Christian Cote, Matija Lah, Dejan Sarka

SQL Server 2017 Integration Services Cookbook

ETL techniques to load and transform data from various sources using SQL Server 2017 Integration Services

Packt

WOW! eBook
www.wowebook.org

- 1: Setting up SSIS
  - b'Chapter 1: Setting up SSIS'
  - b'Introduction'
  - b'SQL Server 2016 download'
  - b'Installing JRE for Polybase'
  - b'Installing SQL Server 2016'
  - b'SQL Server Management Studio'
  - b'SQL Server Data Tools installation'
  - b'Test SQL Server connectivity'
- 2: What Is New in SSIS 2016
  - b'Chapter 2: What Is New in SSIS 2016'
  - b'Introduction'
  - b'Creating SSIS Catalog'
  - b'Custom logging'
  - b'Azure Tasks and Transforms'
  - b'Incremental package deployment'
  - b'Multiple version support'
  - b'Error Column Name'
  - b'Control Flow templates'
- app01: Appendix A: Feedback
  - b'Chapter Appendix A: Feedback'

WOW! eBook
www.wowebook.org

Chapter 1. Setting up SSIS

In this chapter, we will cover the following recipes:

- SQL Server Setup
- JRE installation for polybase
- SQL Server Data Tools (SSDT) installation
- SQL Server Management Studio (SSMS) installation
- Avoiding plain text passwords in scripts

WOW! eBook
www.wowebook.org

Introduction

This chapter will cover the basics on how to install SQL Server 2016 to properly go through the examples in this book. The version of SQL Server used through out this book is the Developer Edition of SQL Server 2016. It's available for free as long as you subscribe to Visual Studio Dev Essentials.

WOW! eBook
www.wowebook.org

SQL Server 2016 download

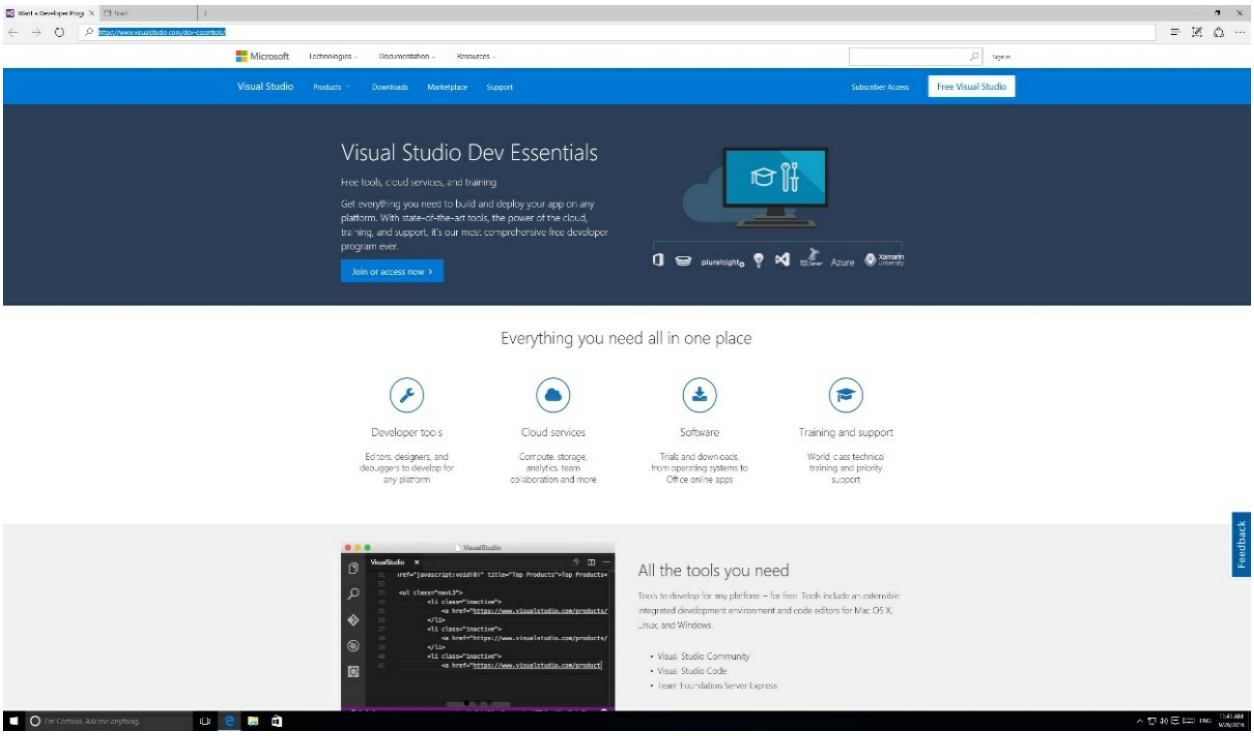
Getting ready

You need to have access to the internet for this recipe.

How to do it...

1. Open your browser and paste this link: https://www.visualstudio.com/dev-essentials/

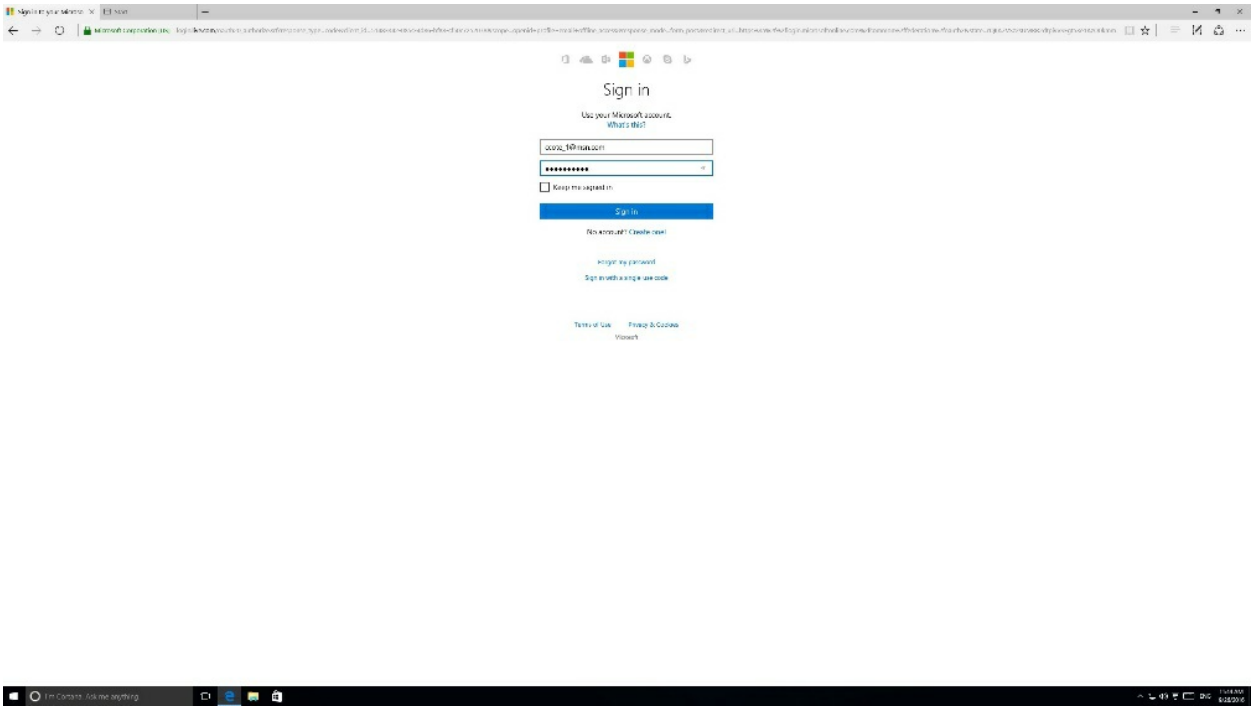
The following page appears in your browser:



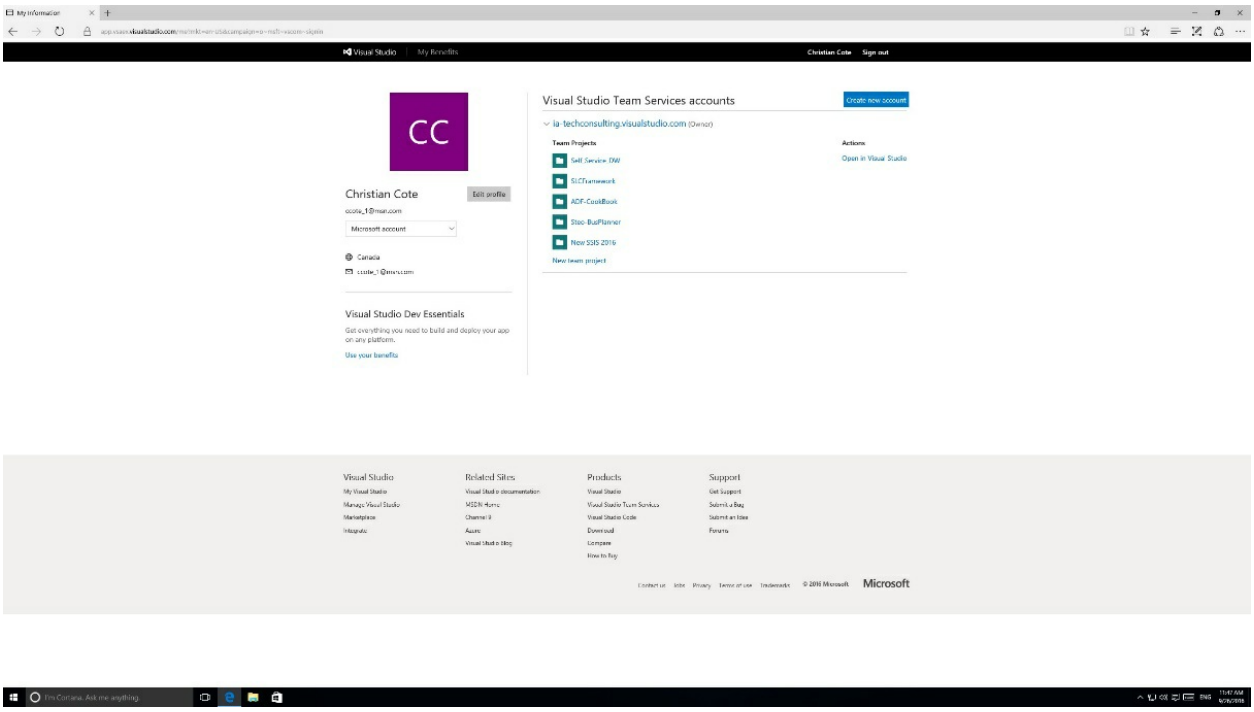
2. Click on the `Sign In` link visible at the tight (top) to log into Visual Studio Dev Essentials. If you don't have an existing subscription, you can create one by clicking on the `Join or access now` button in the

WOW! eBook
www.wowebook.org

middle of the page.

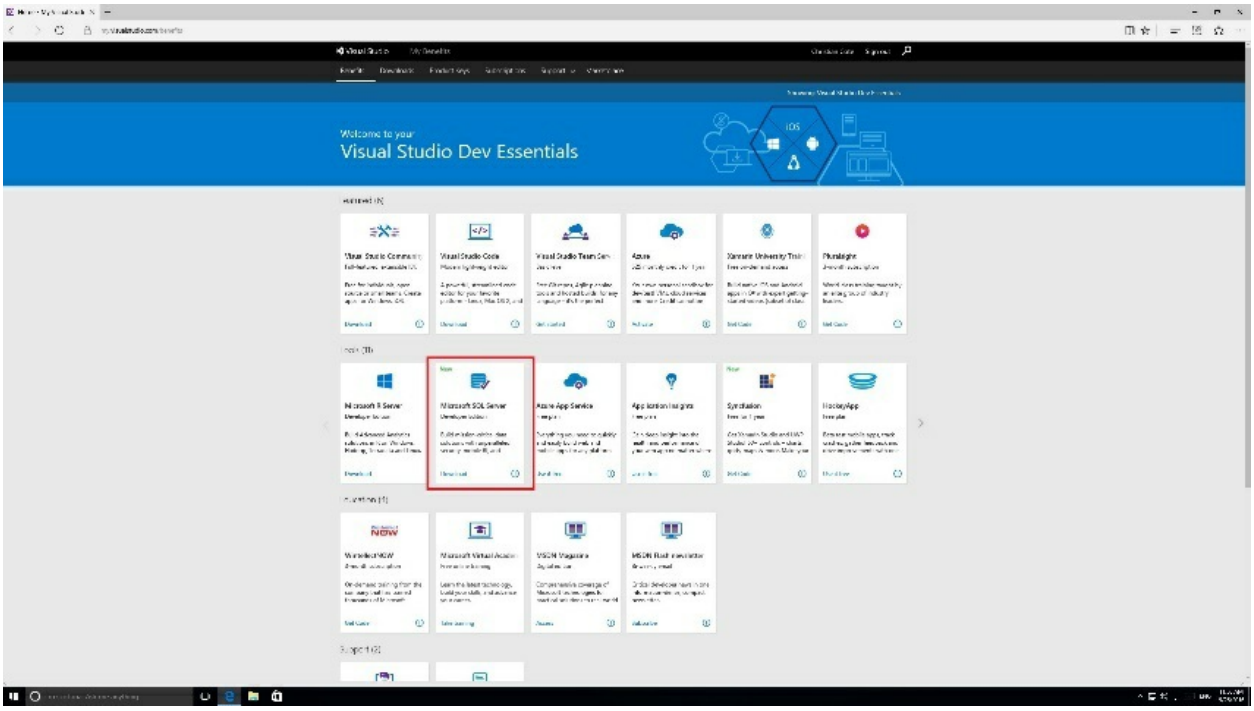


3. You are the directed to the **`My Information`** page. Click on the **`My Benefits`** at the top of the page to access the download section as shown in the following screenshot.



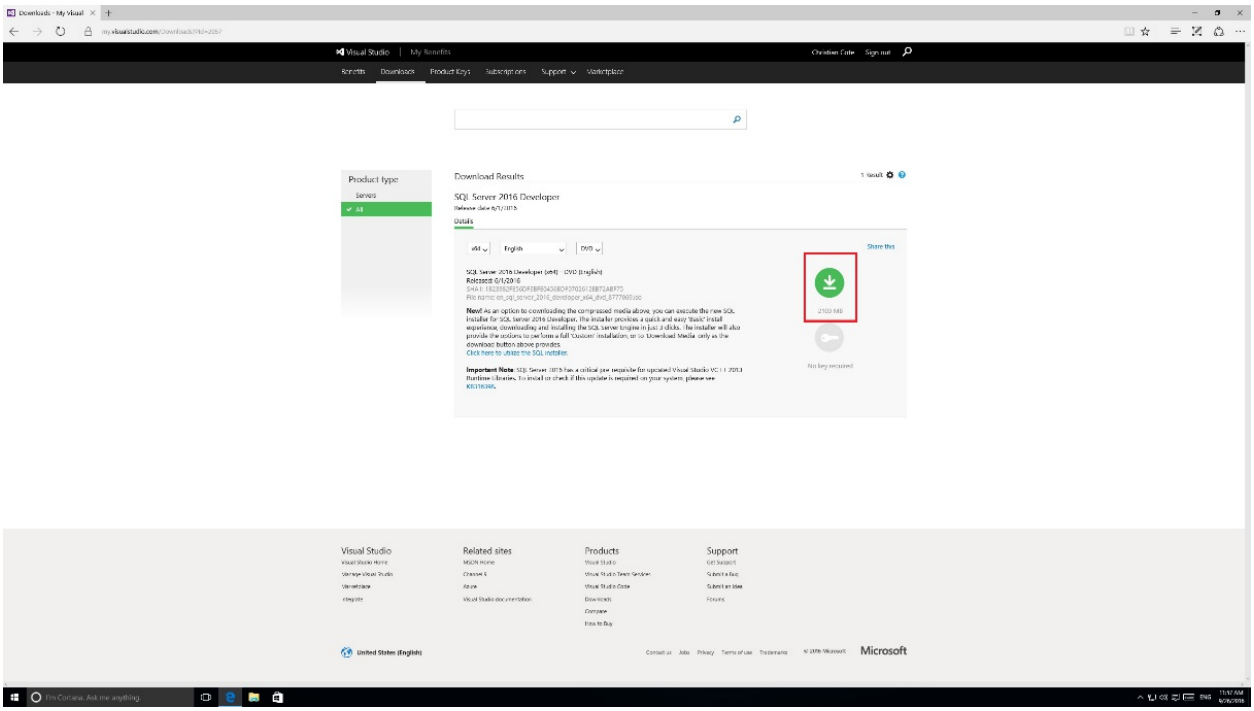
WOW! eBook
www.wowebook.org

4. Click on the download link in the **`Microsoft SQL Server developer edition`** tile as highlighted in the following screenshot.



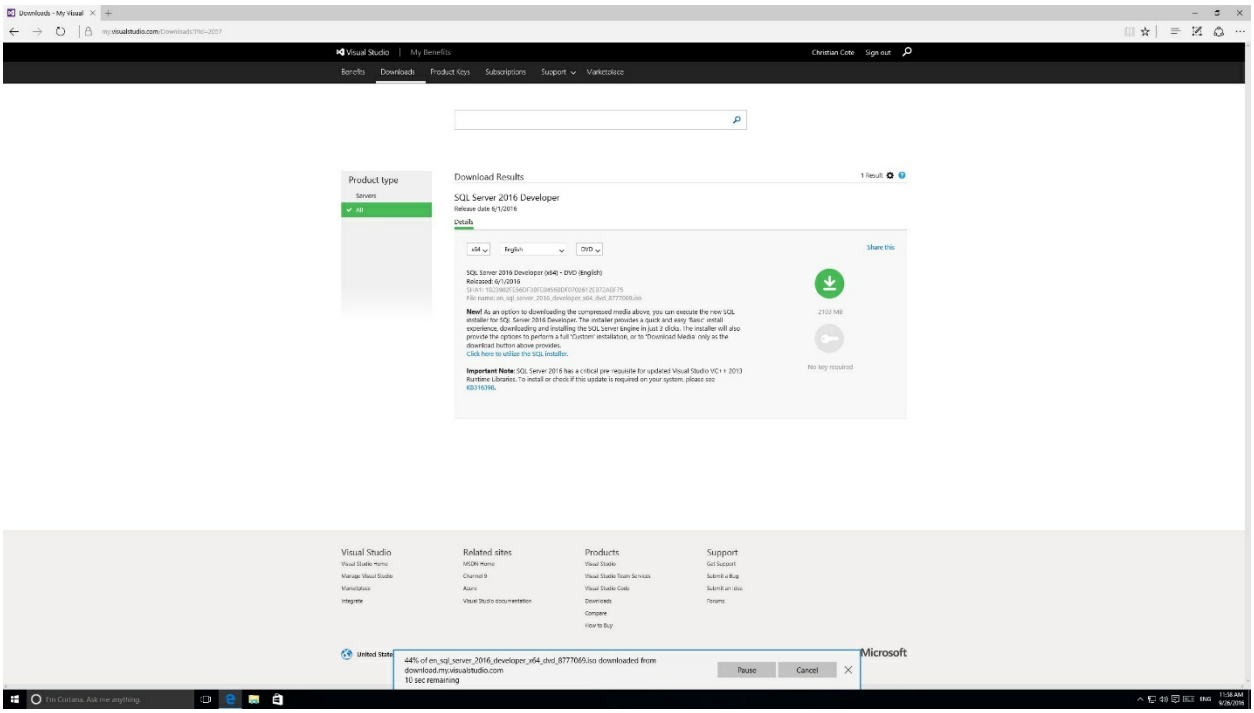
5. This will redirect you to the SQL Server 2016 Developer edition page. Click the download (green arrow) to start downloading the ISO file as shown in the following screenshot.

