists package in the package designer, 19
 - viewing customer records files, 24–25
 - viewing final output, 23–24
 - viewing properties of derived column transformation, 91
 - BIDS (Business Intelligence Development Studio), 8–9
 - defined, 11
 - SQL Server Import and Export Settings Wizard, 8–9
 - Blocking transformations, 73, 313
 - testing packages with, 326–329
 - breakpoints, 176–181
 - setting, 177–181
 - setting and walk through code, 192–194
 - walk through code using, 190–194
 - Breakpoints window, 182
 - breakpoints, creating, 198, 199
 - buffer properties, working with, 323–324
 - buffer settings, 315–316
 - BufferTempStoragePath, 315
 - DefaultBufferMaxRows, 316
 - DefaultBufferSize, 316
 - buffer usage, 311–312
 - columns, 311
 - Data Pipeline engine, 311
 - row size, 311
 - BufferSize Tuning, 337
 - adding log event to, 337
 - BufferTempStoragePath, 315
 - business intelligence, 4
 - Business Intelligence Development Studio, 13–26
 - Data Flow Task – Employee List, 19–20
 - defined, 32
 - docking utility windows, 16–17
 - executing CreateLists package, 24
 - exploring connection managers, 20–21
 - exploring SSIS projects in, 17–26
 - ImportCustomers package, 25–26
 - interface, 13
 - opening the Derived Column – FullName task, 22
 - opening the Flat File Destination – Employees CSV task, 22–23
 - opening the OLE DB Source – Employee Query task, 21
 - opening the Sort – Department Shift Employee task, 22
 - reviewing ImportCustomers package, 25
 - ScrapReason data source view in the Data Source view designer, 18–19
 - Solution Explorer, 15–16
 - viewing CreateLists package in the package designer, 19
 - viewing customer records files, 24–25
 - viewing final output, 23–24
 - Business Intelligence Development Studio (BIDS), 8–9
 - defined, 11
 - SQL Server Import and Export Settings Wizard, 8–9
 - business intelligence solution goals, 346–348
 - combining data from multiple sources, 347
 - providing fast and easy access, 347–348
 - business intelligence solutions
 - choosing, 348
 - data granularity, 349–350
 - dimension, 349
 - level of grain, 349–350
 - supporting business decisions, 350
- ## C
- Call Stack window, 181
 - Catch . . . End Try statements, 146
 - CD-ROM
 - inserting, 3
 - overview, 3
 - changing data, 352
 - changing DelayValidation property, 137
 - changing dimensions, 352
 - Chart Format viewer, 186
 - checkpoints, 263–284
 - benefits of, 263
 - configuring packages for, 264
 - exercises, 264–284
 - implementing, 281–284
 - choosing business intelligence solutions, 348
 - chunked accumulated staging, 373
 - Clear DimCustomer, 177, 180, 185
 - setting breakpoints and, 178, 179
 - colors, SSIS Designer, 180
 - columns
 - compatible data types, 125
 - combining data from multiple sources, 347
 - combining validated source data, 344

- command-line utility, 227–228
- Conditional Split transformation, 73
- conditional split transformation, 85–88
- configuration failure, 210
- configuration files, 208–209
 - determining configuration order, 210
 - direct and indirect configurations, 209
 - environment variable, 208
 - evaluating configuration failure, 210
 - parent package variables, 209
 - registry entry, 209
 - SQL Server tables, 209
 - using multiple configurations, 210
- configuration files, creating
 - testing new package configuration, 218
- configuration files, creating multiple, 210–220
 - configuring environment variable, 213–215
 - creating a database environment, 212–213
 - creating database and OLE DB Connection Manager, 210–212
 - creating environment variable, 212–213
 - creating environment variable configuration, 213–215
 - creating Parent Package variable configuration, 219–220
 - creating ParentProductsDestination parent package, 219–220
 - creating SQL Server configuration, 215–217
 - creating SQL Server table, 215–217
 - exploring parent package, 218–219
 - testing new package configuration, 217, 220
 - viewing ParentPackage.dtsx package, 218–219
- configurations, 201–202
 - configuration benefits, 201–202
 - configuration types, 202
 - direct and indirect configuration, 202
 - environmental variable configuration, 202
 - parent package variable configuration, 202
 - registry entry, 202
 - specifying new XML configuration file location, 202
 - SQL Server table, 202
 - understanding XML configuration file, 202
 - XML, 202
- configuring data flow transformations to ignore errors, 239
- configuring left outer join merge join task, 377–378
- configuring variable settings, 156–157
- Connect To Server dialog box, 290
- connecting to flat file destination, 60
- connection manager, 35–50
 - adding data adapters, 43–49
 - adding Employee connection manager to Excel Destination data adapter, 47–49
 - adding Excel Destination data adapter, 46
 - adding localhost.is2005sbs Connection Manager to OLE DB Source data adapter, 44
 - adding from Data Source, 66
 - adding new, 39–41
 - adding Office Excel connection manager to the Employee.xls file, 40–41
 - adding OLE DB connection manager for is2005sbs database, 39–40
 - adding OLE DB source data adapter, 43–44
 - creating a data flow, 41–42
 - creating SQL Server Integration Services project, 36–38
 - executing the package, 49–50
 - mapping to data adapter, 45–46
 - reviewing available types, 65
 - types, 35–36
 - connection manager settings, 285
 - setting, 257–258
 - connection manager, adding, 82–85
 - connection managers
 - exploring, 20–21
 - ConnectionString, 131, 138
 - ConnectionString property, 35
- containers
 - defined, 10
- containers, 119–125
 - adding sequence containers, 121
 - assigning a value to the variable, 122–123
 - creating a placeholder, 123–125
 - For Loop container, 119
 - Foreach Loop container, 119
 - Sequence container, 120
 - Task Host container, 120
 - testing whether output file is found, 122
- Continue command, 191
- Control Flow
 - creating breakpoints, 198
 - defined, 10
 - implementing custom code in, 173
 - reviewing variable status, 198
 - SSIS, 6
- control flow components, 114–119
 - DelayValidation, 115
 - list of, 114
- Control Flow tab, 29
- control flow, debugging, 175–185
- conventions in book, 4
- copying data from named query to flat file, 57
- Create Table dialog box, 98
- CreateLists package, 24
- CreateLists package in the package designer, 19
- creating a data source, 50
- creating a database environment, 212–213
- creating data flow, 41–42
- creating Data Flow Task, 65
- creating data source views, 52
- creating environment variable configuration, 213–215
- creating environment variables, 212–213
- creating Integration Services package, 65
- creating named queries, 54

