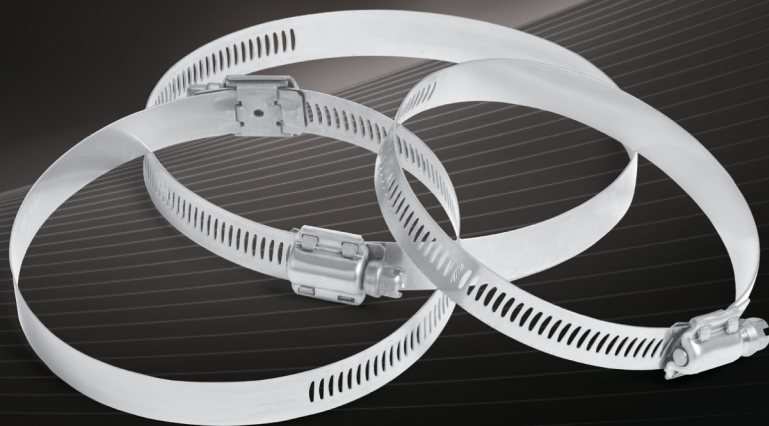


Microsoft®
SQL Server® 2012
Integration Services



Wee-Hyong Tok, Rakesh Parida,
Matt Masson, Xiaoning Ding,
and Kaarthik Sivashanmugam

Microsoft® SQL Server® 2012 Integration Services

Integrate data from across the enterprise—and improve business analysis

Build and manage data integration solutions with expert guidance from the Microsoft SQL Server Integration Services (SSIS) team. See best practices in action and dive deep into the SSIS engine, SSISDB catalog, and security features. Using the developer enhancements in SQL Server 2012 and the flexible SSIS toolset, you'll handle complex data integration scenarios more efficiently—and acquire the skills you need to build comprehensive solutions.

Discover how to:

- Use SSIS to extract, transform, and load data from multiple data sources
- Apply best practices to optimize package and project configuration and deployment
- Manage security settings in the SSISDB catalog and control package access
- Work with SSIS data quality features to profile, cleanse, and increase reliability
- Monitor, troubleshoot, and tune SSIS solutions with advanced features such as detailed views and data taps
- Load data incrementally to capture an easily consumable stream of insert, update, and delete activity



Get code samples on the web

Ready to download at
<http://go.microsoft.com/fwlink/?Linkid=258311>

For **system requirements**, see the *Introduction*.

microsoft.com/mspress

U.S.A. \$59.99

Canada \$62.99

[Recommended]

Programming/
Microsoft SQL Server

ISBN: 978-0-7356-6585-9



About the Authors

As members of the SQL Server Integration Services team, **Wee-Hyong Tok**, **Rakesh Parida**, **Matt Masson**, **Xiaoning Ding**, and **Kaarthik Sivashanmugam** helped develop key SSIS capabilities in SQL Server 2012.

DEVELOPER ROADMAP

Start Here!

- Beginner-level instruction
- Easy to follow explanations and examples
- Exercises to build your first projects



Step by Step

- For experienced developers learning a new topic
- Focus on fundamental techniques and tools
- Hands-on tutorial with practice files plus eBook



Developer Reference

- Professional developers; intermediate to advanced
- Expertly covers essential topics and techniques
- Features extensive, adaptable code examples



Focused Topics

- For programmers who develop complex or advanced solutions
- Specialized topics; narrow focus; deep coverage
- Features extensive, adaptable code examples



Microsoft®

Microsoft® SQL Server® 2012 Integration Services

Wee-Hyong Tok
Rakesh Parida
Matt Masson
Xiaoning Ding
Kaarthik Sivashanmugam

Copyright © 2012 by Wee-Hyong Tok, Rakesh Parida, Matt Masson, Xiaoning Ding, Kaarthik Sivashanmugam. 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.

ISBN: 978-0-7356-6585-9

2 3 4 5 6 7 8 9 10 LSI 7 6 5 4 3 2

Printed and bound in the United States of America.

Microsoft Press books are available through booksellers and distributors worldwide. If you need support related to this book, email Microsoft Press Book Support at mspinput@microsoft.com. Please tell us what you think of this book at <http://www.microsoft.com/learning/booksurvey>.

Microsoft and the trademarks listed at <http://www.microsoft.com/about/legal/en/us/IntellectualProperty/Trademarks/EN-US.aspx> are trademarks of the Microsoft group of companies. All other marks are property of their respective owners.

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

This book expresses the authors' 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 and Developmental Editor: Russell Jones

Production Editor: Melanie Yarbrough

Editorial Production: Stan Info Solutions

Technical Reviewer: boB Taylor

Copyeditor: Teresa Horton

Indexer: WordCo Indexing Services, Inc.

Cover Design: Twist Creative • Seattle

Cover Composition: ContentWorks, Inc.

Illustrator: Rebecca Demarest

Dedicated to my wife, Juliet, and son, Nathaniel, for their love, support, and patience. And to my parents, Siak-Eng and Hwee-Tiang for shaping me into who I am today.

—WEE-HYONG TOK

I would like to dedicate this to my parents, Basanta and Sarmistha, and my soon-to-be-wife, Vijaya, for all their support and encouragement for making this happen.

—RAKESH PARIDA

Dedicated to my family and loving wife, whose patience and support made all this possible.

—MATT MASSON

I would like to dedicate this book to my parents, ChengXian and Dezhen, my wife Yanan, my cute daughter Ruiyi, and my lovely son Ruichen. I love all of you so much.

—XIAONING DING

I dedicate this book to my wife, Devi, and my son, Raghav, for their love and support.

—KAARTHIC SIVASHANMUGAM

Contents at a Glance

Foreword *xxi*

Introduction *xxiii*

PART I **OVERVIEW**

CHAPTER 1	SSIS Overview	3
CHAPTER 2	Understanding SSIS Concepts	27
CHAPTER 3	Upgrading to SSIS 2012	49

PART II **DEVELOPMENT**

CHAPTER 4	New SSIS Designer Features	83
CHAPTER 5	Team Development	115
CHAPTER 6	Developing an SSIS Solution	131
CHAPTER 7	Understanding SSIS Connectivity	161
CHAPTER 8	Working with Change Data Capture in SSIS 2012	195
CHAPTER 9	Data Cleansing Using SSIS	241

PART III **DATABASE ADMIN**

CHAPTER 10	Configuration in SSIS	265
CHAPTER 11	Running SSIS Packages	301
CHAPTER 12	SSIS T-SQL Magic	331
CHAPTER 13	SSIS PowerShell Magic	353
CHAPTER 14	SSIS Reports	365

PART IV **DEEP-DIVE**

CHAPTER 15	SSIS Engine Deep Dive	385
CHAPTER 16	SSIS Catalog Deep Dive	415
CHAPTER 17	SSIS Security	435
CHAPTER 18	Understanding SSIS Logging	463
CHAPTER 19	Automating SSIS	483

PART V	TROUBLESHOOTING	
CHAPTER 20	Troubleshooting SSIS Package Failures	517
CHAPTER 21	SSIS Performance Best Practices	539
CHAPTER 22	Troubleshooting SSIS Performance Issues	569
CHAPTER 23	Troubleshooting Data Issues	589
	<i>Index</i>	605
	<i>About the Authors</i>	637

Contents

<i>Foreword</i>	<i>xxi</i>
<i>Introduction</i>	<i>xxiii</i>

PART I OVERVIEW

Chapter 1 SSIS Overview	3
Common Usage Scenarios for SSIS	4
Consolidation of Data from Heterogeneous Data Sources	4
Movement of Data Between Systems	9
Loading a Data Warehouse	12
Cleaning, Formatting, or Standardization of Data	16
Identification, Capture, and Processing of Data Changes	17
Coordination of Data Maintenance, Processing, or Analysis	18
Evolution of SSIS	20
Setting Up SSIS	21
SQL Server Features Needed for Data Integration	22
SQL Server Editions and Integration Services Features	24
Summary	25
Chapter 2 Understanding SSIS Concepts	27
Control Flow	28
Tasks	28
Precedence Constraints	30
Variables and Expressions	31
Containers	32
Connection Managers	35

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:

microsoft.com/learning/booksurvey

Packages and Projects	36
Parameters	37
Log Providers	38
Event Handlers	40
Data Flow	41
Source Adapters	41
Destination Adapters	42
Transforms	43
SSIS Catalog	44
Overview	45
Catalog	46
Folders	46
Environments	46
References	47
Summary	47
Chapter 3 Upgrading to SSIS 2012	49
What's New in SSIS 2012	49
Upgrade Considerations and Planning	50
Feature Changes in SSIS	50
Dependencies and Tools	52
Upgrade Requirements	52
Upgrade Scenarios	53
Unsupported Upgrade Scenarios	54
Upgrade Validation	55
Integration Services Upgrade	55
Upgrade Advisor	55
Performing Upgrade	61
Addressing Upgrade Issues and Manual Upgrade Steps	69
Conversion to Projects after Upgrade	71
Summary	79

Chapter 4	New SSIS Designer Features	83
	The Integration Services Designer	83
	Visual Studio	83
	Undo and Redo	84
	Getting Started Window	85
	Toolbox	85
	Variables Window	87
	Zoom Control	88
	Autosave and Recovery	89
	Status Icons	89
	Annotations	90
	Configuration and Deployment	90
	Solution Explorer Changes	90
	Parameter Tab	92
	Visual Studio Configurations	92
	Project Compilation	93
	Deployment Wizard	94
	Project Conversion Wizard	95
	Import Project Wizard	96
	New Tasks and Data Flow Components	96
	Change Data Capture	96
	Expression Task	99
	DQS Cleansing Transform	100
	ODBC Source and Destination	100
	Control Flow	100
	Expression Adorners	100
	Connection Managers	101
	Execute SQL Task	101

Data Flow	102
Connection Assistants	102
Improved Column Mapping	103
Editing Components in an Error State	104
Grouping	104
Simplified Data Viewers	105
Row Count and Pivot Transform User Interfaces	105
Flat File Source Changes	106
Scripting	108
Visual Studio Tools for Applications	108
Script Component Debugging	109
.NET 4 Framework Support	111
Expressions	112
Removal of the Character Limit	112
New Expression Functions	112
Summary	113

Chapter 5 Team Development 115

Improvements in SQL Server 2012	115
Package Format Changes	115
Visual Studio Configurations	116
Using Source Control Management with SSIS	117
Connecting to Team Foundation Server	117
Adding an SSIS Project to Team Foundation Server	120
Change Management	124
Changes to the SSIS Visual Studio Project File	127
Best Practices	129
Using Small, Simple Packages	129
One Developer Per Package	129
Consistent Naming Conventions	130
Summary	130

Chapter 6 Developing an SSIS Solution 131

SSIS Project Deployment Models131
 Package Deployment Model131
 Project Deployment Model133
Develop an Integration Services Project.....136
 Creating an SSIS Project136
 Designing an Integration Services Data Flow147
 Using Parameters and the *ForEach* Container.152
 Using the Execute Package Task156
 Building and Deploying an Integration Services Project159
Summary.....160

Chapter 7 Understanding SSIS Connectivity 161

Previous Connectivity Options in SSIS.161
 Providers for Connectivity Technology162
 OLE DB, ADO.NET, and ODBC164
New Connectivity Options in SSIS 2012165
 Introducing ODBC166
 ODBC Components for SSIS.....168
 ODBC Source169
 ODBC Destination174
Connectivity Considerations for SSIS.....177
 64-Bit and SSIS.....177
 SSIS Tools on 64-Bit Architecture178
Connectivity to Other Sources and Destinations184
 Connecting to Microsoft Excel and Access184
 Connecting to Oracle186
 Creating Custom Components189
 Using Script Components.....191
Summary.....193

Chapter 8	Working with Change Data Capture in SSIS 2012	195
	CDC in SQL Server.....	195
	Using CDC in SQL Server	196
	CDC Scenarios in ETLs.....	197
	Stages in CDC	198
	CDC in SSIS 2012.....	202
	CDC State.....	202
	CDC Control Task.....	205
	Data Flow Component: CDC Source	211
	CDC Splitter Component	215
	CDC for Oracle.....	217
	Introduction	217
	Components for Creating CDC for Oracle.....	219
	CDC Service Configuration MMC	219
	Oracle CDC Designer MMC	221
	MSXDBCDC Database.....	233
	Oracle CDC Service Executable (xdbcscvc.exe).....	235
	Data Type Handling	238
	SSIS CDC Components	240
	Summary.....	240
Chapter 9	Data Cleansing Using SSIS	241
	Data Profiling Task	241
	Fuzzy Lookup Transformation	246
	Fuzzy Grouping Transformation.....	251
	Data Quality Services Cleansing Transform	254
	Summary.....	261

Chapter 10 Configuration in SSIS	265
Configuration Basics	266
How Configurations Are Applied	266
What to Configure	266
Changes in SSIS 2012	267
Configuration in SSIS 2012	267
Parameters	268
Creating Package Parameters	268
Creating Project Parameters	271
API for Creating Parameters	273
Using Parameters	274
Configuring Parameters on the SSIS Catalog	281
Configuring, Validating, and Executing Packages and Projects	281
Configuration Through SSMS	281
Configuration Using SQL Agent, DTEXec, and T-SQL	285
SSIS Environments	287
Evaluation Order of Parameters	291
Package Deployment Model and Backward Compatibility	291
Package Deployment Model	292
Best Practices for Configuring SSIS	295
Best Practices with Package Deployment Model	295
Best Practices with Project Deployment Model	299
Summary	300
Chapter 11 Running SSIS Packages	301
Ways to Run SSIS Packages	301
Package Locations	303
Configuring Packages	306
Error Dumps	308
Logging Options	309

Running Packages in the SSIS Catalog	311
Prepare Executions.....	312
Starting SSIS Package Executions	314
View Executions	318
Executions with T-SQL.....	318
Running Packages from SQL Agent.....	320
Create an SSIS Job Step.....	320
Execute Packages from the SSIS Catalog.....	322
Running Packages via PowerShell	325
Creating and Running SSIS Packages Programmatically.....	325
Summary.....	330

Chapter 12 SSIS T-SQL Magic 331

Overview of SSIS Stored Procedures and Views	331
Integration Services Catalog.....	332
SSIS Catalog Properties.....	332
Querying the SSIS Catalog Properties	333
Setting SSIS Catalog Properties.....	333
SSIS Projects and Packages.....	334
Deploy an SSIS Project to the SSIS Catalog.....	334
Learning About the SSIS Projects Deployed to the SSIS Catalog	335
Configuring SSIS Projects	336
Managing SSIS Projects in the SSIS Catalog	339
Running SSIS Packages in the SSIS Catalog.....	341
SSIS Environments.....	345
Creating SSIS Environments	346
Creating SSIS Environment Variables	346
Configuring SSIS Projects Using SSIS Environments	347
Configuring SSIS Projects Using Reference Values.....	348
Package Execution Using SSIS Environments	349
Managing SSIS Environment and Environment Variables.....	349
Summary.....	351

Chapter 13 SSIS PowerShell Magic	353
PowerShell Refresher	353
PowerShell and SQL Server	354
Managing SSIS with PowerShell	357
SSIS Management Object Model	357
PowerShell with SSIS Management Object Model	358
PowerShell and SSIS Using T-SQL	362
Advantages of Using PowerShell with SSIS	364
Summary	364

Chapter 14 SSIS Reports	365
Getting Started with SSIS Reports	365
Data Preparation	367
Monitoring SSIS Package Execution	368
Integration Services Dashboard	368
All Executions Report	370
All Validations and All Operations Reports	371
Using SSIS Reports to Troubleshoot SSIS Package Execution	373
Using the Execution Performance Report to Identify Performance Trends	378
Summary	381

PART IV DEEP-DIVE

Chapter 15 SSIS Engine Deep Dive	385
The Control Flow Engine	385
Overview	385
Load	386
Apply Parameters	388

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:

microsoft.com/learning/booksurvey

Validate	388
Execute.....	390
The Data Flow Engine.....	397
Overview	398
Execution Control.....	401
Backpressure.....	408
Engine Tuning.....	411
Summary.....	414
Chapter 16 SSIS Catalog Deep Dive	415
SSIS Catalog Deep Dive.....	415
Creating the SSIS Catalog.....	415
Unit of Deployment to the SSIS Catalog	417
What Is Inside SSISDB?	418
SQL Server Instance Starts Up	420
SSIS Catalog and Logging Levels.....	422
Understanding the SSIS Package Execution Life Cycle.....	423
Stopping SSIS Package Executions	426
Using the Windows Application Event Log	426
SSIS Catalog Maintenance and SQL Server Agent Jobs.....	427
Backup and Restore of the SSIS Catalog.....	430
Back Up SSISDB	431
Restore SSISDB	432
Summary.....	434
Chapter 17 SSIS Security	435
Protect Your Package.....	435
Control Package Access	435
Package Encryption.....	439
Sensitive Variables and Parameters.....	441
Package Signing.....	442

Security in the SSIS Catalog	443
Security Overview	444
Manage Permissions	446
DDL Trigger	453
Running SSIS with SQL Agent	454
Requirements	454
Create Credentials	454
Create Proxy Accounts	456
Create SQL Agent Jobs	459
Summary	461

Chapter 18 Understanding SSIS Logging 463

Configure Logging Options	463
Choose Containers	464
Select Events	466
Add Log Providers	468
Log Providers	471
Text Files	471
SQL Server	471
SQL Server Profiler	472
Windows Event Log	472
XML Files	473
Logging in the SSIS Catalog	474
Logging Levels	474
Event Logs	476
Event Context Information	477
Advanced Logging Topics	478
Customizing Logging Fields	478
Logging with dtexec Utility	479
Developing Custom Log Providers	479
Summary	481

Chapter 19 Automating SSIS	483
Introduction to SSIS Automation	483
Programmatic Generation of SSIS Packages	483
Metadata-Driven Package Execution	484
Dynamic Package Generation	485
Handling Design-Time Events	486
Samples	488
Metadata-Based Execution	497
Custom Package Runner	498
Using PowerShell with the SSIS Management Object Model . . .	502
Using PowerShell with SQL Agent	505
Alternative Solutions and Samples	508
Samples on Codeplex	508
Third-Party Solutions	509
Summary	513

PART V TROUBLESHOOTING

Chapter 20 Troubleshooting SSIS Package Failures	517
Getting Started with Troubleshooting	517
Data Preparation	519
Troubleshooting Failures of SSIS Package Executions	520
Three Key Steps Toward Troubleshooting Failures of SSIS Package Executions	522
Execution Path	526
Finding the Root Cause of Failure	526
Troubleshooting the Execute Package Task and Child Package Executions	529
DiagnosticEx Events	531
Execute Package Task and Execution Path	532
Troubleshooting SSIS Package Execution Failures Scheduled with SQL Agent	534

Using CallerInfo to Determine SSIS Package Executions That Are Executed by SQL Agent	537
Using SQL Agent History Tables to Determine the SSIS Job Steps That Failed	537
Summary	538

Chapter 21 SSIS Performance Best Practices 539

Creating a Performance Strategy	540
OVAL Technique	540
Measuring SSIS Performance	542
Measuring System Performance	542
Measuring Performance of Data Flow Tasks	546
Designing for Performance	552
Parallelize Your Design	552
Using SQL Server Optimization Techniques	556
Bulk Loading Your Data	558
Keeping SSIS Operations in Memory	561
Optimizing SSIS Lookup Caching	562
Optimizing SSIS Infrastructure	566
Summary	568

Chapter 22 Troubleshooting SSIS Performance Issues 569

Performance Profiling	569
Troubleshooting Performance Issues	570
Data Preparation	571
Understanding SSIS Package Execution Performance	572
SSIS Package Execution Duration	572
Time Spent at Each Task in the SSIS Package	573
Time Spent at Each Phase of the Data Flow Component	573
Elapsed Time for Data Flow Component Phases (Active Time vs. Total Time)	574
Monitoring SSIS Package Execution Performance	576

Per-Execution Performance Counters	578
Interactive Analysis of Performance Data	579
Summary	588
Chapter 23 Troubleshooting Data Issues	589
Troubleshooting in the Design Environment	589
Row Count Values	589
Data Viewers	590
Data in Error Output	592
Breakpoints and Debug Windows	593
Troubleshooting in the Execution Environment	593
Execution Data Statistics	593
Data Tap	596
Error Dumps	600
Summary	603
<i>Index</i>	605
<i>About the Authors</i>	637

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:

microsoft.com/learning/booksurvey

Foreword

In 1989, when we were all much younger, I had a bizarre weekend job: During the week, I was an engineer at Microrim Incorporated, the makers of R:Base—the second most popular desktop database in the world. But on Saturday mornings I would sit completely alone in our headquarters building in Redmond and rebuild the database that ran our call center. This involved getting the latest registered licenses from accounting, the up-to-date employee list from human resources, the spreadsheets from marketing that tracked our independent software vendors, and of course all of the previous phone call history from the log files, and then mashing it all together. Of course none of these systems had consistent formats or numbering schemes or storage. It took me six hours—unless I messed up a step. The process was all scripted out on a sheet of paper. There wasn't a name for it at the time, but I was building a data warehouse.

Anyone who's done this work knows in their heart the message we hear again and again from customers: Getting the right data into the right shape and to the right place at the right time is 80 percent of the effort for any data project. Data integration is the behind-the-wall plumbing that makes a beautiful fountain work flawlessly. Often the fountains get all the attention, but on the SSIS team at Microsoft, we are proud to build that plumbing.

The authors of this book are at the core of that proud team. For as long as I have known him, Kaarthik has been an ardent advocate for this simple truth: You can understand the quality of a product only if you first deeply understand the customers that use it. As the first employee for SSIS in China, Xiaoning blazed a trail. He is one of those quiet geniuses, who, when he speaks, everyone stops to listen to, because what he says will be deep and important. One of my best professional decisions was overriding my manager's advice to hire Matt. You see, he didn't quite fit our mold. Yes, he could write code well, but there was something that just didn't match our expectations. He cared way too deeply about the real world and about building end-to-end solutions to solve business problems; he wouldn't stop talking about it! We made it work. Don't tell Wee Hyong I said this, but he is probably overqualified for his job. His background as a lecturer in academia, and his history as a SQL MVP (our most valuable partners) made him a perfect candidate to be one of the public faces of SSIS. And finally, Rakesh. At the end of his first week on the job, he decided to create a community event for our customers attending a trade show that just happened to be running nearby. He coerced his teammates into helping, found a room at the convention center, and sent out the invite to our customers. In all the authors, you can see a passion for customers and solutions. You are in great hands.

The strategy for the 2012 SSIS release started with a listening tour of those customers. Their priorities were clear: Make the product easier to use and easier to manage. That sounds like a simple goal, but as I read through the chapters of this book I was astonished by just how much we accomplished toward those goals, and just how much better we've made an already great product. If you are new to SSIS, this book is a good way to dive in to solving real problems, and if you are an SSIS veteran, you will find yourself compelled by the authors' enthusiasm to go and try some of these new things. This is the best plumbing we've ever made. I'm proud of it.

When I was asked to write this foreword I was packing my office in Building 34 in Redmond. I looked out the window and I could see Building 21 across the street. Twenty-five years ago that exact same building housed the world headquarters of Microrim Incorporated. I remembered that kid alone on a Saturday. It's a small world.

Jeff Bernhardt

Group Program Manager, SQL Server Data Movement

Shanghai, China

Introduction

Microsoft SQL Server Integration Services is an enterprise-ready platform for developing data integration solutions. SQL Server Integration Services provides the ability to extract and load from and to heterogeneous data sources and destinations. In addition, it provides the ability for you to easily deploy, manage, and configure these data integration solutions. If you are a data integration developer or a database administrator looking for a data integration solution, then SQL Server Integration Services is the right tool for you.

Microsoft SQL Server 2012 Integration Services provides an organized walkthrough of Microsoft SQL Server Integration Services and the new capabilities introduced in SQL Server 2012. The text is a balanced discussion of using Integration Services to build data integration solutions, and a deep dive into Integration Services internals. It discusses how you can develop, deploy, manage, and configure Integration Services packages, with examples that will give you a great head start on building data integration solutions. Although the book does not provide exhaustive coverage of every Integration Services feature, it offers essential guidance in using the key Integration Services capabilities.

Beyond the explanatory content, each chapter includes examples, procedures, and downloadable sample projects that you can explore for yourself.

Who Should Read This Book

This book is not for rank beginners, but if you're beyond the basics, dive right in and really put SQL Server Integration Services to work! This highly organized reference packs hundreds of time-saving solutions, troubleshooting tips, and workarounds into one volume. It's all muscle and no fluff. Discover how experts perform data integration tasks—and challenge yourself to new levels of mastery.

- Gain in-depth understanding of Integration Services capabilities introduced in SQL Server 2012
- Implement Integration Services best practices and design patterns
- Master the ETL tool for data extraction, transformation, and loading
- Manage performance issues using tuning principles and techniques
- Diagnose problems and apply advanced troubleshooting features

Assumptions

This book expects that you have at least a minimal understanding of Microsoft SQL Server Integration Services and basic database concepts. This book includes examples in Transact-SQL, C#, and PowerShell. If you have not yet picked up one of those languages, you might consider reading John Sharp's *Microsoft Visual C# 2010 Step by Step* (Microsoft Press, 2010) or Itzik Ben-Gan's *Microsoft SQL Server 2012 T-SQL Fundamentals* (Microsoft Press, 2012).

With a heavy focus on database concepts, this book assumes that you have a basic understanding of relational database systems such as Microsoft SQL Server, and have had brief exposure to one of the many flavors of the query tool known as SQL. To go beyond this book and expand your knowledge of SQL and Microsoft's SQL Server database platform, other Microsoft Press books offer both complete introductions and comprehensive information on T-SQL and SQL Server.

Who Should Not Read This Book

This book does not cover basic SQL Server concepts, nor does it cover other technologies such as Analysis Services, Reporting Services, Master Data Services, and Data Quality services.

Organization of This Book

This book is divided into five sections, each of which focuses on a different aspect of Microsoft SQL Server Integration Services. Part I, "Overview" provides a quick overview of Integration Services concepts and considerations for upgrading to Microsoft SQL Server 2012 Integration Services. Part II, "Using SSIS," shows how you can leverage the new Integration Services designer features in developing data integration solutions. In addition, Part II shows how you can work with Change Data Capture, and perform data cleansing using Integration Services. Part III, "Configuration/Management and Monitoring" shows how you can configure an Integration Services project. In addition, Part III shows how you can use Transact-SQL and PowerShell with Integration Services. In addition, it provides a walkthrough of the built-in reports. The internals and advanced concepts of Integration Services take center stage in Part IV, "Deep Dive." Finally, Part V, "Troubleshooting," covers topics that show how you can monitor and troubleshoot Integration Services issues, such as package failures, identifying performance bottlenecks, and data issues.

Finding Your Best Starting Point in This Book

The different sections of *Microsoft SQL Server 2012 Integration Services* cover a wide range of concepts and walkthroughs on building data integration solutions. Depending on your needs and your existing understanding of various SQL Server Integration Services capabilities, you might wish to focus on specific areas of the book. Use the following table to determine how best to proceed through the book.

If you are	Follow these steps
New to SQL Server Integration Services	Focus on Parts I and II and on Chapters 10 and 11 in Part III, or read through the entire book in order.
Familiar with earlier releases of SQL Server Integration Services	Briefly skim Part I if you need a refresher on the core concepts. Read up on the new technologies in Parts II, III, and V and be sure to read Chapter 17 in Part IV.
Interested in using Transact-SQL or PowerShell capabilities for using SQL Server Integration Services	Chapter 12 and 13 in Part III provide a walkthrough of the concepts.
Interested in monitoring and troubleshooting SQL Server Integration Services	Read through the chapters in Part V.

Most of the book's chapters include hands-on samples that let you try out the concepts just learned. No matter which sections you choose to focus on, be sure to download and install the sample applications on your system.

Conventions and Features in This Book

This book presents information using conventions designed to make the information readable and easy to follow.

- In most cases, the book includes examples that use Transact-SQL or PowerShell. Each example consists of a series of tasks, presented as numbered steps (1, 2, and so on) listing each action you must take to complete the exercise.
- Boxed elements with labels such as "Note" provide additional information or alternative methods for completing a step successfully.

System Requirements

You will need the following hardware and software to complete the practice exercises in this book:

- .NET 3.5 Service Pack 1 is a requirement for SQL Server 2012
- SQL Server 2012 Standard Edition or higher, with SQL Server Management Studio 2012
- Computer that has a 1.4 GHz or faster processor (2 GHz or faster is recommended)
- SQL Server 2012 requires a minimum of 6 GB of available hard-disk space, and 1 GB of memory (4 GB of memory or more is recommended for optimal performance)
- Refer to <http://msdn.microsoft.com/en-us/library/ms143506.aspx> for operating system requirements for installing SQL Server 2012
- Internet connection to download software or chapter examples

Depending on your Windows configuration, you might require Local Administrator rights to install or configure SQL Server 2012 products.

Code Samples

Most of the chapters in this book include exercises that let you interactively try new material learned in the main text. All sample projects, in both their preexercise and postexercise formats, can be downloaded from the following page:

<http://www.microsoftpressstore.com/title/9780735665859>

Follow the instructions to download the SSIS_2012_examples.zip file.



Note In addition to the code samples, your system should have SQL Server 2012 and SQL Server Management Studio installed.

Most of the samples require sample data from Adventure Works for SQL Server 2012. You can download the sample databases from the following page:

<http://msftdbprodsamples.codeplex.com/releases/view/55330>

Installing the Code Samples

Follow these steps to install the code samples on your computer so that you can use them with the exercises in this book.

1. Unzip the `SSIS_2012_examples.zip` file that you downloaded from the book's website (name a specific directory along with directions to create it, if necessary).
2. If prompted, review the displayed end user license agreement. If you accept the terms, select the Accept option, and then click Next.



Note If the license agreement doesn't appear, you can access it from the same webpage from which you downloaded the `SSIS_2012_examples.zip` file.

Using the Code Samples

The folder structure created by unzipping the sample code download contains folders corresponding to each chapter. In each of the folders, you will see the code examples used in the chapter.

Acknowledgments

The authors would like to thank all the SQL Server professionals who have worked closely with the Integration Services team throughout the years to evolve the product into an enterprise-ready data integration platform, as well as all the members of the SQL Server Integration Services team for their help and contributions to this book. Specifically, the authors would like to thank Jeff Bernhardt for contributing the foreword for the book, and the editorial team at Microsoft Press and O'Reilly (Russell Jones, Melanie Yarbrough, Rani Xavier G, and Teresa Horton) for all their support of the book, from initial proposal to final completion.

Errata & Book Support

We've made every effort to ensure the accuracy of this book and its companion content. Any errors that have been reported since this book was published are listed on our Microsoft Press site:

<http://www.microsoftpressstore.com/title/9780735665859>

If you find an error that is not already listed, you can report it to us through the same page.

If you need additional support, email Microsoft Press Book Support at *mspinput@microsoft.com*.

Please note that product support for Microsoft software is not offered through the addresses above.

We Want to Hear from You

At Microsoft Press, your satisfaction is our top priority, and your feedback our most valuable asset. Please tell us what you think of this book at:

<http://www.microsoft.com/learning/booksurvey>

The survey is short, and we read every one of your comments and ideas. Thanks in advance for your input!