WOW! eBook
www.wowebook.org



6. Due to its pretty large size, the file may take some time to download. The following screenshot is showing 44% done and 10 seconds left to download. This is due to the fact that the file is being downloaded on an Azure VM. It might take longer for you to download it. Depending on your browser, you should see the file downloading as in the following screenshot.

WOW! eBook
www.wowebook.org



7. Don't mount the ISO file for now. We have to install an external component described in the next section before we proceed with the installation of SQL Server.

WOW! eBook
www.wowebook.org

Installing JRE for Polybase

Java Runtime Engine (JRE) is required for Polybase installation. SQL Server Polybase is the technology that allows data integration from other sources other than SQL Server tables. Polybase is used to access data stored in Hadoop File System (HDFS) or Windows Azure Storage Blob (WASB).

As you will see later in this book, SSIS can now interact with these types of storage natively but having Polybase handy can save us valuable time in our ETL.

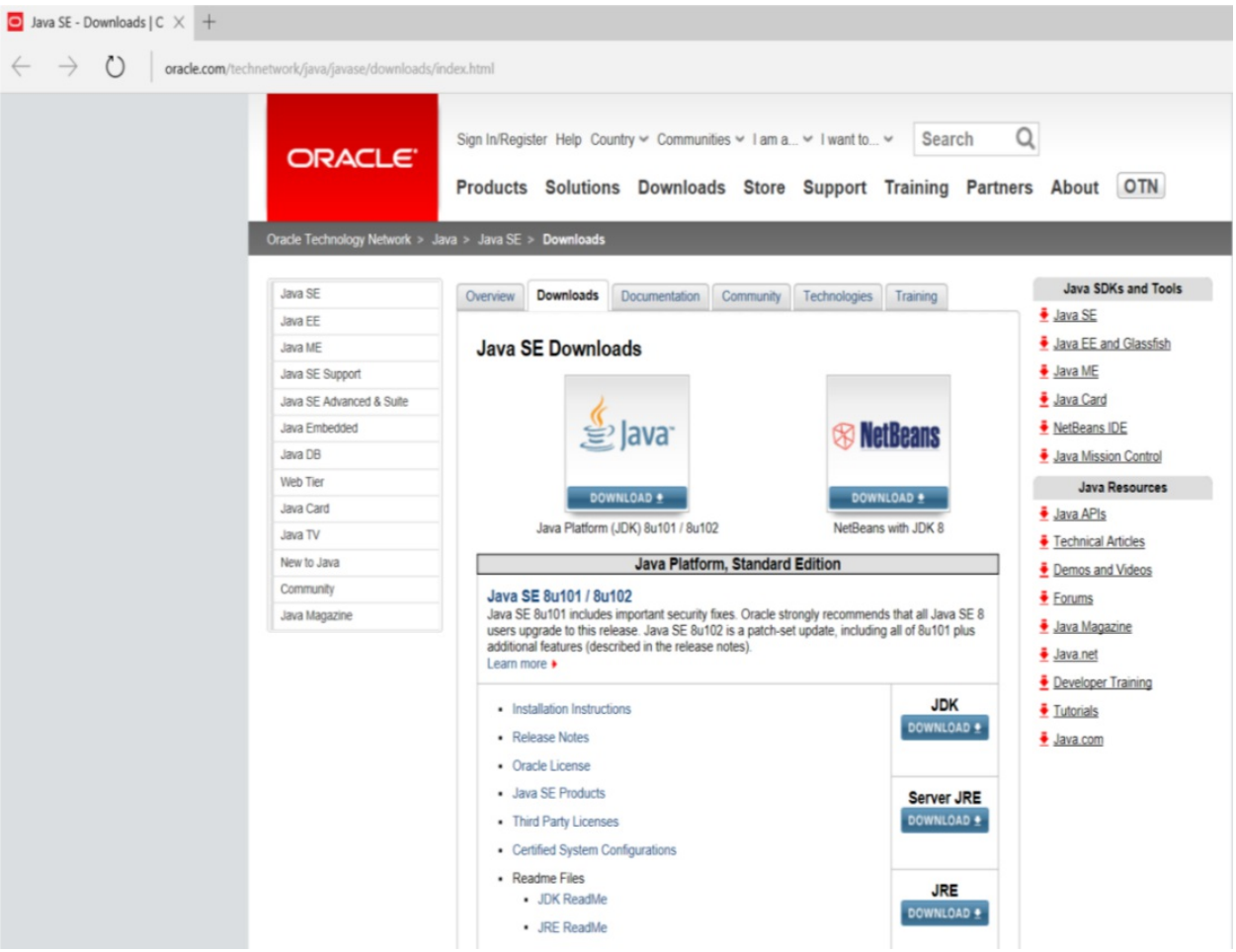
Getting ready

For this recipe you will need to have access to the internet and have administrative rights on your PC to install JRE.

How to do it...

1. To download JRE, follow this link: http://www.oracle.com/technetwork/java/javase/downloads/index.html

WOW! eBook
www.wowebook.org



This directs you to the Java SE Download at Oracle.

2. Click the **`download link`** in the JRE section as shown in the image.

WOW! eBook
www.wowebook.org

Web Tier
Java Card
Java TV
New to Java
Community
Java Magazine

JRE 8u101 Checksum
JRE 8u102 Checksum

Java SE Runtime Environment 8u101

You must accept the Oracle Binary Code License Agreement for Java SE to download this software.

○ Accept License Agreement ◉ Decline License Agreement

Product / File Description	File Size	Download
Linux x86	54.79 MB	jre-8u101-linux-i586.rpm
Linux x86	70.58 MB	jre-8u101-linux-i586.tar.gz
Linux x64	52.68 MB	jre-8u101-linux-x64.rpm
Linux x64	68.49 MB	jre-8u101-linux-x64.tar.gz
Mac OS X	55.99 MB	jre-8u101-macosx-x64.tar.gz
Mac OS X	64.32 MB	jre-8u101-macosx-x64.dmg
Solaris SPARC 64-bit	52 MB	jre-8u101-solaris-sparcv9.tar.gz
Solaris x64	49.85 MB	jre-8u101-solaris-x64.tar.gz
Windows x86 Online	0.71 MB	jre-8u101-windows-i586-iftw.exe
Windows x86 Offline	52.63 MB	jre-8u101-windows-i586.exe
Windows x86	59.42 MB	jre-8u101-windows-i586.tar.gz
Windows x64 Offline	59.17 MB	jre-8u101-windows-x64.exe
Windows x64	62.77 MB	jre-8u101-windows-x64.tar.gz

Java SE Runtime Environment 8u102

You must accept the Oracle Binary Code License Agreement for Java SE to download this software.

○ Accept License Agreement ◉ Decline License Agreement

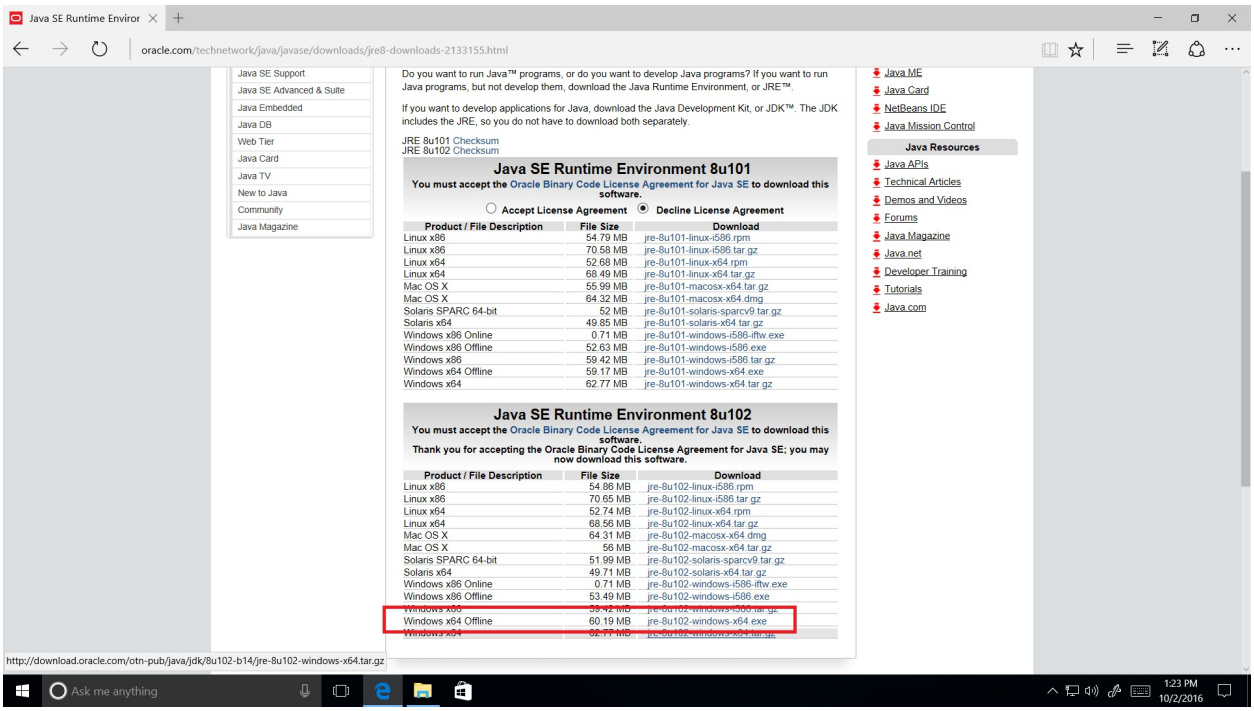
Product / File Description	File Size	Download
Linux x86	54.86 MB	jre-8u102-linux-i586.rpm
Linux x86	70.65 MB	jre-8u102-linux-i586.tar.gz
Linux x64	52.74 MB	jre-8u102-linux-x64.rpm
Linux x64	68.56 MB	jre-8u102-linux-x64.tar.gz
Mac OS X	64.31 MB	jre-8u102-macosx-x64.dmg
Mac OS X	56 MB	jre-8u102-macosx-x64.tar.gz
Solaris SPARC 64-bit	51.99 MB	jre-8u102-solaris-sparcv9.tar.gz
Solaris x64	49.71 MB	jre-8u102-solaris-x64.tar.gz
Windows x86 Online	0.71 MB	jre-8u102-windows-i586-iftw.exe
Windows x86 Offline	53.49 MB	jre-8u102-windows-i586.exe
Windows x86	59.42 MB	jre-8u102-windows-i586.tar.gz
Windows x64 Offline	60.19 MB	jre-8u102-windows-x64.exe
Windows x64	62.77 MB	jre-8u102-windows-x64.tar.gz

Java Resources

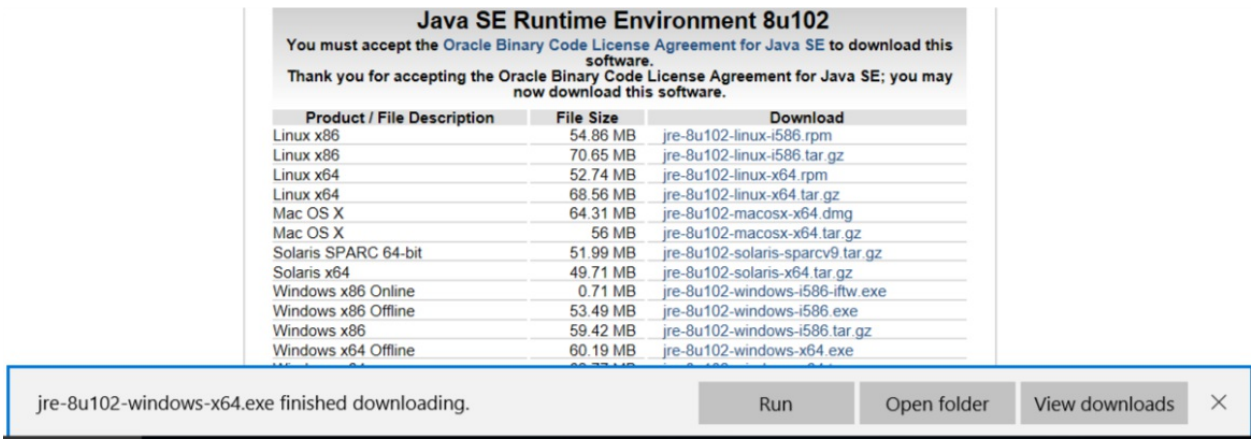
- Java APIs
- Technical Articles
- Demos and Videos
- Forums
- Java Magazine
- Java.net
- Developer Training
- Tutorials
- Java.com

3. You must accept the licence agreement to be able to select a file to download. Select **`Accept License Agreement`** as indicated in the screenshot.

WOW! eBook
www.wowebook.org



4. Since SQL Server 2016 only exists in 64-bit version, download the 64 bit JRE. The version of Java SE Runtime environment might be different than the one that shows in the screen shot. The version in the image is the one available at the time this book was written.



5. Once downloaded, launch the installer. Click on **Run** as shown in Edge browser. Otherwise, go to your download folder and double-click on the file you just downloaded (`jre-8U102-windows-x64.exe` in our case).

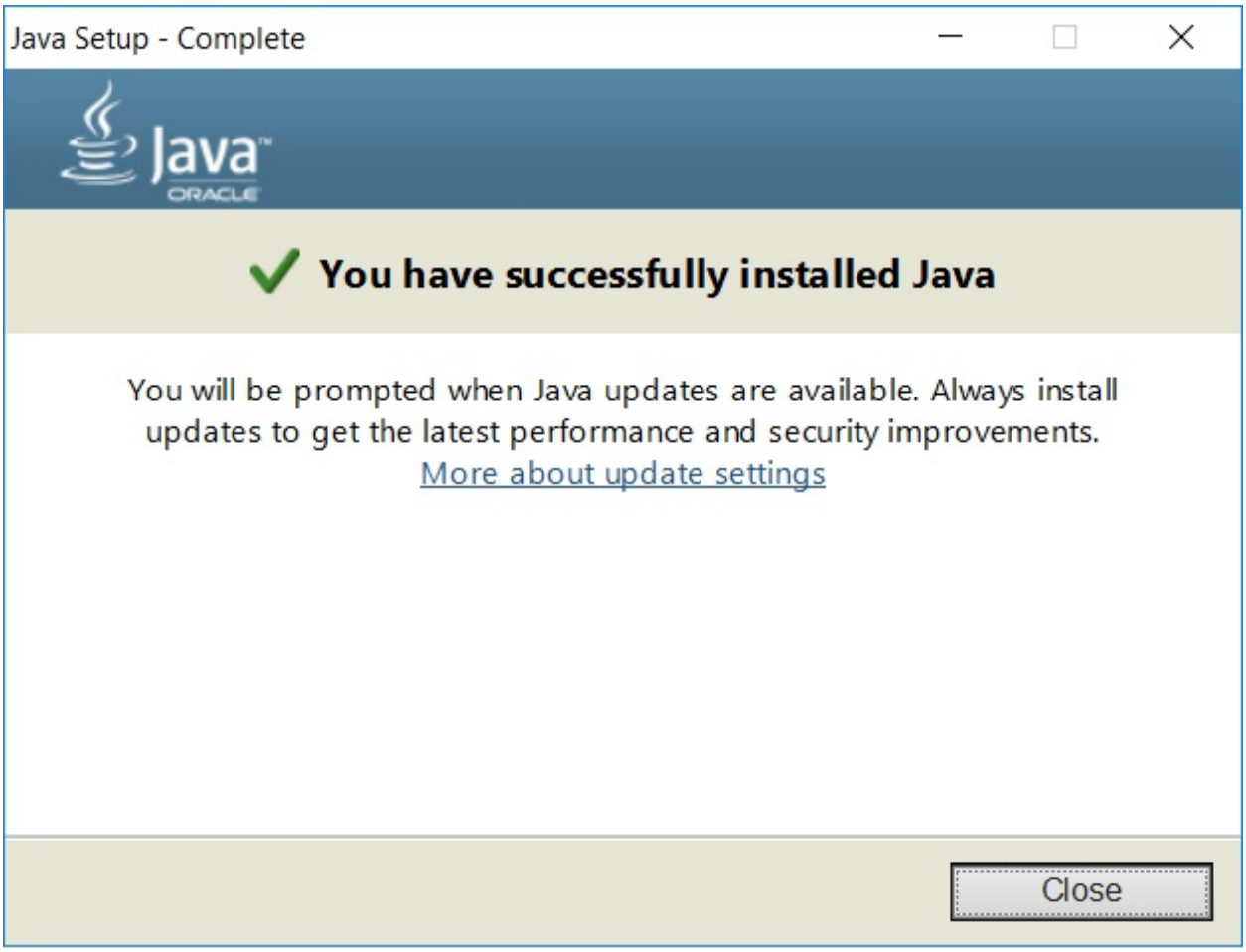
WOW! eBook
www.wowebook.org

6. The Oracle JRE installation starts. Click **`Install`**. The following screen appears. It indicates the progress of the JRE installation.

WOW! eBook
www.wowebook.org

7. Once the installation is completed, click **`Close`** to quit the installer shown as follows:

WOW! eBook
www.wowebook.org



You are now ready to proceed to SQL Server 2016 installation. We'll do that in the next section

How it works...