- creating new databases, 211–212
- creating new project for dimension table load packages, 375–376
- creating new Script task and initiating code, 141–145
- creating Parent Package variable configuration, 219–220
 - creating ParentProductsDestination parent package, 219–220
 - testing package with new configuraton, 220
- creating placeholders, 123–125
- creating SQL Server configuration, 215
 - creating SQL Server table, 215–217
- creating XML configuration file, 204–207
- CurrentRecord field, 390, 391
 - initializing for SCD reference, 391–392
- custom code
 - implementing using ActiveX Script Task, 174
- CustomSSISLog table, 246
 - Lookup Names, 254
 - Lookup Names task, 262
- CustomSSISLogKey, 254

D

- data adapters
 - adding, 43–49
 - adding OLE DB source data adapters, 43–44
 - Excel Destination data adapters, 46
 - mapping connection manager to, 45–46
- data blocking, 313–314
 - blocking transformations, 313
 - partially blocking transformations, 313–314
 - row transformations, 314
- data buffer settings, 315–316
 - BufferTempStoragePath, 315
 - DefaultBufferMaxRows, 316
 - DefaultBufferSize, 316
- Data Conversion task
 - adding, 401–403
- data destination management, 321–322
 - dropping/disabling indexes, 321
 - locking modes, 321
 - using explicit transactions, 321
- data destinations, 6
- data filters, 320
- data flow
 - creating in a package, 67–69
 - data source connections, 68–69
 - defined, 10
 - destinations, 68
 - implementing custom data processing code inside, 173
 - reviewing, 198
 - sources, 67
 - SSIS, 6
 - transformations, 68

- Data Flow designer, 69, 80, 81, 82, 86, 89, 90, 91, 92, 95, 97, 104, 105, 106, 112
- data flow engines, 6
- Data Flow tab, 29
- Data Flow Task – Employee List, 19–20
- Data Flow Task – Lookup Geography task, 104
- Data Flow Task, creating, 65
- data flow transformations, 75–101, 239
 - adding conditional split transformation, 85–88
 - adding connection manager, 82–85
 - adding derived column transformation, 88–90
 - adding flat file destination data adapter, 92–93
 - adding Multicast transformation, 95–96
 - adding SQL Server destination data adapter, 97–99
 - configuring Flat File source, 81
 - configuring transformations to fail when an error occurs, 239
 - configuring transformations to ignore errors, 239
 - configuring transformations to re-route error-causing records, 239
 - creating data adapter, 81–82
 - creating data flow task, 78–80
 - deleting data from table, 99–101
 - executing the package, 94–95, 101
 - opening and exploring the SSIS project, 76–77
 - previewing NewProducts.txt file, 77–78
 - sending output to different destinations, 95–101
 - using Flat File source, 80–82
- data flow, creating, 41–42
- data flow, debugging, 186–190
 - data viewers, 186–190
 - Error Output, 190
 - Progress Messages, 190
 - Row Count, 190
 - setting data viewers and browsing data, 187–190
- data granularity, 349–350
 - dimension, 349
 - level of grain, 349–350
 - supporting business decisions, 350
- data marts
 - characteristics, 342–345
 - combining validated source data, 344
 - fundamentals, 346
 - integrating data from heterogeneous source systems, 344
 - organizing data into nonvolatile, subject-specific groups, 345
 - overview, 341–342
 - providing data for business analysis processes, 343
 - storing data in structures optimized for extraction and queries, 345
- Data Pipeline
 - defined, 11
- data pipeline, 6–7
- Data Pipeline engine, 310, 311

- data quality transformations, 71–72
 - fuzzy grouping, 72
 - fuzzy lookup, 72
- data source connections, 68–69
- data source tuning, 316–317
- data source views, creating, 52, 66
- Data Source Wizard, 50, 66
- data sources, 6, 50–64, 315
 - connecting to flat file destination, 60
 - copying data from named query to flat file, 57
 - creating, 50, 66
 - creating data source view, 52
 - creating named query, 54
 - executing the package, 63–64
 - stop debugging, 64
- data transformation, 6
- data viewers, 186–190
 - setting data viewers and browsing data, 187–190
- data warehouse
 - defined, 366
- data warehouses
 - adding relationships to database, 363
 - additive measures, 353–354
 - BI solution goals, 346–348
 - changing dimensions, 352
 - characteristics, 342–345, 368
 - combining data from multiple sources, 347
 - combining validated source data, 344
 - compared to operational databases, 342
 - components, 359–360
 - creating diagram, 355–358
 - creating new diagram for is2005sbsDW database, 363–364
 - data granularity, 349–350
 - database diagram, 343
 - database diagramming, 354–358
 - defined, 411
 - dimension table characteristics, 360–362
 - fact and dimension tables, 359–360
 - fundamentals, 346
 - historical data, 351–352
 - integrating data from heterogeneous source systems, 344
 - organizing data into nonvolatile, subject-specific groups, 345
 - overview, 341–342
 - providing data for business analysis processes, 343
 - providing fast and easy access, 347–348
 - querying is2005sbsDW database, 364–365
 - relational, 345
 - resetting relationships for is2005sbsDW database, 365
 - reviewing and comparing schema, 362–365
 - saving diagrams, 364
 - snowflake schema dimensions, 362
 - star schema dimensions, 360–361
 - storing data in structures optimized for extraction and queries, 345
 - summary, 365–366
 - surrogate keys, 352–353
 - update frequency and persistence, 350–352
- database administrators
 - getting started, 2
- database diagramming, 354–358
 - creating database diagrams, 355–358
 - is2005sbs database, 355–356
 - organizing the diagram, 356–357
 - querying is2005sbs database, 358
 - saving the diagram, 358
- database environment
 - creating, 212–213
- database schema, reviewing and comparing, 362–365
 - adding relationships to database, 363
 - creating database diagrams, 363–365
 - creating new diagram for is2005sbsDW database, 363–364
 - querying is2005sbsDW database, 364–365
 - resetting relationships for is2005sbsDW database, 365
 - saving the diagram, 364
- databases
 - creating new, 211–212
- data-mining transformations, 72
 - data-mining query, 72
 - term extraction, 72
 - term lookup, 72
- DataReader, 6
- DataTip tool, 195
- db_dtsadmin, 292
- db_dtsltduser, 292
- db_dtsoperator, 292
- Debug Menu, 63
- Debug Windows, reviewing, 181–183
 - Breakpoints, 182
 - Call Stack, 181
 - executing package and, 183
 - Locals, 182
- debugging control flow, 175–185
- debugging data flow, 186–190
 - data viewers, 186–190
 - Error Output, 190
 - Progress Messages, 190
 - Row Count, 190
 - setting data viewers and browsing data, 187–190
- debugging Script component, 198
- debugging Script tasks, 190–197
 - Continue command, 191
 - reviewing state using VSA, 194–197
 - Run To Cursor command, 191
 - Step Into command, 191
 - Step Out command, 191
 - Step Over command, 191
 - walk through code using breakpoints, 190–194
- DefaultBufferMaxRows, 316, 325