Stay in Touch

Let's keep the conversation going! We're on Twitter: *<http://twitter.com/MicrosoftPress>*

PART I

Overview

Chapter 1	SSIS Overview	3
Chapter 2	Understanding SSIS Concepts	27
Chapter 3	Upgrading to SSIS 2012	49



SSIS Overview

In This Chapter

Common Usage Scenarios for SSIS	4
Evolution of SSIS	20
Setting Up SSIS	21
Summary	25

Enterprises depend on data integration to turn data into valuable insights and decisions. Enterprise data integration is a complicated problem due to the heterogeneity of data sources and formats, ever-increasing data volumes, and the poor quality of data. Data is typically stored in disparate systems and the result is that there are differences in data format or schema that must be resolved. The constantly decreasing costs of storage lead to increased data retention and a concomitant increase in the volume of data that needs to be processed. In turn, this results in an ever-increasing demand for scalable and high-performance data integration solutions so organizations can obtain timely insights from the collected data. The diversity of data and inconsistent duplication cause quality problems that can impact the accuracy of analytical insights and thus also affect the quality and value of the decisions. Data integration projects need to deal with these challenges and effectively consume data from a variety of sources (e.g., databases, spreadsheets, files, etc.), which requires that they clean, correlate, transform, and move the source data to the destination systems. This process is further complicated because many organizations have round-the-clock dependencies on data stores; therefore, data integration must often be frequent and integration operations must be completed as quickly as possible.

Microsoft SQL Server Integration Services (SSIS) technology addresses these challenges and provides a platform for building and managing data integration solutions. The flexible, extensible, scalable, high-performance platform and toolset in SSIS caters to enterprise demands for traditional data extract-transform-load (ETL) processes as well as other data integration needs. SSIS is a feature of Microsoft SQL Server that provides seamless interoperability with other features shipped with both SQL Server and with other Microsoft products. Typical data integration scenarios that you can address using SSIS include the following:

- Consolidating data from heterogeneous data sources
- Moving data between systems
- Loading data warehouses
- Cleaning, formatting, or standardizing data
- Identifying, capturing, and processing data changes
- Coordinating data maintenance, processing, or analysis

Some data processing scenarios require specialized technology. SSIS is not suitable for the following types of data processing:

- Processing complex business rules on data
- Coordinating, acquiring, and processing data in inter-business processes
- Processing event messages in real time
- Coordinating data communication between systems
- Building a federated view of data sources
- Unstructured data processing and integration

Common Usage Scenarios for SSIS

In this section, you'll examine some common data integration scenarios in detail and get an overview of how key SSIS features help in each of those scenarios.

Consolidation of Data from Heterogeneous Data Sources

In an organization, data is typically not contained in one system but spread all over. Different applications might have their own data stores with different schema. Similarly, different parts of the organization might have their own locally consolidated view of data, or legacy systems might be isolated, making the data available to rest of the organization at regular intervals. To make important organization-wide decisions that derive value from all this data, it is necessary to pull data from all parts of the organization, massaging and transforming it into a consistent state and shape.

The need for data consolidation also arises during organization acquisitions or mergers. Supporting connectivity to heterogeneous stores and extracting data is a key feature of any data integration software. SSIS supports connectivity to a wide range of data stores using out-of-the-box adapters and extensibility features. Source adapters read data from external sources into SSIS, whereas destination adapters write data from SSIS to external destinations.

Some of the most important built-in source and destination adapters in SSIS are the following:

- OLE DB Source and Destination

- ADO.NET Source and Destination
- ODBC Source and Destination
- Flat File Source and Destination
- Excel Source and Destination
- XML Source



Note Open Database Connectivity (ODBC) source and destination components are available starting with Integration Services 2012 and are not available in earlier versions. In SQL Server 2008 and SQL Server 2008 R2, you can use ADO.NET source and destination components in SSIS to connect to ODBC data sources using the .NET ODBC Data Provider. The ADO.Net Destination component is not available in SQL Server 2005.

Other types of SSIS adapters are as follows:

- Script Source and Destination: These enable SSIS developers to author code to connect to data stores that are not supported by built-in adapters in SSIS.
- Special-purpose adapters: Most of the adapters in SSIS are general purpose, supporting any data store that can be accessed through standard interfaces; however, some of the adapters are specific to a particular data store and depend on a specific application programming interface (API). The SQL Server Destination and Dimension Processing Destination are examples of special-purpose adapters that provide connectivity to SQL Server and Analysis Server, respectively.
- Custom adapters: Using the extensibility mechanism in SSIS, customers and independent software vendors (ISVs) can build adapters that can be used to connect to data stores that do not have any built-in support in SSIS.



Note Scripting in SSIS is powered by Visual Studio for Applications in SQL Server 2005 and Visual Studio Tools for Applications in SQL Server 2008 and later versions. Visual Studio for Applications and Visual Studio Tools for Applications are .NET-based script hosting technologies to embed custom experience into applications. Both of these technologies provide a runtime that executes custom code using a script engine and end-user integrated development environment (IDE) for writing and debugging custom code. Visual Studio for Applications supports VB.Net and Visual Studio Tools for Applications supports both VB.Net and C# programming.

Source and destination adapters that are not a part of SSIS installation but available for download from Microsoft.com include the following:

- Oracle Source and Destination

- Teradata Source and Destination
- SAP BI Source and Destination



Note Oracle, Teradata, and SAP BW connectors are available only for advanced editions of SQL Server. See details on SQL Server editions in a later section in this chapter. Oracle and Teradata connectors are available for download at <http://www.microsoft.com/download/en/details.aspx?id=29283>. Microsoft Connector 1.1 for SAP BW is available as a part of SQL Server Feature Pack at <http://www.microsoft.com/download/en/details.aspx?id=29065>.

SSIS adapters maintain connection information to external data stores using *connection managers*. SSIS connection managers depend on technology-specific data providers or drivers for connecting to data stores. For example, OLE DB adapters use the OLE DB API and data provider to access data stores that support OLE DB. SSIS connectivity adapters are used within a *Dataflow Task*, which is powered by a data pipeline engine that facilitates high-performance data movement and transformation between sources and destinations. Figure 1-1 illustrates flow of data from source to destination through data providers or drivers.

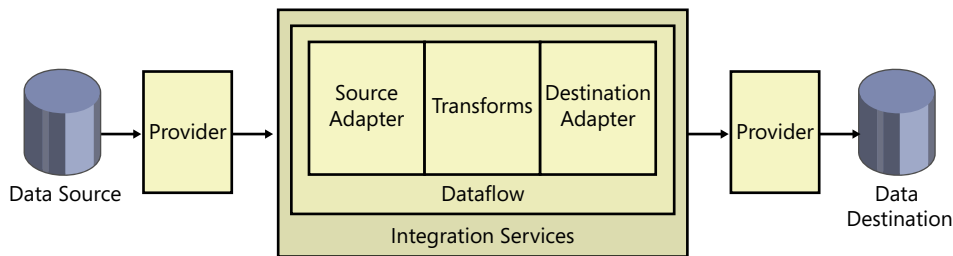


FIGURE 1-1 Representation of data flow from source to destination.

Integration Services offers several options for connecting to relational databases. OLE DB, ADO, NET, and ODBC adapters provide data store generic APIs for connecting to a wide range of databases. The only popular database connectivity option that is not supported in SSIS is Java Database Connectivity (JDBC). SSIS developers are often faced with the challenge of picking an adapter from the choices to connect to a particular data store. The factors that SSIS developers should consider when picking the connectivity options are as follows:

- Data type support
- Metadata exposed by driver or provider
- Driver or provider support in 32 and 64-bit environments
- Performance

Data Type Support

Data type support in relational databases beyond the standard ANSI SQL data types differs; each has its own type system. Data types supported by data providers and drivers provide a layer of abstraction for the type systems in data stores. Data integration tools need to ensure that they don't lose type information when reading, processing, or writing data. SSIS has its own data type system. Adapters in SSIS map external data types exposed by data providers to SSIS data types, and maintain data type fidelity during interactions with external stores. The SSIS data type system ameliorates problems when dealing with data type differences among storage systems and providers, providing a consistent basis for data processing. SSIS implicitly converts data to the equivalent types in its own data type system when reading or writing data. When that is not possible, it might be necessary to explicitly convert data to binary or string types to avoid data loss.



Note See <http://msdn.microsoft.com/en-us/library/ms141036.aspx> for a comprehensive list of SSIS data types.

Metadata Exposed by Provider

SQL Server Data Tools provides the development environment in which you can build SSIS packages, which are executable units in SSIS. Design experience in SQL Server Data Tools depends on the metadata exposed by data stores through drivers or providers to guide SSIS developers in setting package properties. Such metadata is used to get a list of databases, tables, views, and metadata of columns in tables or views during package construction. If a data store does not expose a particular metadata or if the driver does not implement an interface to get some metadata from the data stores, the SSIS package development experience will be affected. Manually setting the relevant properties in SSIS packages could help in those instances.



Note The Integration Services designer in SQL Server 2005, 2008, and 2008 R2 is called Business Intelligence Development Studio. In SQL Server 2012, the SSIS development environment became part of an integrated toolset named SQL Server Data Tools, which brought together database and business intelligence development into one environment.

Support in 32 and 64-Bit Environments

You can execute SSIS packages in either 32-bit or 64-bit modes. If the application is a 32-bit application, SSIS uses the 32-bit data provider.

The 32 and 64-bit versions of data providers usually have the same identifier. Once referenced using the identifier, the version of the data provider loaded during execution time will depend on the application that loads it. The data provider available to SSIS packages will depend on the bit mode under which the package is executed. For example, execution of packages inside SQL Server Data Tools is in 32-bit mode by default; hence the 32-bit provider will be used during execution in SQL Server Data Tools. Packages that successfully execute in 32-bit mode do not necessarily execute successfully in 64-bit mode (and

vice versa). This is because data providers or drivers might not be available in both modes. If the 64-bit driver is not available on the executing machine, execution will fail when attempting 64-bit execution and vice versa. SSIS package developers and administrators have to keep this in mind during package development and execution.



Note You can override 32-bit execution in SQL Server Data Tools by setting the value of the package property *Run64BitRuntime* to *True*. This property takes effect only within SQL Server Data Tools; it has no effect when you execute a package in SQL Server Management Studio or the DTEXec utility. If the package is executed in other contexts, this property is ignored; however, there are other ways to control package execution mode in those contexts.

Performance

Several factors impact the performance of data integration operations. One of the main factors is adapter performance, which is directly related to the performance of the low-level data providers or drivers used by the adapters. Although there are general recommendations (see Table 1-1) for what adapter to use for each popular database, there is no guarantee that you will get the best performance from the recommended adapters. Adapter performance depends on several factors, such as the driver or data provider involved, and the bit mode of the drivers. We recommend that SSIS developers compare performance of different connectivity options before determining which one to use in the production environment.

TABLE 1-1 Recommended adapters for some popular data stores

Database	Recommended adapters
SQL Server	OLE DB Source and Destination
Oracle	Oracle Source and Destination
Teradata	Teradata Source and Destination
DB2	OLE DB Source and Destination
MySQL	ODBC Source and Destination
SAP BW	SAP BI Source and Destination
SAP R/3	ADO.Net Source and Destination



Note Oracle and Teradata connectors are available for download at <http://www.microsoft.com/download/en/details.aspx?id=29283>. Connecting to SAP R/3 requires the Microsoft .NET Data Provider for mySAP Business Suite, which is available as part of the BizTalk Adapter Pack 2.0, available for download at <http://www.microsoft.com/download/en/details.aspx?id=2755>. BizTalk is not required to install the adapter pack or to use the SAP provider. We recommend Microsoft OLE DB Provider for DB2 for connectivity to DB2 and it is available in Microsoft Host Integration Server or in the SQL Server Feature Pack.

Movement of Data Between Systems

The data integration scenario in this section covers moving data between data storage systems. Data movement can be a one-time operation during system or application migration, or it can be a recurring process that periodically moves data from one data store to another. An example of one-time movement is data migration before discontinuing an old system. Copying incremental data from a legacy system at regular intervals to a newer data store, to ensure the new system is a super set of the older one is an example of recurring data movement. These types of transfers usually involve data transformation so that the moved data conforms to the schema of the destination system. The source and destination adapters in SSIS discussed earlier in this chapter can help with connecting to the old and new systems.

You use transform components in SSIS to perform operations such as conversion, grouping, merging, sampling, sorting, distribution, or other common operations on the data that is extracted into the SSIS data pipeline. In SSIS, these transform components take data flow pipeline data as input, process it, and add the output back to the pipeline, which can be of the same shape or different than the input. Transform components can operate on data row-by-row, on a subset of rows, or on the entire data set at once. All transformations in SSIS are executed in memory, which helps with high-performance data processing and transformation. Each data transformation operation is defined on one or more columns of data in the data flow pipeline. To perform operations not supported out of the box, SSIS developers can use scripts or build custom transformations. Built-in SSIS transforms that support some of the most common data operations are as follows:

- **Aggregate** Applies aggregate functions, such as Average, Count, or Group By, to column values and copies the results to the transformation output.
- **Conditional split** Routes data rows to different outputs depending on the data content.
- **Multicast** Distributes every row from input to one or more outputs for branched processing.
- **Lookup** Performs lookups by joining data in input columns with columns in a reference dataset.
- **Merge** Combines two sorted datasets into a single dataset.
- **Sort** Sorts input data in ascending or descending order.

- **Union all** Combines multiple inputs into one output.
- **Data Conversion transform** Converts the data in an input column to a different data type.
- **Derived Column transform** Creates new column values by applying expressions to input columns.

One-time data migrations can range from simple data movement with no transformations to extremely complex movement with more than one source and heavyweight data transformation logic. Packages for complex data movement can evolve from those created for simple one-time data movement to those built from scratch by SSIS developers using SQL Server Data Tools. An employee getting data from a database table and importing it into Excel for further data analysis and processing is an example of simple one-time data movement. Such users do not necessarily have in-depth knowledge of ETL concepts or SSIS features. The Import and Export Wizard in SSIS helps such users build simple data movement solutions. This wizard constructs and uses an SSIS package behind the scenes, hiding the complexity involved in building a package. The packages created by the wizard involve source and destination adapters for the data stores participating in the movement. Figure 1-2 shows a step in the wizard for selecting tables at the source to copy to the destination. After the wizard has created a package, you can save and edit it later using the SQL Server Data Tools environment (discussed in detail later in this chapter). This capability comes in handy for SSIS developers who might have to update packages created by information workers, adding more complex transformations before making those packages available for use by the IT department. Data sources and destinations supported by the Import and Export Wizard include the following:

- Relational databases that support a .NET Framework Provider or OLE DB Provider
- Microsoft Office files: Access and Excel
- Delimited data files in plain text

You can enable simple transformation capabilities in wizard-created packages to carry out data type mapping between a source and a destination. To avoid complexity when dealing with data types, the wizard automatically maps data types of each column selected for data movement at the source to the types of destination columns, using data type mapping files that are part of the SSIS installation for this purpose. SSIS provides default mapping files in XML format for commonly used source and destination combinations. For example, the wizard uses a mapping file called *DB2ToMSSql10.xml* when moving data from DB2 to SQL Server 2008 or a newer version. This file maps each data type in DB2 to the corresponding types in SQL Server 2008 or later. Listing 1-1 shows a portion of this file that maps between the Timestamp data type in DB2 and the SQL Server *datetime2* type.

LISTING 1-1 Data type mapping in *DB2ToMSSql10.xml*

```
<?xml version="1.0" encoding="utf-8" ?>
<dtm:DataTypeMappings
  xmlns:dtm="http://www.microsoft.com/SqlServer/Dts/DataTypeMapping.xsd"
  xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
  SourceType="DB2OLEDB;Microsoft.HostIntegration.MsDb2Client.MsDb2Connection"
  MinSourceVersion="*"
  MaxSourceVersion="*"
  >
```



```

DestinationType="SQLOLEDB;SQLNCLI*;System.Data.SqlClient.SqlConnection"
MinDestinationVersion="10.*"
MaxDestinationVersion="*">
...
<!-- TIMESTAMP 10.* -->
<dtm:DataTypeMapping>
  <dtm:SourceDataType>
    <dtm:DataTypeName>TIMESTAMP</dtm:DataTypeName>
  </dtm:SourceDataType>
  <dtm:DestinationDataType>
    <dtm:NumericType>
      <dtm:DataTypeName>datetime2</dtm:DataTypeName>
      <dtm:SkipPrecision/>
      <dtm:UseSourceScale/>
    </dtm:NumericType>
  </dtm:DestinationDataType>
</dtm:DataTypeMapping>
...
</dtm:DataTypeMappings>

```



Note In SQL Server Integration Services 2012, mapping files are installed to %Program Files%\Microsoft SQL Server\110\DTS\MappingFiles by default. Users can update the default mapping files and also add mapping files to this folder to add support for more sources and destinations in the wizard. The new mapping files must conform to the published XSD schema, and map types between a unique combination of source and destination.

Different usage scenarios of Import and Export Wizard lead to different methods to launch it. For example, a SQL Server database developer or administrator who wants to import data from Microsoft Office Excel to a database could invoke the wizard from SQL Server Management Studio in the context of the destination database. This option allows you to save the package constructed by the wizard and then either execute it later or run it right away.

Another way to invoke the wizard is from SQL Server Data Tools. SSIS beginners who want to start with some basic data-moving package constructs can invoke the wizard from Solution Explorer in SQL Server Data Tools and add a package to the current solution. After adding the package, they can edit it for improvements and then save it just like any other package in the SQL Server Data Tools solution.

One-time migration scenarios often involve copying data objects and data from one instance of SQL Server to another. SSIS supports this scenario with a few tasks that you can use to transfer databases, logins, objects, stored procedures in a master database, user-defined error messages, or SQL Agent jobs between two SQL Server instances. All these transfer tasks use the SQL Management Object (SMO) connection manager to connect to the SQL Servers involved in the transfer.

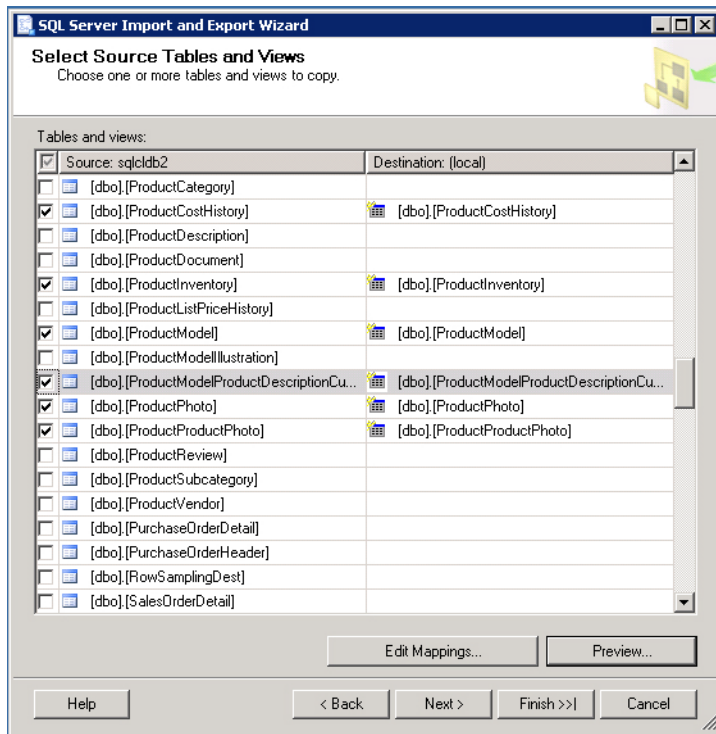


FIGURE 1-2 Make your selections in the Import and Export wizard.

You perform recurring data movements by using a SQL Server Agent job to schedule the execution of the SSIS package that performs the operation.

Loading a Data Warehouse

SSIS is a quintessential ETL tool and platform for traditional data warehouse (DW) loading. Loading DWs is the most popular use for SSIS in enterprises. In data warehousing, data is moved from operational data stores to a central location that's optimized for analysis and reporting. This type of loading can be done either incrementally or as a bulk refresh and it typically involves the following operations:

- Extract all or changed data from multiple sources
- Transform extracted data before loading at the destination
- Load dimension and fact tables at the destination
- Reference data lookup
- Key generation
- Manage historical changes

SSIS can be used effectively to implement all of these operations. Extracting data from multiple sources, transforming data, and loading tables are discussed briefly in the scenarios described earlier in this chapter. Getting changed data is discussed in the next section. This section examines other DW loading operations.

Reference data lookup involves getting data from an external reference dataset. For example, given a customer identifier, a DW loading process might need to retrieve additional data about the customer; for example, getting the customer's zip code from a customer relationship management (CRM) server that stores all customer-related information. In this case, the external dataset on the CRM server is used as the reference to add more data to the ETL pipeline. SSIS supports this step in DW processing using the Lookup Transform component. The reference dataset can be an existing table or view in a relational database, the result of a SQL query, or the Lookup Cache file. SSIS needs reference data in memory for the lookup operation. For large reference sets, the data can be preloaded into a special type of file called Lookup Cache for efficiency and high performance. The Lookup transformation uses either an OLE DB connection manager or a Lookup Cache connection manager to connect to the reference dataset. The Lookup transformation performs lookups by joining data in input columns with columns in the reference dataset. Lookup Transform can use multiple inputs for the lookup operation. Lookup inputs cannot be of certain data types like image or text. For supported string data types, the lookup operation is case sensitive and Character Map transformation has to be used explicitly to convert lookup input to upper or lowercase to match the case in the reference data. Inputs not matched in the reference data can be redirected using the Lookup transform. Connectivity from Lookup transform to relational databases is limited to OLE DB connectivity and the supported data sources are SQL Server, Oracle, and DB2. If the reference data source does not support OLE DB connectivity, a Data Flow task using any supported source adapter and Cache transform is used to create a cache file for lookup operations in another Data Flow task.



Note Lookup transform supports only exact matches when looking up the reference dataset. Fuzzy Lookup transform, discussed in the next section, supports nonexact matching on the reference data. SQL Server 2005 did not support redirection of unmatched inputs.

Generating unique keys and using them to substitute a natural primary key for dimension tables in DWs is a common pattern. These keys are often referred to as *surrogate* or *artificial* keys and they are quite useful in making dimensional updates easier in cases where natural primary keys can change. When the natural key is an alphanumeric or composite surrogate, you use an integer key for better performance. Surrogate key generation during ETL processes is considered better than using key generation at destinations (such as using the identity key feature in SQL Server) during row inserts because referential integrity established at the destination using such destination-generated keys can break during data movement. In contrast, surrogate keys enable data portability. There is no built-in functionality in SSIS to generate keys, but it is quite easy to do. Scripting or custom extensions are popular approaches for building surrogate key generators in SSIS. Surrogate key generation typically involves getting the maximum value of the surrogate key currently used in the table of interest, and using it as a seed to assign a key to each row in the data flow pipeline, with predefined increments.

Some SSIS developers prefer to maintain the seed value within their ETL systems without having to query the database at the beginning of every ETL process.

In DWs, dimensional data can change over time. Such dimensions are commonly referred to as slowly changing dimensions (SCDs). SCD processing and managing historical changes is one of the more difficult steps in DW loading operations. There are three common types of SCDs:

- **Type 1** Old data is overwritten, and historical changes are not preserved. This type is often used with dimensional data that is no longer correct and when the historical values carry no business value.
- **Type 2** Historical changes are preserved, and a row is added for each new value. This is the most common type of change in DW dimensions. Each row (current and historical) of a dimensional value will have its own surrogate key, version number, or timestamp that can be used to get the most recent value.
- **Type 3** Create columns used to maintain current value and history. You would employ this method when changes happen only rarely, or at predefined intervals, and you need to maintain only a few of the most recent historical values.

SSIS supports this operation using the Slowly Changing Dimension transform, which coordinates the update and insertion of records in DW dimension tables. This transformation supports four types of changes:

- **Changing attribute** Supports Type 1 SCD described earlier.
- **Historical attribute** Supports Type 2 SCD described earlier.
- **Fixed attribute** Changes indicate that the column value must not change. Rows that try to change these values are redirected for further processing.
- **Inferred member** A placeholder record is created when dimension values are not yet available.

Type 3 changes are not supported in SCD transforms, but can be handled using a combination of other SSIS components. The Slowly Changing Dimension transformation has one input and up to six outputs. Each output corresponds to the update, insert, or other processing requirements of the record in the dimension table at the destination. During execution, the SCD transform identifies incoming row records with matches in a lookup table using its connection manager. After the match is found, SCD identifies the update type for each row and column, then redirects the row into the appropriate output for handling the change correctly. For example, in Type 1 dimension processing, the SCD transform outputs the row to the Changing Attributes Updates output, which is connected to OLE DB Command transform that updates the record in the dimension table using a SQL UPDATE statement.



Note The OLE DB Command transform in SSIS is a row-based component, and can significantly hurt the performance of SCD processing. The number of dimension value changes to be processed will require an equal number of calls to the database, which is obviously not an efficient approach. There are alternative approaches to design better SCD processing in SSIS, including the following:

- SQL MERGE statement for simple changes
- Custom components for large dimensions and error processing
- The SSIS Dimension Merge SCD Component, available at <http://www.codeplex.com>

Constructing and configuring steps for SCD processing can get complex during ETL design. SSIS supplies a wizard to help developers go through standard steps in SCD management. The wizard produces transformations for SCD processing in the Data Flow task and works only with dimension tables in SQL Server databases. Figure 1-3 shows the typical SSIS components involved in SCD processing. All the components shown (except the source adapter) are added by the wizard.

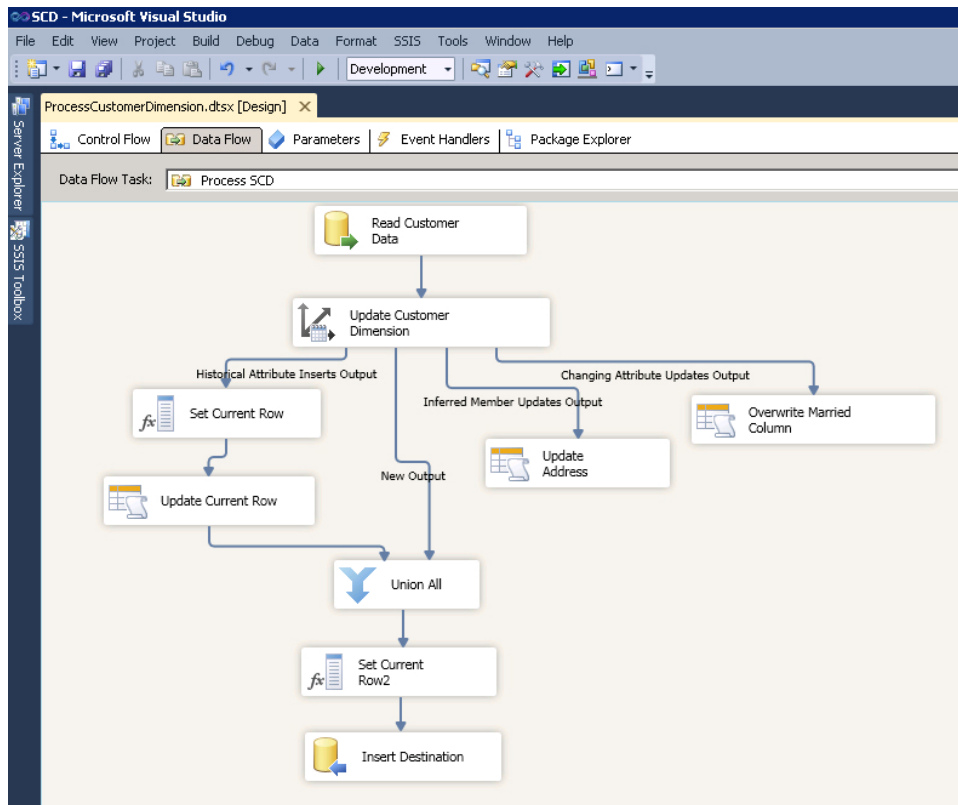


FIGURE 1-3 Slowly changing dimension processing in SSIS.

Cleaning, Formatting, or Standardization of Data

Data in organizations exist in many forms, shapes, and quality levels. Different parts of the organization might use different conventions and formats. During interbusiness transactions, data is acquired from other organizations that might use different data stores and have different data quality standards. Global organizations maintain data in different locales to serve domestic business needs in different geographical regions. Data can also get corrupted during transactions, and such data needs to be isolated during processing.

Data integration processes need to deal with these issues, gathering all the data and ensuring it's in a consistent state before processing the consolidated data in the integration environment or loading it into destinations. Most data integration tools have capabilities to help deal with dirty data, including incorrect spellings, inaccurate or invalid data, duplicates, or unexpected abbreviations.

SSIS provides some data cleansing options suitable for various customer needs, including the Fuzzy Lookup and Fuzzy Grouping transformation components that act as generic data processing operations without requiring an expert collection of domain-specific rules. Fuzzy Lookup helps in matching incoming and potentially low-quality data with a cleaned and standardized reference data set. It outputs the closest match in reference data and quality of the match. Fuzzy Grouping helps in identifying groups of rows in incoming data that are possibly referring to the same entity in a string column leading to duplicate detection in data. SSIS has Data Quality Services (DQS) Cleansing transform in SQL Server 2012. This transform is used to perform data correction and deduplication using knowledge bases built using DQS. During execution time, cleansing work happens in DQS server using the knowledge bases referenced in the transform and data is sent to the DQS server for cleansing. DQS is not a part of SSIS. It is another component in the Microsoft SQL Server product line like SSIS, providing knowledge-driven cleansing functionality.



Note Fuzzy Grouping and Fuzzy Lookup transformation components are not available in all editions of SQL Server. See the later section “SQL Server Editions and Integration Services Features” for details on the SQL Server editions and SSIS features available in each.

In addition to the special-purpose cleansing transforms just described, data standardization and formatting can be done in SSIS using the following features:

- **Character Map transform** Applies string functions to character data.
- **Data Conversion transform** Converts the data in an input column to a different data type.
- **Derived Column transform** Creates new column values by applying expressions on input columns.
- **Data comparisons and replacement** Functions are used in expressions that are computed on input columns.

Cleansing and format manipulation are useful, but in most cases the nature of data has to be well understood before any type of processing happens. SSIS provides a feature called the Data Profiling

task that compiles statistics on the data and can be helpful in identifying cleansing needs and minimizing data quality issues. This task is configured to compute one or more profiles. The results, which are given in XML format, can be stored in a file or in an SSIS variable. Profiling results saved to a file can be viewed in the Data Profiler Viewer. It is possible to control the workflow in SSIS packages using the results of the profiling task.



Note The Data Profiling task works only with data stored in SQL Server 2000 or later versions.

Chapter 9, “Data Profiling and Cleansing Using SSIS,” discusses all these data quality and cleansing capabilities in SSIS in more detail.

Identification, Capture, and Processing of Data Changes

Ever-growing volumes of data in organizations, the need for real-time reports, and a reduced batch window for data processing have all resulted in the demand for change-processing capabilities in data integration tools. Data integration processing is efficient when it can run against an incremental data set instead of the all the data available in the data stores involved. Processing data incrementally reduces the run duration of data integration processes, which in turn can help by increasing the frequency at which these processes run. Using timestamp columns, checksum, or hash-function-based solutions for change detection and capture is a common industry practice. A relatively recent and popular alternative offered by several database vendors provides the native ability to identify changed data. Data integration tools can leverage these features to identify and capture changes for incremental data processing. For example, SQL Server provides Change Data Capture (CDC) and Change Tracking (CT) features, and SSIS provides built-in and custom options to process changed data that can leverage CDC functionality when it is used.

Data integration solution developers can make use of SQL Server CDC infrastructure directly during package design. The CDC feature in SQL Server is powerful, but it is also complex, and involves considerable custom logic. For streamlined CDC processing, SSIS provides the following tasks and components:

- CDC Control task
- CDC Source
- CDC Splitter transform



Note CDC and CT are available in SQL Server 2008 and later versions. The CDC processing tasks and components in SSIS are available in SQL Server 2012.

The CDC Control task is useful for controlling various stages in change data processing in SSIS packages. It requires one connection manager to the database where changes need to be identified and captured, and another (optional) connection manager to persist the state of the CDC processing

operation stored in an SSIS variable. CDC processing stages managed in SSIS using this task include marking the beginning and ending of initial data load, the start of the operation, and the range of data processed. CDC Source extracts changed rows within a specified processing range. Those rows can be obtained by the control task. The source component uses database artifacts generated by SQL Server during CDC setup on the database to be observed for changes. The CDC Splitter transformation routes changed data extracted by CDC Source into three output categories—Insert, Update, and Delete—applying different processing logic to each category. Chapter 8, “Working with Change Data Capture in SSIS 2012,” discusses these components in detail. As mentioned earlier, these CDC components support only SQL Server databases. SSIS supports CDC on Oracle database using a Windows service that mimics Oracle database changes in SQL Server database, enabling change processing using the CDC task and components.