Microsoft integrated Polybase in SQL Server 2016 to connect almost natively to Hadoop and NoSQL platforms. Here are the technologies it allows us to connect to

- HDFS (Hadoop File System): Hortonworks, Cloudera
- Azure blob storage

Since Hadoop is using Java technology, JRE is used to interact with its functionalities.

WOW! eBook
www.wowebook.org

Installing SQL Server 2016

This section will go through the installation of SQL Server engine that will host the database objects used throughout this book.

Here are the features required for the SQL Server Setup:

- Database engine: the core of SQL Server. It manages the various database objects like tables, views, stored procedures, etc.
- Analysis Services: allows us to create a data semantic layer that ease data consumption by the users.
- Reporting Services (native): allow us to create various reports, paginated, mobile and KPI's for data consumption.
- Integration Services: the purpose of this book, SQL Server data movement service.
- Management tools: we'll talk about it in the next section.
- SQL Server Data Tools: we'll talk about it in the next section.

Getting ready

This recipe assumes that you have downloaded SQL Server 2016 Developer edition and you have installed Oracle JRE.

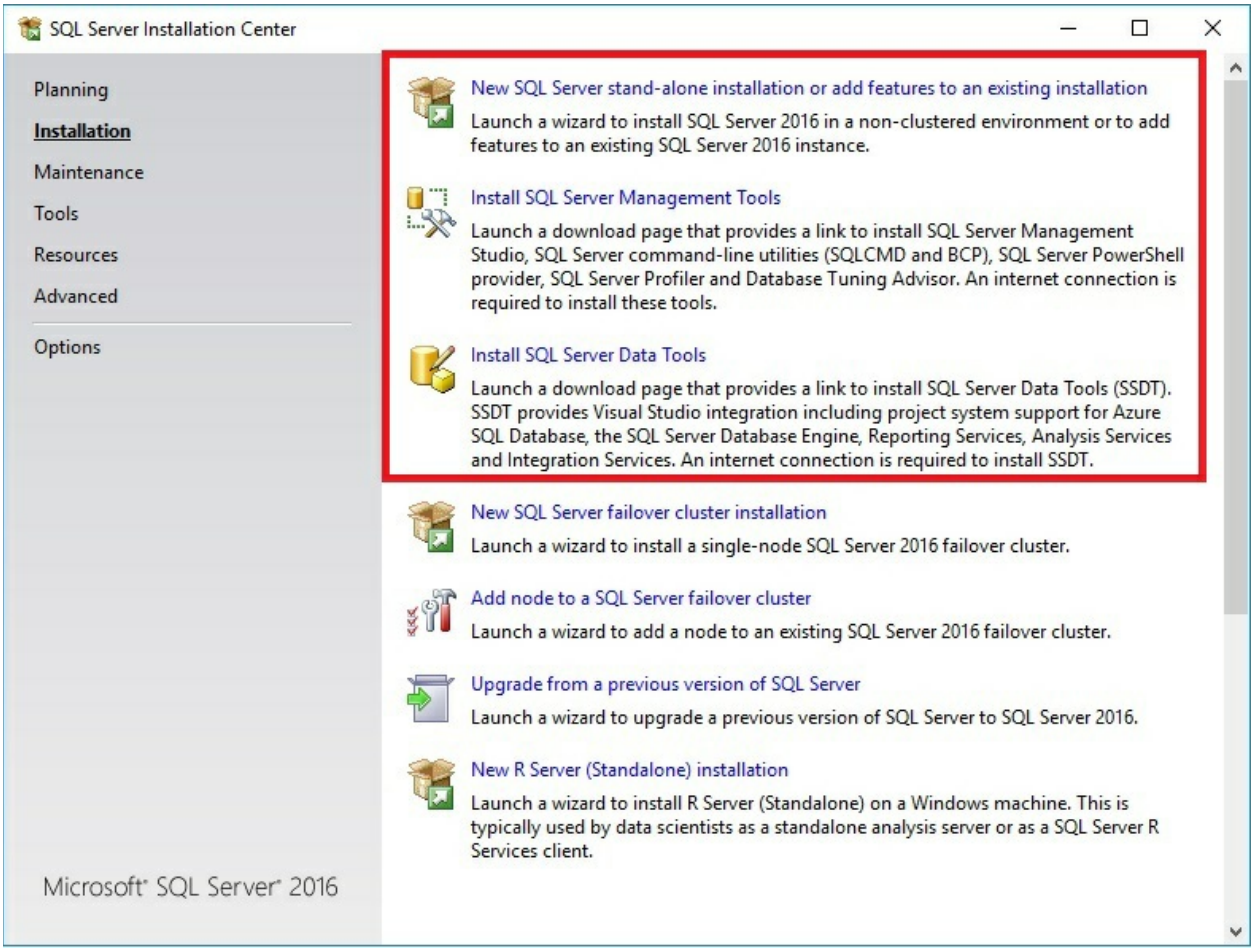
How to do it...

1. The first step is to open the ISO file that you downloaded from Microsoft Visual Studio Dev Essentials website as described in the recipe, *SQL Server 2016 Download*. If you're using Windows 7, you'll need to extract the ISO file into a folder. Third party file compress utilities such as WinRar, Winzip or 7zip (and there are many more) can handle ISO file decompression. The setup files will be uncompressed in the folder of your choice. Other version of Windows such as Windows 8.1, Windows 10 or Windows Server 2012 and beyond, simply double-

WOW! eBook
www.wowebook.org

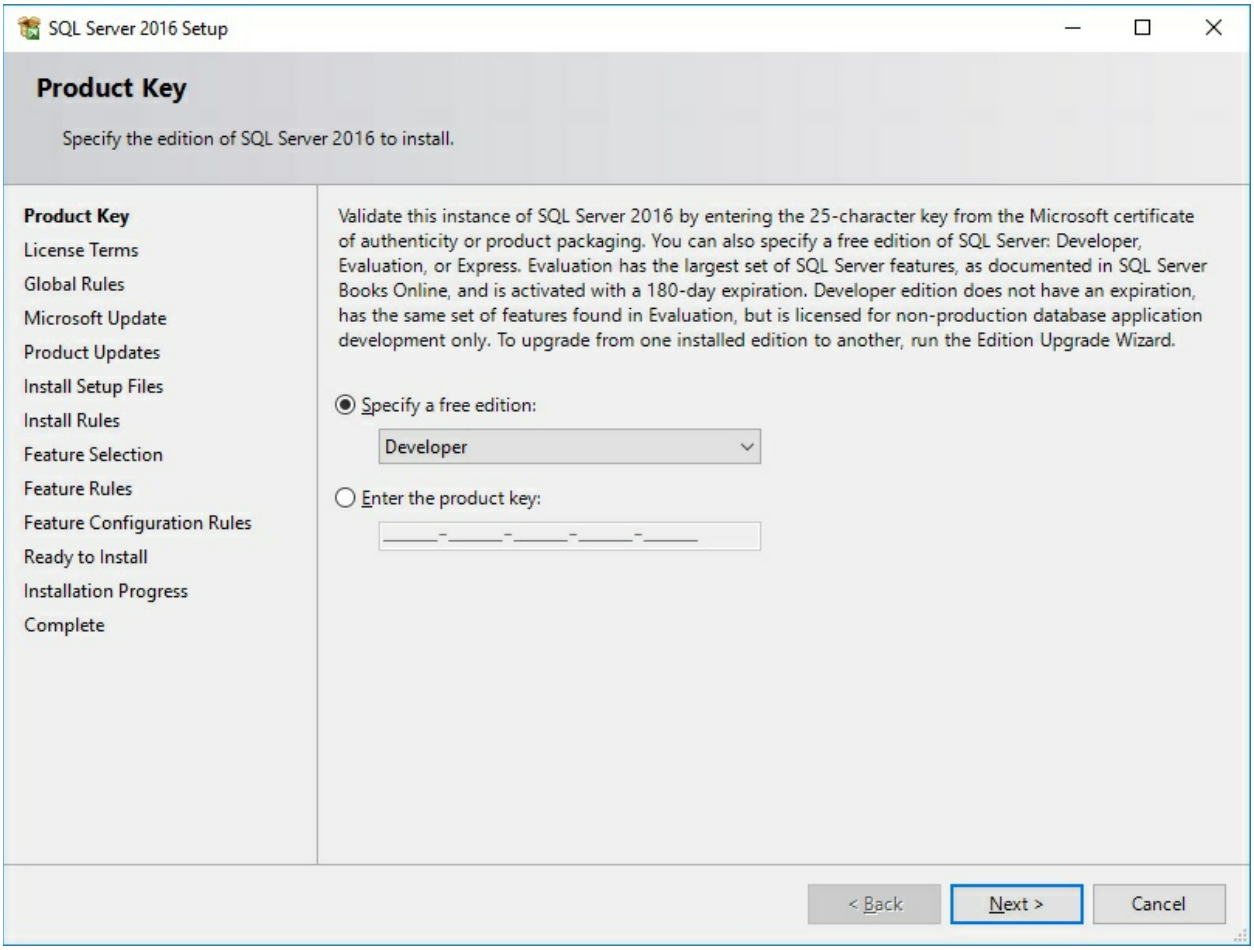
click on the ISO file that you have downloaded previously and a new drive will appear in Windows Explorer.

2. Double-click on the file named `Setup.exe` to start the SQL Server Installation utility. The features we're going to install are highlighted in the image:
    - SQL Server service which corresponds to the **`New SQL Server stand-alone installation`**. This will install a local instance (service) of SQL Server on your PC.
    - SQL Server Management tools: The tool used to create, query and manage SQL Server objects. It's the second option on the screen below: **`Install SQL Server Management Tool`**.
    - SQL Server Data Tool: contains the Visual Studio templates to develop and deploy SQL Server Databases, Integration Services package, Analysis Services cubes and Reporting Services reports. It's the third option in the following screenshot; **`Install SQL Server Data Tools`**.



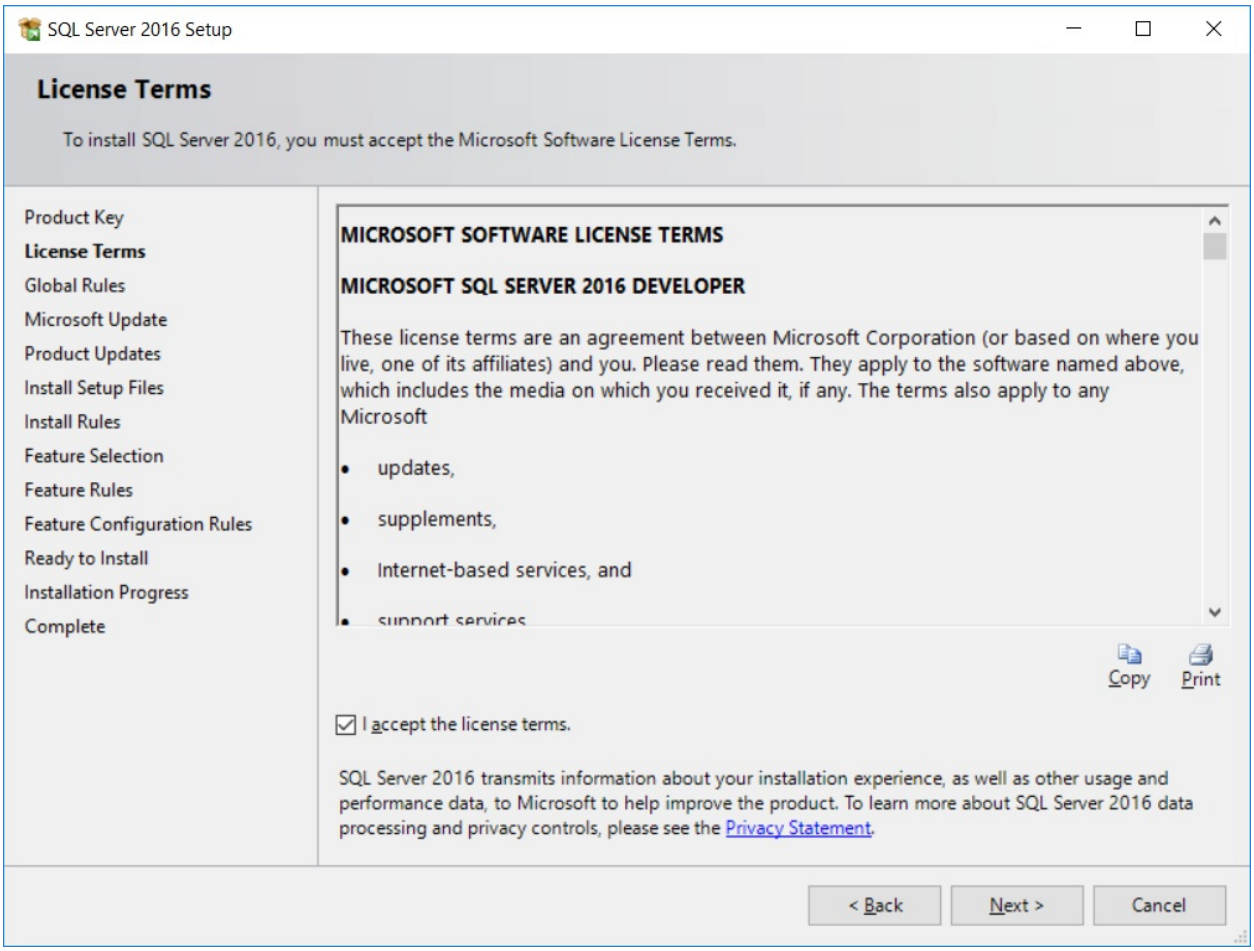
WOW! eBook
www.wowebook.org

3. Insert Image B05695_01_14.pngFrom the installation utility, select the **`New SQL Server stand-alone installation...`** option. A new SQL Server setup window opens.
4. The product key page allows us to specify an edition to install. Since we're going to use the free developer edition, click **`Next`** to go to the next page.



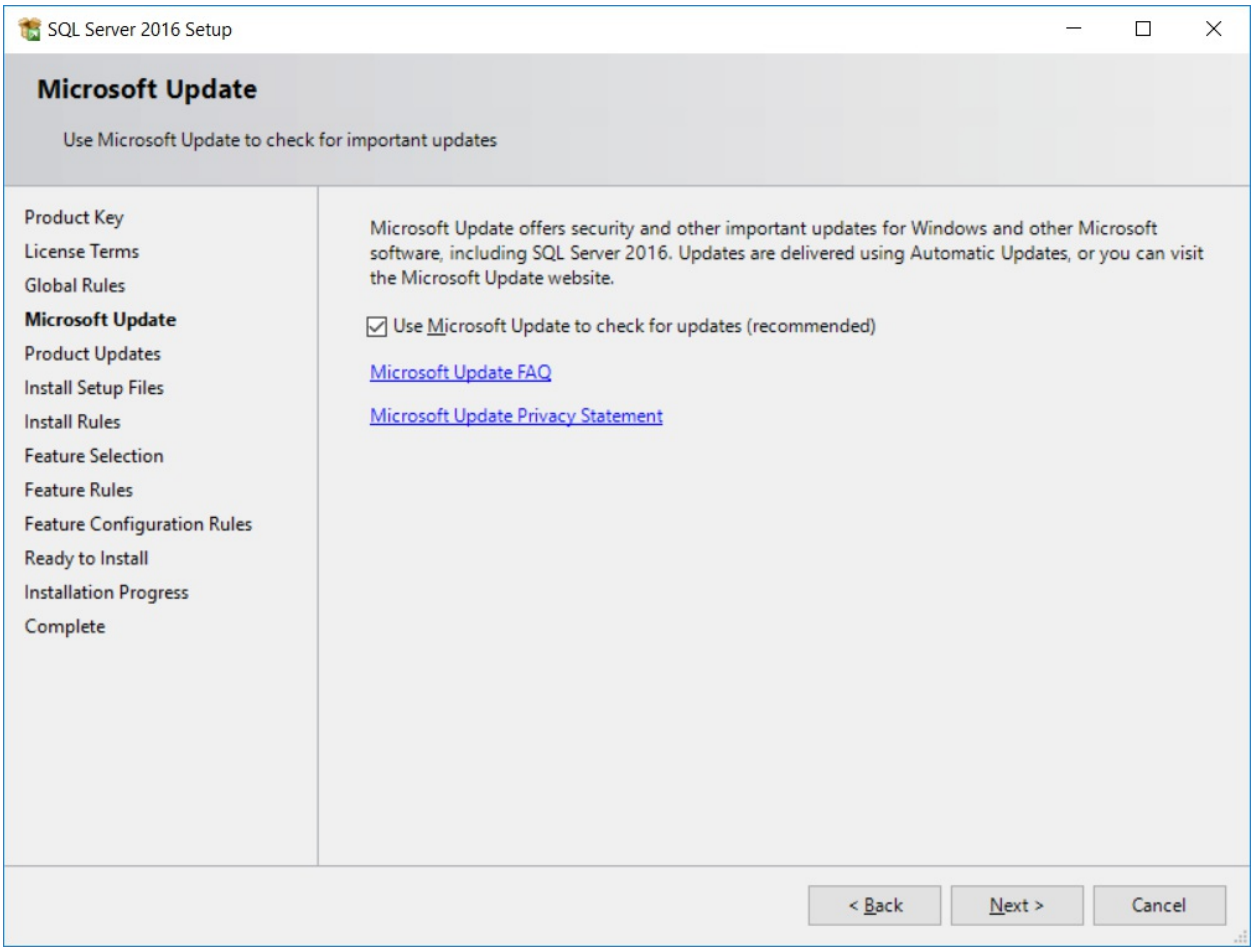
5. Accept the license terms and click **`Next`** to go to the next page.