- changing, 337
- DefaultBufferSize, 316, 325
 - changing, 337
- DelayValidation property, 115
 - changing, 137
 - defined, 115
 - False value, 115
 - True value, 115
 - working with tasks and, 115–119
- deployment options, 294–297
- deployment utility, creating, 288–290
 - enabling deployment utility, 288–289
 - Package Installation Wizard, 288–290
- DeploymentOutputPath, 288, 289, 293
- Derived Column – FullName task, 22
- Derived Column task
 - adding, 400–401
- Derived Column transformation, 74
 - adding, 88–90
 - viewing properties, 91
- Derived Column Transformation Editor, 90
- design considerations, 326–329
- destination considerations, 373
- determining configuration order, 210
- dimension tables, 359–360
 - adding conditional split to find new customers, 378–379
 - adding flat file destination, 379–380, 383–384
 - adding Lookup task process, 381–382
 - adding new package, 380–381
 - adding tables to find new dimension members, 376–377
 - characteristics, 360–362, 368
 - configuring a left outer join merge join task, 377–378
 - creating new project for dimension table load packages, 375–376
 - defined, 366, 411
 - hierarchy levels, 360
 - loading, 380–384
 - loading dimension tables using left outer join, 375–380
 - managing, 374–384
 - snowflake schema dimensions, 362
 - star schema dimensions, 360–361
- direct and indirect configurations, 202, 209
- DontSaveSensitive, 291
- dropping/disabling indexes, 321
- DTEXec, 222–223, 234
- Dtexec.exe, 9
- DTEXecUI, 222, 234
- DTS service
 - LoadDim Prod package, 274
- Dts.Events object, 148, 173
- Dts.Events.FireError, 149
 - event message with verbose error message, 150
 - parameters required, 149
- Dts.Events.FireInformation method, 151, 152
- Dts.Events.FireProgress method, 152

- Dts.Log method, 152, 154, 155, 173
- Dts.TaskResult = Dts.Results.Success, 142, 143
- DTSGlobalVariables.Parent, 169
 - alternatives, 169
 - migration, 169
- Dtutil.exe, 9

E

- editing XML configuration file, 207–208
- Employee connection manager, 47–49
- EncryptAllWithPassword, 291, 292–293
- EncryptAllWithUserKey, 291
- encryption, 290–291
- EncryptSensitiveWithPassword, 291
- EncryptSensitiveWithUserKey, 291
- EntryMethod property, 172, 174
- environment variable, 208
 - configuring, 234
 - creating, 234
- environment variable configuration
 - creating, 213–215
- environment variables
 - creating, 212–213
- environment variables, configuration, 202
- error
 - no index content, 1
- error count properties
 - changing, 260–261
- Error Output, 190
- error output, configuring, 102–111
 - adding data conversion transformation, 105–106
 - adding Flat File Destination for Lookup Errors, 108–109
 - adding Flat File Destination for successful lookups, 109–110
 - adding Lookup transformation, 106–108
 - creating a task, 104
 - creating/naming Flat File Source, 104–105
 - error options, 102
 - executing package/checking results, 110–111
 - exploring LookupGeography package, 102–103
 - fail component, 102
 - ignore failure, 102
 - opening/exploring package, 103
 - redirect row, 102
 - types of errors, 102
- error-handling code
 - implementing in script, 173
- ETL solutions
 - getting started, 2
- evaluating configuration failure, 210
- Event handler
 - defined, 10
- event handler process control, 7–8
- event handlers, 239–262
 - accessing SSIS design environment, 242–243
 - adding a task, 244