Coordination of Data Maintenance, Processing, or Analysis

The Data Flow task that supports data extraction, transformation, and loading described in previous sections is executed in the context of a workflow defined in the Control Flow section of SSIS packages. Control Flow in SSIS is task oriented, and it coordinates execution of data processing tasks in the business workflow. In addition to the special-purpose Data Flow task powered by a buffer-based data processing engine, SSIS has several built-in tasks that you can use when building the control flow. These tasks, executed by the SSIS runtime engine, are useful for operations such as database administrative steps, preparation for dataflow execution, command execution on Analysis Server, and many other operations common in data integration. You can also build custom tasks using the SSIS programming model and use those as a part of the Control Flow. Execution of scripts is also possible in the Control Flow.

Three types of containers are available to host tasks. The Sequence container is used to group tasks and containers to manage them as one unit. The For Each container facilitates repeating steps in a control flow by enumerating files or objects, and the For Loop container provides another option for repeating steps in a Control Flow that uses a conditional expression. These containers can host other containers along with tasks. Tasks and containers in the Control Flow are connected by precedence constraints that determine the execution path in the workflow. Precedence constraints specify the order in which tasks and task containers are executed during runtime, or conditions that determine which part of the workflow will execute next. The simplicity of using constraints makes orchestration of steps in the SSIS workflow easy to build, debug, and manage.

SSIS includes tasks for performing database maintenance operations. These tasks are useful in building database maintenance plans in SQL Server Management Studio and also in SQL Server Data Tools along with other tasks that you can use when constructing control flow in SSIS. Figure 1-4 shows the SQL Server Management Studio designer in the process of building a database maintenance plan using SSIS maintenance tasks. Some of the popular database maintenance tasks are the following:

- **Backup Database task** Performs SQL Server database backups.
- **Rebuild Index task** Rebuilds indexes in SQL Server database tables and views.

- **Update Statistics task** Updates information about the distribution of key values for one or more sets of statistics on the specified table or view.
- **Shrink Database task** Reduces the size of SQL Server database data and log files.

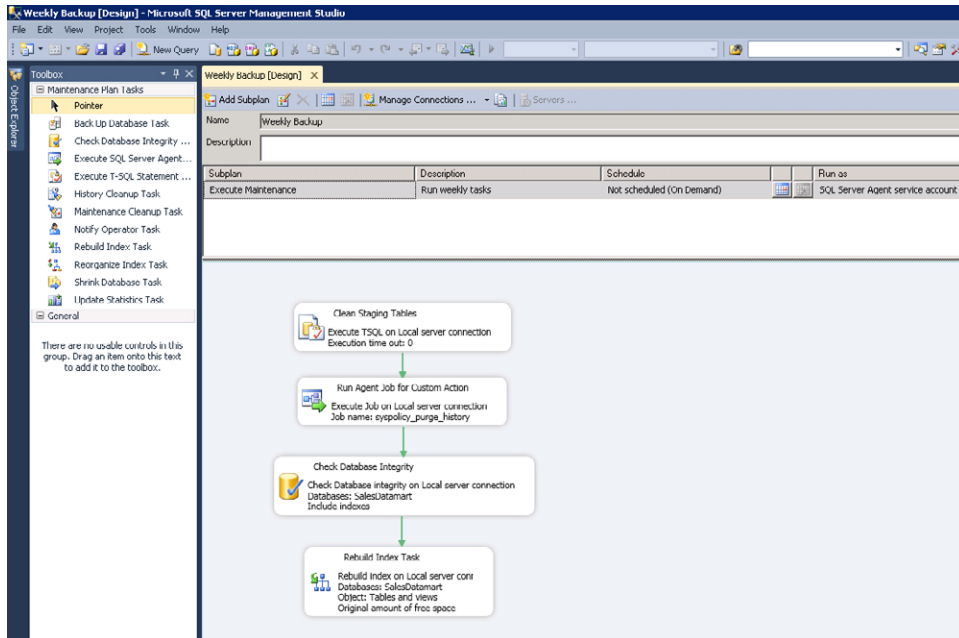


FIGURE 1-4 Database maintenance tasks in SQL Server Management Studio.

Tasks available to build Control Flow are typically used for the preparation of Data Flow task execution. For example, the Execute SQL task is used to run SQL statements on data stores. This task is used for operations like creating or truncating tables, preparing staging database, getting maximum value of an identity column in a table, executing stored procedures, or getting a count of rows in a spreadsheet. The Execute SQL task can be used with a wide range of data sources and supports several connection managers for that. Getting data files from external systems to data integration servers is a common data loading preparation step. SSIS offers a few options for this operation, and the following are the most popular:

- **File System task** Performs operations on files and directories in the file system. For example, this task can be used to get data files from a remote file share and copy them to a newly created directory in the local file system.
- **FTP task** Downloads data files from an FTP server, uploads data files to an FTP server, or manages directories on server. For example, this task can be used to get data from an FTP location for ETL processing and drop the original file from that location after download.
- **Web Service task** Executes a Web service method. For example, this task can be used to get data from a Web service that can be written to a variable or file.

If the data file obtained using one of these tasks is in XML format, you can use the XML task in SSIS to process it. This task can reformat data using XSLT, select XML nodes using XPath queries, or merge multiple XML documents. If the data file is a text or XML file and the data has to be loaded into SQL Server without any transformations, you can use the Bulk Insert task, which wraps the BULK INSERT statement in SQL Server. This provides an efficient way to copy large amounts of data to SQL Server tables or views.

SSIS has two Analysis Services tasks used for executing operations against Microsoft SQL Server Analysis Services databases. The Analysis Services Execute DDL task is used to execute Data Definition Language statements to create, drop, or alter Analysis Services mining models, cubes, or dimensions. The Analysis Services Processing task is used to process these artifacts once they are created.

Any data maintenance or processing functionality not available in SSIS can be implemented using the Script task. For example, if preparation for a Data Flow task involves getting data from a source that does not have a built-in connector from SSIS or if a particular data processing step to be included in Control Flow is not available in SSIS, scripting can help and the Script task is used for that.

SSIS packages can be configured to restart execution from the point of failure if a step in the Control Flow fails. You control package restart using checkpoint files. If a package uses checkpoint files, information about the package execution is written to the file. Therefore, if the package must be restarted after a failure, execution can restart with the failed task or container. SSIS checkpoints are particularly useful to avoid unnecessarily repeating heavyweight operations that have already completed successfully in the Control Flow when rerunning a package, such as copying a huge file, or executing a time-consuming data flow.

Administrative operations and maintenance steps often need to be executed on a regular basis and at scheduled time intervals. SQL Server Agent is used to automate execution of SSIS packages.

Evolution of SSIS

Microsoft introduced Integration Services in Microsoft SQL Server 2005. Data Transformation Services (DTS) in SQL Server 2000 can be considered a predecessor to SSIS; however, SSIS is completely different than DTS in terms of concepts, user interface, feature set, and internal architecture. The transition from a lightweight ETL tool such as DTS to a data integration platform such as SSIS was well received by SQL Server customers. Nearly instant familiarity with the design environment appealed to SSIS developers, and the scalability and performance characteristics of SSIS catered to enterprise needs. The use of SSIS in large-scale data integration projects has been growing steadily ever since its introduction. Figure 1-5 provides some details on the enhancements and features in various versions of SSIS.

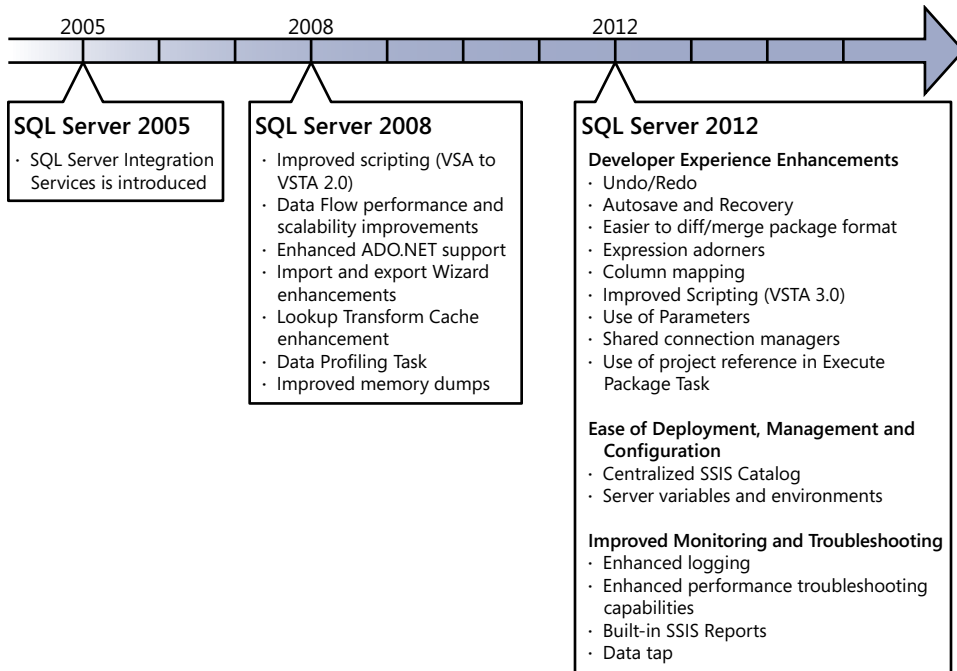


FIGURE 1-5 Evolution of SSIS.



Note Details on data flow engine enhancements in different versions are covered in Chapter 15, “SSIS Engine Deep Dive.”

Setting Up SSIS

You can install all SQL Server features—including SSIS—using a single setup program. The features needed to build, manage, or run SSIS solutions are spread across several SQL Server features, and must be installed appropriately. Figure 1-6 shows the feature selection screen in the SQL Server setup. The main features in SQL Server setup related to SSIS are as follows:

- Integration Services
- SQL Server Data Tools
- Database Engine Services
- Management Tools



Note Setting up a dedicated server for all data integration needs is a best practice. Microsoft recommends installing Integration Services and Database Engine Services on such servers.

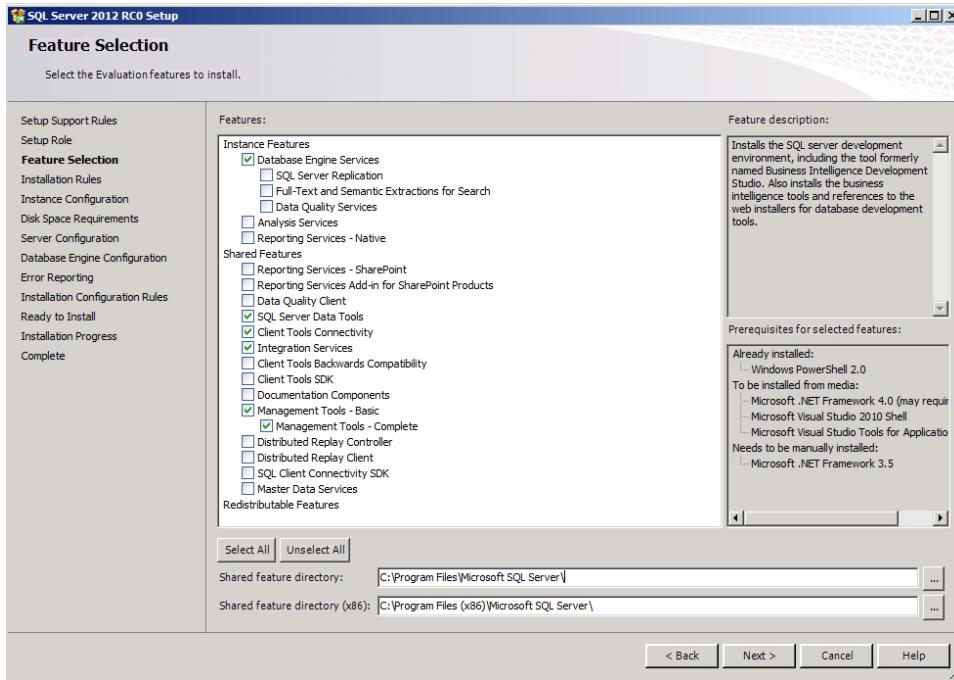


FIGURE 1-6 Feature selection screen in SQL Server Setup.

SQL Server Features Needed for Data Integration

This section takes a brief look at the features that you need to install during SQL Server setup to be able to build and run data integration solutions.

Integration Services

This feature installs the SSIS runtime, some utilities (.exe files), a Windows service named SQL Server Integration Services (followed by a version number), and various libraries (.dll files) used to enable execution of SSIS packages outside of the design environment in SQL Server Data Tools. The Windows service manages locally executed packages and packages stored in msdb in a SQL Server instance. The Integration Services feature is available under Shared Features in the SQL Server feature tree shown in Figure 1-6 because it's not SQL Server database instance specific; that is, it's a feature shared by all instances. So even if a machine has multiple database instances, you need only one copy of Integration Services on that machine. Files installed for this feature in SQL Server 2012 reside in the %Program Files%\Microsoft SQL Server\110\DTS folder. In older versions, the folder hierarchy is

the same but the number indicating SQL Server version (110 in the preceding path) is different. For example, Integration Services in SQL Server 2008 has 100 in the path.

SQL Server Data Tools

The SQL Server Data Tools feature provides the development environment for building SSIS packages. This graphical design tool is an intuitive and easy-to-use environment for composing data integration activities. The designer has separate design surfaces for authoring workflow for data integration and for creating a data flow pipeline. Composition of control or data flows is simple, and involves adding built-in tasks or components into the design surface from SSIS Toolbox, then configuring and connecting them. Integration Services does not have to be installed on the same machine to develop SSIS packages using SQL Server Data Tools. However, if Integration Services is not installed, you cannot execute packages designed in SQL Server Data Tools outside of that feature on the same machine using the DTEXEC utility.

SQL Server Data Tools usage is not limited solely to SSIS development; instead, it is an integrated environment for building SQL Server–based business solutions involving Analysis Services, Reporting Services, or database development. SQL Server Data Tools is hosted in Microsoft Visual Studio. All these technologies support the development experience and need a design environment. Because Visual Studio is the default development environment for Microsoft technologies, it’s the natural choice for SSIS development (as well as others). Visual Studio provides a familiar environment for developers, and the SQL Server Data Tools experience is well integrated with familiar Visual Studio features such as Solution Explorer, the Toolbox, the Properties pane, and the Output and Watch windows. Together, these provide a true, consistent end-to-end experience in building solutions. You can execute SSIS packages in SQL Server Data Tools and leverage other powerful capabilities, such as setting breakpoints and debugging during execution. Business Intelligence Development Studio or SQL Server Data Tools work with specific versions of Visual Studio and SSIS packages. Table 1-2 provides details on this.

TABLE 1-2 Compatibility between Visual Studio and Business Intelligence Development Studio/SQL Server Data Tools in SQL Server

SQL Server version	Name of SSIS development environment	Visual studio version for Business Intelligence Development Studio/SQL Server Data Tools
SQL Server 2005	Business Intelligence Development Studio	Visual Studio 2005 and service packs
SQL Server 2008	Business Intelligence Development Studio	Visual Studio 2008 and service packs
SQL Server 2008 R2	Business Intelligence Development Studio	Visual Studio 2008 and service packs
SQL Server 2012	SQL Server Data Tools	Visual Studio 2010 Service Pack 1+



Note SSIS is a 32-bit application. It runs in WoW64 mode in AMD 64-bit processors. Business Intelligence Development Studio in SQL Server 2005, SQL Server 2008, and SQL Server 2008 R2 is not designed to run in an Itanium 64-bit architecture.

Database Engine Services

An instance of the database engine is installed as a part of this feature. Integration Services packages can be deployed to SQL Server. In SQL Server 2012, packages are deployed to the SSIS catalog, which is a database in a SQL Server instance. In previous versions of SSIS, you could deploy packages to the msdb database in SQL Server. The SSIS catalog is used for management and administration of SSIS packages. The catalog contains system database objects needed for deployment, configuration, and execution of SSIS packages in the context of a SQL Server instance. It also has database objects for monitoring or reporting execution status and troubleshooting data or performance issues. Database engine services features include SQL Server Agent, which is useful for scheduling SSIS package execution. Installing this SQL Server feature installs the components required by the Import and Export Wizard (if that is not already included in the installation through the Integration Services feature during setup). The Import and Export Wizard is included to facilitate getting data into or out of the database engine without explicitly including the Integration Services feature during SQL Server installation.

Management Tools

SQL Server Management Studio is installed as a part of this feature. SQL Server Management Studio is a popular tool among database administrators. It provides a DBA-friendly environment to manage the SSIS catalog, project folders, server variables, and project environment references in Integration Services 2012. After deploying packages to the catalog, you can configure, validate, execute, and monitor them using SQL Server Management Studio. The SSIS catalog is just a database in SQL Server with the name SSISDB, so any feature available in SQL Server Management Studio to interact with or manage a database can be used on the catalog itself. You can also use SQL Server Management Studio to interact with legacy SSIS Windows service or agent jobs that execute SSIS packages. For example, when you design a maintenance plan in SQL Server Management Studio it involves SSIS tasks.

SQL Server Editions and Integration Services Features

Several editions of SQL Server are available, and each edition is associated with a specific usage scenario. The features available in different editions correspond to the target usage scenarios. The functionality of SSIS differs among these various editions. See Table 1-3 for details.

TABLE 1-3 Integration Services features available in different editions in SQL Server 2012

Edition	Features
Express Express Tools Express Advanced Web	Import and Export Wizard Features used by the wizard (SSIS runtime, basic adapters)
Standard Business Intelligence	All SSIS functionality except the advanced features (listed in the next row)
Enterprise Evaluation Developer	Data mining model training destination adapter Dimension processing destination Partition processing adapter SAP BW adapters High-speed Oracle adapters High-speed Teradata adapters Term extraction and lookup transform Fuzzy lookup and groping transform Data mining query transform

The Business Intelligence Edition is available only in SQL Server 2012; it is not available in earlier versions. On the other hand, the Workgroup Edition is not available in SQL Server 2012, but it is available in earlier versions. The SQL Server Datacenter edition available in SQL Server 2008 R2 is not available in SQL Server 2012. If an edition is not available in SQL Server 2012, the SQL Server Installer might change the edition during the upgrade to SQL Server 2012. SQL Server Developer and Evaluation editions have restrictions on usage: The Developer edition is licensed for development purposes only, and the Evaluation edition's license is valid for only 180 days. In editions lower than Standard, the Import and Export Wizard does not allow saving packages or use of other utilities such as the Upgrade Wizard or DTEXec; those are blocked. Some of the high-end editions offer flexible licensing models, which are not discussed in this book.



Note See <http://www.microsoft.com/sqlserver/en/us/editions.aspx> for details about the usage scenarios for the various SQL Server editions.

Summary

This chapter described several common data integration scenarios and provided a high-level overview of the SSIS features that address the requirements for each scenario. SSIS meets most of the requirements for building and managing complex data integration solutions in enterprises. This chapter also discussed the SQL Server features needed for end-to-end SSIS solutions, and the features available in different editions of SQL Server.

New SSIS Designer Features

In This Chapter

The Integration Services Designer	83
Configuration and Deployment	90
New Tasks and Data Flow Components	96
Control Flow	100
Data Flow	102
Scripting	108
Expressions	112
Summary	113

The Integration Services Designer

Visual Studio

The first thing you'll notice when starting a new SSIS project in SQL Server 2012 is that SQL Server Data Tools (formerly Business Intelligence Development Studio) has been upgraded to Microsoft Visual Studio 2010 SP1. Along with the upgrade comes an updated look and feel (shown in Figure 4-1): You'll find many small visual improvements, such as an updated color scheme, new icons, and rounded corners on the tasks and transforms.

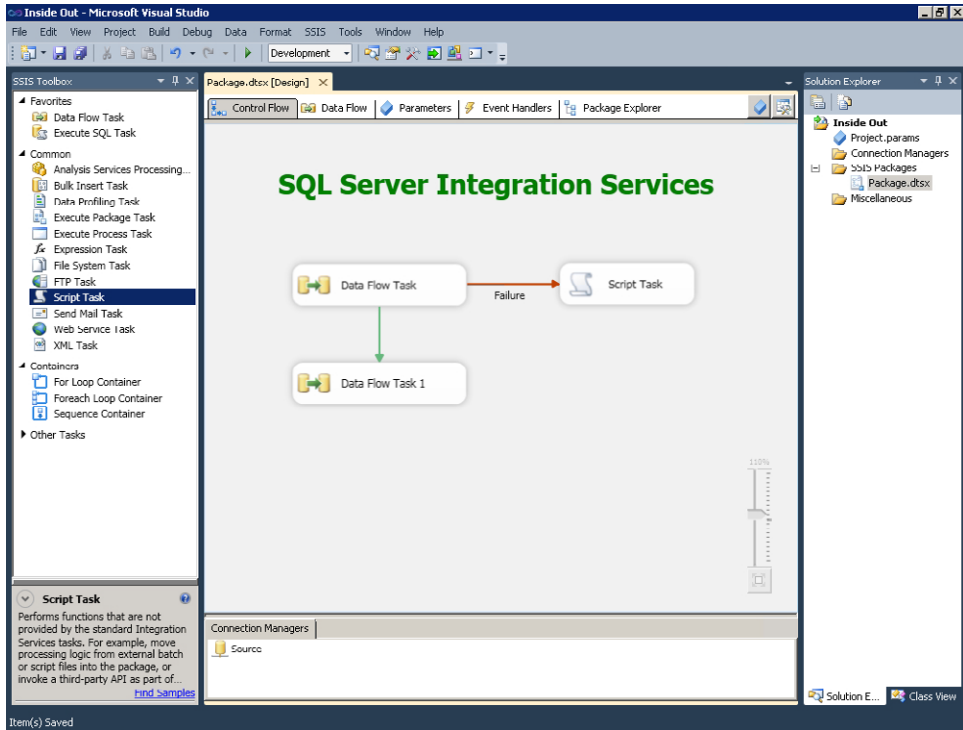


FIGURE 4-1 The SQL Server 2012 designer for Integration Services.



Note Changes were made to way the SSIS Visual Studio project is packaged, so that Microsoft will be able to release an update for SQL Server Data Tools that supports new versions of Visual Studio without waiting for a new release of SQL Server.

Undo and Redo

Perhaps the biggest usability feature missing from previous versions of the product is undo and redo functionality. You can undo most operations within the designer by pressing Ctrl+Z or clicking the Undo button. You can redo changes with Ctrl+Y, or by clicking the Redo button on the toolbar.

Getting Started Window

The Getting Started (SSIS) window (Figure 4-2) appears when creating new SSIS projects. It provides a live feed from Microsoft.com, displaying SSIS news, as well as links to tutorial videos and samples. If the window has been closed, you can open it by clicking the SSIS menu and selecting Getting Started.

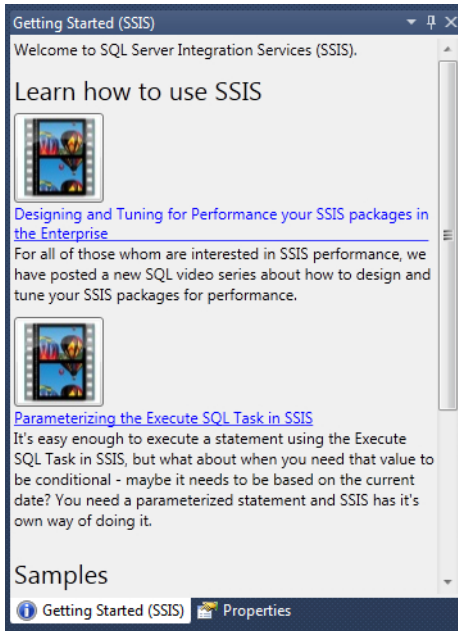


FIGURE 4-2 The Getting Started window.

Toolbox

SSIS now has its own toolbox window. The SQL 2012 toolbox includes significant performance improvements over previous versions, and will automatically display third-party tasks and components installed on your development machine. Items in the toolbox are divided into different categories (which change depending on whether you are looking at the Control Flow or the Data Flow designer). The bottom area of the toolbox displays the name and description of the currently selected item. Most toolbox items will also display in this area a help button and a Find Samples link that will bring up samples that demonstrate how the item can be used within an SSIS package. You can display the toolbox from the SSIS menu by selecting SSIS Toolbox, or by clicking the new SSIS Toolbox button in the upper-right corner of the designer area.

Tasks in the Control Flow toolbox (Figure 4-3) are divided into four categories: Favorites, Common, Containers, and Other Tasks. The Favorites section displays the two most commonly used tasks by default, the Data Flow Task and the Execute SQL Task.

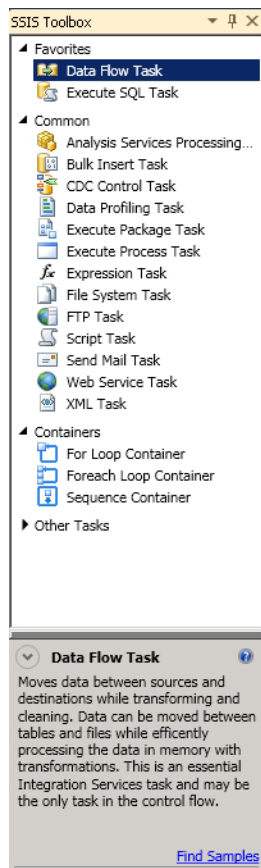


FIGURE 4-3 The toolbox for the Control Flow.

Components in the Data Flow toolbox (Figure 4-4) are divided into five categories: Favorites, Common, Other Transforms, Other Sources, and Other Destinations. The Favorites category contains the new Source and Destination Assistants, which are described later in this chapter.

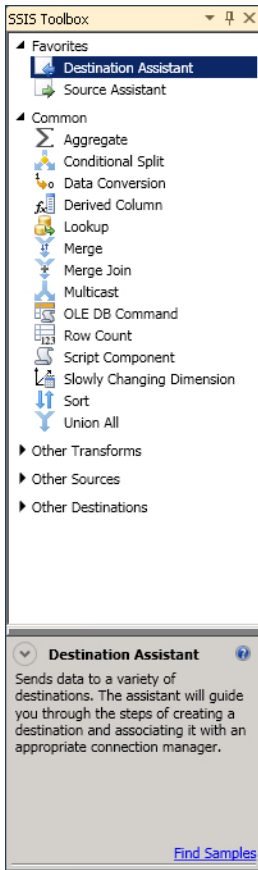


FIGURE 4-4 The toolbox for the Data Flow.

Toolbox items can be moved between categories by right-clicking the item name, and selecting one of the Move To options. The shortcut menu also gives you the Refresh1 Toolbox option, which will look for newly installed third-party tasks and components, and the Reset Toolbox Defaults option, which moves the tasks back to their default categories.

Variables Window

Many usability improvements have been made to the SSIS Variables window (Figure 4-5). It now appears at the bottom of the designer by default, and like most Visual Studio tool windows, it can be docked to any location within the designer. You can display the Variables window from the SSIS Menu by selecting Variables, or by clicking the new Variables button in the top right corner of the designer window.

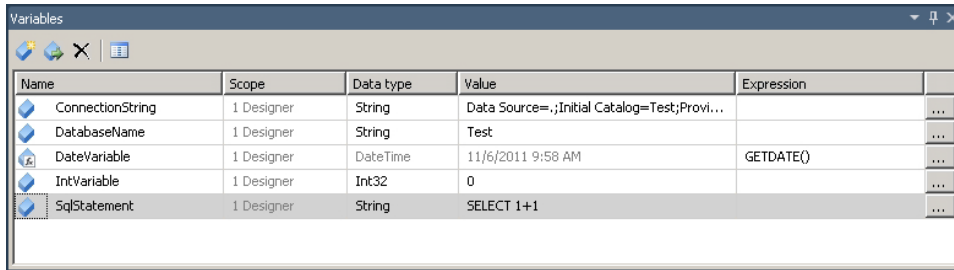


FIGURE 4-5 The SSIS Variables window.

The Variables window shows the default value of a variable. If the variable is set to evaluate using an expression, the expression is displayed in the final column, and the value cannot be modified directly. Clicking the ellipsis button at the end of the row will open the Expression Builder dialog box. Setting a new expression on a variable will automatically set its *EvaluateAsExpression* property to *True*.

In SQL Server 2012, new variables are always created at the Package scope. This is different from previous versions, where the variable would be created in the scope of whichever task or container was currently selected in the designer. The Variables window also contains a new Move Variable button, which allows you to move a variable to a new scope (Figure 4-6).

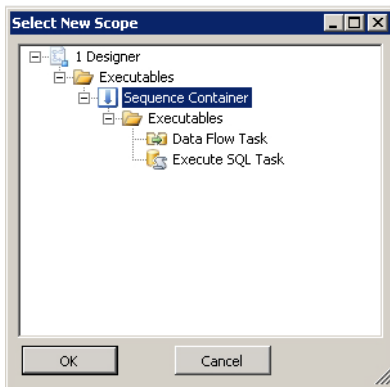


FIGURE 4-6 The Move Variable button allows you to change a variable's scope.

Zoom Control

Both the Control Flow and Data Flow designer surfaces contain a new zoom control in the lower-right corner. This control allows you to easily resize the designer view in (up to 5x size increase) and out (up to 10 percent of the original size). The bottom of the control has a Fit View To Window button that will automatically resize your view so that all package contents fit on one screen.

Autosave and Recovery

SSIS projects now make use of the AutoRecover functionality provided by Visual Studio. This feature allows you to recover your work if SQL Server Data Tools shuts down unexpectedly (due to power failure, crash, or misbehaving third-party add-on). When Visual Studio restarts after an unexpected shutdown, you will be prompted to recover any unsaved changes to your SSIS packages. You can control the AutoRecover settings on the Tools | Options... | Environment | AutoRecover preferences page.

Status Icons

Previous versions of SSIS used colors (green, yellow, and red) to indicate the status of tasks and components while the package was running. The same colors are used in SQL Server 2012, but instead of changing the background color of the task or component, a status icon is displayed in the upper-right corner of the box (Figure 4-7).

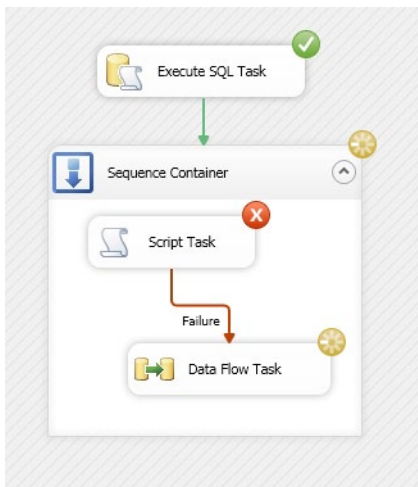


FIGURE 4-7 Status is used to indicate progress instead of changing the background color of the box.



Note Using status icons instead of colors is one of the many accessibility changes that was made for SSIS in SQL Server 2012. Although many people might miss the flashing green and red box colors used in previous versions, not everyone was able to enjoy them, as perception of red and green are affected by the most common form of colorblindness.

Annotations

A number of improvements were made to package annotations, including the following:

- Annotations are persisted in plain text in the package (.dtsx) file. Previous versions of the product had the annotation text encoded in a way that made it very difficult to extract for documentation purposes.
- The annotation window automatically grows as you type.
- Annotations now accept new lines; just press Enter.
- You can easily change the font style, size, and color of an annotation block by right-clicking the annotation block and selecting Set Annotation Text Font.
- You can create a new annotation by clicking the design surface and starting to type.