WOW! eBook
www.wowebook.org



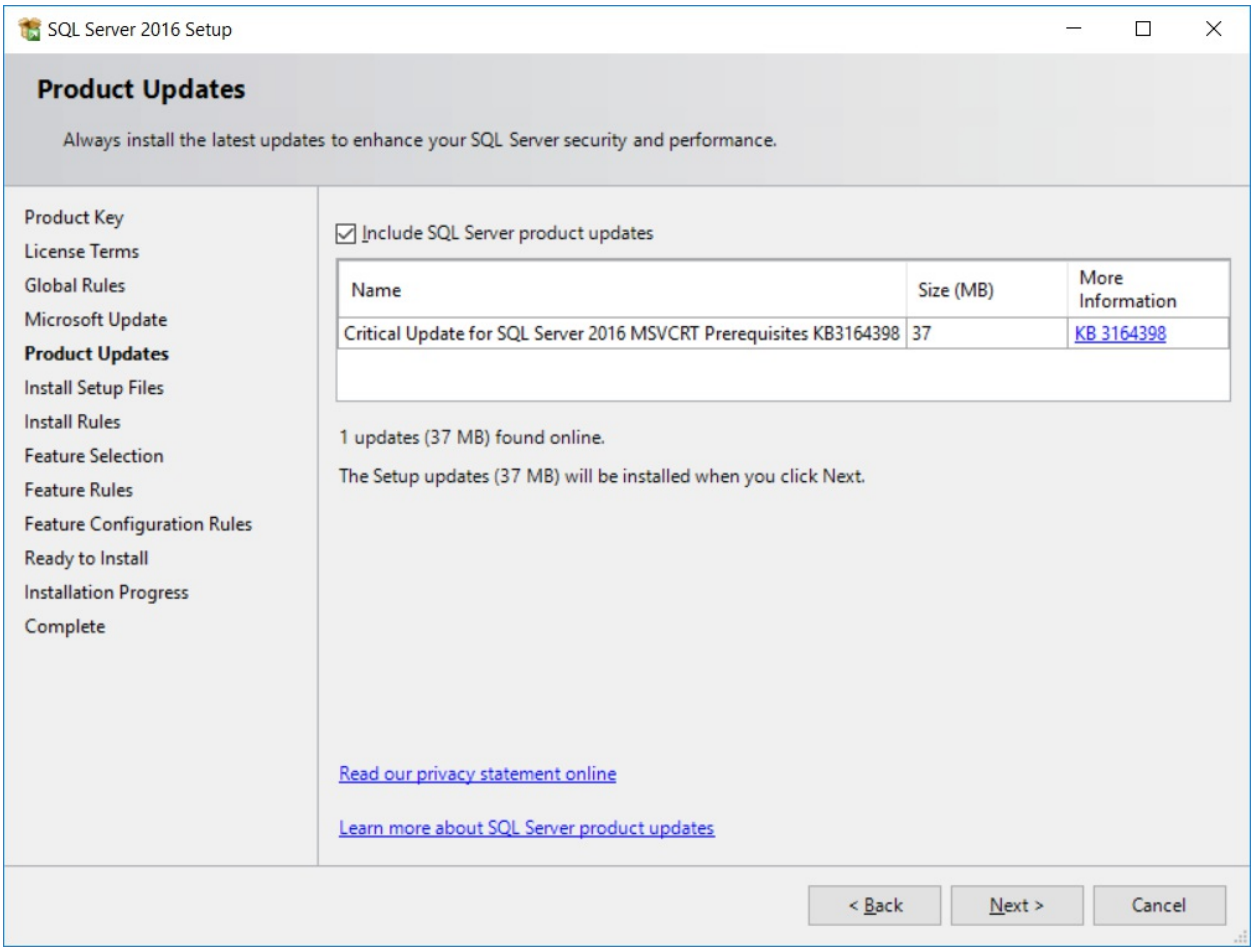
6. At this step, the SQL Server setup will check for product update and will integrate itself in the Windows Updates check that are done regularly on your machine. This step is not mandatory but it's better to use the latest bits of code. Check the **`Use Microsoft Update...`** and click **`Next`**.

WOW! eBook
www.wowebook.org



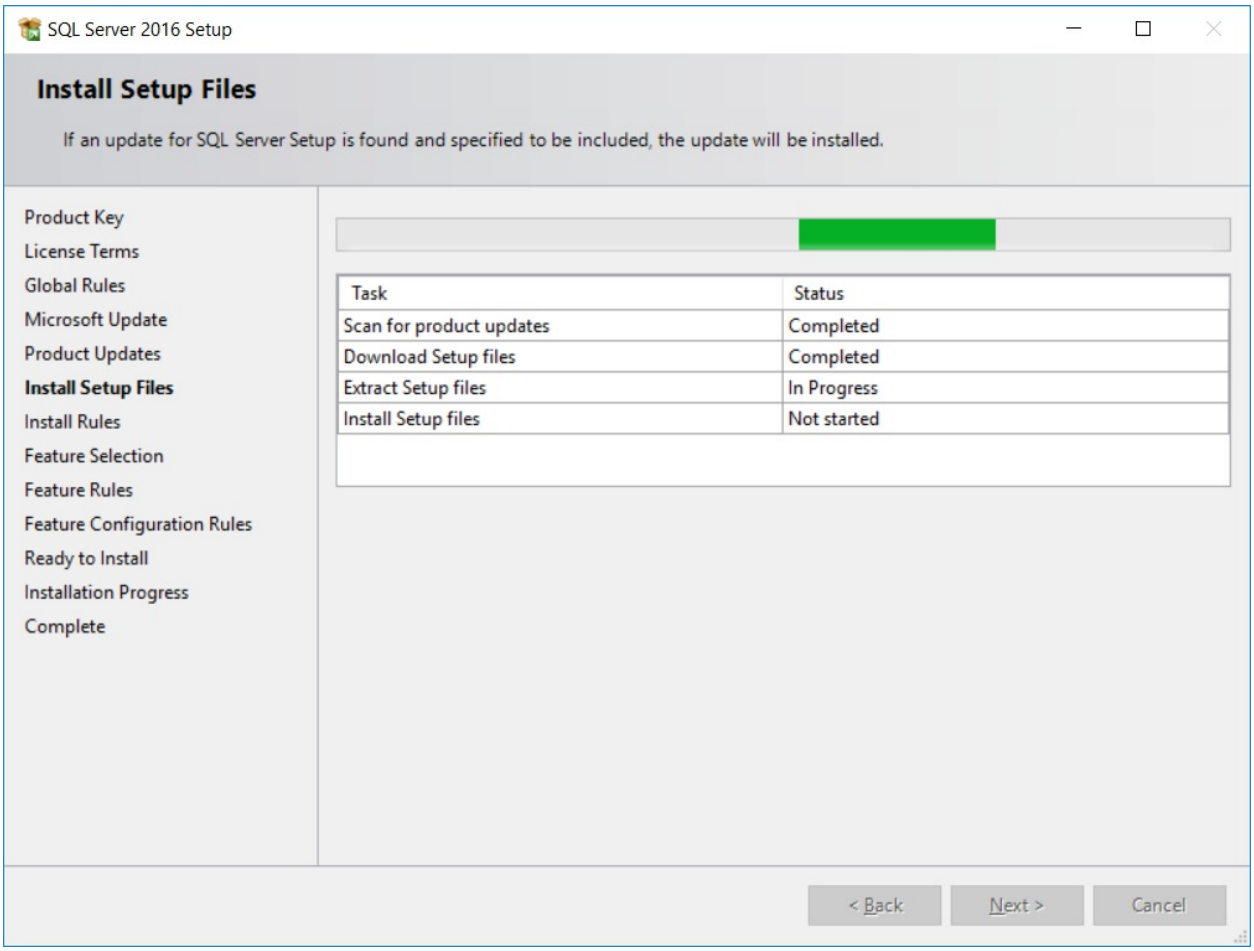
7. Some updates might be found during setup. You can get more information's on these updated by clicking the link in the **`More Information`** column. Click **`Next`** to install the updates.

WOW! eBook
www.wowebook.org



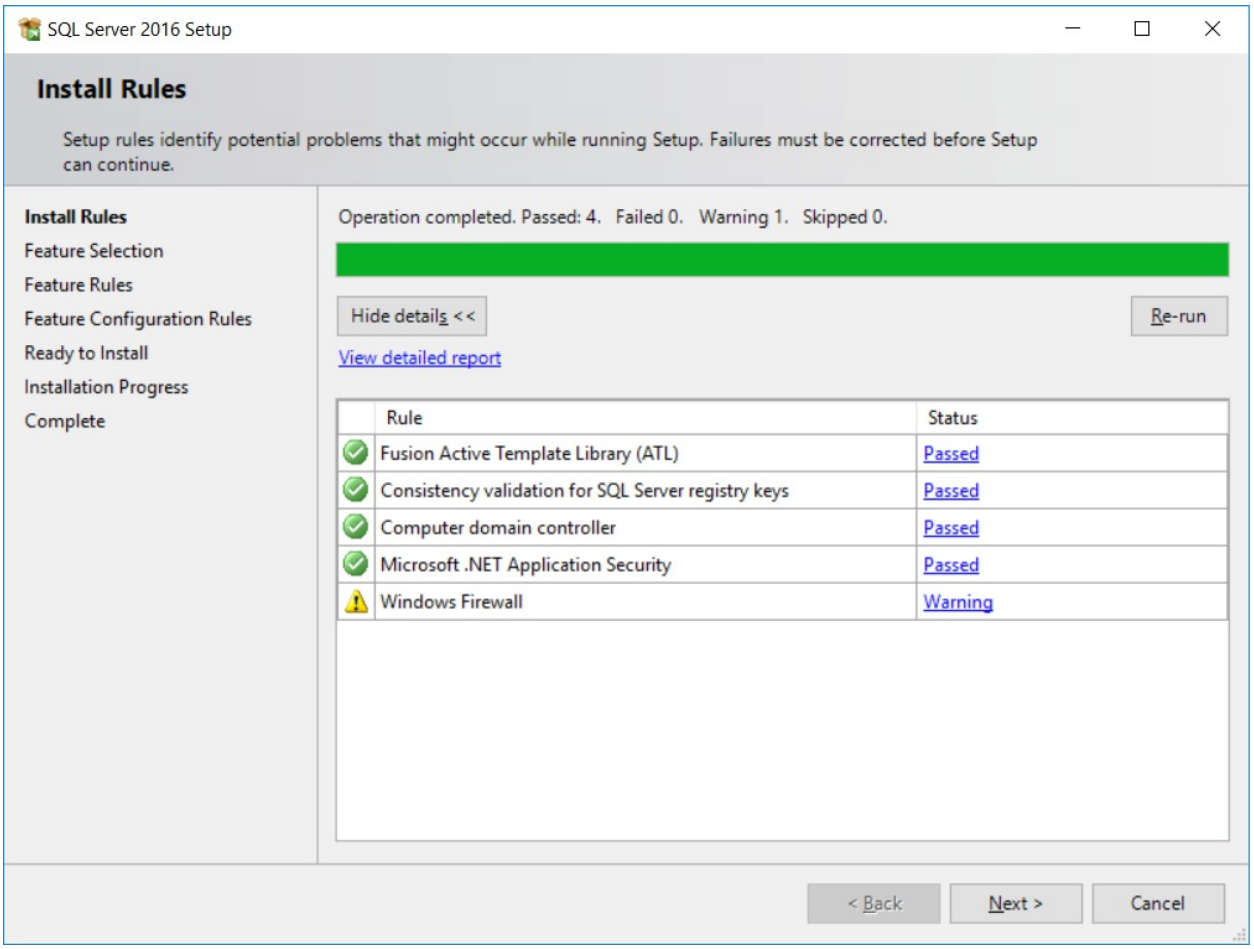
8. This step simply checks to make sure that the latest version of SQL server is installed. Click **`Next`** once the setup files are installed.

WOW! eBook
www.wowebook.org



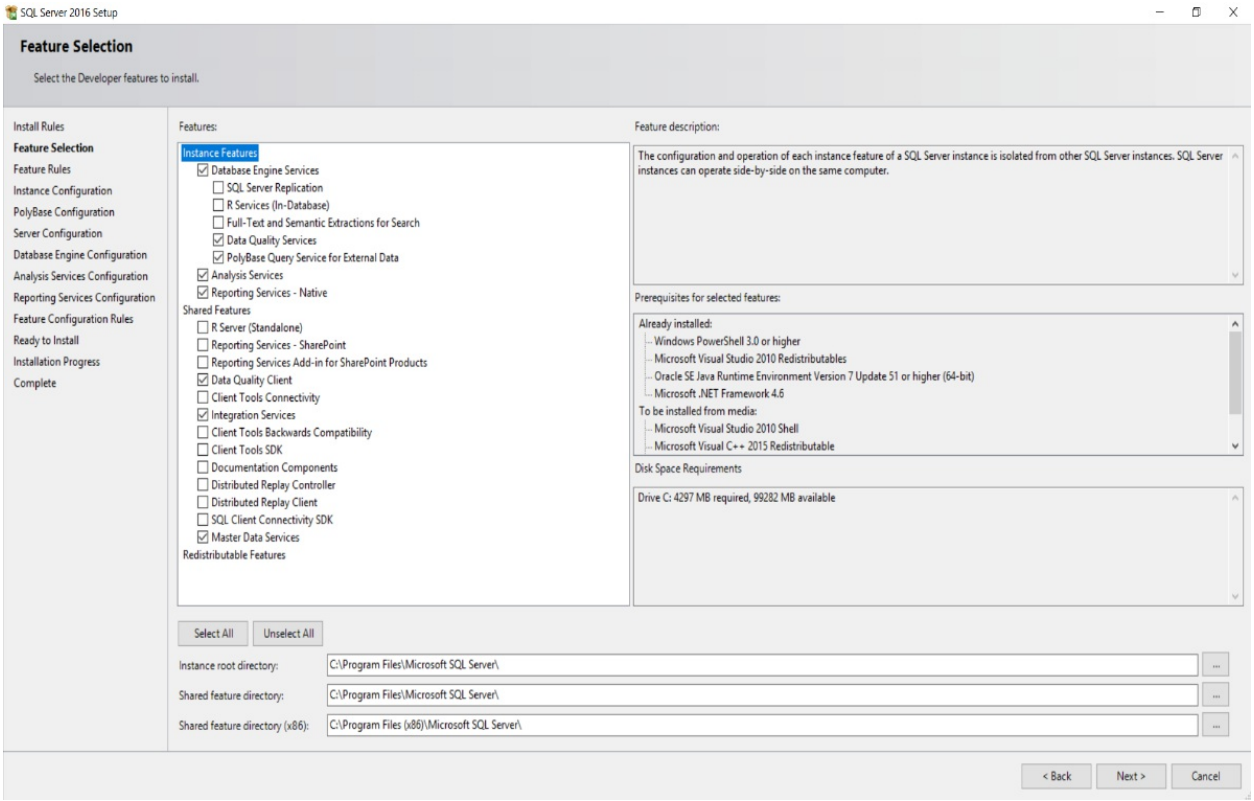
9. The SQL server setup will check several rules to ensure that the computer where we want to install it is setup properly. You might get a warning due to Windows Firewall rules. This tells you that the ports (1433 by default) is not open and SQL Server won't be available from outside your PC. Don't worry about it. Since we'll be using SQL Server from our PC only, we do not need to open any ports for now. Click **`Next`** to advance to the feature selection page.

WOW! eBook
www.wowebook.org



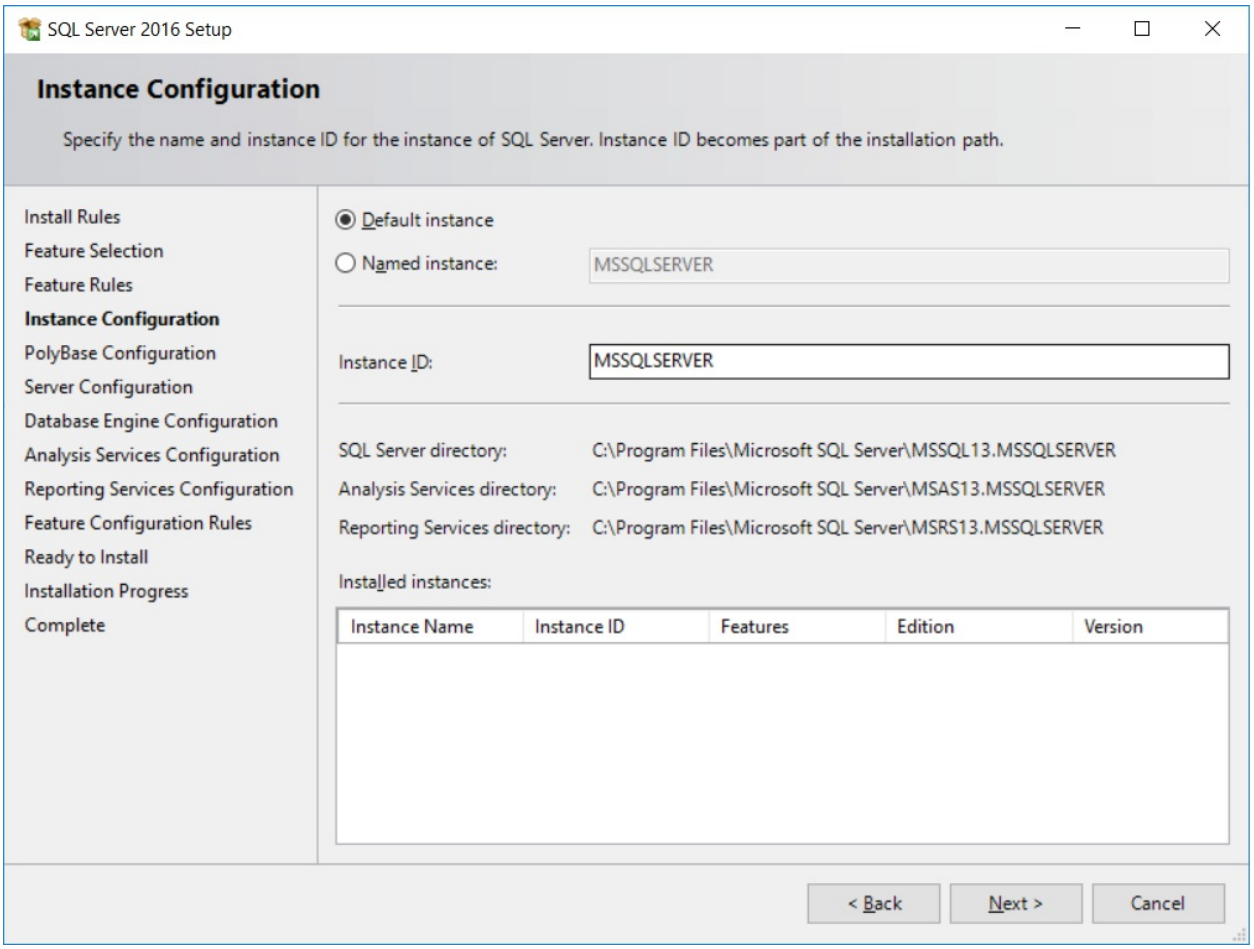
10. Select all features checked in the previous image and click **`Next`**.