- adding Execute SQL task, 244
 - adding Log Finish Execute SQL task to OnPostExecute event handler, 248-249
 - adding tasks to, 285
 - changing error count properties, 260-261
 - configuring the task, 245-247
 - connection manager settings, 257-258
 - creating, 241-262
 - creating a log to finish, 248-249
 - creating log error event handler, 250-251
 - creating task to move files with invalid data, 256-257
 - data flow transformations, 240
 - event handlers provided by SSIS, 241
 - Execute SQL task, 245-247
 - executing NewProducts.dtsx package and viewing results in Management Studio, 251-253
 - executing the package, 251-253, 261-262
 - mapping SSIS variables to SQL statement parameters, 247-248
 - OnError, 241
 - OnPostExecute, 241
 - OnPreExecute, 241, 243-244
 - OnTaskFailed, 241, 255
 - OnWarning, 241
 - preventing events from escalating to containers and packages, 259
 - QuickStartODS database, 242
 - tasks, 240
 - testing package with invalid data, 253-254
 - triggering, 240
 - Excel connection manager
 - adding to the Employee.xls file, 40-41
 - Excel connection manager, adding, 65, 66
 - Excel Destination data adapters, 46
 - adding Employee connection manager, 47-49
 - adding Excel connection manager, 66
 - Excel Destination data adapters, adding, 65
 - Execute Package, 32, 49-50, 66
 - execute package and review debug windows, 183
 - Execute Package task, 221-229
 - command-line utility, 227-228
 - DTEExec, 222-223
 - DTEExecUI, 222
 - executing tasks and containers and then disabling task and executing package, 224-226
 - extending package execution options, 224
 - SQL Agent, 228-229
 - SQL Server Agent, 224-226
 - SQL Server Import and Export Wizard, 221-222
 - SQL Server Management Studio, 223
 - starting Server Wizard from BIDS, 221
 - starting Server Wizard from Management Studio, 221-222
 - using Execute Package Utility, 226-229
 - Execute Package utility, 226-229
 - command-line utility, 227-228
 - SQL Agent, 228-229
 - Execute Process task, 224
 - Execute SQL task
 - adding, 138
 - adding to OnPreExecute event handler, 244
 - configuring, 245-247
 - configuring to run parameterized SQL statements, 247-248
 - Execute SQL tasks
 - adding, 129
 - execution plans, 331-333
 - execution trees, 312, 330
 - explicit transactions, 321
 - Export Column transformation, 72
 - expressions
 - building, 75
 - elements, 74-75
 - using in SSIS, 73-74
 - expressions, using in packages, 73-75
 - extract, transform, and load (ETL) process, 115
- ## F
- fact tables, 359-360
 - adding Data Conversion task, 401-403
 - adding Derived Column task, 400-401
 - adding Multicast task and Aggregate task, 399-400
 - adding new data flow to load fact table from staging table, 403
 - adding new package for fact table load processing, 398-399
 - adding OLE DB destination and configuring Error Output, 405-410
 - adding variable and Row Count task, 403-404
 - aggregating data in, 398
 - characteristics, 368
 - defined, 366, 411
 - loading, 398-410
 - managing, 397-410
 - transaction detail values, 368
 - Fail Component, 102
 - FailPackageOnFailure property, 264
 - FastLoad
 - enabling for OLE DB destination, 337
 - FastLoad option, 321
 - FastParse, 319
 - setting on Flat File Source, 337
 - features of book, 4
 - file system task
 - creating, 285
 - FileFound variable, 123
 - Finally block
 - StreamWriter objects and, 146
 - Flat File Connection Manager Editor, 83, 85, 93, 102, 103, 108, 109, 110
 - Flat File Destination
 - Lookup Errors, 108-109

- successful lookups, 109–110
- Flat File Destination – Employees CSV task, 22–23
- Flat File destination data adapter, 92–93
- Flat File Destination Editor, 93
 - creating and naming, 104–105
- Flat File source, 80–82
 - configuring, 81
- Execute package
 - creating data adapter, 81–82
 - delimited format text files, 80
 - fixed-width format text files, 80
 - ragged-right format text files, 80
- Flat File Source Editor, 82, 85, 104, 105
- Flat File Sources, 319
- flat files
 - connecting to flat file destinations, 60
 - copying data from named query to, 57
- For Loop container, 119
- Foreach container
 - adding to Sequence container, 138
 - preventing Lookup Names task from escalating to, 259
- Foreach containers
 - adding to Sequence Container – File Exists, 132
- Foreach Loop container, 119, 176, 177, 179
- Foreach Loop containers, 130–132
 - adding to Sequence Container – File Exists, 130
- foreign keys
 - defined, 366, 411
- Fuzzy Lookup
 - column width and, 125
- Fuzzy Lookup input
 - adding connection manager for, 138
- Fuzzy Lookup transformation
 - adding, 125–127, 138
 - adding an Execute SQL task, 129
 - adding SQL Server destination, 127–129
- columns, 125
- indexes and, 126
- reference tables and, 125

G

- GeographyKey column, 105
- GetEmployeeData, 192, 193, 197
- getting started, 1–2
 - BI architects, 2
 - building ETL solution, 2
 - database administrators, 2
 - importing/exporting data, 2
 - solution designers, 2
 - system administrators, 2
- GetVariables, 123
- Grid viewer, 186, 188, 189, 190
 - as first option, 187
 - illustration of, 186

H

- hierarchy levels, 360
- Histogram viewer, 186
- historical data, 351–352
- Hit Count options, 177
- Hit Count Type, 177, 179, 183, 198

I

- IDE (Integrated Development Environment), 142
- Ignore Failure, 102
- Immediate window, 195
- implementing ActiveX Script Task, 169–172
- implementing custom code in Control Flow, 173
- implementing custom code using ActiveX Script Task, 174
- implementing custom data processing code inside Data Flow, 173
 - implementing error-handling code in the script, 173
- Import Column transformation, 72
- ImportCustomers package
 - changing package protection level, 292
 - executing, 25–26
 - exporting deployed packages, 296
 - importing to file system, 295
 - reviewing, 25
 - role-based security, 293
 - running deployed packages, 305
- ImportCustomers.dtsx, 302
 - adding configurations, 302–303
 - changing package protection level, 293
 - deleting packages from SSMS, 302
 - executing file system package, 296
- importing/exporting data
 - getting started, 2
- indexes
 - dropping/disabling, 321
 - Fuzzy Lookup transformation and, 126
- Input Output Selection dialog box, 89, 90
- InputBuffer, 161–162
- InputFile, 131
- installing sample files, 3–4
- Integrated Development Environment (IDE), 142
- integrating data from heterogeneous source systems, 344
- Integration Services Object Model, 9
- Integration Services package, creating, 65
- Integration Services, connecting to, 290
- invalid data
 - creating task to move files, 256–257
 - testing package with, 253–254
- is2005sbs database
 - adding OLE DB connection manager, 39–40
 - reviewing with Management Studio, 27
- is2005sbsDW database
 - creating new diagram for, 355–356, 363–364
 - querying, 358, 364–365