Configuration and Deployment

Solution Explorer Changes

The Solution Explorer window has been updated for SSIS projects in SQL Server 2012 to support the introduction of the Project Deployment Model. Two different deployment models are supported in this version of SSIS. Figure 4-8 shows a solution with two projects; the top project (named File Based) is in the Package Deployment Model, whereas the bottom project (named Project Based) is in the Project Deployment Model. New projects created in SQL Server 2012 will be in the Project Deployment Model by default. Existing projects from SQL Server 2005 and SQL Server 2008 will open in the Package Deployment Model. You can switch models by right-clicking the project name, and selecting Convert To Package Deployment Model or Convert To Project Deployment Model from the shortcut menu. The nodes that appear under a project will change depending on the active deployment model for that project.

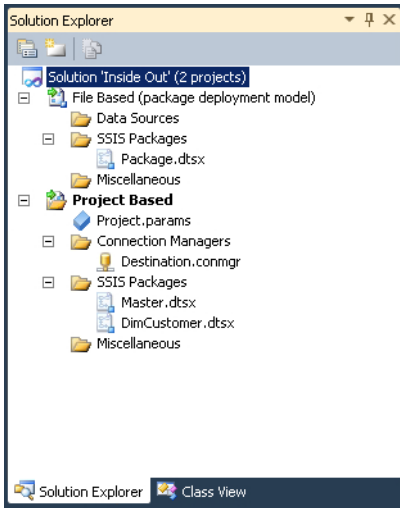


FIGURE 4-8 The SQL Server 2012 Solution Explorer window.

Project Parameters

The Project.params node is a file that stores any project-level parameters. This node will appear for only projects in the Project Deployment Model. Double-clicking this node opens the parameter designer (Figure 4-9).

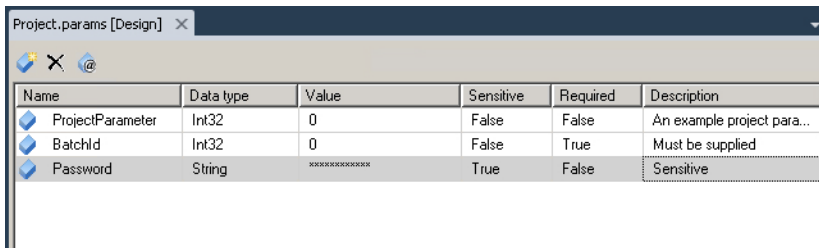


FIGURE 4-9 The designer for project-level parameters.

Connection Managers

SQL Server 2012 allows you to create connection managers at the project level, which get shared among all packages within the project. Each shared connection manager appears as a node under the Connection Managers folder, and is stored in a separate file (with a .conmgr extension). Shared connection managers are available only when using the Project Deployment Model. Shared connection managers replace the use of Data Source (.ds) files from previous versions of the product; the Data Sources folder is shown only when using the Package Deployment Model.

Parameter Tab

Clicking the Parameters tab opens the parameter designer for the current package. This designer allows you to view, create, and edit the parameters exposed by the current package. Parameters function a lot like read-only variables, and they have many of the same properties. Figure 4-10 shows the package parameter designer, with three parameters defined.

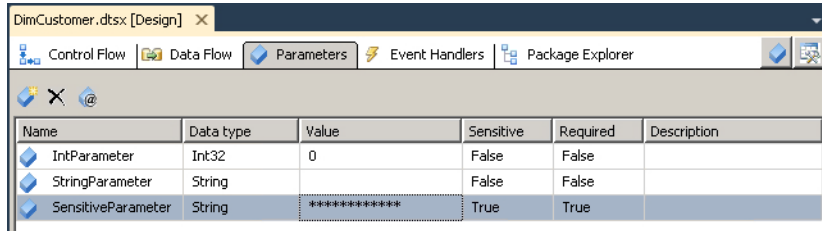


FIGURE 4-10 The package parameter designer.

Visual Studio Configurations

Visual Studio allows a project to contain multiple configurations. In most types of Visual Studio projects, configurations are used to define different build properties (*Debug* vs. *Release*), or settings that affect the way the application runs on a certain platform. SSIS supports the use of Visual Studio configurations in SQL Server 2012, allowing you to externalize your parameter values for different environments. SSIS projects contain a single Development configuration by default. Configurations can be managed by right-clicking the project name in Solution Explorer, selecting Properties, and clicking Configuration Manager.



Note For more information about using Visual Studio configurations, see the Solution Configuration entry in Books Online at <http://msdn.microsoft.com/en-us/library/bb166577.aspx>.

You can externalize a parameter value by clicking the Add Parameter to Configuration button in the parameter designer (Figure 4-11), which opens the Manage Parameter Values dialog box (Figure 4-12). When this is launched from the package parameter designer, you will be able to externalize parameters declared at the package level. Launching the user interface from the project parameter designer allows you to configure project-level parameters.



FIGURE 4-11 The Add Parameter to Configuration button is found in the parameter designer.

The Manage Parameter Values dialog box will show the list of parameters with values that are currently stored within a Visual Studio configuration, with a column for each configuration in the project. To externalize a parameter, click Add and select the parameter from the list. Figure 4-12 shows a package with a single externalized parameter value (*SourceDirectory*). The project has two configurations defined: Monthly and Daily. The current configuration (also known as the active configuration) will be indicated in the column header. In this example, the user has set different values for each configuration. The Monthly configuration (which might represent settings to use for a monthly run of the SSIS package) has a value of C:\InputFiles\Monthly. The Daily configuration (which might represent settings to use for daily runs of the package) has a value of C:\InputFiles\Daily. Clicking Sync will copy the value of the active configuration to all other configurations in the project.

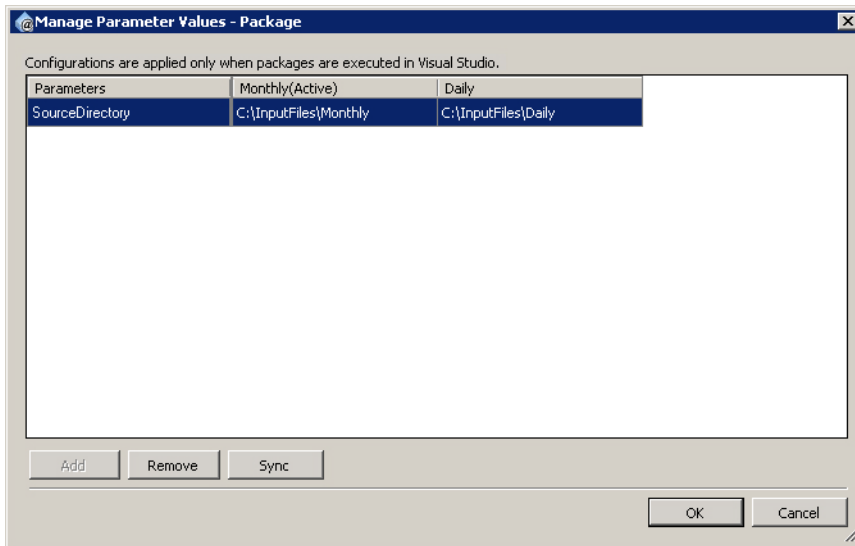


FIGURE 4-12 The Manage Parameter Values dialog box.



Note Parameter configurations are stored within the Visual Studio project file (.dtproj). To save any changes made in the Manage Parameter Values dialog box, you must save the project.

Project Compilation

When using the Project Deployment Model, building an SSIS project produces a project deployment file (.ispac) that contains the packages, shared connection managers, and parameter information for the project. The default values for parameters within the project are taken from the active Visual Studio configuration.

Deployment Wizard

The Integration Services Deployment Wizard (Figure 4-13) is used to deploy project files (.ispac) to an SSIS Catalog. It can also be used to move projects between different SSIS Catalog instances. The wizard can be launched by double-clicking a project file (.ispac), or by running ISDeploymentWizard.exe from the command line. When using the Project Deployment Model, you can also launch the wizard by right-clicking the project in Solution Explorer and selecting Deploy.

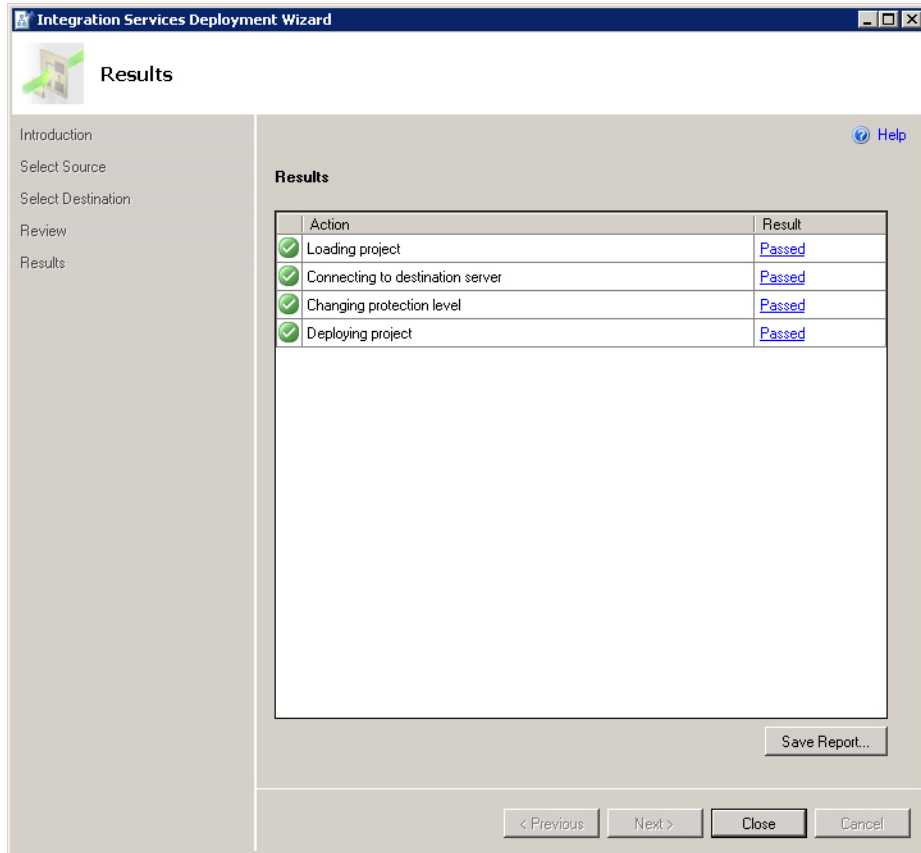


FIGURE 4-13 The Integration Services Deployment Wizard.

Project Conversion Wizard

The Project Conversion Wizard (Figure 4-14) is used to convert SSIS projects in the Package Deployment Model to the Project Deployment Model. One of the main uses of the wizard is to migrate projects created in previous versions of SSIS, so they can take advantage of the new features and functionality in SQL Server 2012.

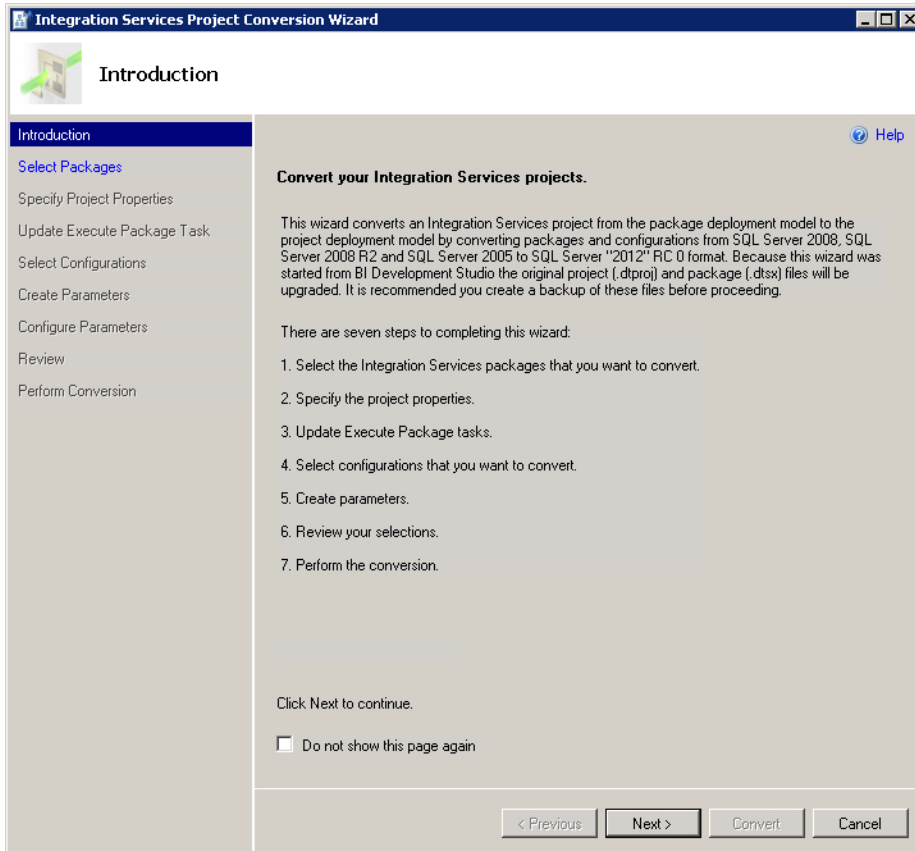


FIGURE 4-14 The Integration Services Project Conversion Wizard.

Import Project Wizard

The Import Project Wizard allows you to create a new SSIS Visual Studio project (.dtproj) from an existing project deployment file (.ispac), or a project in an SSIS Catalog. The wizard can be launched by selecting Integration Services Import Project Wizard when creating a new project in Visual Studio (Figure 4-15). This process is helpful for editing an existing project if you don't have the original source files available.

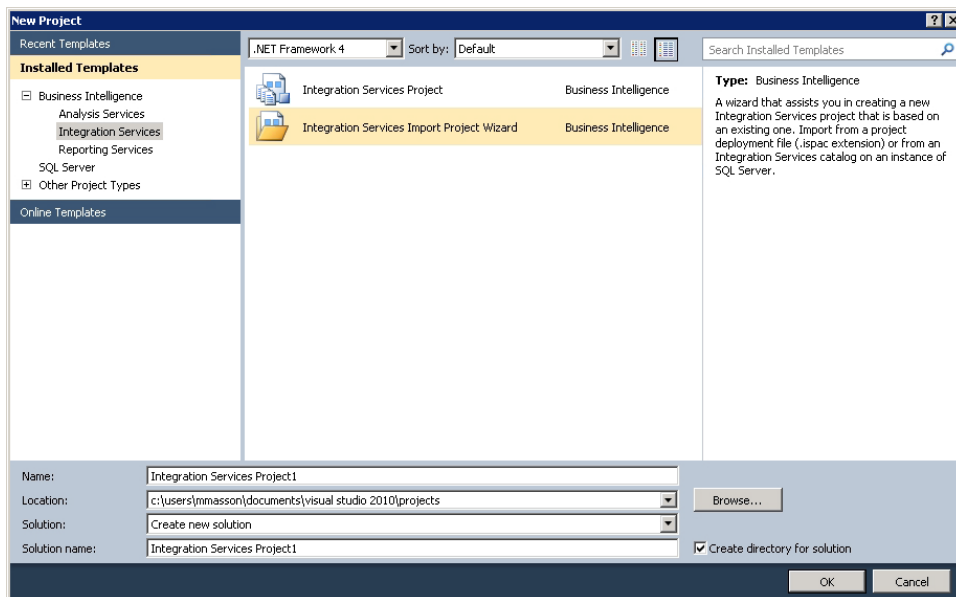


FIGURE 4-15 The Import Project Wizard allows you to create a Visual Studio project from a deployment file.

New Tasks and Data Flow Components

Change Data Capture

SQL Server 2012 introduces a new set of components that make it easier to perform Change Data Capture (CDC) with SQL Server and Oracle. The components include a CDC Control task, a CDC Source component, and a CDC Splitter transformation.



Note More information about the CDC functionality in SQL Server 2012 can be found in Chapter 8, “Working with Change Data Capture.”

CDC Control Task

The CDC Control task (Figure 4-16) is used to set and track the state of your CDC operations.



FIGURE 4-16 The CDC Control task.

You'll typically use this task to mark start and end dates for your CDC process, or to retrieve the next range of dates to process. The task will store all of the state information needed by the CDC Source component in a variable. You can also persist the state information in a database table to maintain state across package executions. Figure 4-17 shows the CDC Control Task Editor user interface.

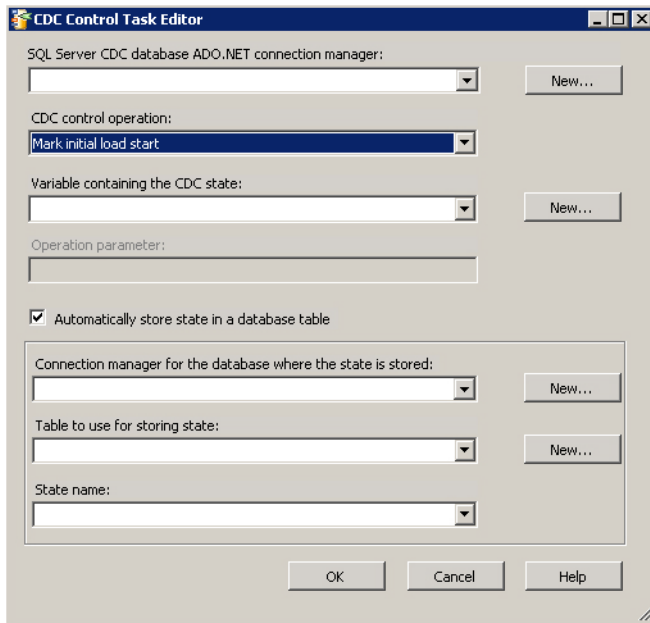


FIGURE 4-17 The editor for the CDC Control task.

CDC Source

The new CDC Source Data Flow component (Figure 4-18) is used to retrieve changed rows from a CDC-enabled table. The CDC Source works with both SQL Server and Oracle sources.

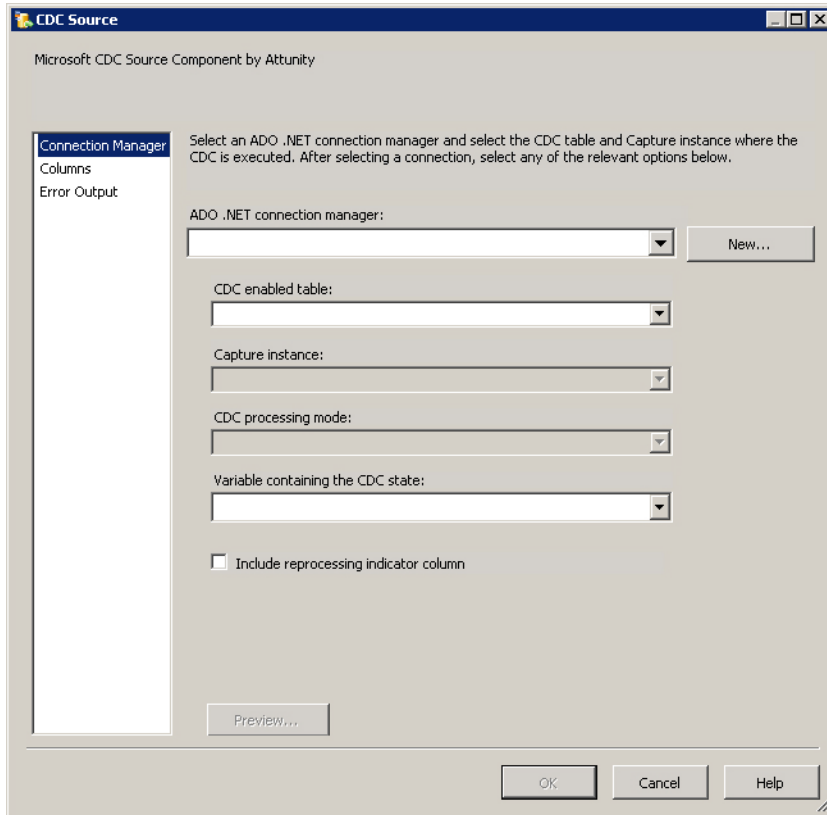


FIGURE 4-18 The CDC Source component.

CDC Splitter

The CDC Splitter transform works a lot like the Conditional Split transform. It will automatically split rows coming from a CDC Source component. Incoming rows will be sent down the Insert, Update, or Delete paths, based on the CDC operation type of the row (Figure 4-19).

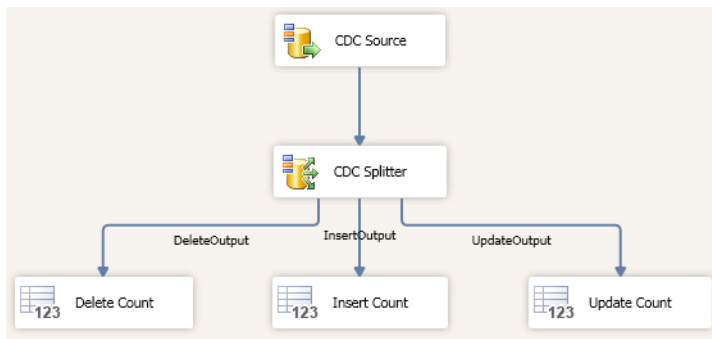


FIGURE 4-19 The CDC Splitter transform processes rows from the CDC Source component.

Expression Task

The Expression task (see Figure 4-20) can be used to explicitly set a variable value using an SSIS expression.



FIGURE 4-20 The Expression task.

The task gives you control over when a value is applied to a variable, and can be used as an alternative to setting the variable's *EvaluateAsExpression* property to *True*. Unlike expressions set directly on the variable (using *EvaluateAsExpression*), the Expression task evaluates only at runtime, not design time. This means that the variable value saved within the package file (.dtsx) is not modified by the Expression task. Figure 4-21 provides an example of an expression being set on the *User::FilePath* variable. Note the syntax is slightly different from regular SSIS property expressions in that it includes the name of the variable to which you are assigning the value.

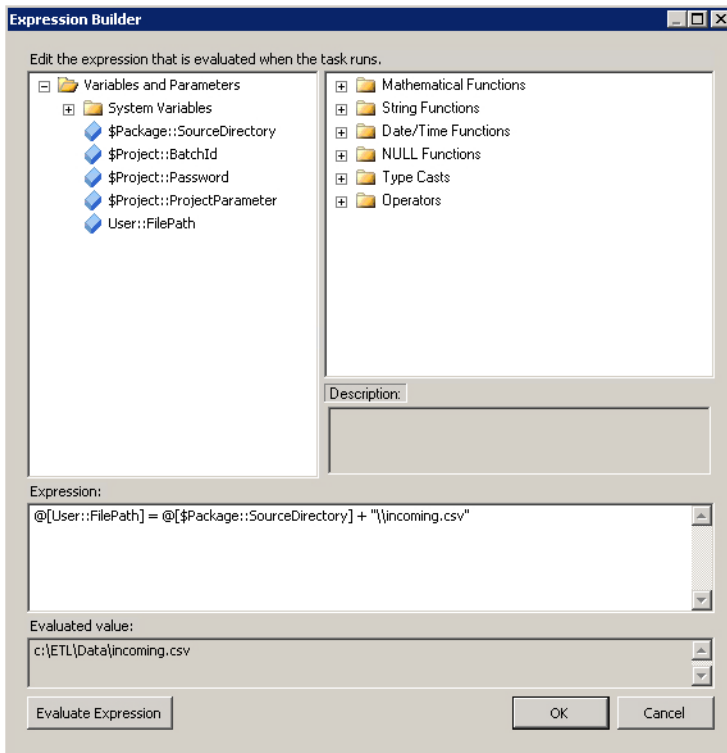


FIGURE 4-21 The Expression task uses the same Expression Builder control as SSIS property expressions.

DQS Cleansing Transform

The DQS Cleansing transform allows you to cleanse data within your data flow using Data Quality Services, a new product in SQL Server 2012. This transform allows you to add one or more columns to domains within a DQS Knowledge Base. At runtime, incoming data will be sent to the DQS server for cleansing.



Note More information about Data Quality Services and the DQS Cleansing transform can be found in Chapter 9, “Data Cleansing Using SSIS.”

ODBC Source and Destination

SQL Server 2008 supported Open Database Connectivity (ODBC) through ADO.NET, but compatibility and performance with certain ODBC providers was not optimal. SSIS has increased its support for ODBC in SQL Server 2012 by providing new Source and Destination components in the Data Flow. Both Source and Destination make use of ODBC connection managers, and the Destination component supports Bulk Insert for most providers.



Note More information about ODBC connectivity can be found in Chapter 7, “Understanding SSIS Connectivity.”

Control Flow

Expression Adorners

There is now a visual indicator when tasks, connection managers, and variables make use of SSIS property expressions. This new feature (called Adorners) adds a white expression icon to the upper-left corner of the object’s regular icon for tasks and variables (Figure 4-22). For connection managers, the expression icon is added to the left of the connection manager’s name.

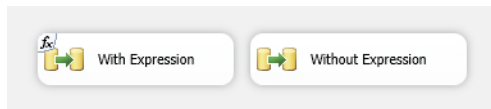


FIGURE 4-22 Tasks with property expressions are now highlighted in the designer.

Connection Managers

The Connection Managers view in the designer window (Figure 4-23) has had a few visual updates in SQL Server 2012. Icons have been updated to show the different connection manager types, and an expression icon is shown for connection managers that use expression properties. Shared connection managers, which are declared at the project level, appear with a (project) prefix before their name.

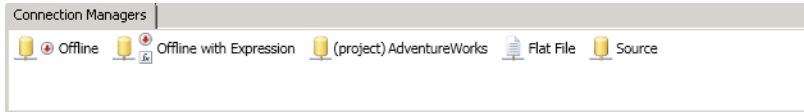


FIGURE 4-23 The new Connection Managers view in the SSIS designer window.

Connection managers can be individually taken offline. This is much like the Work Offline mode, which can be set at the SSIS project level. When a connection manager is offline, all calls to it are essentially short-circuited and return right away. This is useful when a server is offline, as it prevents long waits while connections timeout. A connection manager's connectivity status is validated when a package is opened. If the connection cannot be established, the connection manager is automatically placed in Offline mode. You can put a connection manager online by right-clicking it and selecting Test Connectivity from the shortcut menu. To force a connection manager into an offline state, right-click it and select Work Offline.

Execute SQL Task

The Execute SQL task has a new property called *TypeConversionMode*. This property controls whether the Execute SQL task attempts to perform data type conversion at runtime. For example, imagine the query for the Execute SQL task returns a value of "101" as a VARCHAR(10) data type—a string—and you are assigning it to a package variable with an Int32 data type—a number. In SQL Server 2012, the Execute Package task attempts to convert the value to the package variable type when the *TypeConversionMode* is set to *Allowed*. If the *TypeConversionMode* is set to *None*, strict data type matching is enforced, and you get an error at runtime, the same behavior you'd get in previous versions of SSIS. Packages upgraded from earlier versions of SSIS will have a default *TypeConversionMode* of *None* to preserve compatibility. New Execute SQL tasks will have a default value of *Allowed*.



Note The Execute SQL task uses the .NET Framework to perform the data type conversion. If the conversion is invalid, or there is potential data loss, you will get an error at runtime. For more information about the data type conversion rules, see the *Convert.ChangeType* method entry in Books Online: <http://msdn.microsoft.com/en-us/library/ttf5d382.aspx>.

Connection Assistants

The Source Assistant and Destination Assistant are new Data Flow toolbox items in SQL Server 2012. Dragging them onto the designer will open a wizard that guides you through the process of adding a new source or destination component to your data flow. The first page of the wizard (Figure 4-24) asks you what type of data source you'd like to be connecting to: SQL Server, Excel, or Flat File. After you select the data source type, the wizard automatically determines the best connection manager and source or destination component for you. For example, when you select SQL Server, the wizard will pick the OLE DB provider. You are given the option to create a new connection manager or reuse an existing one. When the wizard completes, it adds a new Source or Destination component to your Data Flow that you can then connect to other transforms.

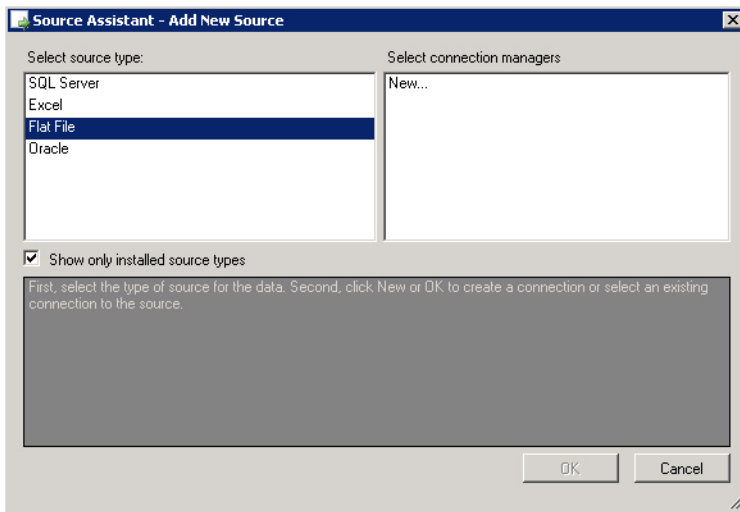


FIGURE 4-24 The first page of the Data Flow Source Assistant Wizard.



Note The Connection Assistant Wizard will display only the 32-bit providers installed on your system. It will not find or recognize any 64-bit providers you have installed.

The Connection Assistant user interface includes a check box that allows the wizard to automatically filter out data sources for which it can't find providers on your system. If you clear this check box, all data sources will be displayed. If you select a data source with no provider installed, a message will be displayed with a hint as to where you can download the preferred provider.



Note The connection assistants are configured through the Connectivity.xml file located under the <Visual Studio Root>\Common7\IDE\PrivateAssemblies folder. You can modify this file to add new connectivity options, as well as change the default provider for each data source.

Improved Column Mapping

A number of usability improvements have been made to the column mapping functionality in the SSIS Data Flow. Data Flow components are now much smarter about the way they map incoming columns. Most will automatically match columns based on name and data type instead of relying on Lineage IDs like they did in previous releases. This means that you can connect a new source component to an existing set of transformations, and SSIS will automatically resolve the column mappings, assuming that all of the expected columns exist. If SSIS is unable to resolve all of the expected column mappings, an error icon is displayed on the path with the missing columns (Figure 4-25).

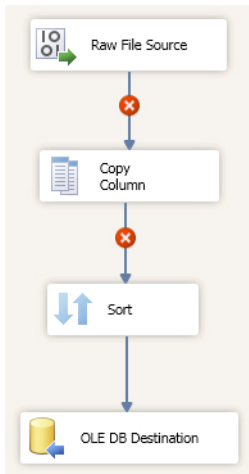


FIGURE 4-25 Error icons are displayed on the Data Flow paths when there are missing columns.

Double-clicking the error icons (or right-clicking the path and selecting Resolve References from the shortcut menu) opens the new Resolve References dialog box (Figure 4-26). This dialog box lets you map output columns to input columns, as well as delete columns that are no longer required. The dialog box has a lot of useful functionality, such as filtering capabilities, automatic mapping based on name and data type, and importing and exporting the column mappings to and from Excel. Resolving column references with this dialog box will fix all column mappings for an entire execution tree.