WOW! eBook
www.wowebook.org



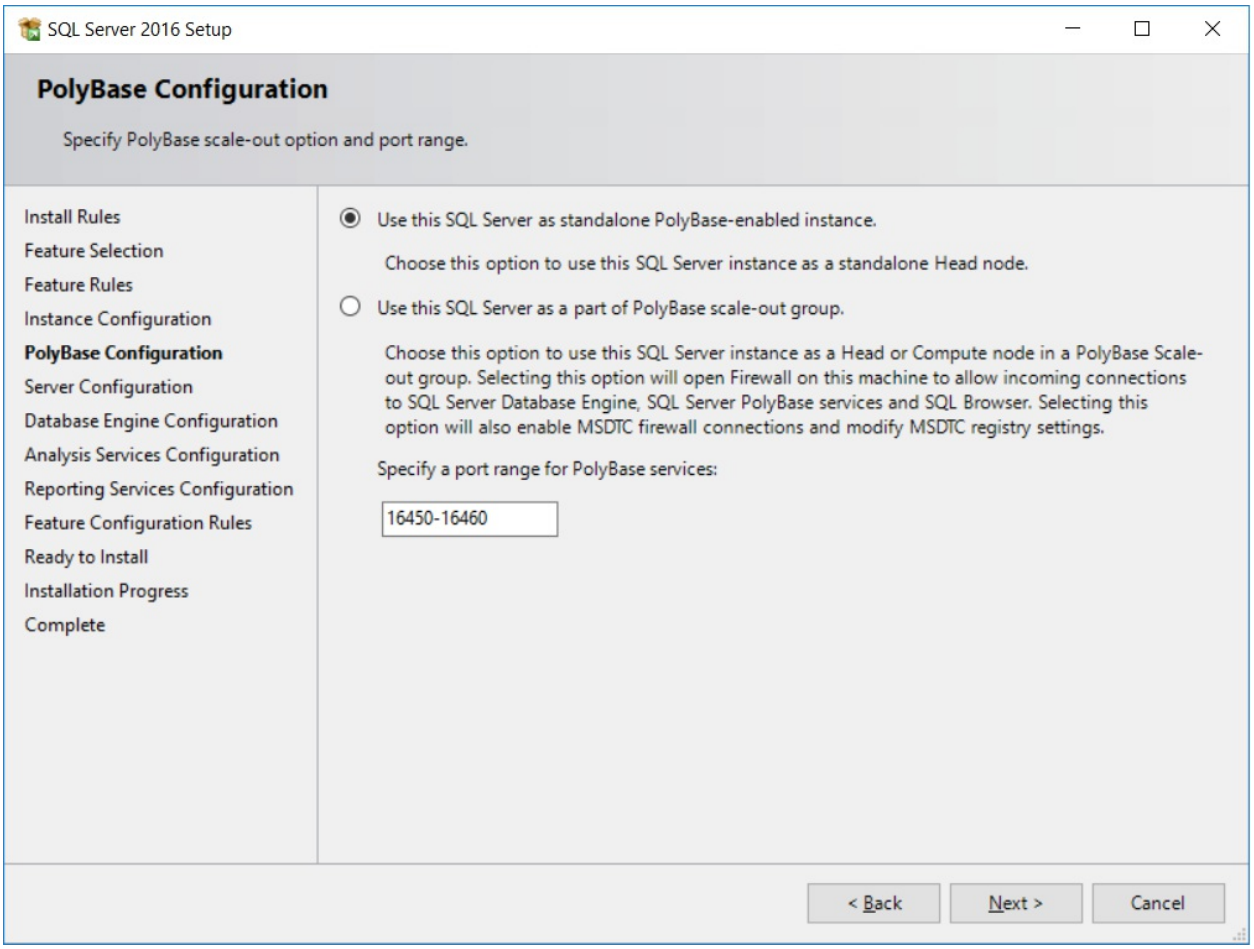
11. The instance configuration allows to specify a name for the SQL Server service. This is done by selecting the **`Named instance`** radio button. Since we'll only use one SQL Server instance, leave the **`Default instance`** selected and click **`Next`**.

WOW! eBook
www.wowebook.org



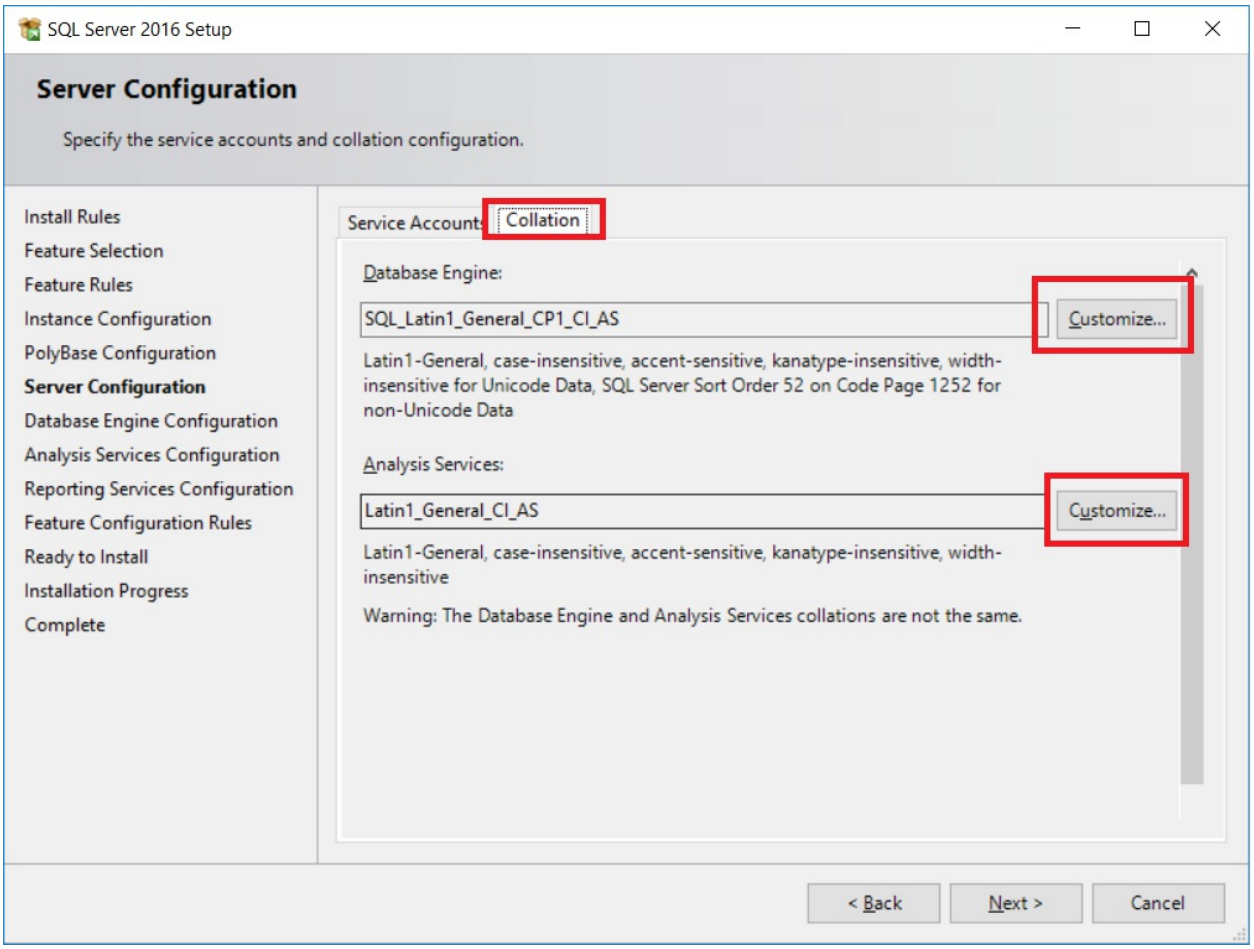
12. This page allows SQL Server to be part of a PolyBase scale out group. Since we're only setting up SQL server PolyBase to be used by one instance, leave the default to **`Use this SQL Server as standalone PolyBase-enabled instance`** and click **`Next`**.

WOW! eBook
www.wowebook.org



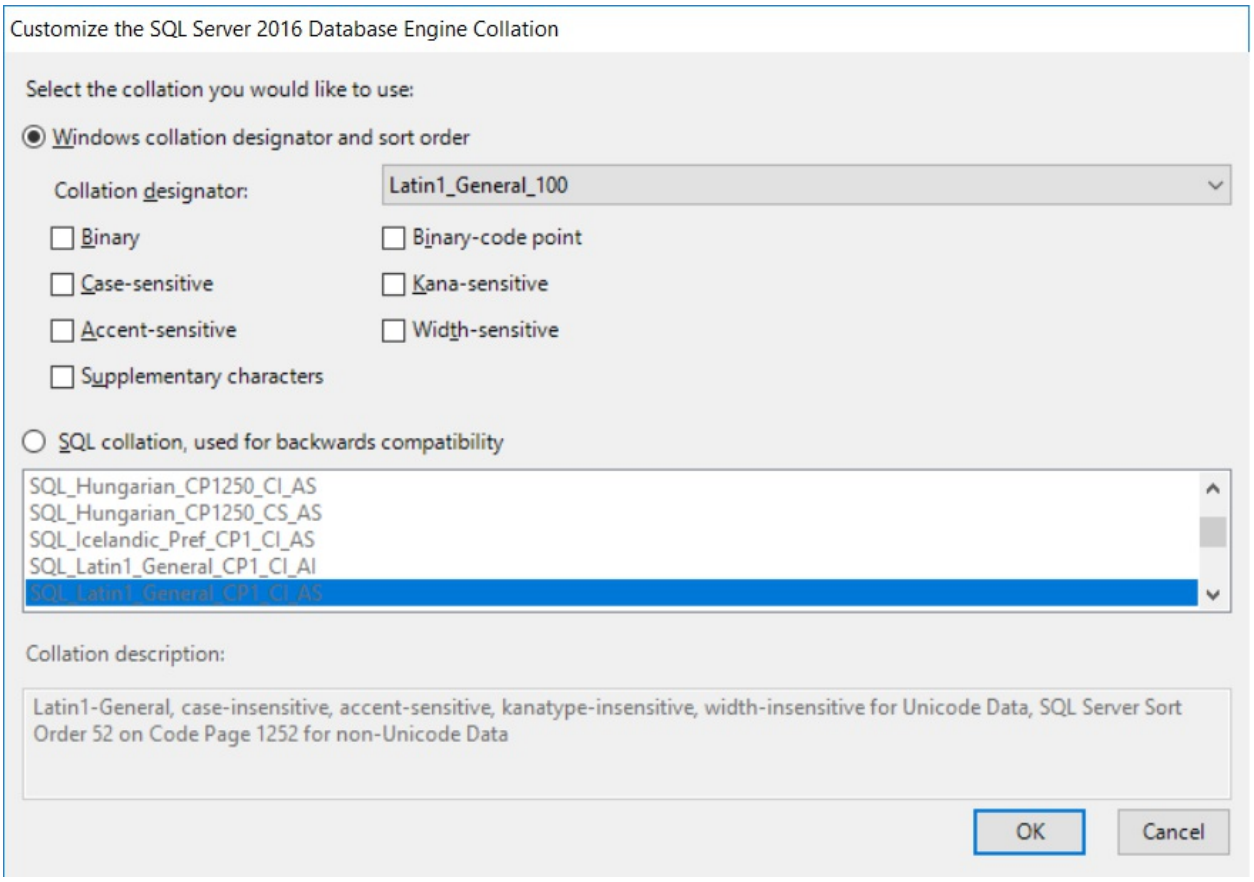
13. This page allows SQL Server to be part of a PolyBase scale out group. Since we're only setting up SQL server PolyBase to be used by one instance, leave the default to **`Use this SQL Server as standalone PolyBase-enabled instance`** and click **`Next`**.

WOW! eBook
www.wowebook.org



14. Server configuration. This step allows us to specify distinct or specific service accounts. Since we're installing SQL Server on a single development machine, we'll use the default accounts.
15. Click on the **`Collation`** tab as highlighted on the following screen shot . The default collation used by SQL server is **`SQL Latin1_General_CP1_CI_AS`**. This is the legacy collation. The choice of the collation is important for text columns. The latest (less bugs) collation is **`Latin1_General_100`**. The last characters **`CI`** and **`AS`** are for **`Case Insensitive`** and **`Accent Sensitive`**.
16. We'll change the collations defaults, click **`Customize...`** at the end of **`SQL Latin1_General_CP1_CI_AI`**.

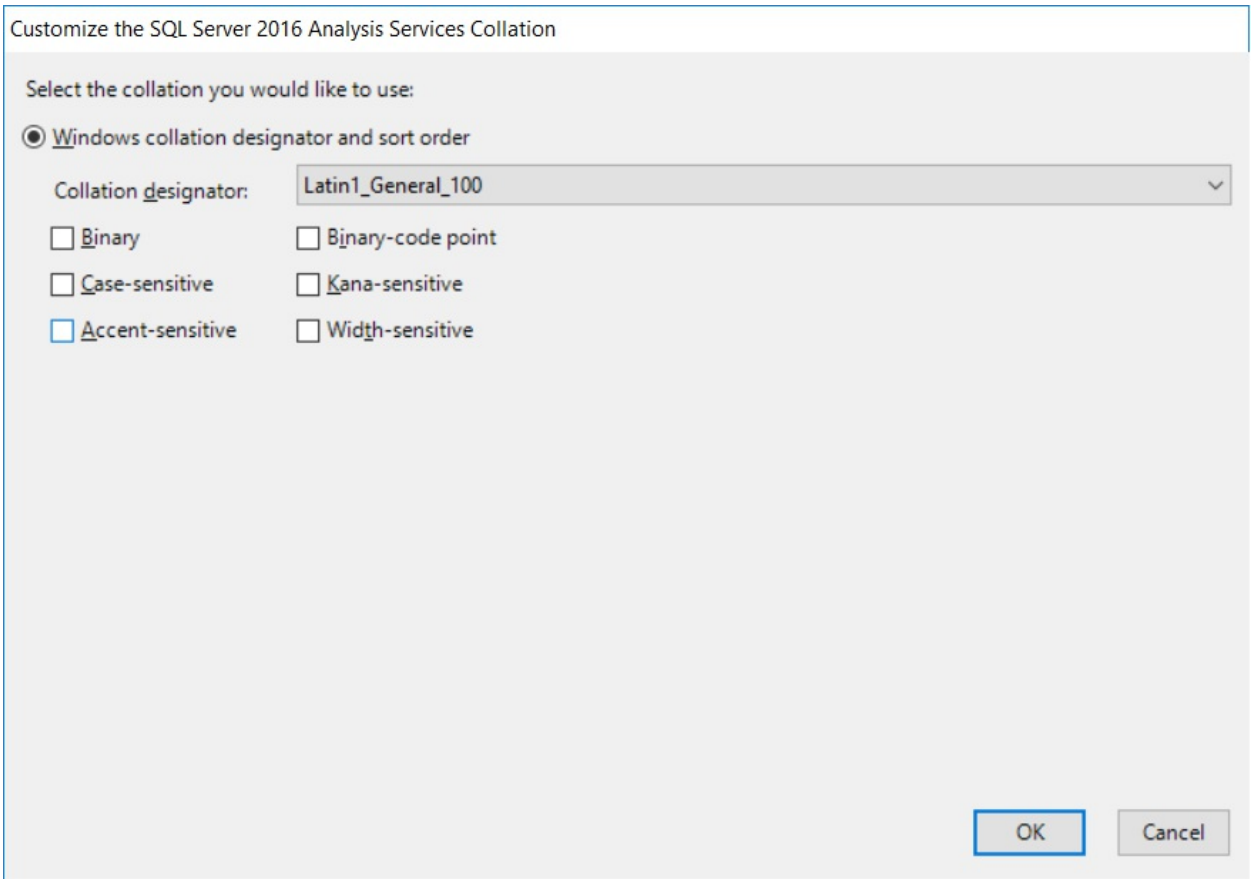
WOW! eBook
www.wowebook.org



17. As stated before, we'll use the Windows collation designator **`Latin1_General_100`**. Uncheck the **`Accent sensitive`**. This allows SQL Server to sort character columns without using accentuated characters. For example, suppose that our application has a FirstName column and we have the following first names:
    - Joel
    - Joël

18. If we query SQL Server filtering on FirstName = 'Joel' with Accent-sensitive collation option, we would end up retrieving the value Joel only. If do not select the Accent-sensitive collation option, we will get both values.
19. Click **`OK`** when done to return to the previous screen. We'll do the same for Analysis Services; click **`Customize`** to customize Analysis Services collation.

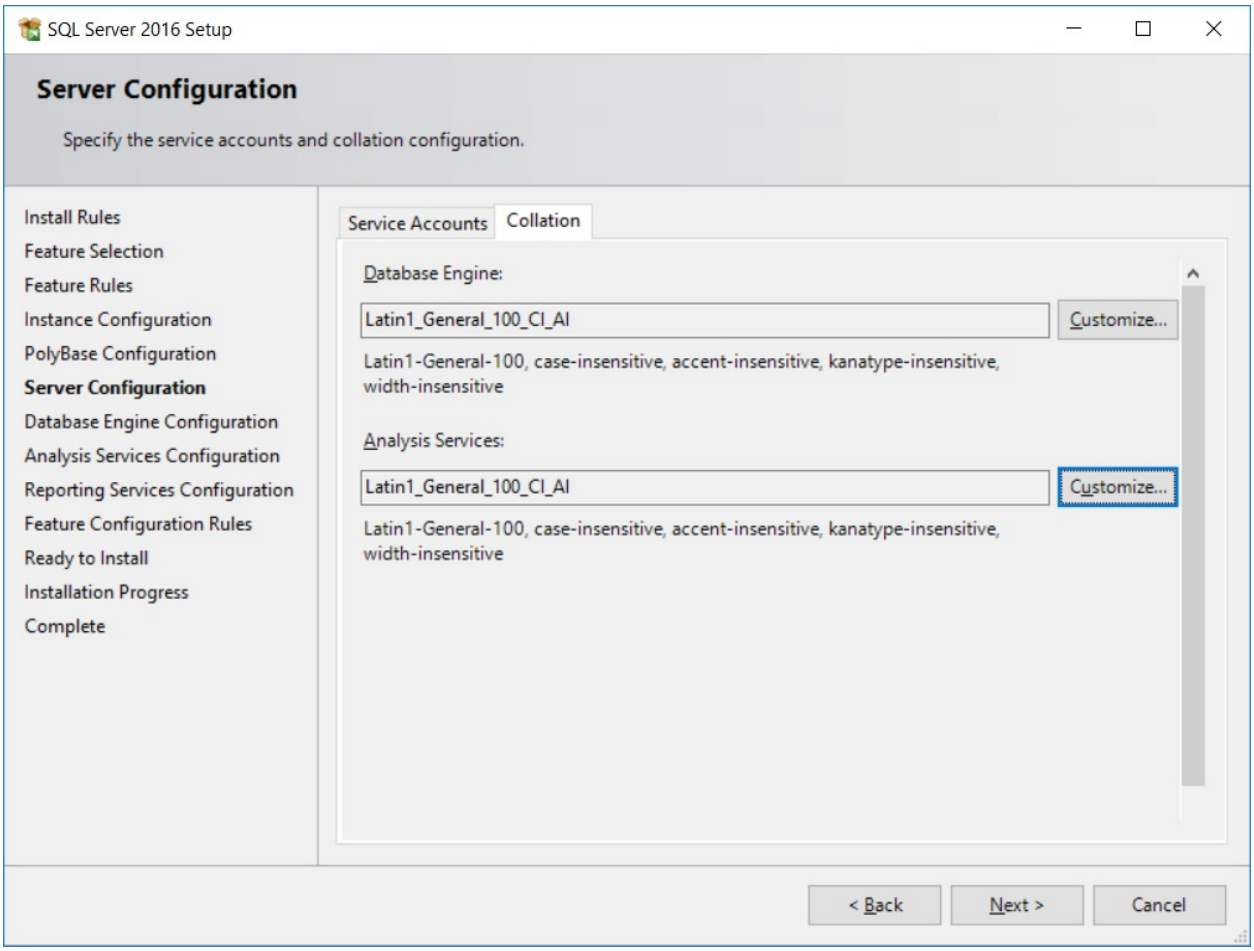
WOW! eBook
www.wowebook.org



20. Again, choose **`Latin1_General_100`** as collation designator and uncheck the **`Accent-sensitive`** checkbox. Click **`OK`** to return to the previous screen.