- resetting relationships for, 365, 366
- iterative design optimization, 333–336
- logging execution plans, 334–336

J

- join
 - defined, 366, 411

L

- LoadDim Prod package
 - deleting DimProd, 268
 - deleting staging tables, 268
 - DTS service, 274
 - loading DimProd, 268
 - loading staging tables, 268
- LoadDimProd package, 267–278
- loading dimension tables using left outer join, 375–380
 - adding conditional split to find new customers, 378–379
 - adding flat file destination, 379–380
 - adding tables to find new dimension members, 376–377
 - configuring left outer join merge join task, 377–378
 - creating new project for dimension table load packages, 375–376
- localhost.is2005sbs Connection Manager, 44
- Locals window, 182
- LockForRead, 123
- LockForWrite, 123
- locking modes, 321
- Log Error Execute SQL task
 - adding to OnError event handler, 250–251
- log events
 - adding to, 337
- Log Files
 - providing verbose information to, 152–156
- Lookup Names, 254
- Lookup Names task
 - event records, 262
 - preventing from escalating to Foreach Loop container, 259
- Lookup transformation, 106–108
- LookupGeography Package, 102–103
- loops, 318–319

M

- Management Studio
 - executing NewProducts.dtsx package and viewing results in, 251–253
- mapping connection manager to data adapter, 65
- master-child packages
 - defined, 437
- MatchedNames table, 127, 128, 129
- MaxConcurrentExecutables property, 316
- maximum data processing, 7
- MaximumErrorCount property, 260

- Me.ComponentMetadata, 198
- Me.ComponentMetadata object, 162
- Me.Log, 198
- Me.Variables object, 162
- memory buffer architecture, 310–312
 - buffer usage, 311–312
 - columns, 311
 - Data Pipeline engine, 311
 - row size, 311
- metadata lineage, 238
- modifying script to read variables, 158–159
- MSDB database roles, 292
 - db_dtsadmin, 292
 - db_dtslduser, 292
 - db_dtsoperator, 292
- Multicast task
 - adding, 399–400

N

- named queries, creating, 54, 66
- NewProducts.dtsx package
 - executing, 251–253
 - invalid data, 253–254
- non-blocking transformations, 314–315
 - data sources, 315

O

- OLE DB connection manager, adding, 65
- OLE DB destinations
 - enabling Fast Load, 337
- OLE DB Source
 - setting from Named Query, 66
- OLE DB Source – Employee Query task, 21
- OLE DB Source data adapters, adding
 - quick reference, 65
- OLE DB source data adapters, adding, 43–44
 - adding localhost.is2005sbs Connection Manager, 44
- OnCustomEvent, 176
- OnError, 176, 177
- OnError event handler, 241
 - adding Log Error Execute SQL task, 250–251
- OnInformation, 176
- OnPostEvent event handler
 - adding Log Finish Execute SQL task, 248–249
- OnPostExecute, 176–177
- OnPostExecute event handler, 241
- OnPreExecute, 176, 177, 178, 179, 184
- OnPreExecute event handler, 241
 - adding Execute SQL task to, 244
 - creating, 243–244, 285
- OnProgress, 176
- OnQueryCancel, 176
- OnTaskFailed, 176

- OnTaskFailed event handler, 241
 - executable lookup names, 255
 - preventing from escalating from Lookup Names task, 259
- OnVariableValueChanged, 176
- OnWarning, 176
- OnWarning event handler, 241
- Open Database Connectivity (ODBC), 68
- operational databases
 - compared to data warehouses, 342
- organizing data into nonvolatile, subject-specific groups, 345
- Output window, 196
- OutputBuffer, 161–162
- OVAL
 - defined, 437

P

- package configuration
 - creating and applying, 297–298
 - enabling/disabling configurations, 205
 - using SSIS BIDS menu to add a configuration, 297–298
- Package Configuration Organizer, 209
 - determining configuration order, 210
 - environment variables, 208
- package configurations, 201–202
 - configuration benefits, 201–202
 - configuration types, 202
 - direct and indirect configuration, 202
 - environmental variable configuration, 202
 - parent package variable configuration, 202
 - registry entry configuration, 202
 - specifying new XML configuration file location, 202
 - SQL Server table configuration, 202
 - understanding XML configuration file, 202
 - XML configuration, 202
- package control flow architecture, 114
- package execution options, 221–229
 - command-line utility, 227–228
 - DTEExec, 222–223
 - DTEExecUI, 222
 - Execute Package utility, 226–229
 - extending, 224
 - SQL Agent, 228–229
 - SQL Server Agent, 224–226
 - SQL Server Import and Export Wizard, 221–222
 - SQL Server Management Studio, 223
 - starting Server Wizard from BIDS, 221
 - starting Server Wizard from Management Studio, 221–222
- package execution, monitoring, 300–306
 - adding configuration to ImportCustomers.dtsx package, 302–303
 - deleting existing package from SSMS, 302
 - deploying package, 304
 - executing by using configuration files, 306
 - executing package, 306

- inspecting alternate configuration files, 304–305
- Package Installation Wizard, 304
 - running deployed package, 305
- Package Installation Wizard
 - building SSIS sample project, 289–290
 - deploying packages, 287, 288
 - enabling deployment utility, 288–289
 - monitoring package installation, 304
 - password options, 293
 - push deployment and, 294
 - quick reference, 307
- package logging
 - configuration, 230–231, 234
 - configuring container and task logging, 232–233
 - executing package and viewing logs, 232–233
 - implementing, 230–234
 - understanding, 229–230
- package variables
 - accessing, 174
- PackagePassword property, 291, 292, 307
- packages
 - data flow task, 30
 - executing partially, 185, 198
 - executing QuickStartIS.dtsc package, 31
 - preparation SQL task, 30
 - reviewing elements, 29–30
 - reviewing QuickStartODS database tables using SSMS, 31
 - stop debugging and close package designer, 31
 - testing, 31
- packages, deploying, 294–297, 298–299
 - DTEExec, 298–299
 - DTEExecUI, 298–299
 - executing file system packages, 296
 - exporting deployed packages, 296
 - importing ImportCustomers package, 295
 - importing to MSDB, 296
 - managing packages on SSIS server, 295–297
 - monitoring running packages, 296–297
 - pull deployment, 295
 - push deployment, 294
- packages, securing, 290–291
- packages, using expressions in, 73–75
 - For Loop container, 74
 - precedence constraints, 74
 - system variables, 74
 - user variables, 74
- parallelism, 316
- parent package
 - exploring, 218–219
 - viewing ParentPackage.dtsx package, 218–219
- Parent Package variable configuration, 219–220
 - creating ParentProductsDestination parent package, 219–220
 - testing package with new configuraton, 220
- parent package variable configuration
 - creating, 234