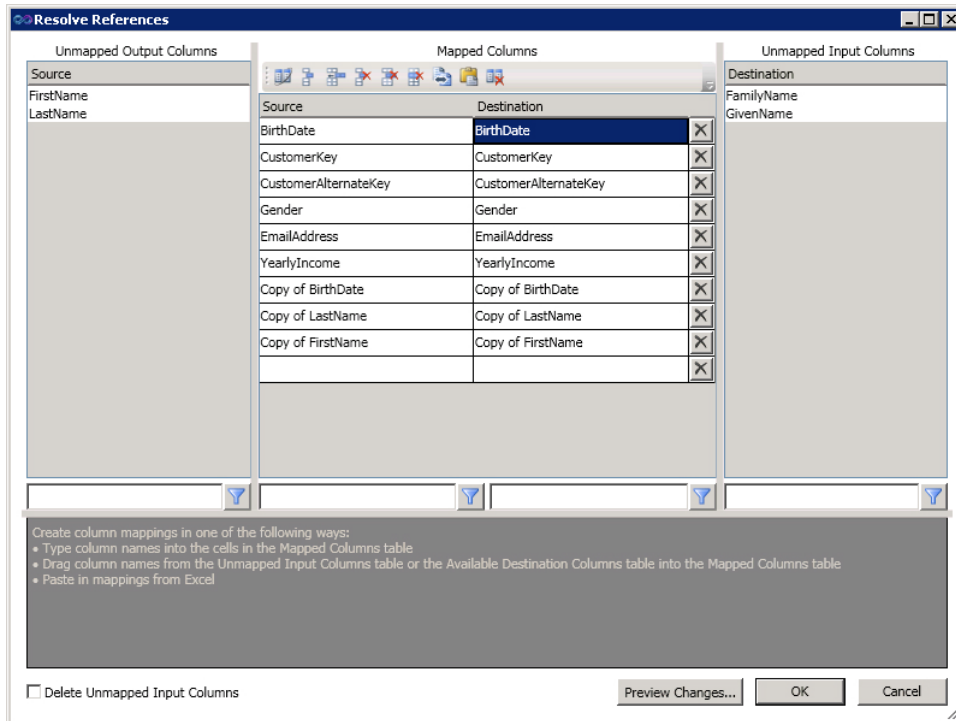


FIGURE 4-26 The Resolve References dialog box.



Note For more information about Lineage IDs, execution trees, and how the SSIS Data Flow does its column mapping, see Chapter 15, “SSIS Engine Deep-Dive.”

Editing Components in an Error State

Previous versions of SSIS did not allow you to open up the editor for a Data Flow component if it was in an error state. You would need to resolve any column mapping errors, or you would be limited to using only the Advanced Editor. Data Flow components remember their metadata in SQL Server 2012, which allows you to edit them even if they are in an error state.

Grouping

SQL Server 2012 allows you to group together one or more objects in the Data Flow. Creating a group in the Data Flow is similar to adding a Sequence container in the Control Flow, except it exists only within the designer. Groups do not affect the way the Data Flow actually runs. Groups can be expanded and collapsed, and are a useful way to hide portions of very complicated Data Flows. Figure 4-27 shows a Data Flow with a group that isolates some of the processing logic. This group could be collapsed to hide the transforms inside.

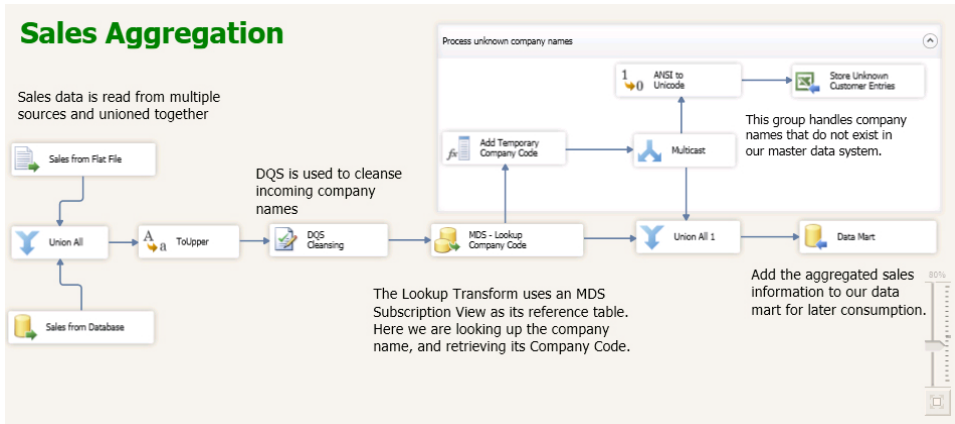


FIGURE 4-27 Complicated Data Flow with a group.

Simplified Data Viewers

Data viewers have been simplified in SQL Server 2012. There is only one type of viewer now, the Grid View. To add a data viewer, right-click the path on which you want to see the data and select Enable Data Viewer. Figure 4-28 shows the icon that indicates an active data viewer. You can configure which columns are displayed by double-clicking the path, and selecting the Data Viewer tab.

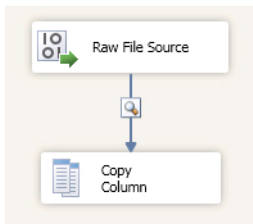


FIGURE 4-28 Data viewers are indicated with an icon on the Data Flow path.



Note All columns in the path are displayed by default when a data viewer is first added. If you change the incoming columns after a viewer has been created, you will need to manually add the new columns to the data viewer for them to be displayed.

Row Count and Pivot Transform User Interfaces

A simple user interface has been added for the Row Count transform (Figure 4-29). The interface has a simple drop-down control and allows you to select any Int32 or Int64 package variables.

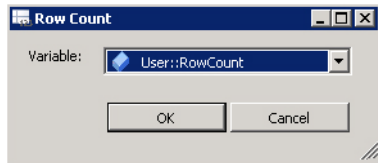


FIGURE 4-29 The Row Count transform now has its own user interface.

The Pivot transform has also been updated to include its own user interface (Figure 4-30), which displays a diagram of a pivot table at the top, allowing you to select the columns to use as the Pivot Key, the Set Key, and the Pivot Value.

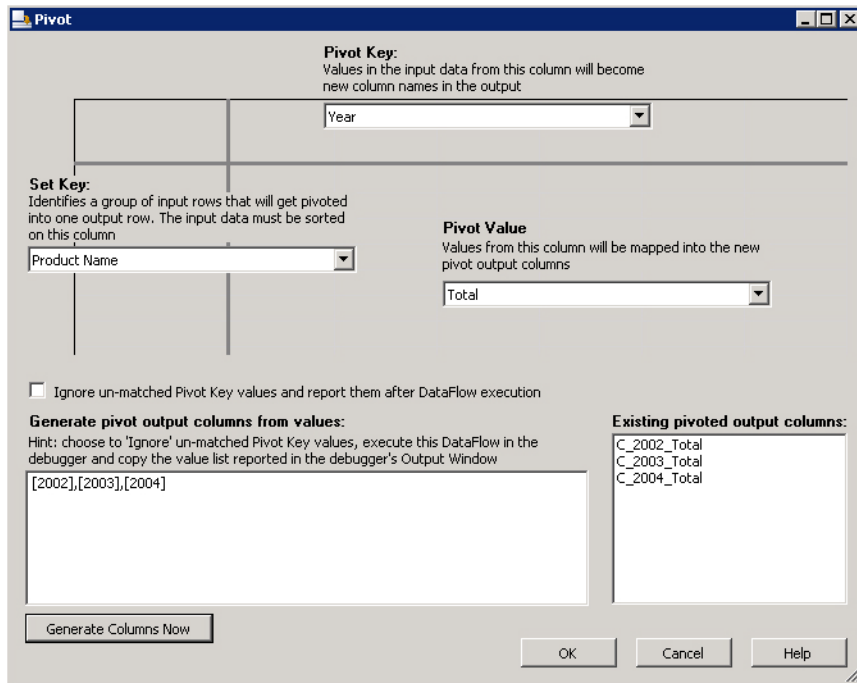


FIGURE 4-30 The Pivot transform now has its own user interface.

Flat File Source Changes

A couple of improvements were made to the Flat File Source parser: It can now support ragged-right delimited files and embedded qualifiers.

Ragged-Right Delimited Files

Ragged-right delimited files can have rows with a varying number of columns. The columns will always appear in the same order, but certain rows will be missing one or more of their trailing columns. These missing columns are meant to be interpreted as *NULL* values. Listing 4-1 shows an example of

this file format. Notice that each row has a *Key* and *AlternateKey* value, but the *SubCategoryKey* and *WeightMeasureCode* fields are optional. The second data row is missing the final field (*WeightMeasureCode*), and the third data row is missing both the final two fields (*SubCategoryKey* and *WeightMeasureCode*).

LISTING 4-1 The ragged-right delimited file format

```
Key,AlternateKey,SubCategoryKey,WeightMeasureCode
211,FT-R92R-F8,14,LB
212,HL-U509-R,31
184,RM-R600
426,FR-M63B-40,12,LB
```

The Flat File Source parser in previous versions of SSIS would not look for row delimiters until it had seen the expected number of columns. Table 4-1 shows how the sample data set would be interpreted in SQL Server 2005 and SQL Server 2008. Notice that the entry for the third data row (key 184) is parsed as a single column.

TABLE 4-1 Results from the ragged-right delimited file in previous versions of SSIS.

Key	AlternateKey	SubCategoryKey	WeightMeasureCode
211	FT-R92R-F8	14	LB
212	HL-U509-R	31	184, RM-R600
426	FR-M63B-40	12	LB

The SSIS Flat File Source now fully supports the ragged-right delimited file format. Table 4-2 shows what the data set looks like when parsed in SQL Server 2012.

TABLE 4-2 Results from the ragged-right delimited file when parsed correctly.

Key	AlternateKey	SubCategoryKey	WeightMeasureCode
211	FT-R92R-F8	14	LB
212	HL-U509-R	31	
184	RM-R600		
426	FR-M63B-40	12	LB



Note The new parsing behavior to support ragged-right delimited files is enabled by default, but can be disabled by setting the *AlwaysCheckForRowDelimiters* property on the Flat File Connection Manager to False. If you do not need this functionality, disabling it might give you a slight performance bonus when working with large flat files.

Embedded Qualifiers

If a field value in a flat file contains the column delimiter (for example, a comma or tab), you can use a qualifier mark when the field value starts and ends. This is typically an apostrophe (') or double quote (") character. To use the qualifying character as a literal within the string, you would typically "escape" it by doubling the character. Listing 4-2 shows a sample data set that uses double quote characters as a text qualifier. The first data row contains a double quote literal character, which is escaped using a second double quote character.

LISTING 4-2 A data set that uses embedded qualifiers

```
Key,AlternateKey,EnglishDescription
396,HS-3479,"High-quality 1"" thread-less headset, with a grease port"
252,FR-R92R-56,"Our lightest and best quality aluminum frame"
261,FR-R38B-62,"The LL Frame provides a safe, comfortable ride"
```

Although text qualifiers were supported, the Flat File Source did not support embedded qualifiers in previous versions of SQL Server. In fact, it was unable to parse data sets like the one in Listing 4-2, and would return an error while parsing the first data row due to the lack of a delimiter immediately after the double quote qualifier. Table 4-3 shows how the Flat File Source correctly interprets embedded qualifiers in SQL Server 2012.

TABLE 4-3 Results from the data containing embedded qualifiers when parsed in SQL Server 2012.

Key	AlternateKey	WeightMeasureCode
396	HS-3479	High-quality 1" thread-less headset, with a grease port
252	FR-R92R-56	Our lightest and best quality aluminum frame
261	FR-R38B-62	The LL Frame provides a safe, comfortable ride

Scripting

Visual Studio Tools for Applications

SSIS uses Visual Studio Tools for Applications to provide support for its .NET scripting functionality. Visual Studio Tools for Applications provides an integrated development environment (IDE), allowing users to embed VB.NET and C# code within their SSIS packages. SQL Server 2012 contains a new version of Visual Studio Tools for Applications (3.0), which is based on the Visual Studio 2010 shell, and fully supports .NET 4.

In SQL Server 2008, Visual Studio Tools for Applications was used both at design time (when the script is created) and at runtime (when the package is run). A number of changes were made in SQL Server 2012 that allow SSIS to fully embed the script and execute the .NET assembly inline without the need to spin up the Visual Studio Tools for Applications runtime. This improvement leads to some minor performance benefits, and also allows the product to support a much requested feature: Script component debugging.

Script Component Debugging

The Script task has always supported debugging, but users were limited to using message boxes and log messages to debug their Script components. Script component debugging is now supported in SQL Server 2012, allowing you to set breakpoints and walk through your code within the Visual Studio Tools for Applications designer. Figure 4-31 shows the debugging of a Script component source.

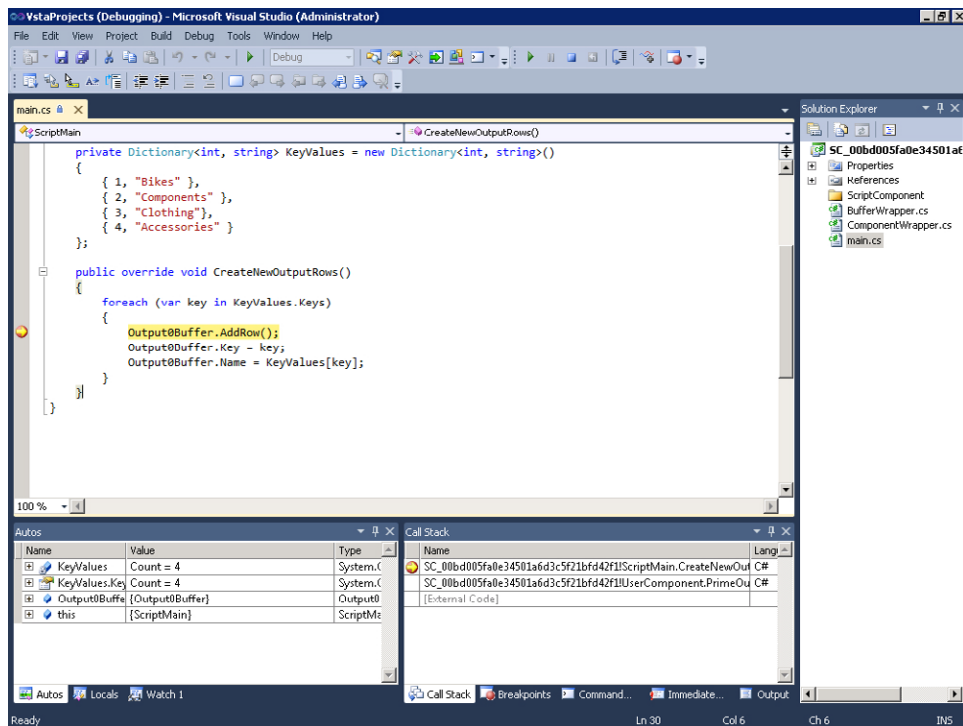


FIGURE 4-31 Debugging a Script component.

To enable debugging, your project must be set to run in 32-bit mode within the Visual Studio designer. This is done by opening the properties for your SSIS project and setting the *Run64BitRuntime* property on the Configuration Properties | Debugging tab to False (Figure 4-32). Because Visual Studio is a 32-bit application, debugging doesn't work when the package runs in a 64-bit mode. This same limitation exists for debugging the Script task as well.

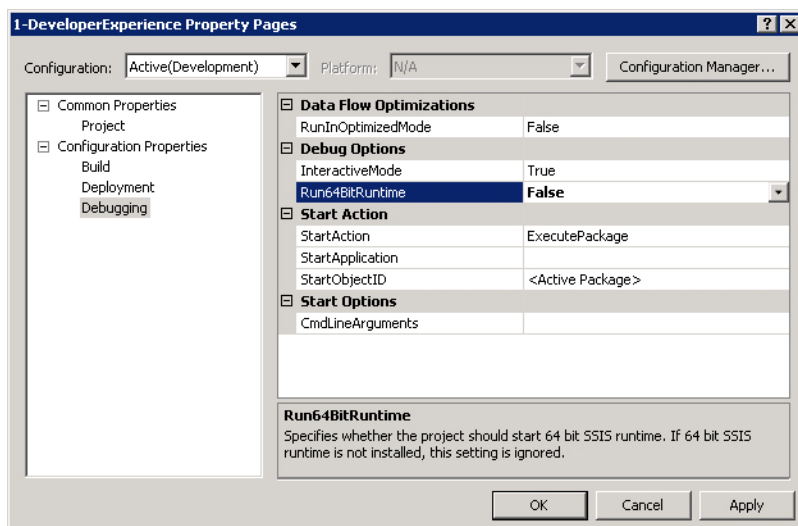


FIGURE 4-32 Enable debugging by setting the project's *Run64BitRuntime* to *False*.



Note Unlike the Script task, there is no visual indicator on the Data Flow design surface when a Script component contains a breakpoint. To remove a breakpoint from a Script component after it has been set, click Edit Script to launch the Visual Studio Tools for Applications editor. You can delete or disable breakpoints individually, or all at once using the Debug | Delete All Breakpoints menu option.

.NET 4 Framework Support

New Script tasks and Script components will target .NET Framework 4 by default. The target framework can be changed in the script project's properties dialog box from within the Visual Studio Tools for Applications designer (Figure 4-33).

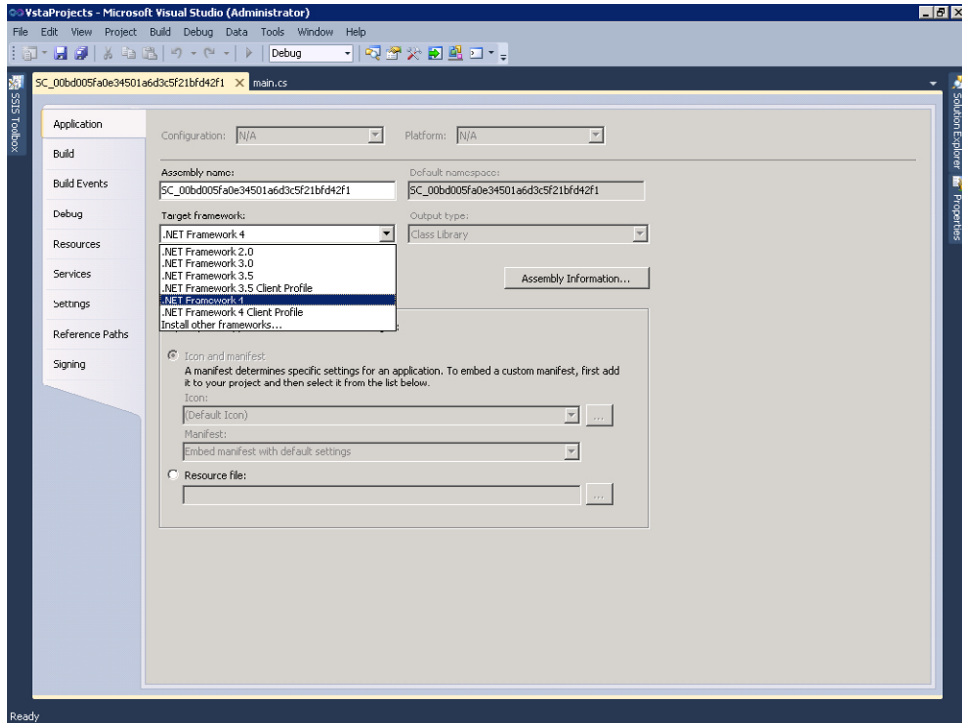


FIGURE 4-33 Enable debugging by setting the project's *Run64BitRuntime* to *False*.

Removal of the Character Limit

SQL Server 2012 has removed the 4,000-character limit in SSIS expressions that existed in previous versions of the product. This limit affected users in two ways; the physical length of the expression string had to be fewer than 4,000 characters, and strings created by the expression would be truncated if they went over the limit. It was possible to set a string greater than 4,000 characters in length to a variable using a Script task, but this workaround is no longer needed. The character limit is gone from all of the places that use the SSIS expression engine, including property expressions, the Derived Column transform, and the Conditional Split transform.

New Expression Functions



Note For examples on how to use the new SSIS expression functions, please see the Functions entry in Books Online: [http://msdn.microsoft.com/en-us/library/ms141671\(v=SQL.110\).aspx](http://msdn.microsoft.com/en-us/library/ms141671(v=SQL.110).aspx).

LEFT Function

The *LEFT* function returns the specified number of characters from the beginning of a string.

```
LEFT ( <string>, <number> )
```

This is similar to the *RIGHT* function, which returns characters from the end of the string. In previous versions of SSIS, you'd accomplish this with the *SUBSTRING* function, starting at the first character.

REPLACENULL Function

The *REPLACENULL* function returns the value of the second expression argument if the first expression argument evaluates to *NULL*.

```
REPLACENULL ( <expression_1>, <expression_2> )
```

If the first expression argument is not *NULL*, then it is returned instead. This function is useful for substituting a *NULL* value with a default value in a Derived Column transform. *REPLACENULL* provides an alternate, easier to use syntax than performing a *NULL* check with the Conditional (? :) operator.



Note The *REPLACENULL* function works with most data types (everything except the LOB types—*DT_IMAGE*, *DT_TEXT*, and *DT_NTEXT*), but both expressions should evaluate to the same data type. If there is a data type mismatch between the arguments, the function will attempt to cast the value of the second expression to the data type of the first. If the cast cannot be done, the expression will return an error.

TOKEN Function

The *TOKEN* function splits the given string (first argument) into a set of tokens using the delimiters (second argument), and returns the specified occurrence (third argument).

```
TOKEN ( <string>, <delimiters>, <occurrence> )
```

Note that the occurrence index starts at 1, and not 0. To get the first token occurrence in the string, you'd use a value of 1. If the specified occurrence is higher than the number of actual tokens in the string, the function returns an empty string. You can use the *TOKENCOUNT* function to determine how many tokens exist in the string. The delimiter argument is a string containing one or more characters. Each character in the string is treated as a separate delimiter.



Note The underlying implementation of the *TOKEN* and *TOKENCOUNT* functions is based on the *strtok* C++ function. For more information on how the function determines how to split the string into tokens, see the entry in Books Online: <http://msdn.microsoft.com/en-us/library/2c8d19sb.aspx>.

TOKENCOUNT Function

The *TOKENCOUNT* function splits a string the same way that the *TOKEN* function does.

```
TOKENCOUNT ( <string>, <delimiters> )
```

Instead of returning individual tokens from the string, it returns the total number of tokens within that string. It can be used to easily determine the value to use to retrieve the last token in a string. For example, when splitting a file path, *TOKENCOUNT* can be used to determine the occurrence value to use to return the file name.

Summary

This chapter provided an overview of the new features and productivity enhancements in the Integration Services designer for SQL Server Data Tools. It explored the new tasks and Data Flow components, and some of the long-awaited features like Undo and Script component debugging. Many of the topics introduced here are further explored in later chapters of the book.

Index

Symbols

- 0 (CDC state), 204
- 32-bit runtime adapters
 - ACE and Jet providers, requirement for, 186
 - Execute Package dialog box and, 181
 - running on 64-bit machines, 178
 - SQL Agent, configuring for, 182
 - T-SQL and, 181
- 64-bit vs. 32-bit architecture, 177–183
 - Package Execution Tools in SSIS Catalog, 180–183
 - SSIS and, 177–178
 - WOW64 application, 178, 178–184
- _operation column (CDC), 216
- <ce-lsn> (CDC state string token), 203
- <cs-lsn> (CDC state string token), 203
- /Env[Reference] (dtexec option), 302, 306
- <ir-end> (CDC state string token), 203
- <ir-start> (CDC state string token), 203
- /IS[Server] (dtexec option), 302, 305
- ##MS_SSISServerCleanupJobLogin## login
 - Catalog maintenance job, as owner of, 430
 - SSISDB restore and, 432
- .NET event handlers vs. SSIS object model events/event handlers, 488
- .NET libraries, PowerShell integration with, 364
- /Pack[age] (dtexec option), 302
- /Par[ameter] (dtexec option), 302
- /Proj[ect] (dtexec option), 302
- <short-error-text> (CDC state string token), 203
- <state-name> (CDC state string token), 203
- <timestamp> (CDC state string token), 203

A

- absolute project references, creating, 348
- AccessMode (ODBC Source component property), 172
- ACE providers, 185–186
- Active Transactions property (CDC instance), 232
- ActiveX scripting, 51
- Add Parameter to Configuration button (parameter designer), 92
- ADO.NET connection manager, Data Profiling task, 241
- ADO .NET Connection Manager property (CDC Source component), 213
- ADO.NET destination
 - Bulk Insert option, 558
 - Data Flow task analysis with, 552
 - destination adapter, 42
 - measuring performance of, 551
- ADO.NET OLE DB bridge provider, 162
- ADO.NET provider dialog box, 162
- ADO.NET Source and Destination (source adapter), 5
- Advanced Editor (Fuzzy Grouping transformation)
 - Exhaustive and MaxMemoryUsage properties, 252
 - overriding default names with, 254
- Advanced Editor (Fuzzy Lookup transformation), 248
- Advanced Editor (SSIS designer), 172, 176
- Advanced tab (DQS Cleansing Transformation Editor), 259
- Advanced tab (Execute dialog box), 317
- Advanced tab (for CDC instance), 232
- AGGREGATE operation, optimization of, 557–568
- Aggregate transformation, 9
 - Comparison Flag options for, 249
- All Connections (SSIS report), 366
- All Executions (SSIS report), 366, 370–371
 - action columns in, 371
 - filters, based on launching point, 370
- All Messages action column (All Executions report), 371

All Messages report, View Context action in

- All Messages report, View Context action in, 375
- All Operations (SSIS report), 366, 371–372
 - filters for, 372
 - SSIS startup stored procedure, using to check for, 421
- All processing mode (CDC), 213
 - net processing mode vs., 214–215
- All Validations (SSIS report), 366, 371–372
 - filters for, 372
 - launching, 371
- All with Old Values processing mode (CDC), 213
 - all (parameter), 215
- Analysis Services Execute DDL task, 20
- Analysis Services Processing task, 20
- Analysis services, standalone server for, 566
- Analysis Services tasks, 20
- Analysis Wizard (Upgrade Advisor), 56
- AND-based multiple constraints, parallelization and, 555
- AND precedence constraints, parallelization and, 555
- Application.LoadPackage() method (Application class), 488
- application programming interfaces (APIs). *See also* T-SQL API
 - managing usage of, 357–358
 - PowerShell vs. managed, 358
- Application step (OVAL technique), 541
- archive mode
 - Create Oracle CDC Instance Wizard, 229
 - enabling, for Oracle database, 229
- Associated SQL Server Instance field (Oracle CDC service), 221
- asymmetric key password
 - creating, 237
 - updating, 236
- AsymmetricKeyPassword (REG_BINARY) registry value, 221
- Attunity connectors (Oracle), 188–189
 - fast loading in, 188
 - LOB data types, support for, 189
 - source for download of, 189
- auto-correction threshold values, DQS Cleansing transformation and, 259
- Automatically Store State In A Database Table (CDC Control Task Editor property), 206
- automation (SSIS), 483–514
 - Codeplex, as source of sample code, 508–509
 - creating packages in bulk, 484
 - custom package runners, creating, 498–502
 - Data Flow, creating (sample code), 491
 - dynamic package generation of, 485–497
 - ETL patterns, applying consistently with, 484
 - Execute Package tasks, using (sample code), 496–497
 - handling design-time events, 486–488
 - metadata-based execution, 497–507
 - package execution, metadata-driven, 484–485
 - packages, creating/saving/loading (sample code), 488–490
 - packages, generating programatically, 483–484
 - PowerShell scripts and, 502–507
 - PowerShell scripts, running in SQL Agent, 505–507
 - PowerShell scripts, using MOM, 502–505
 - projects, working with (sample code), 490–491
 - tasks, inserting from another application (sample code), 495
- Auto Suggest (cleansing status), 258
- auto-suggestion threshold values, DQS Cleansing transformation and, 259
- Average Disk sec/Transfer counter, 545

B

- Backup Database task (database maintenance), 18
- bandwidth (network), 545
- baseline in performance-tuning strategies, 540
- Basic (Logging Level property value), 475
- Batch (ODBC Destination bulk insert), 175
- Batch (ODBC Source bulk fetch), 171
- BatchSize (ODBC Destination component property), 176
- BatchSize (ODBC Source component property), 172
- Ben-Gan, Itzik, xxiv
- BFILE data type, mapping in SQL Server, 238
- BIDS Helper (third-party Visual Studio extension), 509–510
- BINARY_DOUBLE type, mapping in SQL Server, 238
- BINARY_FLOAT type, mapping in SQL Server, 238
- binary large object (BLOB)
 - BLOB type, mapping in SQL Server, 238
 - columns and memory usage, 562
 - temporary storage, 544
- BindCharColumnsAs (ODBC Destination component property), 177
- BindCharColumnsAs (ODBC Source component property), 173
- ANSI option, 173

- BindNumericsAs (ODBC Destination component property), 177
- BindNumericsAs (ODBC Source component property), 173
 - Char option, 173
 - Numeric option, 173
 - Unicode option, 173
- BI xPress (Pragmatic Works), 511–512
- BLOBs. *See* binary large object (BLOB)
- BLOBTempStoragePath property (Data Flow task)
 - modifying, 544
 - pointing to fast drives with, 562
- BlockedSignatureStates key (system registry), 442
- blocking transforms and memory usage, 561
- Books Online
 - Functions entry, 112
 - Solution Configuration entry, 92
 - SQL Server 2008 file format, 116
 - SSIS object model in, 486
- bottlenecks (performance)
 - active Data Flow components as source of, 574
 - identifying, 380
- buffers
 - Data Flow task options for size, 567
 - pipeline task performance and, 561
- Buffers Spooled counter, 543–544
- BufferTempStoragePath property (Data Flow task)
 - modifying, 544
 - pointing to fast drives with, 562
- bulk fetch logic (ODBC Source), 171
- bulk insert logic (ODBC Destination), 175
- Bulk Insert task (OLE DB exclusive component), 164
- Bulk Insert task (SQL Server), 558–560
- Bulk Loading option
 - Bulk Insert task, 558–560
 - data, 558–561
 - partitions and, 560–561
- Business Intelligence Development Studio, 7
 - behavior during upgrade process, 62
 - data viewers and, 592
- Business Intelligence Markup Language (Biml)
 - BIDS Helper and, 509–510
 - Varigence Mist, 510–511
- Bytes Total/Sec counter, 545