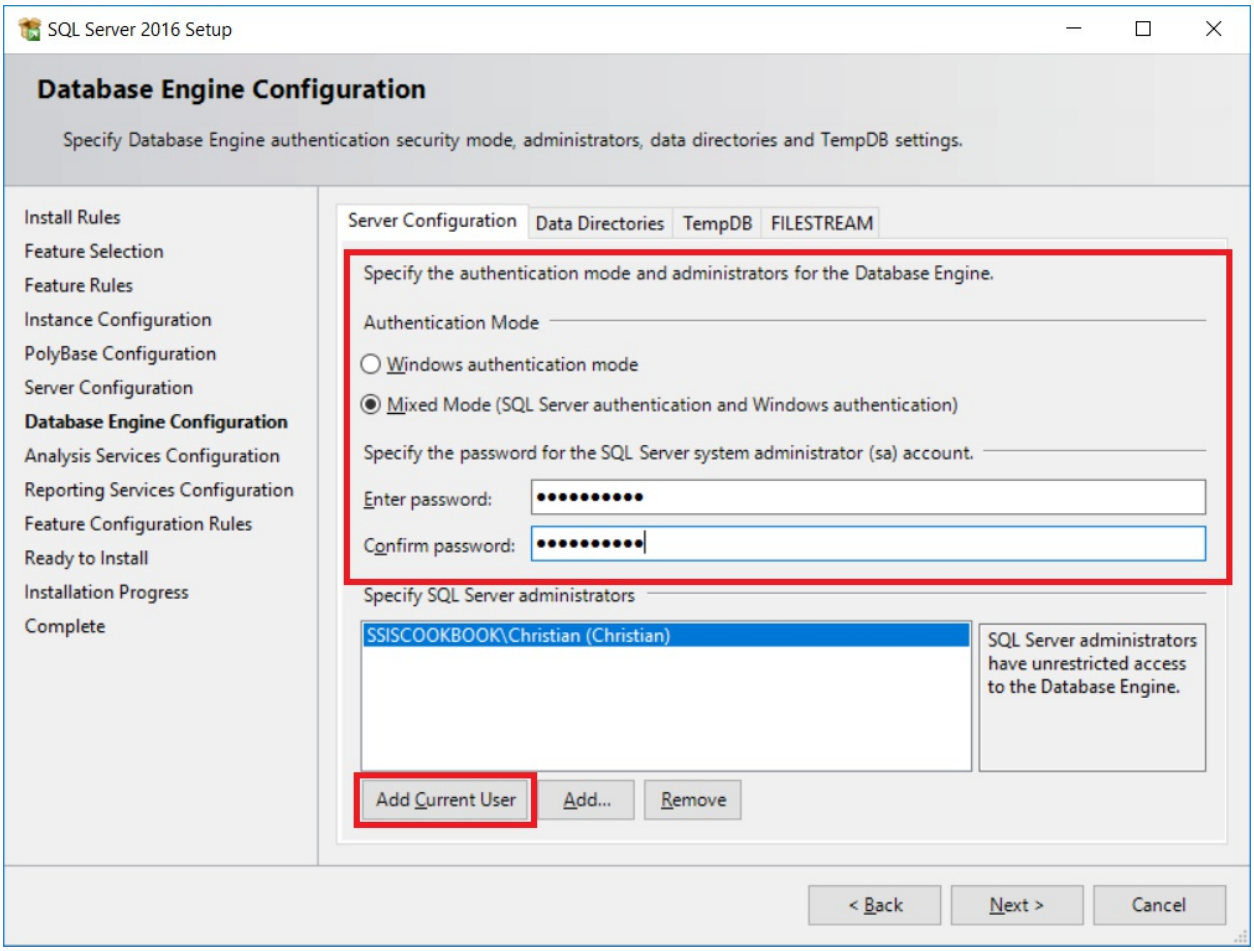
Your screen should look like this:

WOW! eBook
www.wowebook.org



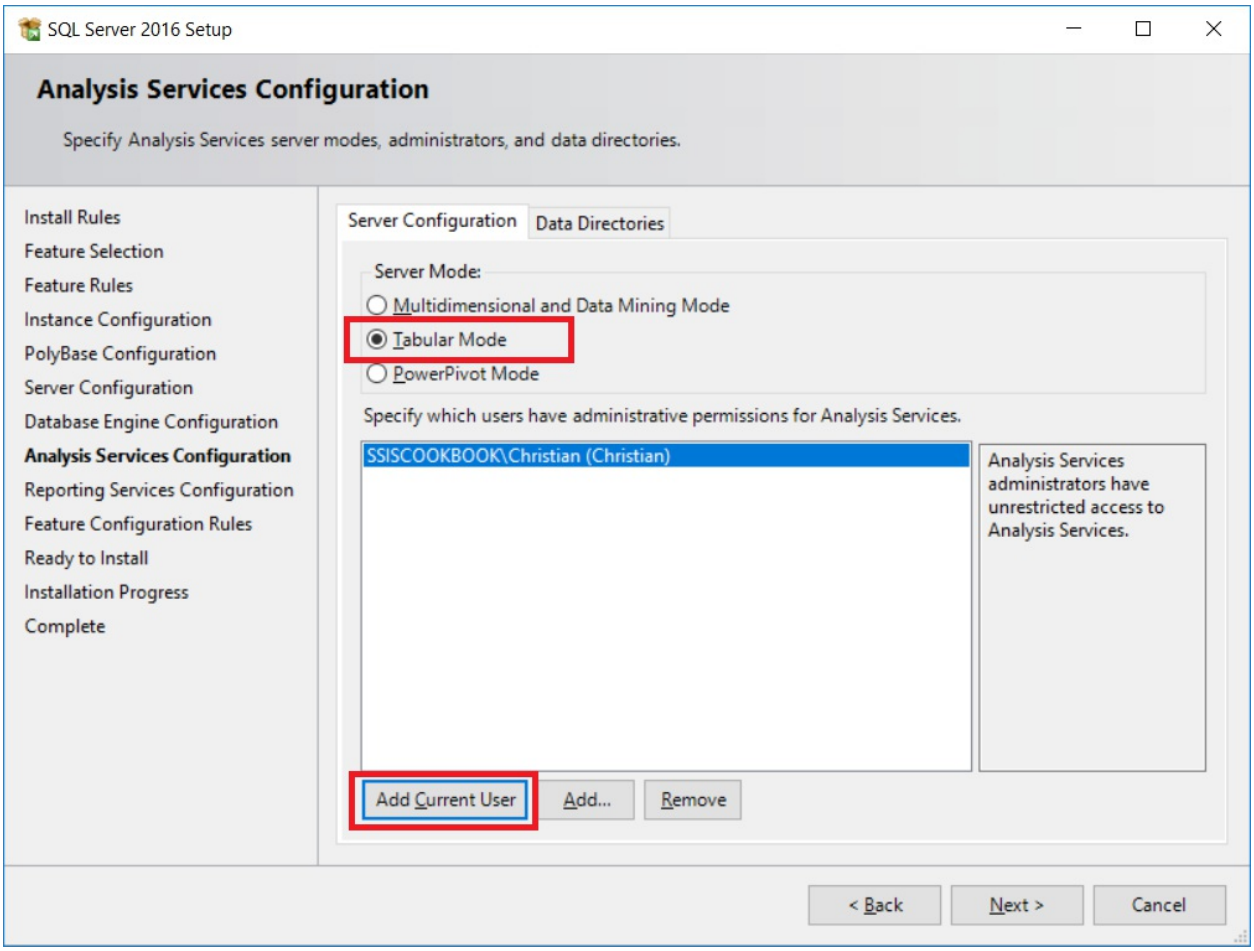
This will direct you to the following screen.

WOW! eBook
www.wowebook.org



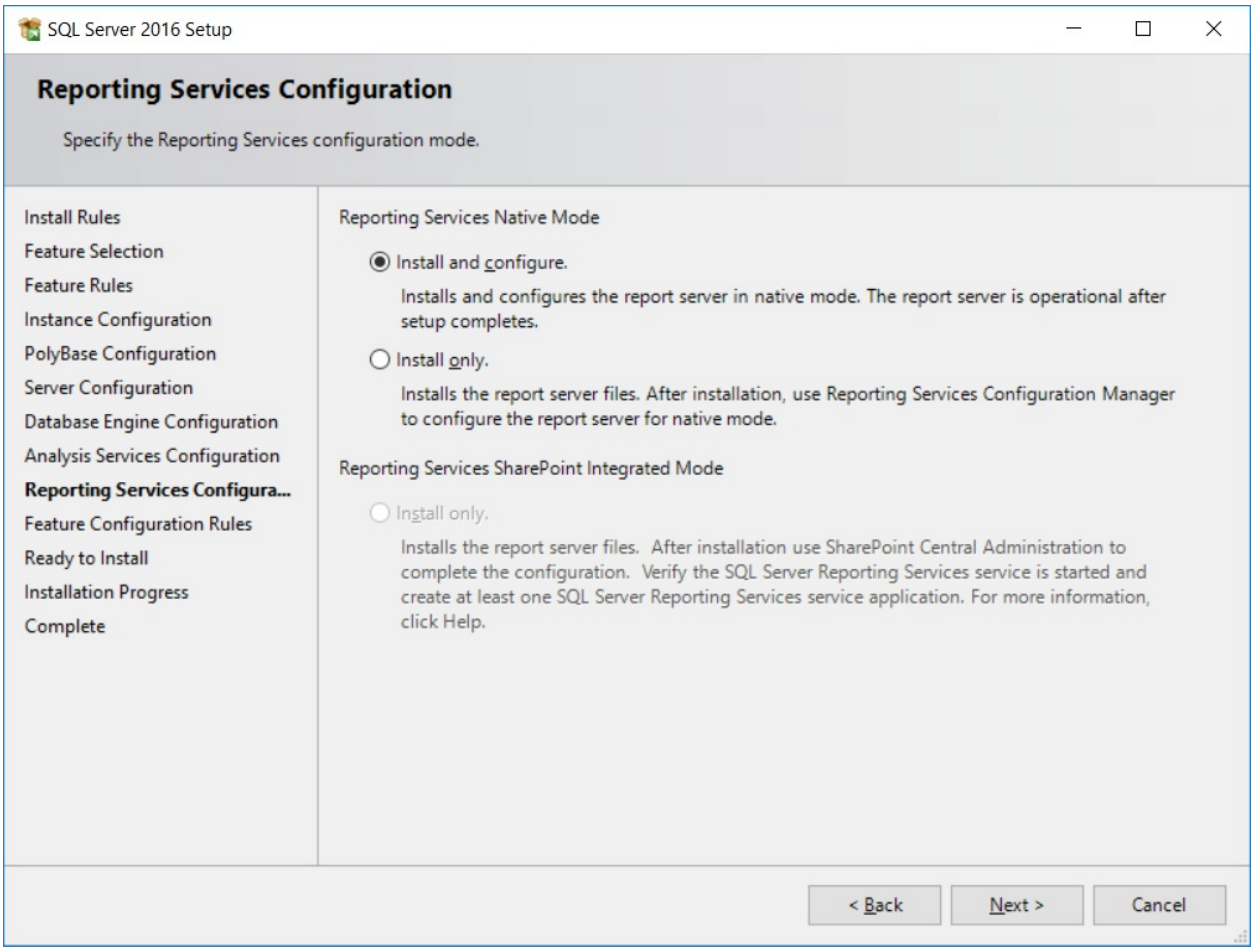
21. For the database engine configuration, we'll use **`Mixed Mode`** that will allow us to use SQL Server logins and Windows logins. Select the **`Mixed Mode`** radio box and enter a password for the SA account. Click on **`Add Current User`** as shown in the screen shot to add your windows account as an administrator of the instance. You'll have every right on it. Click **`Next`**.

WOW! eBook
www.wowebook.org



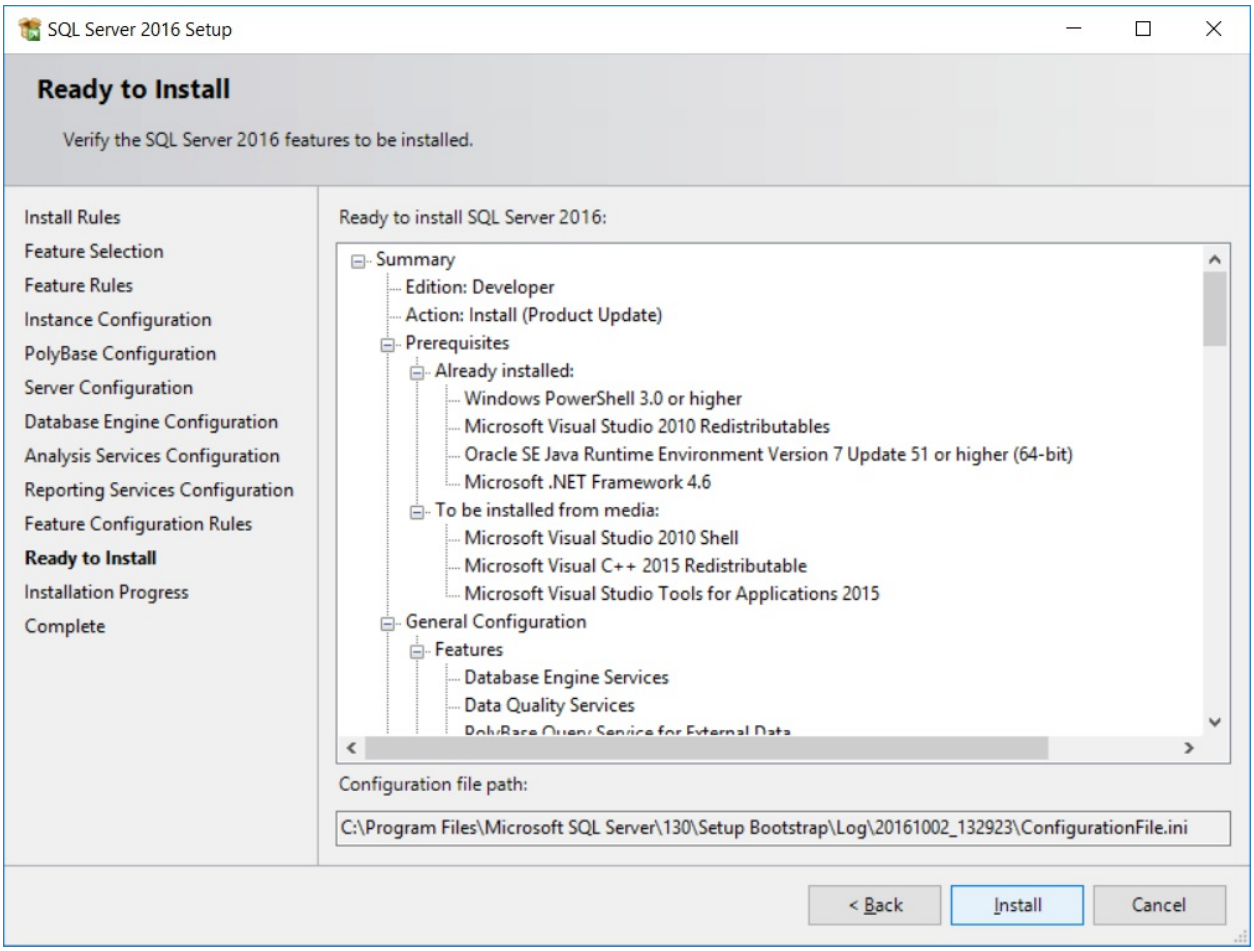
22. Analysis Services configuration. The server mode we'll use is **`Tabular Mode`** and again, click on **`Add Current User`** as shown in the previous screen shot to add your windows account as an administrator of the service. Click **`Next`**.