- parent package variable, configuration, 202
- parent package variables, 209
- ParentPackage.dtsx package, 218–219
- ParentProductsDestination parent package, 219–220
- partially blocking transformations, 73, 313–314
- password protection, 291
- Percentage Sampling transformation, 387–388
- performance management
 - data destination management, 321–322
 - data filters, 320
 - design considerations, 326–329
 - dropping/disabling indexes, 321
 - Flat File Sources, 319
 - locking modes, 321
 - loops, 318–319
 - performance-tuning exercises, 322–323
 - testing DefaultBufferMaxRows and DefaultBufferSize, 325
 - using explicit transactions, 321
 - using SQL Server connection manager, 325
 - variable scope, 320–321
 - variables, 320–321
 - Variables collection, 320
 - working with buffer properties, 323–324
 - working with SQL Server destination, 324–325
- persisted staging, 371
- PipelineExecutionPlan event, 331, 336
 - Log Events window, 336
- PipelineExecutionTree event, 330
- placeholders, creating, 123–125
- PrdCode, 85
- PrdName, 85
- Precedence Constraint objects, 5
- Precedence constraints
 - applying, 138
 - defined, 10
- precedence constraints, 132–136, 238
 - constraint values, 132
- PrepLoadDimProd.dtsx package, 264–267
- Progress messages, 183–184, 190
- Progress tab
 - providing messages to, 148–152, 173
- project
 - defined, 32
- Project Property Pages dialog box, 289
- projects
 - executing, 234
- ProtectionLevel, 291, 292, 307
- ProtectionLevel property, 291
 - DontSaveSensitive, 291
 - EncryptAllWithPassword, 291
 - EncryptAllWithUserKey, 291
 - EncryptSensitiveWithPassword, 291
 - EncryptSensitiveWithUserKey, 291
 - ServerStorage, 291

- providing fast and easy access to data, 347–348
- providing messages to Progress tab, 148–152, 173
- providing verbose information to Log File, 152–156
- pull deployment, 295
- push deployment, 294

Q

- Query Designer toolbar, 306
- QuickStartIS SSIS project
 - creating QuickStart solution to contain, 28
- QuickStartODS database
 - creating, 242
 - creating the database, 242
 - determining whether QuickStartODS database is on your computer, 242
 - CustomSSISLog table, 246
 - importing tables into, 28–29
- QuickWatch dialog box, 194

R

- Redirect Row, 102
- Regex.IsMatch method, 167, 174
- registry entry, 209
- registry entry, configuration, 202
- Regular Expressions, 167, 147
- Relational Data Sources list, 52
- relational data warehouses, 345
- reviewing available connection manager types, 65
- reviewing Debug Windows, 181–183
 - Breakpoints, 182
 - Call Stack, 181
 - executing package and, 183
 - Locals, 182
- reviewing state using debug windows
 - step by step, 196–197
- reviewing state using VSA features, 194–197
 - Autos window, 194
 - DataTip tool, 195
 - Immediate window, 195
 - Output window, 196
 - QuickWatch dialog box, 194
 - Watch window, 194
- role-based security, 291–294
 - applying, 292–294
 - assigning reader/writer roles, 293–294
 - changing protection level, 292–293
- Row Count, 190
- Row Count transformation, 72
- row transformations, 70, 314
 - character map, 70
 - copy column, 70
 - data conversion, 70
 - derived column, 70
 - OLE DB command, 70

- lookup, 71
- merge, 71
- merge join, 71
- multicast, 71
- union all, 71
- SQL Agent, 228–229
- SQL Destination Editor, 98, 99
- SQL Server Agent, 224–226
- SQL Server configuration
 - creating, 215, 234
 - creating SQL Server table, 215–217
- SQL Server destinations, 324–325
 - adding, 127–129, 138
 - testing DefaultBufferMaxRows and DefaultBufferSize, 325
 - using SQL Server connection manager, 325
- SQL Server Import and Export Settings Wizard, 8
- SQL Server Import and Export Wizard, 221–222
 - starting Server Wizard from BIDS, 221
 - starting Server Wizard from Management Studio, 221–222
- SQL Server Integration Services project, 36–38
- SQL Server Management Studio (SSMS), 9, 223
 - defined, 11
- SQL Server table, 215–217
- SQL Server tables, 209
- SQL Server tables, configuration, 202
- SQL statement parameters, mapping SSIS variables to, 247–248
- SSIS
 - .dtsx (DST package) files, 9
 - BIDS, defined, 11
 - common applications, 4
 - components, 8–9
 - container types, 119–120
 - containers, defined, 10
 - control components, 114
 - control flow, 6
 - control flow elements, 114
 - control flow, defined, 10
 - data flow, 6
 - data flow, defined, 10
 - data pipeline, 6–7
 - data pipeline, defined, 11
 - Dtexec.exe, 9
 - Dtutil.exe, 9
 - event handler, 7–8
 - event handler, defined, 10
 - Integration Services Object Model, 9
 - Microsoft products and, 4
 - objects and process control components, 4–5
 - Package Migration Wizard, 10
 - packages, defined, 10
 - precedence constraints, defined, 10
 - process control, 5–8
 - SQL Server 2000 DTS migration, 10
 - SQL Server environments and, 4
 - SSMS and, 9
 - SSMS, defined, 11
 - tasks, defined, 10
 - validation, 115
 - variables, defined, 10
- SSIS design environment, 242–243
- SSIS Designer, 188, 189, 190
 - breakpoints, 186
 - breakpoints and, 176
 - color coding, 180
 - Data Flow Task, 180
 - Hit Count type, 177
 - package execution, 183, 185
 - viewer types, 186–187
- SSIS Designer toolbox, 114
- SSIS engines, 310
 - Data Pipeline engine, 310
 - Runtime engine, 310
- SSIS Import and Export Wizard, 26–29
 - creating tables in new database, 26–29
 - creating QuickStart solution to contain QuickStartIS SSIS project, 28
 - defined, 32
 - importing tables into new QuickStartODS database with a new package, 28–29
 - reviewing is2005sbs database using Management Studio, 27
 - running wizard in BIDS, 27
- SSIS log reports, 337
- SSIS package
 - defined, 10, 32
- SSIS package design, 414–436
 - adding Data Flow tasks to child packages, 426
 - adding Execute Package tasks, 424–426
 - adding Row Count task and variable, 426–429
 - creating master–child package, 424–430
 - database snapshots, 420
 - designing for performance and maintenance, 420–422
 - defining best practices, 423
 - defining project folders, 433
 - designing for deployment, 434–435
 - disabling Execute Package task, 429–430
 - fast parse, 423
 - logging reports, 436
 - Lookup task vs. Merge Join, 419–420
 - managing buffers and memory, 434
 - managing CPU use, 434
 - managing multiple schemas, 435
 - managing performance and debugging, 433
 - managing SSIS application deployment, 434–436
 - opening project/building packages, 424
 - organizing package components, 430–434
 - OVAL principles, 414–417
 - using prefixes to identify package components, 430–433
 - using SSIS components, 417–423