C

- C++
 - strtok function, 113
 - TOKEN function, 113
- cache
 - creating, 562
 - misses in, 564–565
 - modes, 563–564
 - sharing a, 565–566
- cache connection manager
 - creating shared cache with, 565
- Cache transform manager, creating cache files
 - with, 562
- CALLER_INFO (execution-specific parameters), 343
- Canceled/Stopped (package execution status), 523
- Candidate Key profile, 242
- Candidate Key request, 243
- canonical value columns (Fuzzy Grouping transformation), 253
- Capture Instance property (CDC Source component), 213
- catalog.add_data_tap_by_guid stored procedure (data taps), 597, 600
- catalog.add_data_tap stored procedure (data taps), 597
- catalog.catalog_properties view, 333
- catalog.configure_catalog stored procedure, 333
- catalog.create_environment stored procedure, 346
- catalog.create_environment_variable stored procedure, 346
- catalog.create_execution stored procedure, 319, 342, 349, 424
- catalog.create_execution_dump stored procedure, 319, 603
- catalog.data_taps view, 599
- catalog.deny_permission stored procedure, 453
- catalog.deploy_project stored procedure, 334–335
- catalog.dm_execution_performance_counters (TVF), 571
- catalog.effective_object_permissions view, 452
- catalog.environment_references view, 349
- catalog.environments view, 349
 - security enforcement on, 346
- catalog.environment_variables view, 349
- catalog.event_message_context view, 477
 - troubleshooting with, 520
- catalog.event_messages view, 476
 - troubleshooting with, 520, 526
- catalog.executable_statistics view, 571, 573
 - troubleshooting with, 520, 525

catalog.executables view

- catalog.executables view, 520
- catalog.execution_component_phases view
 - aggregations, performing on, 595
 - package failures, troubleshooting with, 520
 - performance issues, troubleshooting with, 571, 573, 594–596
- catalog.execution_data_statistics view, 594
 - columns included in, 594
 - logging level required by, 595
- catalog.execution_data_taps stored procedure, 597
- catalog.execution_parameter_values view, 601
 - error dumps and, 343
 - SQL Agent jobs, troubleshooting with, 537
 - troubleshooting with, 520, 524
- catalog.executions view, 319, 340
 - execution duration, finding with, 572
 - troubleshooting with, 520, 523, 525
- catalog.grant_permission stored procedure, 451
- catalog.move_project stored procedure, 339
- catalog.object_parameters view, 337
 - connection managers and, 337
- catalog.object_permissions view, 452
- catalog.operation_messages view, 319
- catalog.packages view, 336
- catalog.projects view, 335–336
- Catalog Properties dialog box (SSMS), 476
- catalog.remove_data_tap stored procedure, 597
- catalog.restore_project stored procedure, 341
- catalog.revoke_permission stored procedure, 451
- catalog schema (SSISDB), 418
- Catalog (securable type), 445
- catalog.set_execution_parameter_value stored procedure, 319, 344, 345, 424
- catalog.set_execution_property_override_value stored procedure, 319
- catalog.set_object_parameter_value stored procedure, 338
- Catalog (SSIS). *See* SSIS Catalog
- catalog (SSISDB schema), 444–445
- catalog.start_execution stored procedure, 319, 342, 424
- catalog.startup stored procedure (SSISDB), 420
 - SQL Server Agent job, running as, 421
- catalog.stop_operation stored procedure, 319, 345, 424, 426
- catalog.set_execution_property_override_value stored procedure, 343
- CDC. *See* Change Data Capture (CDC)
- CDC Control Operation (CDC Control Task Editor property), 206
- CDC Control task, 17, 205–211
 - configuring, 206–211
 - function, 202
 - in SSIS 2012, 202
 - operations of, 207–211
- CDC Control Task Editor, 206
- CDC Enabled Table property (CDC Source component), 213
- CDC in Oracle, 219
- CDC, Oracle databases and, 217–240
 - components for creating, 219
 - data type mapping with SQL Server, 238–239
 - functions, 217–218
 - MSXDBCDC database and, 233–235
 - Oracle CDC Designer MMC, 221–233
 - Service Configuration MMC, 219–221
 - SSIS components and, 240
 - xdbcscvc.exe service executable, 235–238
- CDC Processing Mode property (CDC Source component), 213
- CDC processing range
 - defined, 205
 - determination of, 208
- CDC run, 202
- CDC Service Configuration Microsoft Management Console (MMC), 219–221
 - installation, 219
 - service creation results, 221
 - using, 220–221
- CDC Source component, 17
 - editor for, 212
 - function, 202
 - in SSIS 2012, 202
 - properties, 213
- CDC Source Data Flow component, 211
- CDC Source task
 - components, 211–215
 - configuring, 212–214
 - net vs. all changes in, 214–215
- CDC Splitter component
 - components, 215–217
 - function, 202
 - types of changes, 216
 - using, 216–217
- CDC Splitter transform (SSIS), 17–18
- CDC, SSIS 2012 and, 202–217
 - CDC Control task, 205–211
 - CDC Source Data Flow components, 211–215
 - CDC Splitter components, 215–217
 - CDC state, 202–205

- components in Oracle and, 240
- CDC states, 202–205
 - defined, 202
 - format, 202–203
 - storing, 204–205
 - tokens in, 203
 - types, 204
- cdc_states table, creating, 205
- cdc.xdbcdc_config table (CDC instance configuration data), 222
- cdc.xdbcdc_state table (CDC state of Oracle instance), 222
- CE (CDC state string token), 203
- change data
 - consumption scenarios, 197–198
 - Oracle tables, capturing from, 218
 - querying, 197
 - SSIS 2012 consumption, 202–217
- Change Data Capture (CDC), 17–18, 96–98, 195–240
 - components, 240
 - Control task, 97
 - enabling, 196
 - ETL processes and, 195, 197–198
 - functions, 217–218
 - Oracle and, 18, 217–240
 - Oracle creation components, 219
 - Source Data Flow component, 97–98
 - Splitter transform, 98
 - SQL Server support for, 195
 - SSIS and, 202–217
 - stages, 198–202
 - states, 202–205
 - syncing changes in, 200–202
- change data capture in SQL Server, 195–201
 - CDC in Oracle and, 217
 - CDC stages, 198–202
 - CDC use, 196–197
 - ETLs and, 197–198
- change management. *See* Source Control Management (SCM)
- Change tables, 195
- Change Tracking (CT), 17
- Changing attribute (SCD), 14
- Character Map transform (data integration), 16
- CHAR type, mapping in SQL Server, 238
- CHECK operations and bulk loading, 559
- child package executions
 - DiagnosticEx events and, 531
 - troubleshooting, 529–533
- child packages
 - data taps for, 600
 - storing list of outside package, 485
 - troubleshooting, 377
- cleansed columns (DQS Cleansing transformation), 257
- cleansing status (DQS)
 - column-level, 257
 - descriptions, 258
- CLOB type, mapping in SQL Server, 238
- cmdlets
 - PowerShell, 353
 - SQL Server 2008, 354
- Codeplex
 - BIDS Helper, 509–510
 - EzAPI, alternative package creation tool, 508–509
 - MetaData Driven ETL (MDDE) Studio project, 508
 - package generation sample code on, 508
 - Vulcan project, 509
- code samples
 - downloading, xxvi
 - installing, xxvii
 - using, xxvii
- Collect Diagnostics action (CDC instance), 231
- Column Length Distribution profile, 242
- column-level cleansing status (DQS Cleansing transformation), 257
- column-level confidence score (DQS Cleansing Transformation), 258
- column-level output reason (DQS Cleansing transformation), 258
- column-level similarity score
 - Fuzzy Grouping transformation, 253
 - Fuzzy Lookup transformation, 249
- Column Null Ratio profile
 - Data Profiling task and, 244
 - defined, 242
- Column Null Ratio request, 243
- Column Pattern profile, 242
- Columns property (profile request), 243
- Columns tab (Fuzzy Grouping transformation), 252
- Column Statistics profile, 242
- Column Value Distribution profile, 242
- COM. *See* Component Object Model
- Common component (Control Flow toolbox), 86
- Common component (Data Flow toolbox), 86
- Comparison Flags options (Fuzzy Lookup transformation), 249
- Completed (package execution status), 523

Component Object Model (COM), PowerShell compatibility with

- Component Object Model (COM), PowerShell compatibility with, 353
- components
 - missing, troubleshooting, 518
 - performance measurement of, 542
- composite domains (DQS), 256
- Conditional split (data operation), 9
- Conditional Split (transform component), 43
 - character limit on, 112
 - Data Flow pipelines and, 258
- confidence scores, column-level, 250, 258
- Config command (Oracle CDC service executable), 235
- ConfigFile option (dtexec utility), 307
- configuration files, property paths in, 342
- configuration (Project Deployment Model), 90–96
 - Add Parameter to Configuration button, 92
 - connection managers, 91
 - Manage Parameter Values dialog box, 92–93
 - Parameters tab, 92
 - Project.params node, 91
 - Visual Studio, 92–93
- configuration version
 - updating database, 235
 - updating Oracle service, 235
- Configure SSIS Logs dialog box (SSMS), 478
- Configure SSIS Logs: Package dialog box (SSIS), 464–466
- connection assistants
 - Destination Assistant, 102–103
 - Source Assistant, 102–103
- connection information available in Integration Services Dashboard, 368
- Connection Manager dialog box (SSIS), 163
- Connection Manager For The Database Where The State Is Stored (CDC Control Task Editor property), 207
- Connection Manager property (profile request), 243
- connection managers (Control Flow), 35–36, 161–194
 - Add SSIS Connection Manager dialog box, 161–162
 - location of in file system, 36
 - packages, sharing connections between, 36
 - parameters, catalog.object_parameters and, 337
 - Retain Same Connection property, 36
 - sharing data between tasks with, 36
- Connection Managers tab
 - DQS Transformation Editor, 255
 - Execute dialog box, 316
- ConnectionString property, 528
- connection strings (error context), 527
- connection type (cache mode option), 564
- connectivity, 161–194
 - 64-bit vs. 32-bit architecture, 177–183
 - ACE providers, 185–186
 - ADO.NET OLE DB bridge provider, 162
 - considerations for, 177–183
 - custom components, creating, 189–191
 - Jet provider, 184–185
 - Microsoft Excel and Access, connecting to, 184–186
 - ODBC, 166–177
 - ODBC Destination component, 174–177
 - ODBC Source component, 169–174
 - Oracle, 186–189
 - performance considerations with, 164–165
 - providers for, 162–163
 - script components, using, 191–193
 - task/transform supportability, 164
- connectors and infrastructure optimization, 567
- connect strings
 - creating, 237
 - updating, 236
- Connect to Oracle Source Database screen (Create Oracle CDC Instance Wizard), 225
- ConsoleLog option (dtexec utility), 309, 310, 479
- Containers component (Control Flow toolbox), 32–35, 86
 - Foreach Loop, 33
 - Foreach Loop enumerator, 33
 - For Loop, 34
 - nesting, 35
 - Sequence, 34–35
- context depth, 527
 - logging and, 477
- Control Flow (SSIS), 18–20, 28–40, 100–101
 - Adorners, 100
 - Analysis Services tasks, 20
 - connection managers, 35–36
 - Connection Managers, 101
 - containers, 32–35
 - container types, 18
 - Control Flow toolbox, 86–87
 - Data Flow task and, 18
 - event handler, 40
 - Execute SQL task, 19, 101
 - log provider, 38–39
 - packages, 36
 - parameters, 37–38

- precedence constraints on tasks, 30–31
- projects, 36–37
- tasks, 28–30
- task types, 19–20
- variables/expressions in, 31–32
- zoom control in, 88
- Convert To Project Deployment Model selection (SSDT), 71
- Copy data (data viewer control), 591
- CopyReferenceTable property (Fuzzy Lookup transformation), 248
- CoraclDC Instance Deployment Script action (CDC instance), 231
- Correct (cleansing status), 258
- Corrected (cleansing status), 258
- counters
 - Buffers Spooled, 543–544
 - LogicalDisk, 545
 - Memory performance, 546
 - Network, 545
 - Processor, 545
 - SSIS Catalog, 546
 - Windows, 542
- Counters (CDC instance), 232
- CozyRoc (third party extensions), 512
- CPU utilization, measuring, 546
- Create a new CDC service function (CDC Service Configuration MMC), 220
- Create CDC Database screen (Create Oracle CDC Instance Wizard), 224
- Create CDC Service Master Password field (new Oracle CDC service window), 221
- Create command (Oracle CDC service executable), 236
- Created (package execution status), 523
- CREATE LOGIN T-SQL statement, 432
- Create Oracle CDC Instance Wizard, 225
 - Connect to Oracle Source Database screen, 225
 - Create CDC Database screen, 224
 - creating instances with, 223
 - enabling supplemental logging, 228
 - executing setup steps in, 229
 - Select Oracle Tables and Columns screen, 226
 - Table Properties dialog box, 228
 - Table selection dialog box, 227
- Create Parameters page (Project Conversion Wizard), 77
- Create, Read, Update, and Delete (CRUD) operations, 558

- Create Relationships editor (Fuzzy Lookup transformation), 248
- Create Relationship window (Fuzzy Lookup transformation), 250
- Create statement (cdc_states table), 205
- CS (CDC state string token), 203
- Current Bandwidth counter, 545
- Current Change Number property (CDC instance), 232
- Currently Processing property (CDC instance), 232
- custom connectors, vendor generated, 165
- Custom Properties tab (ODBC Source Advanced Editor), 172, 176

D

- data
 - changes, identification, capture, and processing of, 17–18
 - consolidation from heterogeneous sources, 4–9
 - data cleansing, 241–262
 - increasing retention of, 3
 - integration, 16–17
 - maintenance, processing, analysis of, 18–20
 - movement between systems, 9–12
 - operations on, common, 9
 - troubleshooting, 589–604. *See also* troubleshooting data
 - Database Engine Services feature, 52
 - database maintenance, 18–19
 - database migration
 - backup/restore procedure, using, 430
 - database snapshots (SQL Server), 200
 - data cleansing, 241–262
 - Data Profiling task, 241–246
 - Data Quality Services, 254–261
 - Data Quality Services (DQS) Cleansing transform, 16
 - Fuzzy Grouping transformation, 16, 251–254
 - Fuzzy Lookup transformation, 16, 246–251
 - data consolidation, 4–9
 - acquisitions/mergers and, 4
 - aggregate (data operation), 9
 - Conditional split (data operation), 9
 - Data Conversion transform (data operation), 10
 - data type support, 7
 - Derived Column transform (data operation), 10
 - lookup (data operation), 9
 - mapping files, 10–11

data consolidation, *continued*

- merge (data operation), 9
- metadata and, 7
- Multicast (data operation), 9
- Open Database Connectivity (ODBC), 5
- performance considerations with, 8–9
- Slowly Changing Dimension transform, 14–15
- sort (data operation), 9
- source adapters, 4
- SQL Management Object (SMO) connection manager, 11
- union all (data operation), 10
- Data Conversion transform, 10, 16, 43
- data dependency and parallelization, 553
- data errors, popular solutions for, 593
- Data Flow components
 - active vs. total time elapsed, finding, 574–576
 - catalog.execution_component_phases view, troubleshooting performance issues with, 573
 - creating/configuring dynamically, 491–495
 - design-time component, adding, 493
 - Execution Performance reports and, 380
 - IDTSCoordinateEvents interface and, 492
 - metadata component, adding, 492
 - ODBC Destination, 174–177
 - OLE DB Destination component and, 494
 - phases of, 571
 - time spent on, finding, 573–574
- Data Flow Components Information (Execution Performance report), 379
 - logging level requirements for, 380
- dataflow_path_name column (catalog.execution_data_statistics view), 594
- Data Flow paths
 - identification string, finding, 598
 - monitoring in SSDT, 591–592
 - troubleshooting, 593–603
- Data Flow Plus extension (CozyRoc), 512
- Data Flow (SSIS), 41–43, 102–108
 - column mapping in, 103–104
 - components, editing in error state, 104
 - Data Flow toolbox, 86–87
 - data viewers, 105
 - destination adapter, 42
 - Destination Assistant, 102–103
 - DQS Cleansing transformation in, 260
 - Flat File Source, 106–108
 - Fuzzy Grouping transformation in, 254
 - Fuzzy Lookup transformation component, 251
 - grouping objects in, 104–105
 - Lineage IDs, 103
 - ODBC Source (source component), 169–174
 - Pivot Transform UI, 105–106
 - precedence constraints and, 30–31
 - Refresh1 Toolbox option, 87
 - Resolve References dialog box, 103
 - Row Count UI, 105–106
 - source adapters, 41
 - Source Assistant, 102–103
 - SSIS data flow data type, 41
 - transforms, 43
 - zoom control, 88
- Data Flow tasks
 - analysis, 551–552
 - Control Flow and, 18
 - counters for, 542
 - data staging with, 557
 - data viewers, troubleshooting with, 590
 - Destination component performance, 550–551
 - instrumenting, 547–548
 - OLE DB commands and performance of, 557
 - overall performance, 546–547
 - performance measurements, 546–552
 - performance-tuning options, 567
 - Source component performance, 549
 - Transformation component performance, 549–550
 - troubleshooting, 589–590
- data insertion queries, optimizing, 559
- data integration, 16–17
 - CDC Control task (SSIS), 17
 - CDC Source (SSIS), 17
 - CDC Splitter transform (SSIS), 17–18
 - Change Data Capture (CDC), 17–18
 - changes and, 17–18
 - Change Tracking (CT), 17
 - Character Map transform, 16
 - cleansing, data, 16
 - Data Conversion transform, 16
 - Data Flow tasks and, 41–43
 - data profiling and, 241
 - Data Profiling task, 16–17
 - Derived Column transform, 16
- data integrity constraints and partitions, 560
- data loading stage (CDC), 198–200
- data mart loading, parallelizing, 553
- Data Mining Model Training destination adapter, 42
- data operations, 9
- data processing, CDC for changes in, 197
- Data Profile Viewer, 245–246

- Data Profiling Task Editor
 - Profile Requests page, 244
 - profiling single dataset with, 242
- Data Profiling task (SQL Server 2000), 241
- Data Profiling task (SSIS), 16–17, 241–246
 - configuring profiling for, 243–245
 - Data Profile Viewer and, 245–246
 - execution, 244
 - operations, 241–242
 - output, 244
 - uses, 241
- Data Protection API (DPAPI), 221
- data quality management. *See* data cleansing
- Data Quality Services Client
 - DQS Cleansing transform vs., 255
 - DQS installation, 259
- Data Quality Services (DQS), 254–261
 - Cleansing transform, 16
 - configuration, 255
 - execution, 260–261
 - inputs and outputs, 256–259
 - uses, 254–255
- Data Row toolbox
 - Destination Assistant, 102–103
 - Source Assistant, 102–103
- data sources during upgrade process, 70
- data staging
 - OLE DB commands vs., 557
 - SQL Server optimization with, 556–557
- data taps, 596–600
 - adding by GUID, 600
 - adding by name, 597–599
 - catalog.data_taps view (SSIS Catalog), 599
 - destination file for, 598
 - permissions required to set, 599
 - T-SQL, creating with, 597
- Data Transformation Service (DTS), migration
 - from, 51
- data types (SSIS)
 - comprehensive list of, source for, 7
 - mapping, in SQL Server and Oracle, 238–239
 - mismatched, troubleshooting, 517
- data viewers
 - Business Intelligence Development Studio and, 592
 - controls available for, 591
 - discontinued support for, 51–52
 - Grid View, 105
 - information provided by, 592
 - troubleshooting data with, 590–592
- data warehouse(s)
 - loading with SSIS, 12–15
 - Lookup Cache file, 13
 - Lookup Transform, 13
 - slowly changing dimensions (SCDs), 14
 - Slowly Changing Dimension transform, 14
- DATE type, mapping in SQL Server, 238
- db_dtsadmin role (msdb database), 437
- db_dtsltduser role (msdb database), 437
- db_dtsoperator role (msdb database), 437
- dbo.sysssislog table (MSDB database), 437
- dbo.sysssispackagefolders table (MSDB database), 437
- dbo.sysssispackages table (MSDB database), 437
- dbo.xdbcadc_add_service(servicename,sqluser) procedure (MSXDBCDC database), 234
- dbo.xdbcadc_databases table (MSXDBCDC database), 233
- dbo.xdbcadc_disable_db(dbname) procedure (MSXDBCDC database), 234
- dbo.xdbcadc_enable_db(dbname,svcname) procedure (MSXDBCDC database), 234
- dbo.xdbcadc_remove_service(servicename) procedure (MSXDBCDC database), 234
- dbo.xdbcadc_reset_db(dbname) procedure (MSXDBCDC database), 234
- dbo.xdbcadc_services table (MSXDBCDC database), 233
- dbo.xdbcadc_start(dbname) procedure (MSXDBCDC database), 235
- dbo.xdbcadc_stop(dbname) procedure (MSXDBCDC database), 235
- dbo.xdbcadc_trace table (MSXDBCDC database), 233
- dbo.xdbcadc_update_config(dbname) procedure (MSXDBCDC database), 235
- ddl_cleanup_object_permissions (DDL trigger), 453
- Debug mode (SSDT), 589, 591
- DefaultBufferMaxRows property (Data Flow task), 567
- DefaultBufferMaxSize property (Data Flow task), 567
- Delete action (CDC instance), 231
- DELETE changes
 - CDC for, 195
 - CDC Splitter component, 216
 - syncing, 200
- Delete command (Oracle CDC service executable), 237
- Delimiters property
 - Fuzzy Grouping transformation, 252
 - Fuzzy Lookup Transformation, 248

DENSE_RANK function (SQL Server)

- DENSE_RANK function (SQL Server), 341
- Deny (permission type), 452–453
- deployment
 - Integration Services Deployment Wizard, 94
 - project compilation, 93
 - Project Deployment Model, 90–96
 - Visual Studio configurations, using to target specific environments, 116
- Derived Column transform (data integration), 10, 16, 43
 - character limit on, 112
- design_default_value (catalog.object_parameters view), 337
- design, performance-optimized. *See* performance-optimized design
- design-time events, handling dynamically, 486–488
- destination adapters, 42
- Destination component (Data Flow task), 550–551
- destination_component_name column (catalog.execution_data_statistics view), 594
- Detach (data viewer control), 591
- developer community (PowerShell), 364
- DiagnosticEx events, 531–532
 - log entries for, 442
 - value of, example, 531
- digital signatures, 442–443
 - BlockedSignatureStates key (system registry), 442
- Digital Signing dialog box (SSIS), 443
- Dimension Processing destination model, 42
- dimensions, CDC for changes in, 197
- disk bottlenecks, 545
- disk I/O rate, 545
- disk transfer rate, average, 545
- domains, DQS, 256
- Domain Value (output reason), 258
- DontSaveSensitive (package protection level), 440
- DontSaveSensitive (protection level), 74
- DQS. *See* Data Quality Services (DQS)
- DQS Cleansing Connection Manager, 255
- DQS Cleansing (output reason), 258
- DQS Cleansing Transformation Editor
 - Advanced tab, 259
 - Connection Manager tab, 255
 - Enable Field Level Columns, 258
 - Mapping tab, 256
- DQS roles, 260
- drill-down option (Data Profile Viewer), 246
- DropExistingMatchIndexproperty (Fuzzy Lookup transformation), 248
- DT_BOOL (data flow data type), 41
- DTEExec utility, 301–302
 - configuring packages, 306–308
 - ConsoleLog option, 310
 - error dump options for, 308–309
 - file system, executing packages from, 303
 - IServerExec.exe vs., 311
 - Logger option, 310–311
 - logging and, 309–312
 - logging with, 479
 - msdb databases, executing packages from, 304
 - options for, 302
 - package store, executing packages from, 305
 - Parameter option, 307–308
 - Run64BitRuntime package property, 8
 - VLog option, 311
- DT_I4 (data flow data type), 41
- DT_WSTR (data flow data type), 41
- DUMP_EVENT_CODE, error dump execution parameter, 601
- DUMP_EVENT_CODE (execution-specific parameters), 343
- DUMP_ON_ERROR, error dump execution parameter, 601
- DUMP_ON_ERROR (execution-specific parameters), 343
- DumpOnError option (dtexec utility), 308
- Dump On Errors option (Advanced tab of the Execute Package dialog box), 317
- DUMP_ON_EVENT, error dump execution parameter, 601
- DUMP_ON_EVENT (execution-specific parameters), 343
- Dump option (dtexec utility), 308
- duplicate identifier (Fuzzy Grouping transformation), 253

E

- Edit distance parameter (Fuzzy Lookup transformation), 250
- Edit Mappings option, Fuzzy Lookup transformation, 248
- Edit window (Create Oracle CDC Instance Wizard), 227
- Enable Automatic Execution Of Integration Services Stored Procedure At SQL Server Startup option (SSIS Catalog), 420
- Enable Field Level Columns (DQS Cleansing Transformation Editor), 258

- EncryptAllWithPassword (package protection level), 439, 440
- EncryptAllWithPassword (protection level), 74
- EncryptAllWithUserKey (package protection level), 440
- EncryptAllWithUserKey (protection level), 74
- encryption
 - backup/restore issues with, 430–434
 - keys created for, on package execution startup, 424
 - master key, backing up with T-SQL script, 431
 - PackagePassword (package property), 439
 - packages, of, 439–441
 - protection levels (of package encryption), 439–441
- ENCRYPTION_ALGORITHM (SSIS Catalog properties), 332
 - valid values for, 334
- EncryptSensitiveWithPassword (package protection level), 74, 439, 440
- EncryptSensitiveWithUserKey (package protection level), 74, 440
- EngineThreads property (Data Flow task), 567
- entry points (SSIS packages), 336
- Environment check box (Execute dialog box), 315
- environments (data flow), 46–47
- Environment (securable type), 445
- Environments (SSIS), 345–350
 - absolute project references, creating, 348
 - catalog.create_environment stored procedure, 346
 - catalog.create_environment_variable, 346
 - catalog.create_execution, 349
 - catalog.environment_references view, querying to identify, 349
 - catalog.environment_variables view, 349
 - catalog.environment view, managing with, 349
 - as containers for environment variables, 345
 - creating, 346
 - creating project references to, 347–348
 - creating, with SSIS Management Object Model, 361
 - deleting, 350
 - environment variables, creating, 346–347
 - finding identifier in SSMS, 306
 - managing, 349–350
 - managing, with SSIS Management Object Model, 359
 - moving, 350
 - package execution using, 349
 - performance measurement and, 542
 - projects, configuring with, 347–348
 - reference values, configuring projects with, 348
 - relative project references, creating, 347–349
 - specifying, 349
 - SSMS, managing in, 313
 - environment variables
 - catalog.environment_variables view, 349
 - creating, with SSIS Management Object Model, 361
 - EPTs. *See* Execute Package task (EPT)
 - ER (CDC state string token), 203
 - ERROR (CDC state), 204
 - Error Code (system variable), 40
 - error context(s), 527–529
 - context depth, 527
 - stack trace vs., 527
 - Error Description (system variable), 40
 - error dumps, 308–309
 - catalog.create_execution_dump stored procedure, 603
 - catalog.execution_parameter_values view, 601
 - execution parameters for, 601
 - files generated by, 602
 - location of file for, 308
 - registry key for, 308
 - troubleshooting data, as tool for, 600
 - error output
 - CDC Splitter component, 216
 - troubleshooting data with, 592–593
- ETL processes. *See* extract-transform-load processes
- Evaluate As Expression property (variables), 32
- evaluation operation (precedence constraint), 555
- event handler (control flow), 40
- Event Handlers tab (package designer), 40
- Event Handler Start Time (system variable), 40
- event sink (SSIS object model), 486–488
- events (SSIS), logging, 344–345
- Excel Source and Destination (source adapter), 5
- Execute dialog box (SSMS), 315–317
- Execute_Objects (permission type), 445
- Execute Package dialog box, configuring to run as 32-bit application, 181
- ExecutePackageTask class (Microsoft.SqlServer.ExecPackageTaskWrap assembly), 497
- Execute Package task (EPT), 29
 - DiagnosticEx events and, 531
 - execution path and, 532–533
 - SSIS Project Conversion Wizard, updating with, 74

Execute Package task (EPT), *continued*

- troubleshooting, 529–533
 - troubleshooting child packages with, 377
 - Execute (permission type), 445
 - Execute SQL task (Control Flow), 19, 29
 - data staging with, 557
 - instrumenting Data Flow with, 547
 - TypeConversionMode property, 101
 - execution identifier (ID)
 - finding status of with T-SQL script, 523
 - package execution, of, 424
 - Execution Information available in Integration Services Dashboard, 368
 - Execution Information (Execution Performance report), 379
 - execution log (SSISDB), 423
 - Execution Overview
 - troubleshooting child packages with, 377
 - troubleshooting with, 374
 - execution path
 - Execute Package tasks and, 532–533
 - interpreting results of queries to, 533
 - troubleshooting with, 526
 - T-SQL queries, obtaining with, 533
 - execution_path column (catalog.execution_data_statistics view), 594
 - Execution Performance action column (All Executions report), 371
 - troubleshooting with, 374
 - Execution Performance report
 - sections of, 379
 - troubleshooting performance issues with, 380
 - execution plans and precedence constraints, 554
 - execution siloes and parallelization, 553
 - execution-specific parameters, 343
 - execution-specific reports, getting to from the Integration Services Dashboard, 369
 - Exhaustive property
 - Fuzzy Grouping transformation, 252
 - Fuzzy Lookup transformation, 248
 - ExposeCharColumnsAsUnicode (ODBC Source component property), 174
 - Expression Builder dialog box (Variables window), 88
 - expressions
 - Adorners, 100
 - character limit on, 112
 - LEFT function, 112
 - REPLACENULL function, 112
 - SSIS native language for, 32
 - TOKENCOUNT function, 113
 - TOKEN function, 113
 - Expression task vs. EvaluateAsExpression property, 99
 - external systems, performance measurements of, 542
 - extract-transform-load (ETL) packages, 483–485
 - applying patterns consistently, through automation, 484
 - creating in bulk, 484
 - extract-transform-load (ETL) processes
 - CDC in, 195
 - CDC scenarios in, 197–198
- ## F
- Failed (package execution status), 523
 - Favorites component (Control Flow toolbox), 86
 - Favorites component (Data Flow toolbox), 86
 - FetchMethod (ODBC Source component property), 172
 - File System task (data loading), 19
 - filtering options (Create Oracle CDC Instance Wizard), 226
 - Fixed attribute (SCD), 14
 - flashback query (Oracle), 200
 - Flat File Connection Manager, 107
 - Flat File Source and Destination (source adapter), 5, 106–108
 - Destination component performance measurement, 550
 - embedded qualifiers, 108
 - loading data from, 558
 - ragged-right delimited files, 106–107
 - Flat File Source parser, 107
 - FLOAT([1-24] type, mapping in SQL Server, 238
 - FLOAT([25-53] type, mapping in SQL Server, 238
 - FLOAT([54-126] type, mapping in SQL Server, 238
 - Folder (securable type), 445
 - folders (SSIS Catalog)
 - creating/dropping, 359
 - permission inheritance and, 452
 - For Each container (Control Flow), 18
 - Foreach Loop container, 33
 - creating custom enumerators, 33
 - Foreach Loop enumerator, 33
 - For Loop container (Control Flow), 18, 34
 - For Loop Editor, 34–35
 - FTP task (data loading), 19
 - full vs. partial cache mode, 563
 - fully blocking operations, 557

Functional Dependency profile, 242
 functions (PowerShell), 353
 Fuzzy Grouping transform (OLE DB exclusive component), 16, 164, 251–254
 Comparison Flag options for, 249
 configuration, 252–253
 inputs and outputs, 251–252, 253–254
 properties, 252
 uses, 251
 Fuzzy Lookup transform (OLE DB exclusive component), 16, 164, 246–251
 configuring, 248–249
 Fuzzy Grouping's use of, 254
 inputs and outputs, 246, 249–251
 properties, 248
 uses, 246

G

GetProcessingRange operation (CDC Control Task), 207, 208
 goals for performance-tuning strategies, 540
 Grid format (data view), 52
 GROUP BY clause, 558

H

Help command (Oracle CDC service executable), 238
 Historical attribute (SCD), 14
 HKLM\Software\Microsoft\XDBCDCSVC\

I

IDTSComponentEvents interface (SISS object model), 486–488
 IDTSEvents interface (SSIS object model), 486–488
 Ignore case comparison flag, 249
 Ignore character width comparison flag, 249
 Ignore kana type comparison flag, 249
 Ignore nonspacing characters comparison flag, 249
 Ignore symbols comparison flag, 249
 ILEND state (CDC)
 CDC Control Task operations and, 207
 defined, 204
 ILSTART state (CDC)
 CDC Control Task operations and, 207

 defined, 204
 ILUPDATE state (CDC), 204
 IMPERSONATION SQL Server permission, 448
 Import and Export Wizard (SSIS), 10–12
 data sources/destinations supported by, 10
 launching, 11
 mapping files, 10–11
 Import Project Wizard, 96
 IncludeInDebugDump property (parameters/variables), 441
 Include Reprocessing Indicator Column property (CDC Source component), 213
 Indexing operations and bulk loading, 559
 Inferred member (SCD), 14
 information from reference data services, DQS
 Cleansing transformation and, 259
 infrastructure in OVAL technique, 541
 infrastructure optimization
 connector selection and, 567
 SSIS Catalog and, 566
 initial data loading stage (CDC), 198–200
 initial processing range, 205
 input/reference dataset matches, cache mode and, 564
 INSERT changes
 CDC for, 195
 CDC Splitter component, 216
 syncing, 200
 InsertMethod (ODBC Destination component property), 176
 instrumentation, adding to Data Flow tasks, 547–548
 Integration Services Catalogs node (SQL Server), 312
 Integration Services Catalog (SSIS Catalog). *See* SSIS Catalog
 Integration Services Dashboard (SSIS report), 366, 368–370
 troubleshooting with, 368
 Integration Services Deployment Wizard, 94
 Integration Services Designer (ISD), 83–90
 annotations, 90
 AutoRecover functionality in Visual Studio, 89
 Connection Managers view, 101
 Control Flow toolbox, 86–87
 Data Flow toolbox, 86–87
 Deployment Wizard, 94
 detailed information available in, 368
 Find Samples link, 85
 Getting Started window, 85
 launching, 368
 launching other reports from, 369