WOW! eBook
www.wowebook.org



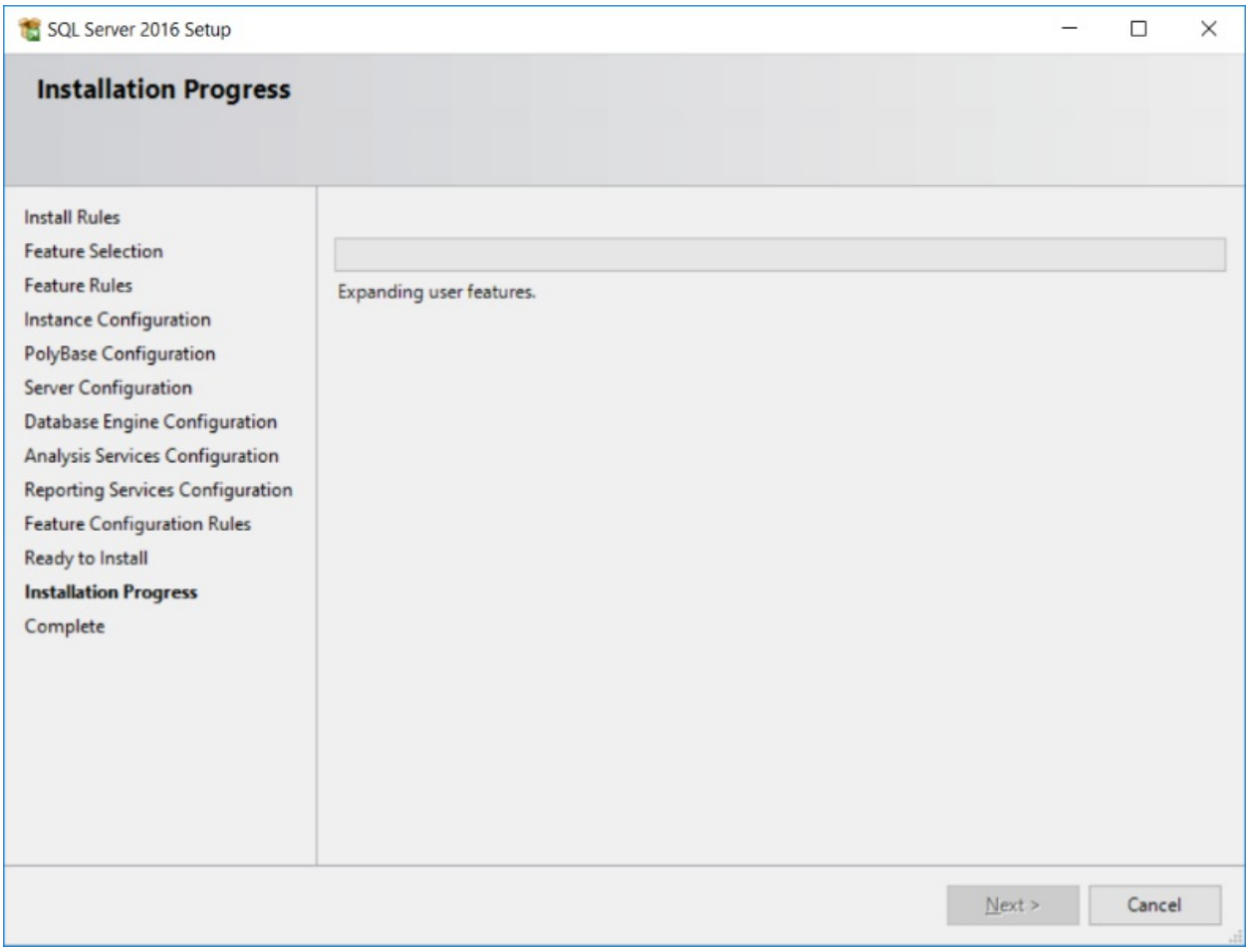
23. Reporting Service configuration, leave the default values and click **Next**.

WOW! eBook
www.wowebook.org



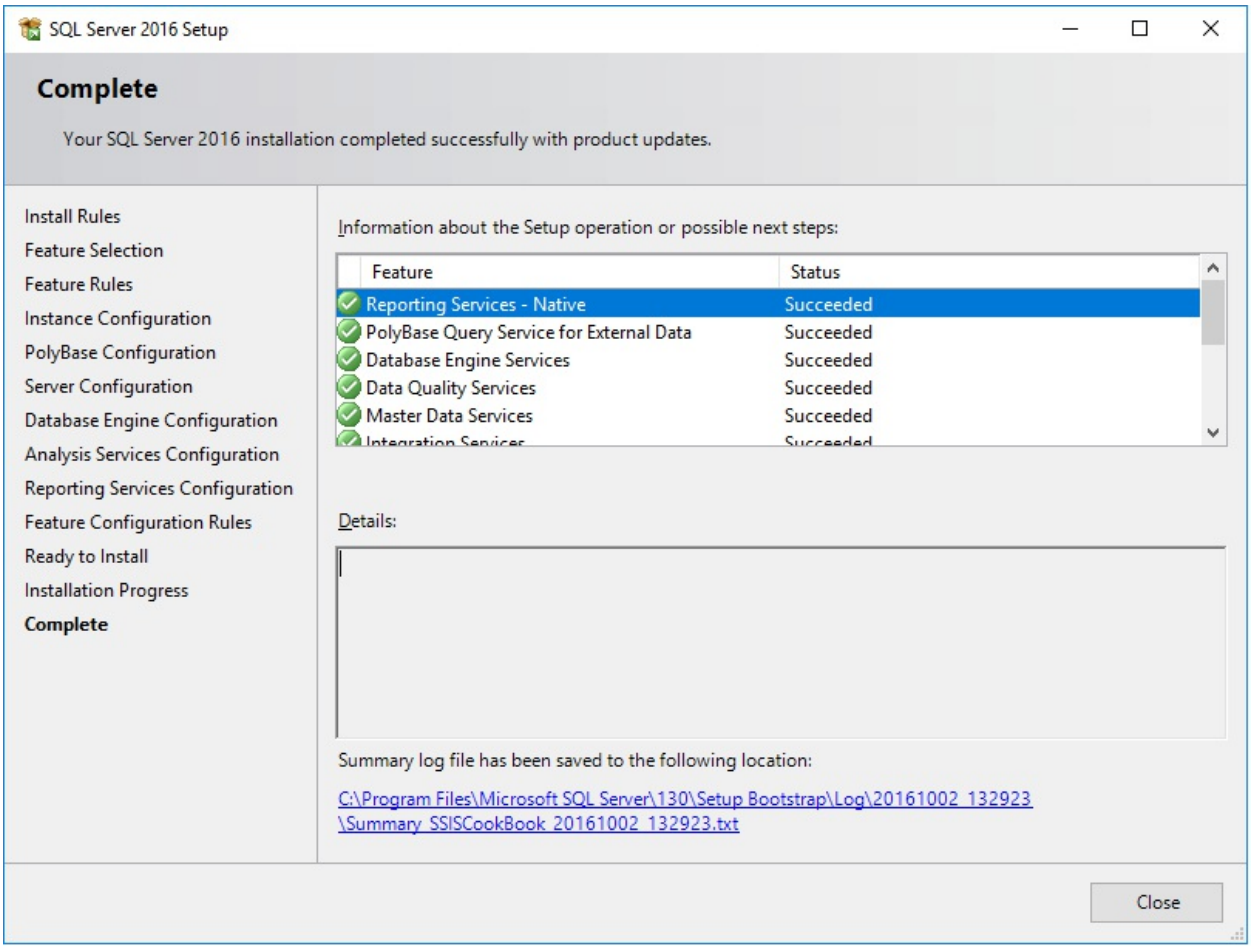
24. We're finally ready to install, click **`Install`** to start the installation process.

WOW! eBook
www.wowebook.org



25. Once the installation is completed, you get the following screen:

WOW! eBook
www.wowebook.org



We're done. We just installed the SQL Server 2016! In the next section, we'll install SQL Server Management Studio.

WOW! eBook
www.wowebook.org

SQL Server Management Studio

SQL server Management studio is a separate download from SQL server. This program will allow us, among other things, to create database objects and query SQL server. Without this tool, we wouldn't be able to manage SQL Server databases easily.

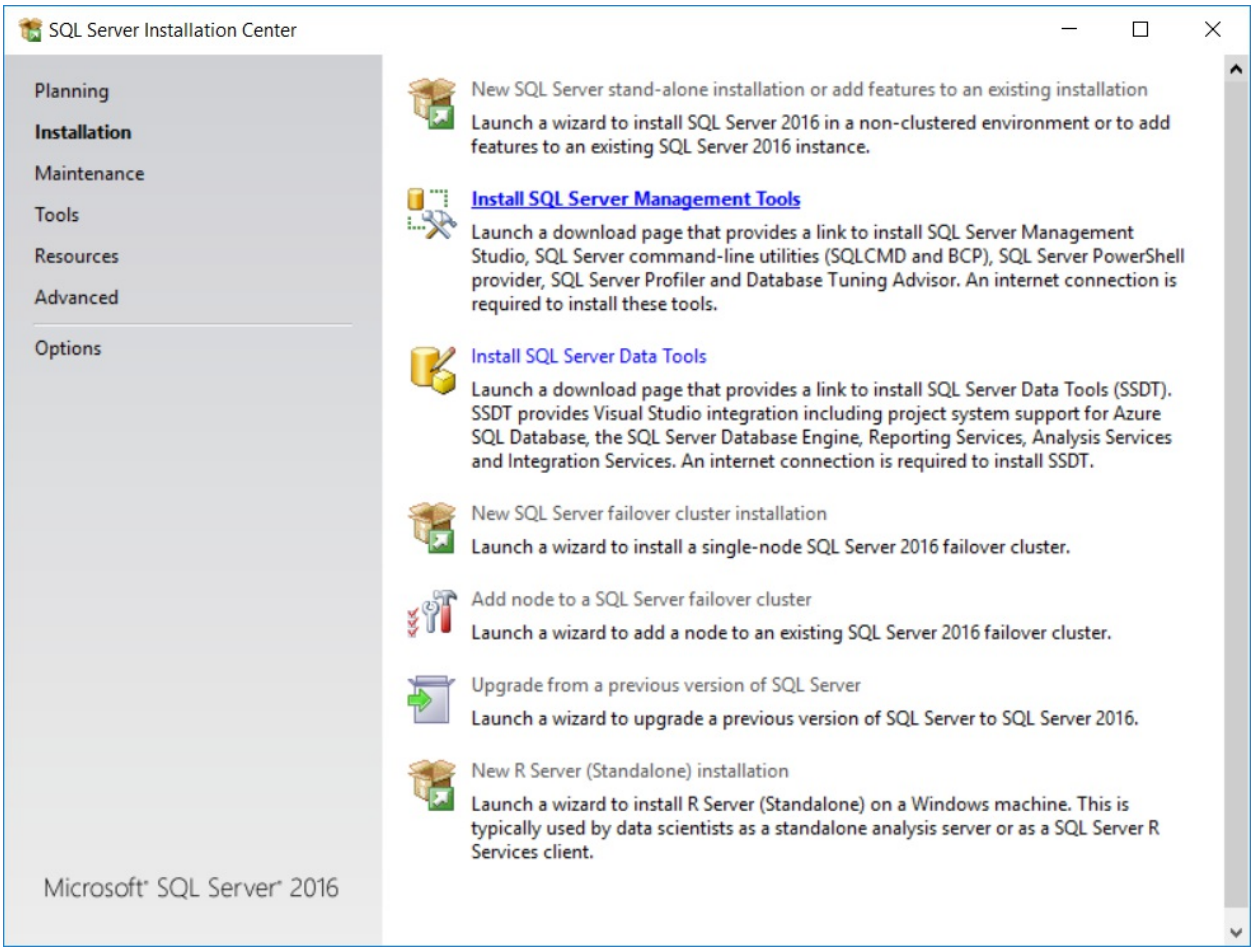
Getting ready

This section assumes that you have installed SQL Server 2016.

How to do it...

1. To download and install it, click on **`Install SQL Server Management Tools`**

WOW! eBook
www.wowebook.org



2. The SSMS download page opens in your browser. Click **`Download SQL Server Management Studio`** (<latest version>) to start the download process. Once downloaded, run the installation as shown in the following screenshot:

WOW! eBook
www.wowebook.org

er Tools > SQL Server Management Tools (including SSMS)

Download SQL Server Management Studio (SSMS)

Updated: September 23, 2016

SQL Server Management Studio (SSMS) is an integrated environment for accessing, configuring, managing, administering, and developing all components of SQL Server. SSMS combines a broad group of graphical tools with a number of rich script editors to provide developers and administrators of all skill levels access to SQL Server. This release features improved compatibility with previous versions of SQL Server, a stand-alone web installer, and toast notifications within SSMS when new releases become available.

Download SQL Server Management Studio (SSMS)

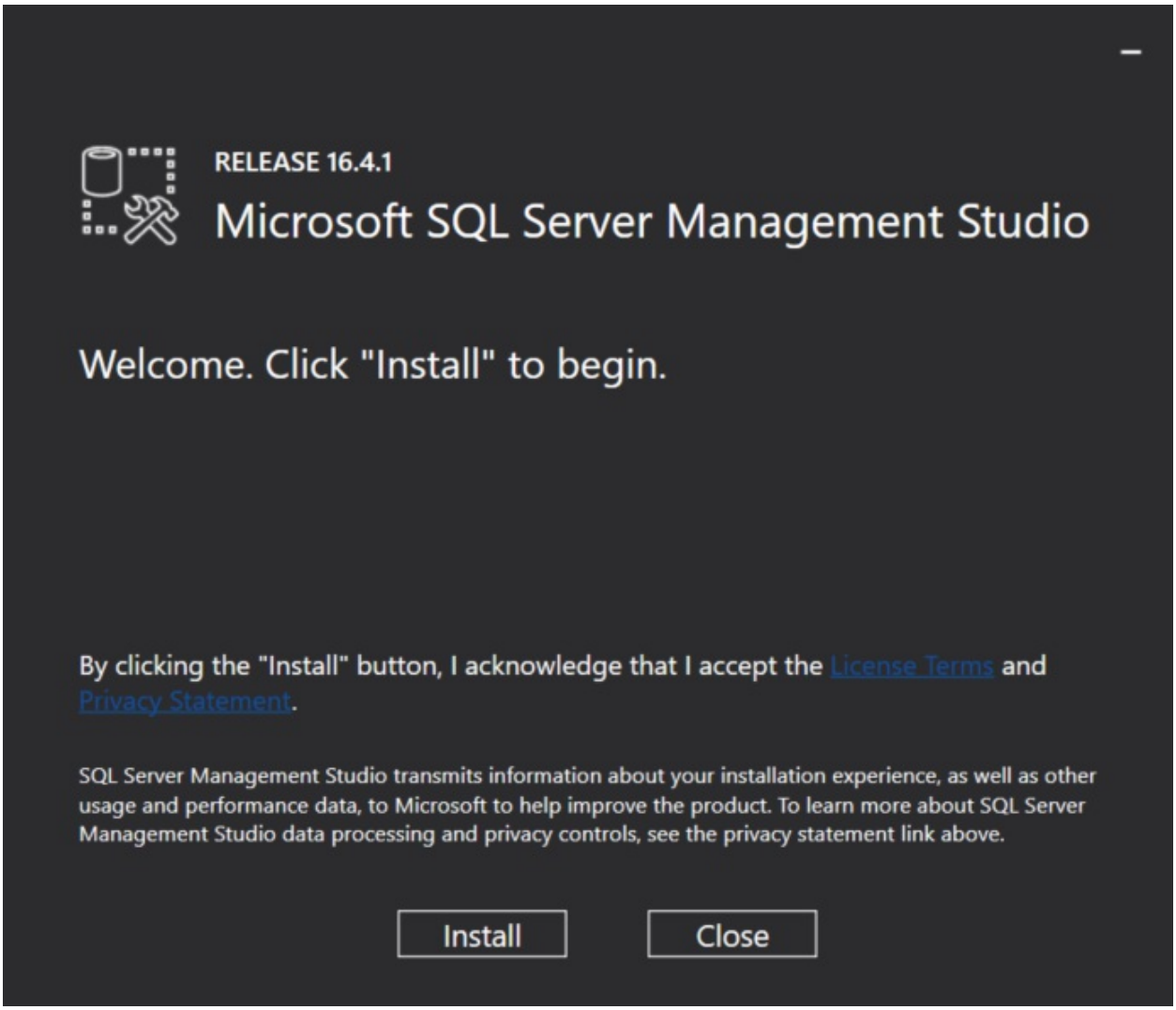
Download SQL Server Management Studio (16.4.1)

Note

- SSMS releases are now branded numerically, not by months.
- This generally available release of SSMS is free and does not require a SQL Server license to install and use.

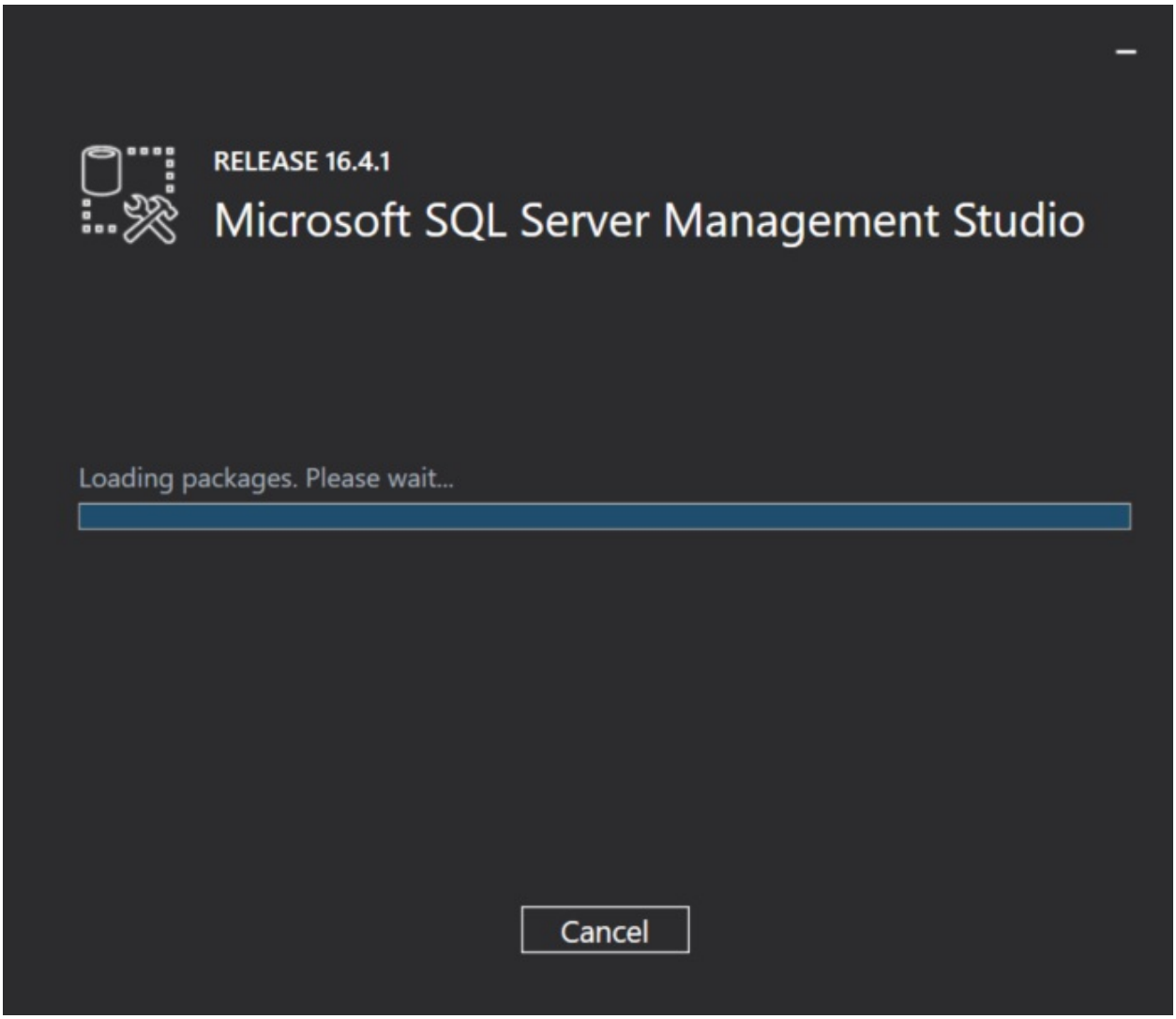
Click **`Install`**.