- using SSIS package configurations, 435
 - variables, 418–419
 - SSIS packages
 - storage, 14
 - SSIS project
 - creating new, 203–204, 234
 - SSIS Sample Solution.sln, 288, 302
 - SSIS transformations, 69–73
 - asynchronous transformations, 73
 - audit, 72
 - blocking transformations, 73
 - data quality transformations, 71–72
 - data-mining query, 72
 - data-mining transformations, 72
 - export column, 72
 - fuzzy grouping, 72
 - fuzzy lookup, 72
 - import column, 72
 - memory buffers, 69
 - partially blocking transformations, 73
 - row count, 72
 - row transformations, 70
 - rowset transformations, 70–71
 - slowly changing dimension, 72
 - split and join transformations, 71
 - synchronous transformations, 73
 - term extraction, 72
 - term lookup, 72
 - SSIS variables
 - mapping to SQL statement parameter, 285
 - SSIS variables, mapping to SQL statement parameters, 247–248
 - SSISDeploymentManifest file
 - build process and, 289
 - building SSIS sample project, 289
 - changing security levels, 293
 - creating deployment utility, 288
 - monitoring package installation, 304
 - Package Installation Wizard and, 307
 - push deployment and, 294
 - SSMS
 - defined, 11
 - staggered staging, 371
 - staging data from multiple sources, 370
 - staging schemes, 370–373
 - accumulated staging, 372
 - chunked accumulated staging, 373
 - destination considerations, 373
 - persisted staging, 371
 - staggered staging, 371
 - staging data from multiple sources, 370
 - staging tables
 - implementing, 369–370
 - uses, 369
 - star schema
 - defined, 366
 - star schema dimensions, 360–361
 - star schema model
 - characteristics, 368
 - defined, 411
 - dummy record members, 375
 - Step Into command, 191
 - Step Out command, 191
 - Step Over command, 191
 - Stop Debugging, 64
 - storing data in structures optimized for extraction and queries, 345
 - streaming transformation, 312
 - StreamWriter objects
 - Finally block and, 146
 - If statements and, 147
 - Try block and, 147
 - string data
 - validating, 174
 - surrogate keys, 352–353, 374
 - characteristics, 368
 - defined, 366, 411
 - defining dimension member rows, 374
 - uses, 374
 - synchronous and asynchronous processing, 312–313
 - synchronous tasks
 - defined, 437
 - synchronous transformation, 312
 - Synchronous transformations, 73
 - system administrators
 - getting started, 2
 - system requirements, 3
 - System.IO, 123
- ## T
- Task Host container, 120
 - Tasks
 - defined, 10
 - tasks
 - color coding, 31
 - defined, 114
 - testing new package configuration, 208, 217–218, 220
 - testing whether output file is found, 122
 - thumbtack icon, 17
 - TransactionOption property
 - Not Supported, 263
 - Required, 263
 - Supported, 263
 - transactions, 263
 - configuring, 263
 - exercises, 264–284
 - TransactionOption property, 263
 - transformations
 - asynchronous, 312
 - blocking, 313

- data sources, 315
- non-blocking, 314–315
- partially blocking, 313–314
- row, 314
- Sort transformation, 326
- streaming, 312
- synchronous, 312
 - testing packages, 326–329
- Try . . . Catch . . . Finally statements, 145–146
 - Catch block, 145, 146
 - Finally block, 145–146
 - StreamWriter object, 146
 - Try block, 145

U

- understanding package configurations, 201–202
- understanding XML configuration file, 202
- update frequency and persistence, 350–352
 - historical data, 351–352
- using containers, 119–125
 - adding sequence containers, 121
 - assigning a value to variables, 122–123
 - For Loop container, 119
 - Foreach Loop container, 119
 - Sequence container, 120
 - Task Host container, 120
 - testing whether output file is found, 122
- using explicit transactions, 321
- using multiple configurations, 210
- using sample files, 3–4
- using variables, 156–159
- utility windows, 16–17

V

- validating string data, 174
- validation, 238
- VariableDispenser, 123
- VariableDispensers, 123
- Variables
 - defined, 10
- variables, 156–159, 320–321
 - adding, 137
 - assigning a value to, 122–123
 - configuring settings, 156–157
 - modifying at run time, 159–160
 - modifying script to read, 158–159
 - variable scope, 320–321
 - Variables collection, 320
- Variables collection, 320
- verbose information
 - providing to Log File, 152–156
 - writing out to log file, 173
- viewing ParentPackage.dtsx package, 218–219
- Visual SourceSafe