- ODBC Source basic editor dialog box, 170
- tast/component status icons, 89
- toolbox window, 85–87
- undo/redo feature, 84
- Variables window, 87–88
- zoom control, 88
- Integration Services Project Conversion Wizard, 71
- Integration Services SQL Server feature, 52
- interactive cleansing, 255
- internal (SSISDB schema), , 418
- INTERVAL DAY [(1-9)] TO SECOND [(1-9)] type, mapping in SQL Server, 238
- INTERVAL YEAR [(1-9)] TO MONTH type, mapping in SQL Server, 238
- Invalid (cleansing status), 258
- Invoke-SqlCmd cmdlet (PowerShell script), 502
- I/O bottlenecks, 546
- I/O considerations, OVAL technique and, 541
- IR (CDC state string token), 203
- ISD. *See* Integration Services Designer (ISD)
- IsSensitive property (parameters), 442
 - DiagnosticEx (log entry) and, 442
- ISServerExec.exe, 419
 - dtexec utility vs., 311
 - limits of, 566
 - system performance and, 542

J

- Java Database Connectivity (JDBC), 6
- Jet provider, 184–185
- jobs (SQL Server Agent)
 - creating, 320–322
 - SSIS Catalog and, 427–430
 - viewing status of, 537

K

- KBs. *See* knowledge bases (KBs)
- key(s)
 - asymmetric, creating, 237
 - AsymmetricKeyPassword (REG_BINARY) registry value, 221
 - asymmetric, updating, 236
 - backing up master, 431
 - created on package execution startup, 424
 - surrogate, generation of, 13
- KeyStrengthThreshold property (Candidate Key request), 243

- knowledge bases (KBs), 254

L

- Last Change Timestamp property (CDC instance), 232
- Last Transaction Timestamp property (CDC instance), 232
- Lineage IDs, 103
- LobChunkSize (ODBC Destination component property), 177
- LOB data types, support for in Attunity connectors, 189
- Locals debug window, 593
- Location step (OVAL technique), 541
- Log File Viewer and SQL Agent job errors, 536
- Logger option (dtexec utility), 309, 310–311, 479
- logging, 463–482
 - capturing events with, for performance troubleshooting, 570
 - catalog.event_message_context view (SSIS Catalog), 477
 - catalog.event_messages, as container for, 423
 - catalog.event_messages view, 476
 - Configure SSIS Logs: Package dialog box (SSIS), 464–466
 - configuring, 463–470
 - containers, choosing which to track, 464–466
 - context depth and, 477
 - custom, for troubleshooting, 518
 - customizing fields, 478
 - custom providers, developing, 479–480
 - DiagnosticEx entry, 442
 - dtexec utility options for, 309–312
 - dtexec utility, with, 479
 - event context information, finding, 477
 - event logs, 476–477
 - events, selecting for, 466–468
 - improvements in, for troubleshooting, 518–519
 - levels of, 422
 - Logging Level property (SSIS Catalog), 474–476
 - ODBC tracing and, 167–169
 - performance effects of, 423
 - providers, 471–474
 - providers, adding, 468–470
 - SERVER_LOGGING_LEVEL property, 422
 - SSIS Catalog and, 474–477
 - supplemental, 228
 - supplemental, enabling in Create Oracle CDC Instance Wizard, 228

- sysssislog table and, 471
- LOGGING_LEVEL (execution-specific parameters), 344
- Logging Level option (Advanced tab of the Execute Package dialog box), 317
- Logging Level property (SSIS Catalog), 474–476
 - setting in SSMS, 476
 - values of, 475
- logging levels, setting in SSIS Catalog, 422–423
- Logging menu (package designer), 38
- LogicalDisk counter, 545
- Logical Sequence Number (LSN), 340–341
 - and CDC Control Task, 205
 - calculating, 207
 - changes between, 196
 - database snapshot with, 200
 - defined, 196
 - DENSE_RANK function (SQL Server), 341
 - encoding of, 203
 - Object Version LSN, 341
- log providers, 471–474
 - adding, 468–470
 - ProgIDs/Class IDs of, listed, 474
 - SQL Server, 471–472
 - SQL Server Profiler, 472
 - text files as, 471
 - Windows Event, 472–473
 - XML files as, 473–474
- log providers, custom
 - building functionality into, 479–480
 - Microsoft.SqlServer.Dts.Runtime.
 - LogProviderBase, as base for, 479
 - registering with SQL Server, 480
- LONG RAW type, mapping in SQL Server, 238
- LONG type, mapping in SQL Server, 238
- Lookup Cache (SSIS)
 - cache modes, 563–564
 - file location of, 13
 - Lookup Transformation, 562–563
 - memory and, 561
 - optimizing, 562–566
 - partial cache mode, 564–565
 - sharing a cache, 565–566
- lookup match operation (Fuzzy Lookup transformation), 246
- Lookup (OLE DB exclusive component), 9, 164
- Lookup Transformation, 562–563
 - alternatives to, for non OLE DB sources, 13
 - connectivity limitations on, 13
- Lookup Transformation Editor, 562

- Lookup Transform component (data warehouses), 13
- LSN. *See* Logical Sequence Number (LSN)

M

- maintenance tasks
 - maximum versions per project, 427–430
 - Operation records, 427–430
 - SSIS Catalog and, 427–430
- Manage_Object_Permissions (permission type), 445
- Manage Parameter Values dialog box (parameter designer), 92–93
- Manage_Permissions (permission type), 445
- mapping files, 10–11
 - location of, 11
- Mapping tab (DQS Cleansing Transformation Editor), 256
- MarkCdcStart operation (CDC Control Task), 207
- MarkCDCStart operation (CDC Control Task), 208
- MarkInitialLoadEnd operation (CDC Control Task), 207, 208
- MarkInitialLoadStart operation (CDC Control Task), 207
 - in active databases, 208
- MarkProcessedRange operation (CDC Control Task), 211
- MatchIndexNameproperty (Fuzzy Lookup transformation), 248
- MatchIndexOptionsproperty (Fuzzy Lookup transformation), 248
- maximum number of rows per buffer (Data Flow task), 567
- Maximum versions per project maintenance (SSIS Catalog), 427–430
- MaxMemoryUsage property (Fuzzy Lookup transformation)
 - disk space and memory use and, 251
 - Fuzzy Grouping transformation, 252
 - Fuzzy Lookup transformation, 248
- MaxNumberOfViolations property (Candidate Key request), 243
- MAX_PROJECT_VERSIONS property (SSIS Catalog), 332, 340, 417
- memory
 - buffer spooling and, 544
 - cache modes and, 564
 - keeping SSIS operations in, 561–562
 - measuring utilization of, 546
 - running out of, 561
 - SSIS pipeline engine performance and, 561

Memory performance counter

- Memory performance counter, 546
- merge (data operation), 9
- metadata columns, 197
- MetaData Driven ETL (MDDE) Studio project, 508
- Microsoft Access, connecting to, with Jet provider, 184–185
- Microsoft Data Access Components, 187
- Microsoft Developer's Network (MSDN), 479
- Microsoft Excel
 - connecting to, with Jet provider, 184–185
 - PowerPivot and, 579–588
- Microsoft Management Console (MMCs)
 - CDC Service Configuration, 219–221
 - Oracle CDC Designer, 221–233
- Microsoft .NET 7 Framework
 - PowerShell and, 353
 - scripting support, 111
- Microsoft Office 2007 files, connecting to, 185–186
- Microsoft Press Book Support, xxviii
- Microsoft SQLNcli team blog, 164
- Microsoft SQL Server 2012 T-SQL Fundamentals (Itzik Ben-Gan), xxiv
- Microsoft SQL Server Analysis Services, 20
- Microsoft.SqlServer.Dts.Design assembly (SSIS object model), 485
- Microsoft.SqlServer.DTSPipelineWrap assembly (SSIS object model), 325, 485
- Microsoft.SqlServer.Dts.Runtime.Application class
 - Upgrade method, 67
- Microsoft.SqlServer.Dts.Runtime.LogProviderBase, as base for custom log providers, 479
- Microsoft.SqlServer.DTSRuntimeWrap assembly (SSIS object model), 325, 485
- Microsoft.SqlServer.ManagedDTS assembly (SSIS object model), 325, 485
- Microsoft.SqlServer.Management.IntegrationServices assembly (SSIS object model), 357, 358, 485
- Microsoft.SqlServer.PipelineHost assembly (SSIS object model), 485
- Microsoft TechNet, 165
- Microsoft Visual C# 2010 Step by Step (John Sharp), xxiv
- Microsoft Visual Studio
 - Add SSIS Connection Manager dialog box, 161–162
 - ADO.NET provider dialog box, 162
 - AutoRecover functionality for SSIS projects, 89
 - BIDS Helper (third-party extension), 509–510
 - configurations and team development, 116–117
 - configurations in, 92–93
 - Import Project Wizard, 96
 - Pragmatic Works BI xPress (third party add-on), 511–512
 - project compilation, 93
 - Project Conversion Wizard, 95
 - Solution Explorer, 90–91
 - solutions, upgrading, 68
 - SourceDirectory parameter, 116
 - SSDT and, 83–84
 - SSDT environment for, 23
 - SSIS project files, changes that for modifications to, 127–129
 - SSIS Variables window, 87–88
- Microsoft Visual Studio Team Explorer 2010, 117
- MinSimilarity property
 - column-level similarity score and, 249
 - Fuzzy Grouping transformation, 252
 - Fuzzy Lookup transformation, 248, 249, 250
 - row-level similarity score and, 250
- mirroring component, 219
- mirror tables
 - CDC for Oracle and, 217
 - data mapping for, 238
- Mist (Varigence), 510–511
- MMC. *See* Microsoft Management Console
- Modify_Objects (permission type), 445
- Modify (permission type), 445
- MOM. *See* SSIS Management Object Model (MOM)
- monitoring component (CDC in Oracle), 219
- msdb database
 - package execution and, 303–304
 - Package Roles dialog box, 438
 - package security and, 437–438
 - related roles in, 437
 - SSIS Catalog vs., 44
 - SSISDB vs., 416
 - user-defined roles, creating, 437–438
- MSI installer
 - CDC Service Configuration MMC, 219
 - Oracle CDC Designer MMC, 222
- MSXDBCDC database, 233–235
 - adding entries for xdbc_cdc_services table, 234
 - associating SCL Server instance with, 221
 - initializing, 234
 - removal of CDC database from, 234

- stored procedures, 233–235
- tables, 233–235
- Multicast (data operation), 9
- multiple constraints evaluation logic (precedence constraints), 555
- multiselect option window (Fuzzy Grouping transformation), 252

N

- NCHAR type, mapping in SQL Server, 238
- NCLOB type, mapping in SQL Server, 239
- net changes
 - all vs., 214–215
 - CDC Splitter component and, 217
- Net processing mode (CDC)
 - all vs., 214–215
 - defined, 213
- Net with Merge processing mode (CDC), 213
- Net with Update Mask processing mode (CDC), 213
- network
 - bandwidth monitoring, 545
 - bottlenecks, measuring, 546
 - transfers and performance optimization, 566
- Network counter, 545
- New (cleansing status), 258
- New Credential dialog box (SSMS), 455
- New Value (output reason), 258
- No Cache mode, 563
- None (Logging Level property value), 422, 475
- NTFS file system
 - package execution and, 303
 - package security and, 436
 - Project Deployment Model and, 303
- NUMBER (1-4,0) type, mapping in SQL Server, 239
- NUMBER (5-9,0) type, mapping in SQL Server, 239
- NUMBER(10-18,0) type, mapping in SQL Server, 239
- NUMBER(19-38,0) type, mapping in SQL Server, 239
- Number of tokens parameter (Fuzzy Lookup transformation), 250
- NUMBER(P, S) type, mapping in SQL Server, 239
- NUMBER type, mapping in SQL Server, 239
- Numerals column (Fuzzy Grouping transformation), 252
- NVARCHAR2 type, mapping in SQL Server, 239

O

- object_name column (catalog.object_parameters view), 337
- object_type column (catalog.object_parameters view), 337
- Object Version LSN, 341
 - DENSE_RANK function (SQL Server), 341
- ODBC Destination component, 174–177
 - advanced properties, 176–177
 - basic editor dialog box, 174–175
 - Batch Size for, 175
 - bulk insert logic, 175
 - Connection Manager for, 175
 - Data Access Mode for, 175
 - error output format for, 175
 - set up, 174–175
- ODBC driver, 187
- ODBC Source and Destination (source adapter), 5
- ODBC Source component, 169–174
 - advanced properties, 172–174
 - bulk fetch logic, 171
 - Connection Manager for, 170
 - Data Access Mode for, 170
 - error output of, 171
 - parameters and, 171
 - set up, 170–171
- ODBC, tracing, 167–169
- OLAP destination adapter, 42
- OLE DB
 - commands vs. data staging, 557
 - Microsoft SQLNcli team blog, 164
 - providers, 162–163
 - support for ADO.NET and ODBC connections with, 164
- OLE DB Command transform, 15
- OLE DB connection manager
 - creating shared cache with, 565
 - Fuzzy Grouping transformation connection with, 252
 - Fuzzy Lookup transformation connection with, 246
- OLE DB Source and Destination (source adapter), 4–5, 42
 - Data Flow task analysis with, 552
 - measuring performance of, 551
- OLTP. *See* Online Transaction Processing (OLTP)
- OnError event (SSIS), 468
- OnInformation event (SSIS), 468
- Online Transaction Processing (OLTP), 566

OnPostExecute event (SSIS)

- OnPostExecute event (SSIS), 468
- OnPostValidate event (SSIS), 468
- OnPreExecute event (SSIS), 468
- OnPreValidate event (SSIS), 468
- OnQueryCancel event (SSIS), 468
- Open Database Connectivity (ODBC), 162, 166–177
 - database vendors and, 166
 - distributed transaction support in, 177
 - drivers, 166–167
 - drivers and performance considerations, 166
 - ODBC Destination component, 174–177
 - ODBC Source component, 169–174
 - tracing, 167–168
- operation (CDC Control Task), 207
- OPERATION_CLEANUP_ENABLED (SSIS Catalog properties), 333
- operation_messages view, querying,, 362
- OperationParameter property (CDC Control Task), 208
- Operation records maintenance (SSIS Catalog), 427–430
- Operation (securable type), 445
- Operations step (OVAL technique), 541
- Options property group (bulk loading), 560
- Oracle, 186–189
 - Attunity connectors for, 188–189
 - CDC components and, 18, 217–240
 - CDC creation components, 219
 - CDC functions, 217–218
 - CDC Service Configuration MMC, 219–221
 - data type mapping, 238–239
 - data type mapping issues with, 187
 - Microsoft OLE DB and .NET providers for, 186
 - MSXDBCDC database and, 233–235
 - ODAC components, 187
 - ODBC Connections, 187–188
 - OLE DB provider, 187
 - xdbcscvcs.exe service executable, 235–238
- Oracle CDC Designer Microsoft Management Console (MMC), 221–233
 - installation, 222
 - location, 223
 - Oracle CDC instance, 222
 - Oracle CDC instances, creating, 223–233
- Oracle CDC instances
 - clearing data contents of, 234
 - configuration information, 222
 - creating with Oracle CDC Designer Microsoft Management Console (MMC), 223–233
 - managing, 230
 - properties, 232
- Oracle CDC Service Configuration MMC
 - MSXDBCDC database creation, 233
 - xdbcscvcs.exe and, 235
- Oracle Data Access Components (ODAC), 187
- Oracle flashback query, 200
- Oracle Logging Script action (CDC instance), 231
- Oracle log miner, changing data capture with, 218
- Oracle ODBC driver, 187
- Oracle OLE DB provider, 187
- Oracle Source and Destination (source adapter), 5
- Oracle tab (for CDC instance), 232
- ORDER BY clause, 558
- ORDER hint and bulk loading, 559
- ordering of changes with CDC Splitter, 217
- OR precedence constraints and parallelization, 555
- Other Tasks component (Control Flow toolbox), 86
- Other Transforms component (Data Flow toolbox), 86
- output reason (DQS Cleansing transformation)
 - column-level, 258
 - descriptions, 258
- OVAL technique, 540–541
- Overall time spent (performance measurement), 546
- Overview action column (All Executions report), 371
- Overview report
 - child packages and, 377
 - troubleshooting with, 374

P

- Package class (Microsoft.SqlServer.Dts.Runtime namespace), 490
- Package Deployment Model
 - converting to, 71–79
 - converting to Project Deployment Model projects from, 95
- Package Designer (SSIS)
 - Event Handlers tab, 40
 - Logging menu, 38
 - Parameters tab, 37
 - Precedence Constraint Editor, 30–31
 - precedence constraints, modifying in, 30
 - precedent constraints, representation of, 30
 - Toolbox window, 28
- package execution (SSIS), 301–330, 341–345
 - All Executions report, 370–371
 - asynchronous vs. synchronous executions, 345

- catalog.create_execution stored procedure, 342
- catalog.execution_parameter_values view, 343
- catalog.set_execution_parameter_value stored procedure, 344
- catalog.start_execution stored procedure, 342
- catalog.stop_operation stored procedure, 345, 424, 426
- configuring for the SSIS Catalog, 312–314
- counters for, 542
- creating, with SSIS Management Object Model, 361–362
- custom package runners, using, 498–502
- detail discovery, 343–344
- dtexec utility, configuring for, 306–308
- duration of, 572
- environments, using, 349
- error dumps, 308–309
- Execute dialog box (SSMS), 315–317
- execution/operation ID, 522
- external processes generated by, 425
- life cycle of, 423–427
- locations of, 303–306
- LOGGING_LEVEL parameter (packages SSIS), 344
- logging options for, 309–311
- logging, setting up, 344
- metadata-driven, 484–485, 497–507
- monitoring with reports, 368–372
- msdb database and, 303–304
- NTFS file system and, 303
- package store and, 304–305
- performance, monitoring, 576–578
- PowerShell and, 325
- PowerShell scripts, using, 502–507
- programmational, 325–330
- property overrides, 342–343
- SQL Agent and, 320–324
- SQL Agent jobs, creating, 320–322
- SQL Agent jobs, viewing status of all, 537
- SSIS Catalog and, 311–319
- SSIS Dashboard, monitoring with, 537
- starting, from SSMS, 314–317
- starting in the Catalog, 314–317
- starting/stopping, 342
- states of, 425
- status of, troubleshooting, 523
- status report of, 318
- stopping, 345, 426
- SVT steps, troubleshooting with, 522
- termination in case of lost connection with SSIS instance, 425
- troubleshooting failures, 520–529
- troubleshooting performance issues of, 572–576
- troubleshooting, when scheduled with SQL Agent, 534–538
- troubleshooting, with reports, 373–377
- T-SQL and, 318–319
- viewing in SSIS Catalog, 318
- Windows Application Event Log and, 426–427
- with PowerShell over SSIS T-SQL API, 363
- package generation
 - handling design-time events, 486–488
 - sample code on Codeplex for, 508
- Package.LoadFromXML() method (SSIS Packages), 488
- package parameters, 343
 - creating, with SSIS Management Object Model, 361–362
 - package parameter designer, 92
- Package Roles dialog box (SSMS), 438
- packages (SSIS)
 - catalog.set_execution_property_override_value stored procedure, 343
 - changes to, during upgrade process, 68–69
 - controlling access to, 435–439
 - CozyRoc and script reuse, 512
 - creating in bulk, 484
 - creating, programatically, 325–330
 - creating templates with Pragmatic Works BI xPress, 511–512
 - detailed information available in Integration Services Dashboard, 368
 - dynamic generation of, 485–497
 - encryption, 439–441
 - entry points and, 336
 - executing with Performance Logging Level, 378
 - execution identifier (ID), 424
 - execution location, SSMS vs. T-SQL as source, 424
 - extract-transform-load (ETL) packages, 483–485
 - EzAPI, alternative creation tool for, 508–509
 - generating programatically, 483–484
 - LOGGING_LEVEL parameter, 344
 - logs when creating programatically, 474
 - metadata, SSISDB as container for, 417
 - PackagePassword (property), 439
 - parallelization of multiple, 553
 - parallelization within, 554
 - reports, launching from in SSMS, 365–367
 - Run64BitRuntime property, 8
 - runtime behavior of, 424–425
 - security and, 435–443

packages (SSIS), *continued*

- signing, 442–443
- SSIS Catalog, configuring for, 312
- storage locations for, 303
- troubleshooting child packages, 377
- Vulcan, using to describe, 509
- XML format of, 115–116
- package store. *See* SSIS package store
- parallelization, 552–556
 - precedence constraints in, 554–556
 - solution architecture, 553–554
- Parameter option (dtexec utility), 307, 307–308
- parameter(s)
 - IncludeInDebugDump property, 441
 - IsSensitive property, 442
 - object types of, 343
 - ODBC Source and, 171
 - securing, 441–442
 - troubleshooting values of, 524
 - T-SQL, finding the values of with, 524
 - values, externalizing with Visual Studio, 116–117
- Parameters tab (Execute dialog box), 315
- Parameters tab (Package Designer), 37
- Parameter tab (package parameter designer), 92
- partial cache mode
 - full vs., 563
 - Lookup Cache, 564–565
- Partition Processing destination model, 42
- partitions
 - Bulk Loading and, 560–561
 - SQL Server optimization with, 558
- PARTITION SWITCH technique, 558
- passwords
 - asymmetric key, 237
 - CDC service master password, 221
 - DPAPI encryption for, 221
- Past Execution(s) Information (Execution Performance report), 379
- Pending (package execution status), 523
- per-execution performance counters, 578–579
 - SQL Server Dynamic Management Views (DMVs) vs., 578
- performance
 - counters, troubleshooting with, 570
 - Execution Performance Report, indentifying trends with, 378–380
 - Execution Performance report, launching, 379
 - goals, setting, 540
 - interactive analysis of, 579–588
 - logging, effects of, 423
 - per-execution performance counters, 578–579
 - profiling, 569–570
 - troubleshooting, 569–588. *See also* troubleshooting performance issues
- Performance (Logging Level property value), 422, 475
- performance measurements, 542–552
 - Data Flow tasks, 546–552
 - system-level, 542–546
- performance optimization, 539–568
 - creating performance-tuning strategies, 540–541
 - designing for performance, 552–567
 - measuring performance, 542–552
- performance-optimized design, 552–567
 - Bulk Loading of data, 558–561
 - parallelization, 552–556
 - SQL Server optimization techniques and, 556–558
 - SSIS infrastructure in, 566–567
 - SSIS Lookup Cache in, 562–566
 - SSIS operations in memory and, 561–562
- performance profiler, 570
- performance-tuning options
 - applying, 540
 - hypothesizing about, 540
 - remeasuring performance after applying, 540
- performance-tuning strategies
 - creating, 540–541
 - OVAL technique in, 540–541
- permissions
 - catalog.deny_permission stored procedure (SSIS Catalog), 453
 - catalog.effective_object_permissions view, 448
 - querying with, 452
 - Catalog.object_permissions view and, 452
 - ddl_cleanup_object_permissions and, 453
 - Deny, 452–453
 - enforcement, 448–449
 - granting/revoking, 450–451
 - IMPERSONATION SQL Server permission, 448
 - inheritance, 451–452
 - initialization of, 446–448
 - initial settings for, 447
 - managing, 446–453
 - NTFS file system, setting in, 436
 - package store and, 439
 - types of, 445
- physical/virtual memory, availability of, 425
- pipeline

- components, command processing by, 557
 - tasks, buffers and performance of, 561
- PipelineComponent time, 571
- PowerPivot for Microsoft Excel, 579–588
 - analyzing performance data with, 582–588
 - Diagram view, 585
 - importing data, how to, 579–581
- PowerShell, 353–364
 - advantages of, 364
 - building blocks of, 353–354
 - functions, 353
 - runtime behavior of, 341
 - SQL Server and, 354–357
 - SSIS Catalog instances, creating with, 332
 - SSIS Management Object Model and, 358–362
 - SSIS management with, 357–363
 - T-SQL and, 362–363
- PowerShell scripts, 502–507
 - advantages of, 364
 - defined, 353
 - SQL Agent and, 505–507
 - SSIS Management Object Model and, 502–505
- Pragmatic Works BI xPress (third party Visual Studio add-on), 511–512
- Precedence Constraint Editor, 30–31
- precedence constraints (tasks), 30–31
 - modifying in package designer, 30
 - parallelization with, 554–556
 - Precedence Constraint Editor, 30–31
- Prepare SQL Server function (CDC Service Configuration MMC), 220
- Private Bytes counter, 546
- processing range (CDC), 205
- processor cores, parallelizing for efficient use of, 552
- Processor counter, 545
- processor time, monitoring, 545
- Profile Requests page
 - Data Profiling Task Editor, 244
 - editing profiling options with, 243
- Profiles pane (Data Profile Viewer), 245
- profiling operations, 241–242
- profiling requests. *See* profiling operations
- Programmatic (unattended) upgrade, 67
- Project Conversion Wizard, 71
- Project Conversion Wizard (Visual Studio), 95
- Project Deployment Model, 90–96
 - configuration/deployment of, 90–96
 - converting Package Deployment Model projects to, 95
 - executing packages created with, 311–312
 - packages stored in the filesystem and, 303
 - protection level, setting, 440–441
- Project Properties dialog box (SSMS), 450
- projects, converting to, 71–79
 - Project Conversion Wizard, 71
 - SSDT, converting with, 71–79
- Project (securable type), 445
- projects (SSIS), 334–345
 - adding to Team Foundation Server, 120–123
 - backwards compatibility of packages, 116
 - catalog.projects view, 335–336
 - catalog.set_object_parameter_value stored procedure, configuring with, 338
 - collections of packages as, 37
 - configuring in SSIS Catalog, 336–338
 - configuring with SSIS environments, 347–348
 - consecutive version numbers, obtaining, 341
 - converting to, after upgrade, 71–79
 - created_time value, 340
 - debugging scripts, 110
 - deleting from SSIS Catalog, 339
 - deploying/managing, with SSIS Management Object Model, 360
 - deploying to SSIS Catalog, 334–335
 - deploying, with PowerShell and SSIS T-SQL API, 363
 - Enable Data Viewer, 105
 - existing project deployment file, converting to, 96
 - file extensions for, 123
 - format of, as XML files, 124–129
 - Getting Started window, 85
 - Import Project Wizard, 96
 - internal file format of, 37
 - locating information on in SSIS Catalog, 335–336
 - Logical Sequence Number (LSN), 340
 - managing in SSIS Catalog, 339–341
 - MAX_PROJECT_VERSIONS catalog property, 340
 - metadata, SSISDB as container for, 417
 - moving in SSIS Catalog, 339
 - multiple configuration support, 92
 - object_type column, 337
 - Object Version LSN, 341
 - OLE DB providers, pre-installed, 162–163
 - parameters, 343
 - parameters of, finding, 337–338
 - parameters, project vs. object, 337
 - parameters, setting, 338
 - Project Conversion Wizard, 95
 - reports, launching from in SSMS, 365–367

projects (SSIS), *continued*

- restoring in SSIS Catalog, 340–341
 - Run64BitRuntime property, 180
 - Solution Explorer window, 90–91
 - storing, 491
 - streamName parameter, requirement for when
 - adding items to, 491
 - T-SQL query for obtaining project information, 336
 - value_set column, 337
 - value_type column, 337
 - Visual Studio SSIS projects vs., 36
 - properties, encrypting, 439–441
 - property (Fuzzy Lookup transformation), 248
 - property overrides, 343
 - property_path, setting, 424
 - set_execution_property_override_value stored procedure (SSIS), 424
 - Property Overrides option (Advanced tab of the Execute Package dialog box), 317
 - protection levels (of package encryption), 439–441
 - providers
 - offloading of processing by, 567
 - PowerShell, 353
 - Proxy Accounts (SQL Agent feature), 505–507
 - proxy accounts (SSIS), 456–458
 - permissions for, managing, 458
 - SSIS, making available to, 457
- ## Q
- Quick Profile button (Data Profiling Task Editor), 242
 - Quick Profiling, configuring, 243
- ## R
- ragged-right delimited files, 106–107
 - disabling parsing behavior for, 107
 - RAW file format, 566
 - RAW type, mapping in SQL Server, 239
 - Read Bytes/Sec counter, 545
 - Read_Objects (permission type), 445
 - Read (permission type), 445
 - REAL type, mapping in SQL Server, 239
 - Reason Alias column (DQS), 258
 - Rebuild Index task (database maintenance), 18
 - ref_count column (MSXDBCDC.xdbcdc_services table), 234
 - reference dataset
 - Fuzzy Lookup transformation, 246
 - knowledge base, 255
 - references (data flow), 47
 - environments and folders, binding together with, 47
 - package executions with, 361–362
 - references, location change of, during upgrade, 71
 - reference table
 - cache mode and changes in, 564
 - cache mode and size of, 564
 - ReferenceTable property (Fuzzy Lookup transformation), 248
 - reference values, configuring projects with, 348–351
 - REF type, mapping in SQL Server, 239
 - registry key setting for error dumps, 308
 - relative project references
 - creating, 347
 - using, 348
 - Relative token frequency parameter (Fuzzy Lookup transformation), 250
 - remote data source connections,
 - troubleshooting, 517
 - Reporting services, standalone server for, 566
 - reports (SSIS), 365–382
 - Execution Performance Report, indentifying trends with, 378–380
 - Integration Services Dashboard, 368–370
 - launching from SSMS, 365–367
 - package execution, monitoring with, 368–372
 - package execution, troubleshooting with, 373–377
 - relationship between, 370
 - viewing in SharePoint or Reporting Services, 367
 - Report Viewer (Upgrade Advisor), 56
 - Request ID property
 - Data Profiling Task Editor, 244
 - Reset action (CDC instance), 231
 - ResetCDCStateoperation (CDC Control Task), 207
 - Resolve References dialog box (Data Flow), 103
 - Results pane (Data Profile Viewer), 245
 - Resume (data viewer control), 591
 - Retain Same Connection property (connection manager), 36
 - RETENTION_WINDOW (SSIS Catalog properties), 333
 - roles
 - permission inheritance and, 451
 - SSIS Catalog Security model and, 444

Row by row (ODBC Destination bulk insert), 175
 Row by row (ODBC Source bulk fetch), 171
 Row Count transformation, troubleshooting data with, 590
 row identifier (Fuzzy Grouping transformation), 253
 ROWID type, mapping in SQL Server, 239
 row-level cleansing status (DQS Cleansing transform), 258
 row-level similarity score
 Fuzzy Grouping transformation, 253
 Fuzzy Lookup transformation, 250
 rows displayed (data viewers, information available in), 592
 rows_sent column (catalog.execution_data_statistics view), 594
 Run64BitRuntime package property, 8
 Running (package execution status), 523

S

SAP BI Source and Destination (source adapter), 6
 scalar-value functions, SSISDB as container for, 418
 SCDs. *See* Slowly Changing Dimension transform (SCD)
 SCHEMA_BUILD (SSIS Catalog properties), 333
 SCHEMA_VERSION (SSIS Catalog properties), 333
 Script Component Plus extension (CozyRoc), 512
 scripting, 108–111
 ActiveX, 51
 changing target framework for, 111
 errors, troubleshooting, 518
 expressions and, 112
 .NET 7 Framework support, 111
 PowerShell, 364, , 502–514
 Script task, debugging, 109–110
 Visual Studio Tools for Applications and, 108–109
 Script Source and Destination (source adapter), 5
 Script task, 29
 Script Task Plus extension (CozyRoc), 512
 security, 435–462
 access to packages, controlling, 435–439
 catalog.environments view and, 346
 catalog.projects view and, 336
 ddl_cleanup_object_permissions, removing users and, 453
 digital signatures and, 442–443
 encryption, packages, 439–441
 msdb database roles and, 437