- defined, 32
- VSA Code Editor, 190, 191, 192, 193, 196, 197
 - Continue command, 191
 - reviewing state by using, 194–197
 - Run To Cursor command, 191
 - Step Into command, 191
 - Step Out command, 191
 - Step Over command, 191
- VSA Integrated Development Environment (IDE), 142

W

- Watch window, 194
- working with tasks and DelayValidation, 115–119
- WriteOutEmployeeData, 192, 193, 196, 197
- writing out verbose information to a log file, 173

X

- XML configuration file, 202
 - creating, 204–207, 234
 - creating a new SSIS project, 203–204
 - creating and editing, 203–208
 - editing, 207–208
 - product records, 208
 - specifying new location, 202
 - testing package with new configuration, 208
 - understanding, 202
- XmlDocument class, 140

Z

- zero keys, 375

Paul Turley

Paul Turley is an architect for Hitachi Consulting and has been managing and developing business solutions for about 14 years for many companies, such as Hewlett-Packard, Walt Disney, and Microsoft. He manages the BI Training group for Hitachi's national External Education Services. Paul has taught application development and database design courses for a number of colleges and private training facilities. He has been a Microsoft Certified Solution Developer since 1996 and holds MCDBA, MCSDB, MCT, MSF, and IT Project+ certifications. Paul has presented at various conferences and industry associations, including Microsoft SQL PASS in 2004, 2006, and 2007. He has authored and co-authored several books for Wrox Press/Wiley Publishing on Reporting Services, Analysis Services, TSQL, and Access. He is the primary author of Beginning Transact-SQL for SQL Server 2000 and 2005 and was the lead author for Professional SQL Server Reporting Services (2000 and 2005). He is also a contributing author for Beginning SQL Server 2005 Administration. He lives in Vancouver, Washington, with his wife, Sherri, four kids, one dog, two cats, and a bird.

Joe Kasprzak

Joseph Kasprzak is a manager of business intelligence (BI) solutions for Hitachi Consulting in Boston. He has over 14 years of comprehensive business, technical, and managerial experience, providing consulting services for clients in the financial services, retail, telecommunications, health care, hospitality, manufacturing, and government industries. He has helped architect, integrate, develop, and manage full life cycle implementations of strategic BI analytical systems. Joe is a leader in providing BI best practices, proven BI methodologies, BI technology assessments, retail marketing analytics, business performance score cards, labor analytics, KPI executive dashboards, corporate performance management and reporting, financial analytics, and database modeling and design. Joe resides seaside in Newburyport, Massachusetts, where he and his wife, Liz, enjoy sailing and local volunteering. Joe holds a Bachelor of Science degree in mathematics/chemistry from Assumption College in Worcester, Massachusetts, and has performed post-graduate studies in computer science at MIT in Cambridge, Massachusetts.

Scott Cameron

Scott Cameron, a Senior BI Architect at Hitachi Consulting, has been developing BI solutions for nine years and has over 20 years' data analysis experience. He has over five years' experience working with SQL Server 2005 BI components and has taught SQL Server BI courses in the United States and Europe. He has experience in the health care, software, retail, insurance, legal, vocational rehabilitation, travel, and mining industries. He has helped several large companies perform their initial implementation of Microsoft Analysis Services 2005 and helped several independent software vendors integrate Analysis Services into their products. He holds a Bachelor of Arts degree in economics and Asian studies from Brigham Young University; his Master of Arts degree in Economics is from the University of Washington. Scott lives in the Seattle, Washington, metropolitan area with his wife, Tarya, and beagles Hunter and Si.

Satoshi Iizuka

Satoshi Iizuka, an engineer with Hitachi Ltd. in Tokyo, has over nine years of database custom development experience and significant BI development experience. He is a member of the Windows COE initiative at Hitachi Ltd., which manages and educates best practices for Microsoft products (including Microsoft .NET technologies). He programs with almost all Microsoft programming languages and Java and is proficient with Microsoft major server products and multiple software development methodologies. He is a Microsoft Certified Systems Engineer (MCSE), Microsoft Certified Database Administrator (MCDBA), and Microsoft Certified System Developer (MCSD) for .NET. Satoshi lives in Tokyo, Japan, with his wife and favorite Nikon cameras.

Pablo Guzman

Pablo Guzman, a BI consultant at Hitachi Consulting, has been developing BI solutions for over seven years. Prior to joining Hitachi Consulting, Pablo worked as a BI consultant for around a year and then worked for three years in a software-development company where he built BI tools for the banking industry. After that, he worked three years at the largest bank in Quito, Ecuador, where he was the BI program manager. He has taught SQL Server BI courses and developed SQL 2005 BI training material. He has engaged with multiple clients in business groups that include data warehousing, IT, insurance, manufacturing, and banking. He has worked in numerous projects that involved performing complex analysis on large data sets. He received an engineering degree in computer and information systems from the National Polytechnic School of Ecuador. Pablo lives in the Seattle, Washington, metropolitan area, where he has been an active volunteer participant in some nonprofit organizations and plays guitar, bass, and drums in several bands in Seattle.

Supporting Author

Anne Bockman Hansen

Anne has ten years' project-based experience in technical writing and editing, instructional design, and project management. She is experienced in designing curriculum for a wide variety of content areas, including Microsoft Windows Server, Windows Small Business Server, Microsoft Exchange Server, SQL Server, and Microsoft Office. She is experienced in designing curriculum for a variety of learning levels, including Web developers, Microsoft Certified Solution Providers, Solution developers, technical implementers and decision makers, corporate developers, site administrators, senior support professionals, and technical consultants. Anne received a Master of Science degree in technical communication from the University of Washington College of Engineering in 1996 and a Bachelor of Science degree in cognitive psychology in 1980. Anne resides in the country in Fall City, Washington, with her husband, Barry.