MSDB system database and package
 security, 437–438
 NTFS file system, packages stored in, 436
 packages, 435–443
 package signing, 442–443
 package store and, 438–439
 permissions, managing, 446–453
 permissions, types of, 445–446
 protection levels (of package encryption), 439–441
 schemas, 444–445
 securables, types of, 445–446
 SQL Agent and, 454–460
 SSIS Catalog and, 443–453
 users/roles and, 444
 variables/parameters and, 441–442
 security roles (CDC), 227
 Select Oracle Tables and Columns screen (Create Oracle CDC Instance Wizard), 226
 sensitive metadata in Data Profiling output, 245
 Sequence container (Control Flow), 18, 34–35
 grouping tasks by property with, 34
 packages as, 34
 sequential designs, parallelization for, 555
 SERVER_LOGGING_LEVEL property (SSIS Catalog), 333, 422
 server security model, SSIS Catalog folders and, 46
 Service Account field (new Oracle CDC service dialog box), 220
 service name, creating/configuring, 236
 Service Name field (new Oracle CDC service dialog box), 220
 services
 CDC creating, with Service Configuration MMC, 220
 in Oracle CDC Designer MMC, 223
 set_execution_property_override_value stored procedure (SSIS), 424
 SET option (dtexec utility), 307
 shared cache, 565–566
 Sharp, John, xxiv
 Shrink Database task (database maintenance), 19
 similarity score
 column-level, 249, 253
 row-level, 250, 253
 similarity threshold (Fuzzy Grouping transformation), 254
 single domains (DQS), 256

Slowly Changing Dimension transform (SCD)

- Slowly Changing Dimension transform (SCD), 14–15, 164
 - Changing attribute, 14
 - Fixed attribute, 14
 - Historical attribute, 14
 - Inferred member, 14
 - wizard for, 15
- SMO. *See* SQL Server management object model
- snapshots, database (SQL Server), 200
- solution architecture, parallelization, 553–554
- Solution Explorer, 90–91
 - change management and, 124
 - connection managers, 91
 - Parameters tab, 92
 - Project.params node, 91
- SORT operations, 9
 - performance optimization and, 557
- Sort transformation, Comparison Flag options for, 249
- source adapters (data flow), 4–6, 41
 - ADO.NET Source and Destination, 5
 - custom adapters, 5
 - Excel Source and Destination, 5
 - Flat File Source and Destination, 5
 - limits on availability of, 6
 - ODBC Source and Destination, 5
 - OLE DB Source and Destination, 4–5
 - Oracle Source and Destination, 5
 - SAP BI Source and Destination, 6
 - Script Source and Destination, 5
 - special-purpose adapter, 5
 - standard, 41
 - Teradata Source and Destination, 6
 - XML Source, 5
- Source component (Data Flow task), 549
- source_component_name column (catalog.execution_data_statistics view), 594
- Source Control Management (SCM), 117–129
 - change management, 124–127
 - determining when changes were made, 126
 - SSIS project file, changes to, 124–129
 - Team Foundation Server and, 117–120
 - Visual Studio Team Foundation Server 2010, 117–129
- SourceDirectory parameter (Visual Studio), 116
- Split transformation (Fuzzy Lookup transformation), 247
- sp_ssis_startup stored procedure (SSIS), 420
 - generating when moving SSIS Catalog to new server, 431
- SQLAgentOperatorRole (msdb database role), 454
- SQLAgentReaderRole (msdb database role), 454
- SQLAgentUserRole (msdb database role), 454
- SQL CLR
 - enabling for SSIS Catalog creation, 415
 - SSISDB restoration, requirement for, 432
- SqlCommand (ODBC Source component property), 172
- SqlConnectionString (REG_BINARY) registry value, 221
- SQL Management Object (SMO) connection manager, 11
- SqlPassword (REG_BINARY) registry value, 221
- sqlps mini shell, 354
- SQL Server
 - addressing nodes, 355
 - CDC state storage, 204
 - partitioning, 558
 - query performance, optimizing, 542
 - replication, syncing CDC changes with, 201
- SQL Server 2008
 - CDC for, 217
 - CDC in, 195
 - PowerShell in, 354
- SQL Server 2012
 - 32-bit, installing runtime and tools for, 179
 - CDC in Oracle and, 217
 - CDC scenarios in ETLs, 197–198
 - CDC stages in, 198–202
 - CDC techniques in, 195–201
 - CDC use in, 196–197
 - Database Engine Services, 24
 - database snapshots, 200
 - Data Tools, 23
 - data type mapping with Oracle, 238–239
 - DENSE_RANK function, 341
 - editions that can support SSIS, 24–25
 - installation of DQS Cleansing transform and, 255
 - installation requirement for, 178
 - Integration Services feature, 22–23
 - Management Tools, 24
 - multiple instances on one server, 54
 - Object Version LSN, 341
 - PowerShell and, 354–357
 - preparing, with CDC Service Configuration MMC, 220
 - required features for SSIS, 22–24
 - sp_ssis_startup stored procedure (SSIS), 420
 - SSIS, set up for, 21
- SQL Server 2012, required features for SSIS in database engine services, 24

- Integration Services, 22–23
- SQL Server Data Tools feature, 23
- SQL Server Management Studio, 24–25
- SQL Server Agent
 - creating, steps for, 454
 - credentials, creating, 454–456
 - history tables, 537–538
 - jobs, creating, 459–460
 - PowerShell scripts and, 505–507
 - Proxy Accounts and PowerShell scripts, 505–507
 - proxy accounts, creating, 456–458
 - proxy accounts, executing jobs under, 459–461
 - role members who can create jobs, 454
 - security and, 454–460
 - service, starting, 321
 - SQL Agent history tables, 537–538
 - sqlps mini shell in, 354
 - troubleshooting job step issues, 518
 - troubleshooting package executions, scheduled with, 534–538
 - T-SQL script, creating with, 534
- SQL Server Agent jobs
 - package execution, configuring jobs for, 322
 - package execution from Catalog, 322–330
 - package execution in, 320–324
 - proxy accounts, source for detailed instructions, 323
 - SSIS Catalog and, 306
- SQL Server Analysis Services cubes
 - Analysis Services Execute DDL task, 20
 - Analysis Services Processing task, 20
 - Varigence Mist, support for, 510
- SQL Server authentication credentials
 - creating Oracle service with, 237
 - updating, 236
- SQL Server CDC Database ADO .NET Connection Manager (CDC Control Task Editor property), 206
- SQL Server Data Tools (SSDT), 23
 - breakpoints, 593
 - Business Intelligence Development Studio, 7
 - debug windows, 593
 - engine threads in, 567
 - Import and Export Wizard, launching, 11
 - in-line upgrade to, 67
 - maintenance operations in, 18
 - row count values, as tools for troubleshooting data issues, 589–590
 - Run64BitRuntime package property, 8
 - running in 64-bit architectures, 179–180
 - sqlps mini shell in, 354
 - SSIS, as requirement for, 23
 - troubleshooting data issues with, 589–593
 - troubleshooting in, 518
 - troubleshooting large projects in, 529
 - Visual Studio and, 83–84
- SQL Server destination (OLE DB exclusive component), 164
- SQL Server Developer edition, 195
- SQL Server Dynamic Management Views (DMVs), 578
- SQL Server Enterprise edition, 195
- SQL Server Evaluation edition, 195
- SQL Server Installation Center user interface, 62–69
- SQL Server Integration Services (SSIS), 3–26
 - 32-bit/64-bit environments, support for, 7–8
 - best practices for installation of, 22
 - catalog, 24–25
 - CDC Control task, 205–211
 - CDC Source Data Flow components, 211–215
 - CDC Splitter components, 215–217
 - CDC state, 202–205
 - change data consumption, 202–217
 - data cleansing, 16–17
 - data consolidation, 4–9
 - data maintenance, processing, analysis of, 18–20
 - data movement between systems, 9–12
 - data standardization, 16–17
 - data warehouses, loading, 12–15
 - evolution of, 20–21
 - formatting data, 16–17
 - Import and Export Wizard, 10–12
 - infrastructure, optimizing, 566–567
 - operations, 561–562
 - setting up, 21–25
 - source adapters, 4–6
- SQL Management Object (SMO) connection manager, 11
- SQL Server editions, available in, 24–25
- SQL Server, required features for, 22–24
 - user's perspective, 27
 - WOW64 application, 178–184
- SQL Server log provider, 471–472
 - ProgID/Class ID of, 474
 - sysssislog table and, 471
- SQL Server LSN/snapshot database name (CDC Control Task Editor property), 206
- SQL Server Management Object Model (SMO), 356

SQL Server Management Studio (SSMS)

- SQL Server Management Studio (SSMS)
 - configuring packages in, 312–314
 - credentials for SQL Agent proxy accounts, creating in, 455–456
 - Deny permission and the permissions dialog box, 453
 - environment folders and, 347
 - Execute dialog box, 315–317
 - generating sp_ssis_startup with, 431
 - Integration Services Dashboard report, 368
 - launching SSIS reports from, 365–367
 - maintenance job, generating in, 431
 - maintenance operations in, 18
 - package execution, starting from, 314–317
 - Package Roles dialog box, 438
 - package store and, 438
 - permissions, granting/revoking in, 450
 - permissions, viewing/modifying in, 447–448
 - Project Properties dialog box, 450
 - proxy accounts, creating in, 456–458
 - proxy accounts, organization of, 458
 - restoring SSISDB with, 432
 - Run64BitRuntime package property, 8
 - Script functionality and SSIS Catalog in, 339
 - SSIS Catalog instances, creating with, 332
 - SSIS service, viewing in, 44
 - standard logging reports for troubleshooting, 519
- SQL Server optimization techniques, 556–558
 - aggregation operations, 557–558
 - data staging, 556–557
 - OLE DB commands and, 557
 - SQL Server partitioning, 558
- SQL Server Performance (blog), 165
- SQL Server PowerShell, package execution via, 325
- SQL Server Profiler log provider, 472
 - ProgID/Class ID of, 474
- SQL Server Provider, 354
- SQL Server security, 446
- SQL Server:SSIS Pipeline 11.0 counter group, 542
 - Buffers Spooled counter, 543–544
- SQL Server:SSIS Service 11.0 counter group, 542
- SQL Server table, querying, 356
- SQL Server Upgrade Advisor, 55–61
 - components, selecting, 56–57
 - log file, location of, 60
 - parameters for, 57
 - parameters required by, 57
 - parameters, restrictions on, 58
 - prerequisites for, 56
 - reports, location of, 60
 - restrictions on, 60
 - rules for, 59
 - SSIS Parameter page, 57–58
 - Upgrade Advisor Report Viewer, 60
- SqlUsername (REG_BINARY) registry value, 221
- ssis_admin role (SSISDB), 444
- SSIS application programming interfaces (APIs), 483–485
- SSIS Catalog, 24, 44–47, 415–434
 - backing up, 431
 - catalog.catalog_properties view, 333
 - catalog.configure_catalog stored procedure, 333
 - catalog.create_environment stored procedure, 346
 - catalog.create_environment_variable, 346
 - catalog.create_execution stored procedure, 342
 - catalog.create_execution, 349
 - catalog.create_execution stored procedure, 424
 - catalog.deploy_project stored procedure, 334–335
 - catalog.execution_parameter_values view, 343
 - catalog.executions view, 340
 - catalog.move_project stored procedure, 339
 - catalog object, 46
 - catalog.object_parameters view, 337
 - catalog.packages view, 336
 - catalog.restore_project stored procedure, 341
 - catalog.set_execution_parameter_value stored procedure, 344
 - catalog.set_object_parameter_value stored procedure, 338
 - catalog.start_execution, 342
 - catalog.start_execution stored procedure, 342
 - catalog.startup stored procedure, 420
 - catalog.stop_operation, 345
 - configuring projects in, 336–338
 - counters in, 546
 - creating, 415–416
 - creating/dropping, 358–359
 - creating/dropping folders of, 359
 - DDL trigger, 453
 - deleting projects from, 339
 - deploying/managing projects in, 360
 - deploying SSIS projects to, 334–335
 - disabling automatic execution of, 421
 - dtexec utility, package execution from, 305–306
 - Enable Automatic Execution Of Integration Services Stored Procedure At SQL Server Startup option, 420

- environments, 46–47
- environments/environment variables in, 361
- /Env[Reference] (dtexec option), 306
- event context information, finding, 477
- event logs, 476–477
- finding package information within, 335–336
- folders, 46
- infrastructure for, 541
- infrastructure optimization with, 566
- Integration Services Project Deployment file, 417
- /IS[Server] (dtexec option), 305
- logging in, 474–477
- Logging Level property, 474–476
- logging levels, 422–423
- Logical Sequence Number (LSN), 340
- maintenance tasks and, 427–430
- Management Object Model (MOM) and, 341
- managing projects in, 339–341
- manipulating folders in, 446
- Maximum versions per project
 - maintenance, 427–430
- MAX_PROJECT_VERSIONS property, 340, 417
- moving projects in, 339
- objects in, 357
- Object Version LSN, 341
- Operation records maintenance, 427–430
- package execution in, 311–319, 341–345
- package execution, life cycle of, 423–427
- package executions/parameters in, 361–364
- Package Execution Tools in, 180–183
- package execution, using PowerShell scripts
 - through MOM, 502–505
- parameters of projects, finding, 337–338
- performance data in, 548
- permissions, managing, 446–453
- PowerShell and, 341
- project parameters, setting, 338
- projects/packages and, 334–345
- properties of, 332–333
- properties, setting/changing, 333–334
- querying properties of, 333
- Read permissions and, 449
- references, 47
- relationships between objects in, 45
- restoring from backup, 432–434
- restoring projects in, 340–341
- security in, 443–453
- SERVER_LOGGING_LEVEL property, 422
- shutdown/restart, behavior after, 420–421
- SQL Agent and, 306
- SQL Server agent jobs and, 427–430
- SQL Server Agent jobs, enabling for maintenance
 - jobs, 428
- SSISDB as container for, 416
- SSISDB database and, 444
- SSIS Server Maintenance Job, 428–430
- startup behaviors, 420–421
- startup stored procedure, 416
- troubleshooting in, 593–603
- T-SQL and, 332–334
- users, types of, 444
- SSIS Catalog Security model, 444–453
 - SQL Server security and, 446
 - Windows Authentication and, 444
- SSIS Dashboard, 537
- SSIS data flow data type, 41
- SSIS Data Flow engine, 552
- SSISDB, 331
 - adding authorized users to, 419
 - anatomy of, 418–419
 - backing up, 431
 - catalog schema, objects belonging to, 418
 - catalog.startup stored procedure, 420
 - encryption of, 417
 - execution log generated in, 423
 - internal schema, objects belonging to, 418
 - launching SSIS reports from, in SSMS, 365–367
 - login requirements for, during restore, 432
 - mapping database users to login, 433
 - public views, documentation for, 418
 - restoring from backup, 432–434
 - shutdown/restart, behavior after, 420–421
 - SQL CLR, requirement for during restoration, 432
 - SSIS Catalog and, 416, 444
 - Windows authentication requirements for, 419
- SSISDB node
 - All Operations report and, 371
 - Integration Services Dashboard report, launching
 - from, 368
- SSISDB restore, 432–434
 - login requirements for, 432
 - master keys/passwords and, 433
 - [##MS_SSISServerCleanupJobLogin##] login, 432
 - SQL CLR and, 432
 - SSMS, with, 432
 - T-SQL, with, 432
- SSIS Flat File Source. *See* Flat File Connection
 - Manager; *See* Flat File Source and Destination (source adapter); *See* Flat File Source parser
 - ragged-right delimited files and, 107

SSIS Management Object Model (MOM)

- SSIS Management Object Model (MOM)
 - creating/dropping folders, 359
 - creating/dropping SSIS catalogs, 358–359
 - creating environments/environment variables, 361
 - creating package parameters/executions, 361–362
 - deploying/managing projects, 360
 - loading, 358
 - PowerShell and, 358–362
 - PowerShell and SSIS Management Object Model, 358
 - PowerShell scripts and, 502–505
 - runtime behavior of, 341
 - T-SQL and PowerShell, 362–363
 - uses of, 357–358
 - uses of SSIS Management Object Model, 357
- SSIS object model
 - assembly files, location of, 486
 - automation and, 485–497
 - event sink, 486–488
 - IDTSEvents interface, 486–488
 - required assemblies for, 485
- SSIS package store, 304–305
 - location of, 305, 438
 - security and, 438–439
- SSIS Package Upgrade Wizard, 63, 63–69
 - common problems, 66
 - invalid packages and, 66
 - launch options for, 66
 - options for, 65
 - password-protected fields and, 67
 - programmatic (unattended) upgrade, 67
 - script issues, 67
- SSIS Project Conversion Wizard (SSDT), 71–79
 - Create Parameters, 77
 - Execute Package tasks, updating with, 74
 - Protection levels, 74
- SSIS Projects. *See* projects (SSIS)
- SSIS Server Maintenance Job (SSIS Catalog), 428, 428–430
 - owner of, 430
- SSIS service vs. SSIS Catalog, 44–45
- stack trace vs. error context, 527
- Staged Transactions property (CDC instance), 232
- standalone SQL Servers, 566
- Start action (CDC instance), 231
- startup stored procedure (SSIS Catalog), 416
- state (CDC)
 - initial data loading and source database, 199
 - source database, 199
- StatementTimeout (ODBC Destination component property), 176
- StatementTimeout (ODBC Source component property), 172
- State Name (CDC Control Task Editor property), 207
- status (data viewers, information available in), 592
- Status property (CDC instance), 232
- Stop action (CDC instance), 231
- Stopping (package execution status), 523
- stored procedures (SSIS), 331–332
 - MSXDBCDC database, 233–235
 - SSISDB, 331
 - SSISDB as container for, 418
- string data types, Fuzzy Grouping transformation for, 251
- strtok function (C++), 113
- Successful (package execution status), 523
- surrogate key generation, 13
- SVT steps, troubleshooting package execution with, 522
 - Step S, 523–524
 - Step T, 525–526
 - Step V, 524–525
- SYNCHRONIZED (execution-specific parameters), 344
- syncing CDC changes stage, 200–202
- sys.fn_cdc_get_max_lsn system function, 207
- sys.sp_cdc_enable_table procedure, 196
- sys.sp_replincrementlsn procedure
 - LSN values from, 208
 - requirements, 207
- sysssislog table (SQL Server), 471
- system performance
 - Buffers Spooled counter, 543–544
 - LogicalDisk counter, 545
 - measuring, 542–546
 - Memory performance counter, 546
 - Network counter, 545
 - Processor counter, 545
 - SSIS Catalog counters, 546
- system requirements, xxvi
- system variables, 32

T

- Table Lock option, bulk loading and, 560
- TableName (ODBC Destination component property), 176

- TableName (ODBC Source component property), 172
- Table or View property (profile request), 243
- Table Properties dialog box (Create Oracle CDC Instance Wizard), 228
- tables
 - MSXDBCDC database, 233–235
 - references, CDC for changes in, 197
- Table Selection dialog box (Create Oracle CDC Instance Wizard), 227
- Tables tab (for CDC instance), 232
- Table To Use For Storing State(CDC Control Task Editor property), 207
- Table-Valued Functions, SSISDB as container for, 418
- tasks (control flow), 28–30
 - built-in tasks, location of, 28
 - connecting, 30–31
 - custom, writing, 30
 - Execute Process task, 28
 - native tasks, location of, 29
 - precedence constraints on, 30–31
 - Script task, 29
 - variables and, 31–32
- tasks (data flow components), 96–100
 - Change Data Capture (CDC), 96–98
 - DQS Cleansing transform, 100
 - Expression task, 99
 - ODBC Source and Destination support, 100
- tasks (SSIS)
 - catalog.executable_statistics view,
 - troubleshooting performance issues with, 573
 - connection managers, sharing data with, 35
 - durations of, finding, 573
 - insterting, from another package, 495
 - self-modifying packages and, 28
- team development, 115–130
 - best practices for, 129–130
 - package format improvements for, 115–116
 - Source Control Management (SCM), 117–129
 - Visual Studio configurations and, 116–117
- Team Foundation Server. *See* Visual Studio Team Foundation Server 2010
- TechNet
 - digital signature article in, 443
 - PowerShell scripts on, 364
- tempdb database
 - Data Profiling task and, 244
 - tables in Fuzzy Lookup transformation, 247
- Teradata Source and Destination (source adapter), 6
- Term extraction (OLE DB exclusive component), 164
- Terminated unexpectedly (package execution status), 523
- Term lookup (OLE DB exclusive component), 164
- text file
 - log provider, as, 471
 - ProgID/Class ID of, 474
- TFEND state (CDC)
 - CDC Control Task operations and, 207
 - defined, 204
- TFREDO state (CDC), 204
- TFSTART state (CDC), 204
- ThresholdSetting property (Candidate Key request), 243
- TIMESTAMP([0-7]) type, mapping in SQL Server, 239
- TIMESTAMP([8-9]) type, mapping in SQL Server, 239
- Timestamp, flashback query using, 200
- TIMESTAMP WITH LOCAL TIME ZONE type, mapping in SQL Server, 239
- TIMESTAMP WITH TIME ZONE type, mapping in SQL Server, 239
- Token order parameter (Fuzzy Lookup transformation), 250
- Toolbox window (Package Designer), 28
- total buffers (data viewers, information available in), 592
- total rows (data viewers, information available in), 592
- trace
 - end, instrumenting package with, 548
 - start, instrumenting package with, 548
- transaction completion, in Oracle, 232
- Transaction Log Head Change Number property (CDC instance), 232
- Transaction Log Tail Change Number property (CDC instance), 232
- TransactionSize (ODBC Destination component property), 177
- Transformation component (Data Flow task), 549–550
- transformation editor (Fuzzy Grouping transformation), 254
- transforms, buffers for, 43
- troubleshooting, 517–538
 - child package executions, 529–533
 - child packages, 377
 - common errors, 517–522
 - custom logging for, 518
 - DiagnosticEx events, with, 531–532
 - error contexts, 527–529

troubleshooting, *continued*

- event names in Verbose logs, obtaining with T-SQL, 530
- Execute Package task, 529–533
- execution path, with, 526, 532–533
- Integration Services Dashboard report and, 368
- package execution failures, 520–529
- package executions, scheduled with SQL Agent, 534–538
- performance issues/bottlenecks with Execution Performance report, 380
- SQL Agent history tables and, 537–538
- SQL Server 2008 R2 Books Online (BOL) article on, 518
- SQL Server Data Tools and, 518
- SQL Server Management Studio and, 519
- SVT steps, with, 522–526
- troubleshooting data, 589–604
 - breakpoints, as tools for, 593
 - catalog.execution_data_statistics view (SSIS Catalog), 594
 - data errors, popular solutions for, 593
 - Data Flow paths as a tool for, 593
 - data taps as tools for, 596–600. *See also* data taps
 - data viewers as tools for, 590–592
 - debug windows as tools for, 593
 - dtexec utility as tool for, 600–603
 - error dumps, 600–603
 - error outputs for, 592–593
 - Row Count transformation, as tool for, 590
 - row count values, as tools for, 589–590
 - SSDT, troubleshooting in, 589–593
 - SSIS Catalog, troubleshooting in, 593–603
- troubleshooting performance issues, 569–588
 - capturing events with logging, 570
 - catalog.dm_execution_performance_counters, 571, 578
 - catalog.executable_statistics view (SSIS Catalog), 571, 573
 - catalog.execution_component_phases, 571
 - catalog.executions view (SSIS Catalog), 572
 - Data Flow components, finding time spent on, 573–576
 - interactive analysis of performance data, 579–588
 - methods for, 570–571
 - package execution duration, finding, 572
 - package execution, monitoring, 576–578
 - per-execution performance counters, 578–579
 - performance counters, 570
 - possible culprits beyond SSIS, 578
 - PowerPivot, analyzing performance data with, 579–588
 - profiling, 569–570
 - SSIS package execution, 572–576
 - tasks, finding duration of, 573
 - T-SQL, using to monitor package executions, 576
- TS (CDC state string token), 203
- T-SQL, 331–352
 - configuring/executing packages with, 341–345
 - configuring to run in 32-bit, 181
 - ConnectionString property, finding with, 528
 - data taps, creating with, 597
 - environments, creating with, 345–350
 - event names, obtaining with, 530
 - execution paths, obtaining with, 533
 - finding failed executables with, 525
 - Integration Services Catalog, 332–334
 - logging level, setting with, 344
 - monitoring package execution performance with, 576
 - overriding properties with, 343
 - package execution with, 318–319
 - permissions, granting/revoking with, 451
 - projects/packages and, 334–345
 - restoring SSISDB with, 432
 - Script functionality (SSMS) and SSIS Catalog in, 339
 - SQL Agent job, creating with, 534
 - SSIS management with, 362–363
 - SSIS operations and, 558
 - stored procedures (SSIS), 331–332
 - troubleshooting with, 519
 - using to find broken project references to moved/deleted environments, 350
- T-SQL API
 - deploying projects with PowerShell and, 363
 - package execution with PowerShell and, 363
 - PowerShell and, 362
 - use of, with PowerShell, 362
- T-SQL statements
 - for All changes, 215
 - for MSXDBCDC database, 233
 - for Net changes, 215
 - for OLE DB command, 213
- TypeConversionMode property (Execute SQL task), 101

U

- unexpected data values, catching with data viewers, 590
- Unicode strings, specifying in SSIS catalog, 339
- union all (data operation), 10
- unmatched rows caching feature, 565
- UPDATE changes
 - CDC for, 195
 - CDC Splitter component, 216
 - syncing, 200
- Update Statistics task (database maintenance), 19
- Upgrade Advisor Report Viewer, 60
- Upgrade method (Microsoft.SqlServer.Dts.Runtime.Application class), 67
- upgrade path (SQL Server 2012 Integration Services), 49–80
 - considerations/planning for, 50–55
 - duration of upgrade, 53
 - features, new in, 49–50
 - hardware requirements for, 53
 - manual upgrade steps, 69–71
 - software requirements, 52
 - SQL Server Installation Center user interface, 62–69
 - SQL Server Upgrade Advisor, 55–61
 - validation, 55
- upgrade, performing, 61–69
 - DTEXec, upgrading in-memory, 67
 - installed files/objects, 62–63
 - runtime behavior of installer, 62
 - SSDT, in-line upgrade of, 67
 - SSIS Package Upgrade Wizard, 63–69
 - SSIS projects/packages, 63–69
 - Upgrade method (Microsoft.SqlServer.Dts.Runtime.Application class), 67
 - user-created artifacts and, 70–71
 - Visual Studio Conversion Wizard, 63–69
 - Visual Studio solutions, 68
- upgrade scenarios, 53–54
 - edition upgrade, 54
 - side-by-side install, 54
 - unsupported, 54–55
 - version upgrade, 53–66
- upgrading, considerations/planning for, 50–55
 - ActiveX scripting and, 51
 - addressing issues, 69–71
 - Database Engine Services feature, 52
 - data viewers and, 51–52

- dependencies/tools for, 52
- DTS support, 51
- feature changes in SSIS, 50
- Integration Services SQL Server feature, 52
- SQL Server Upgrade Advisor and, 55–61
- UROWID type, mapping in SQL Server, 239
- User defined type, mapping in SQL Server, 239
- users and the SSIS Catalog Security model, 444–461

V

- Value Inclusion profile, 242
- value_set column (catalog.object_parameters view), 337
- value_type column (catalog.object_parameters view), 337
- VARCHAR2 type, mapping in SQL Server, 239
- Variable Containing property (CDC Source component), 213
- Variable Containing The CDC State (CDC Control Task Editor property), 206
- variables
 - Evaluate As Expression property, 32
 - expressions and, 32
 - IncludeInDebugDump property, 441
 - scope, default settings, 88
 - securing, 441–442
 - system variables, 32
 - tasks and, 31–32
- Variables window (Expression Builder dialog box), 88
- Varigence Mist, 510–511
- VARRAY type, mapping in SQL Server, 239
- Verbose (Logging Level property value), 422, 475
- VERSION_CLEANUP_ENABLED (SSIS Catalog properties), 332
- View Context link (All Messages report), 375–377
- VIRTUAL type, mapping in SQL Server, 239
- Visual Studio Conversion Wizard, 63–69
- Visual Studio for Applications, 5
- Visual Studio Team Foundation Server 2010, 117
 - additional information source for, 117
 - changing encoding of files in, 124
 - connecting to, 117–120
 - creating project for, 117–118
 - diff utility, 126
 - History dialog box, 125
 - Microsoft Visual Studio Team Explorer 2010 and, 117
 - SSIS project, adding to, 120–123

Visual Studio Tools for Application (VSTA)

Visual Studio Tools for Application (VSTA), 108–109
 Script task, as basis for, 29
VLog option (dtexec utility), 309, 311, 479
Volume step (OVAL technique), 541
Vulcan project, 509

W

WarmCaches property (Fuzzy Lookup transformation)
 described, 248
 disk space/memory use and, 251
WarnAsError option (dtexec utility), 308
Watch debug window, 593
Web Service task (data loading), 19
Windows Application event log, 426–427
Windows Authentication
 data connection via, in Data Profile Viewer, 246
 SSIS Catalog Security model and, 444
Windows Azure Marketplace, 255
Windows credentials, creating Oracle service with, 237
Windows Event log provider, 472–473
 ProgID/Class ID of, 474
Windows Event Viewer, 426–427

Windows Management Instrumentation (WMI), 353
Windows performance counters, 542
Windows Performance Monitor, 542
Windows service account credentials, 236
Windows Service Control Manager, registration of
 new CDC Oracle service with, 221
Windows Service Manager API, 237
Working Set counter, 546
WOW64 application, 178
Write Bytes/Sec counter, 545

X

xdbcscv.exe (Oracle), 235–240
xdbcscv.exe service executable, 235–238
 commands, 235–238
 installation, 235
XML
 log provider, as, 473–474
 Vulcan project, describing SSIS packages with, 509
XML log provider, ProgID/Class ID of, 474
XML Source (source adapter), 5
XMLTYPE type, mapping in SQL Server, 239

About the Authors



DR. WEE HYONG TOK is a Program Manager on the SQL Server Integration Services team. Since joining the team in 2008, he has worked on many aspects of the product. Prior to joining Microsoft, Wee Hyong is a Microsoft Most Valuable Professional (MVP), and frequently presents at various technology conferences.



RAKESH PARIDA is a Software Development Engineer on the SQL Server Integration Services team. Rakesh joined the team during the development of SQL Server 2012 and has worked on many new features of this release. He has been working with SSIS since SQL Server 2005 and has extensively worked with ETL frameworks, data warehousing and SSIS based solutions.



MATT MASSON is a Senior Program Manager on the SQL Server Integration Services team. Matt joined the team shortly after the SQL Server 2005 release, and has worked on many aspects of the product. A frequent presenter at Microsoft conferences, he is heavily involved with the SSIS community, and maintains the SSIS Team Blog (<http://blogs.msdn.com/b/mattm/>).



DR. XIAONING DING is a Software Development Engineer on the SQL Server Integration Services team. Xiaoning joined the team in 2007 shortly before the release of SQL Server 2008. After spending some time on the development of SSIS Connector for SAP BW, he started working on SQL Server 2012. Xiaoning has worked on many aspects of the product.



KAARTHIK SIVASHANMUGAM is a Lead Software Engineer in SQL Server Data Curation team. Kaarthik joined the Integration Services team during the development of SQL Server 2005 SP1 and involved in several feature improvements in SSIS until SQL Server 2012.

What do you think of this book?

We want to hear from you!

To participate in a brief online survey, please visit:

microsoft.com/learning/booksurvey

Tell us how well this book meets your needs—what works effectively, and what we can do better. Your feedback will help us continually improve our books and learning resources for you.

Thank you in advance for your input!

Microsoft[®]
Press