WOW! eBook
www.wowebook.org



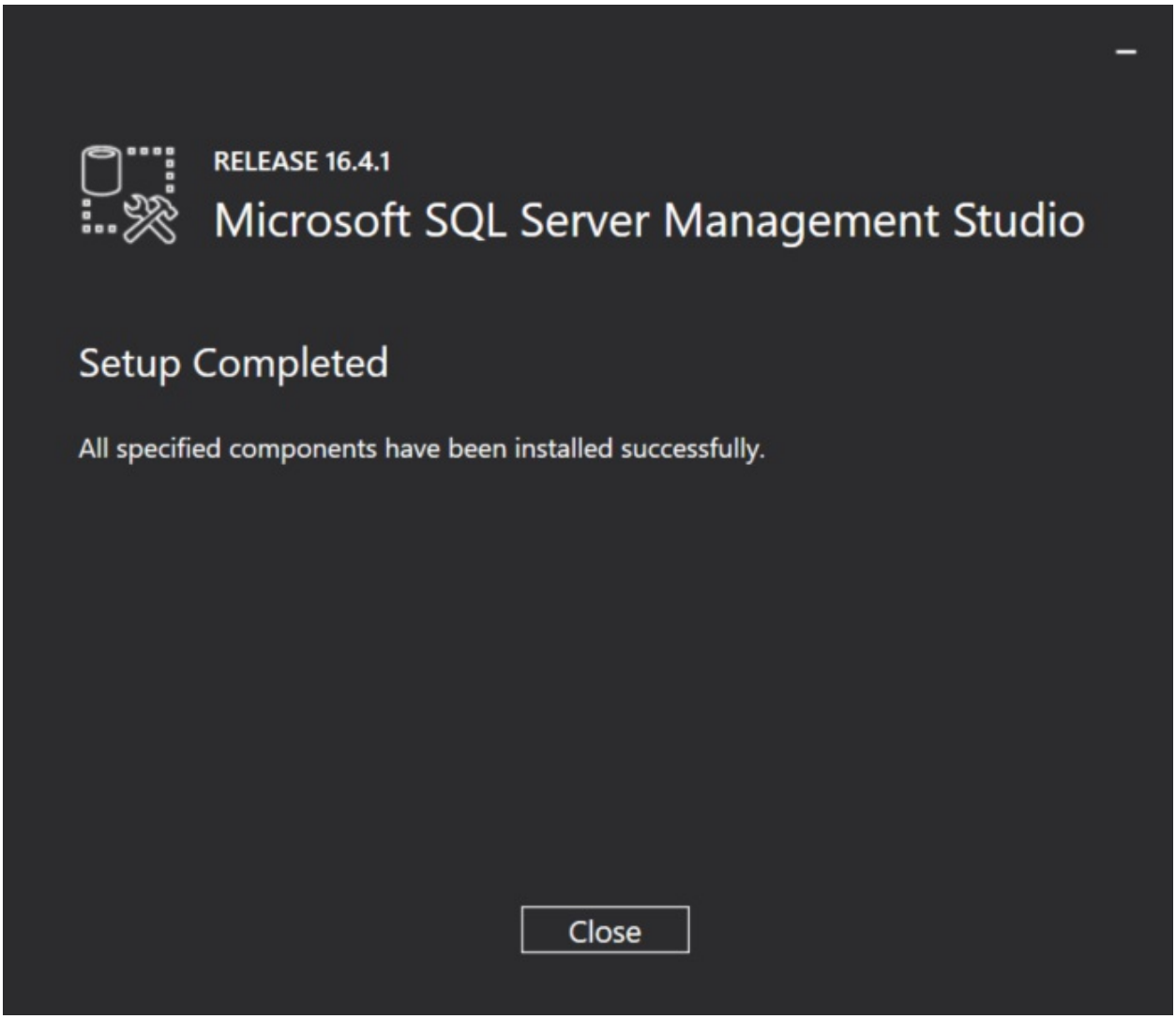
This will direct you to the Microsoft SQL Server Management Studio installation screen shown as follows

WOW! eBook
www.wowebook.org



3. The installation is in progress; it may take several minutes to complete. Click **`Close`** to close the installation wizard.

WOW! eBook
www.wowebook.org



WOW! eBook
www.wowebook.org

SQL Server Data Tools installation

The last part of our SQL Server 2016 setup is to install SQL Server Data Tools (SSDT). This will install a Visual Studio Shell that contains BI templates necessary for the following:

- SQL Server Integration Services
- SQL Server Analysis Services
- SQL Server Reporting Services
- Database objects management

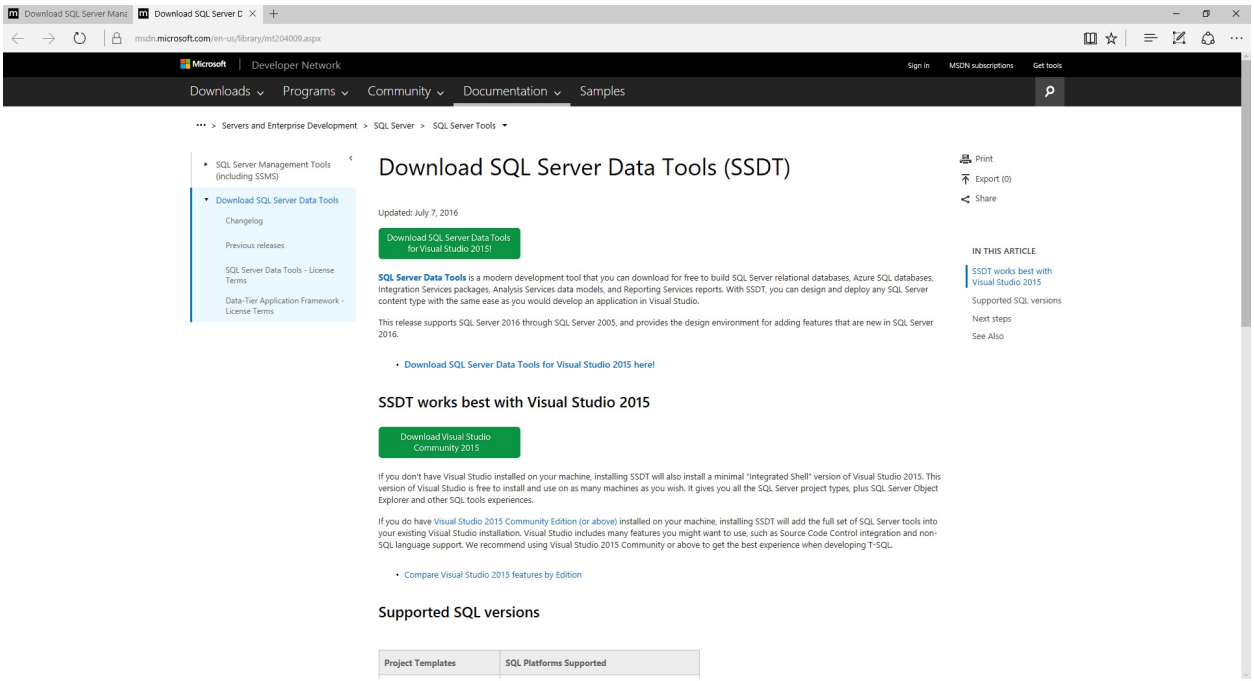
Getting ready

We'll use SSDT throughout this book to create, deploy and maintain our SSIS packages and some databases.

How to do it...

1. From the SQL Server 2016 Setup utility, click on SQL Server Data Tools (SSDT) This will open a the **`Download SQL Server Data Tools (SSDT)`** download page in your browser as shown in the following screenshot:

WOW! eBook
www.wowebook.org

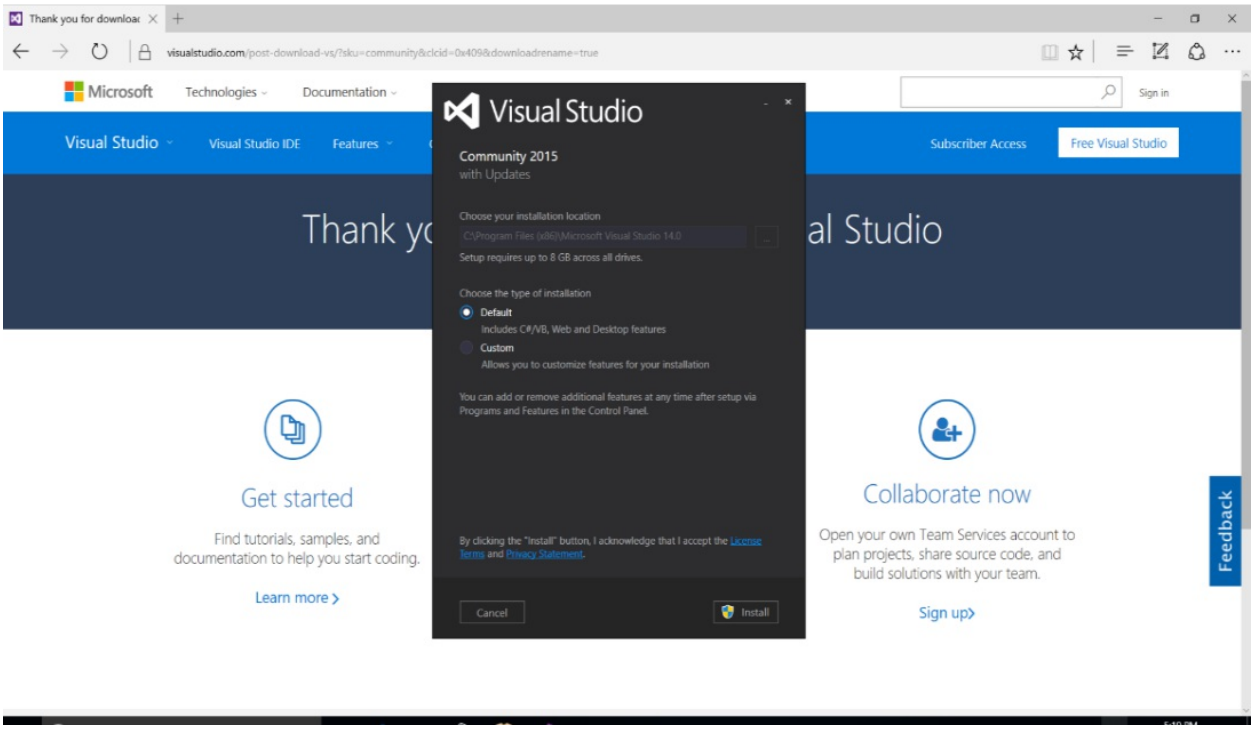


There are two choices:

- Install SSDT only: This is the simplest scenario. It only installs SSDT and a development shell.
- Install Visual Studio and SSDT: You do it if you plan to use source control inside Visual Studio or when you want to make other type of development (.Net, Python, etc.) than SSIS/SSAS/SSRS development. Since we'll talk about custom components in this book, we'll install Visual Studio Community Edition. This version is free for individuals.

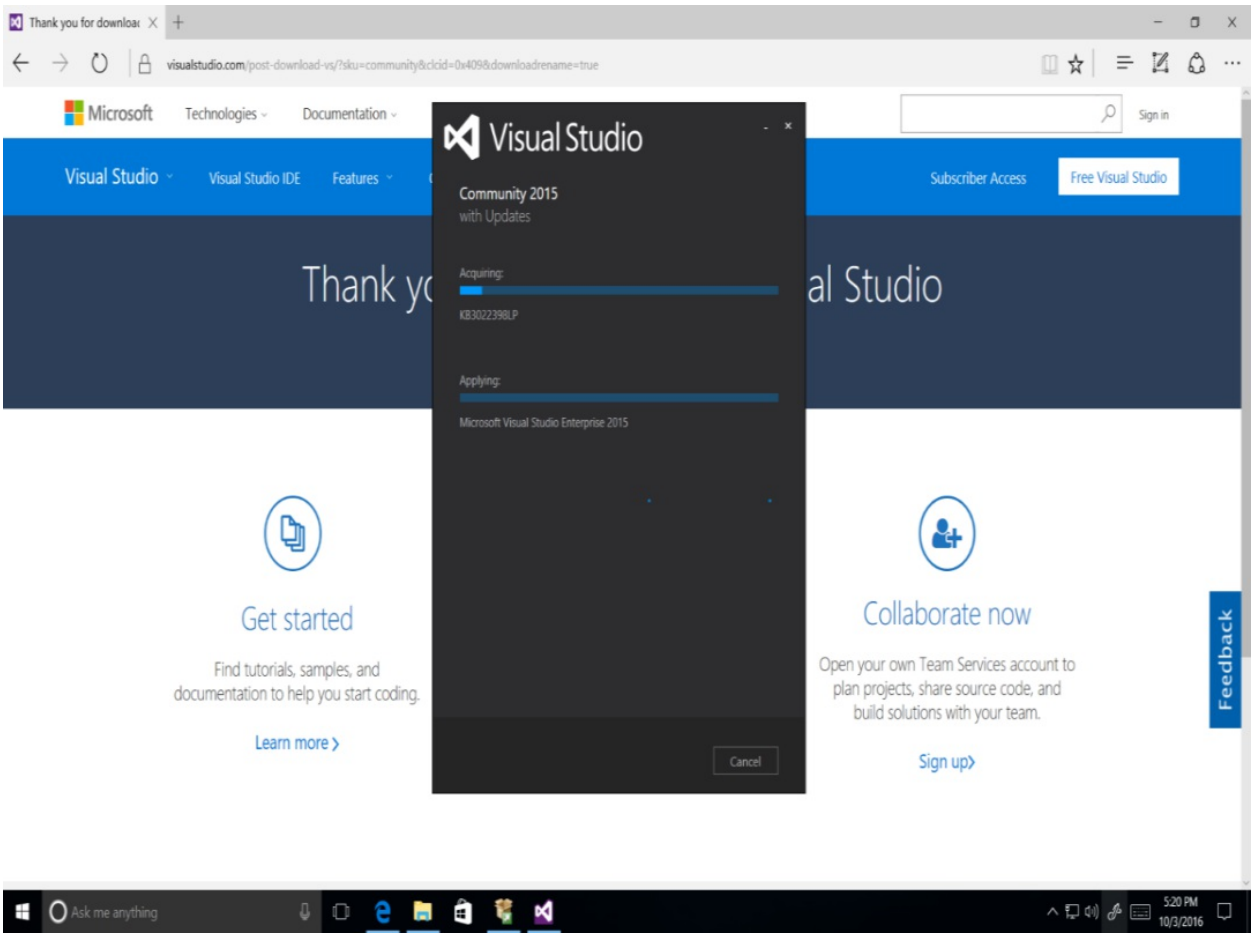
2. Click on **`Download Visual Studio Community 2015`** link to download the Visual Studio installer.
3. Once downloaded, click on **`Run`** to launch the Visual Studio installer.

WOW! eBook
www.wowebook.org



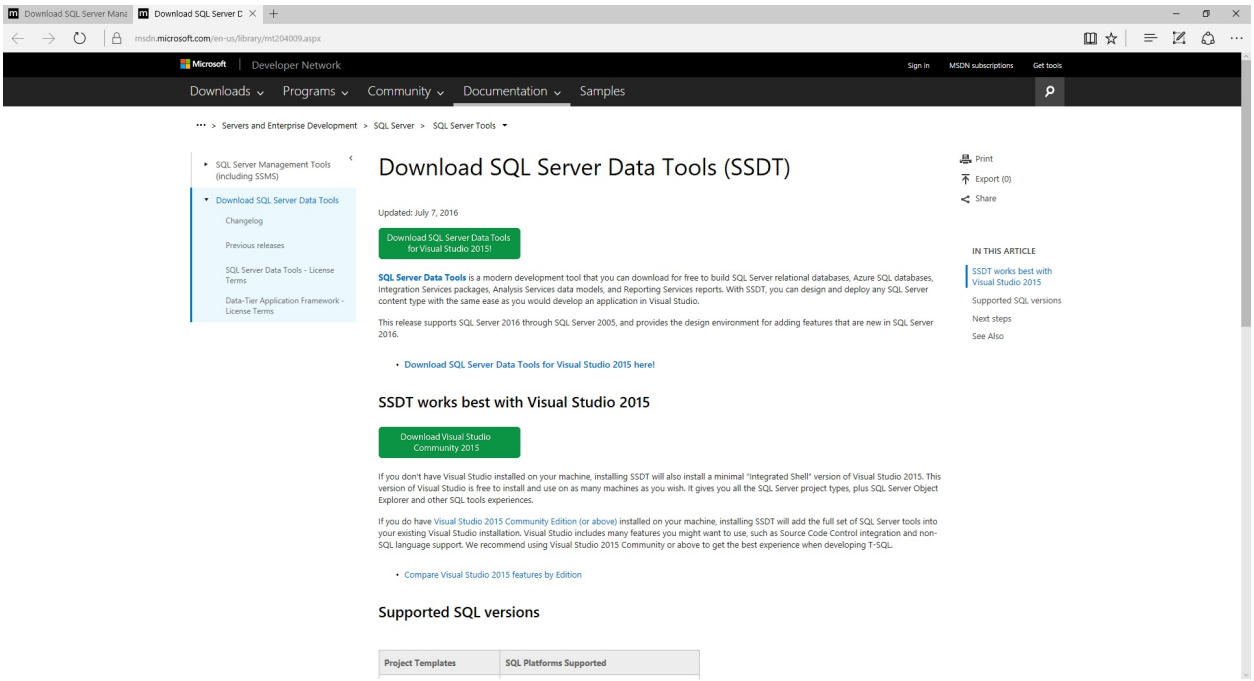
4. Accept the default installation type and click **`Install`** to start the installation process.

WOW! eBook
www.wowebook.org

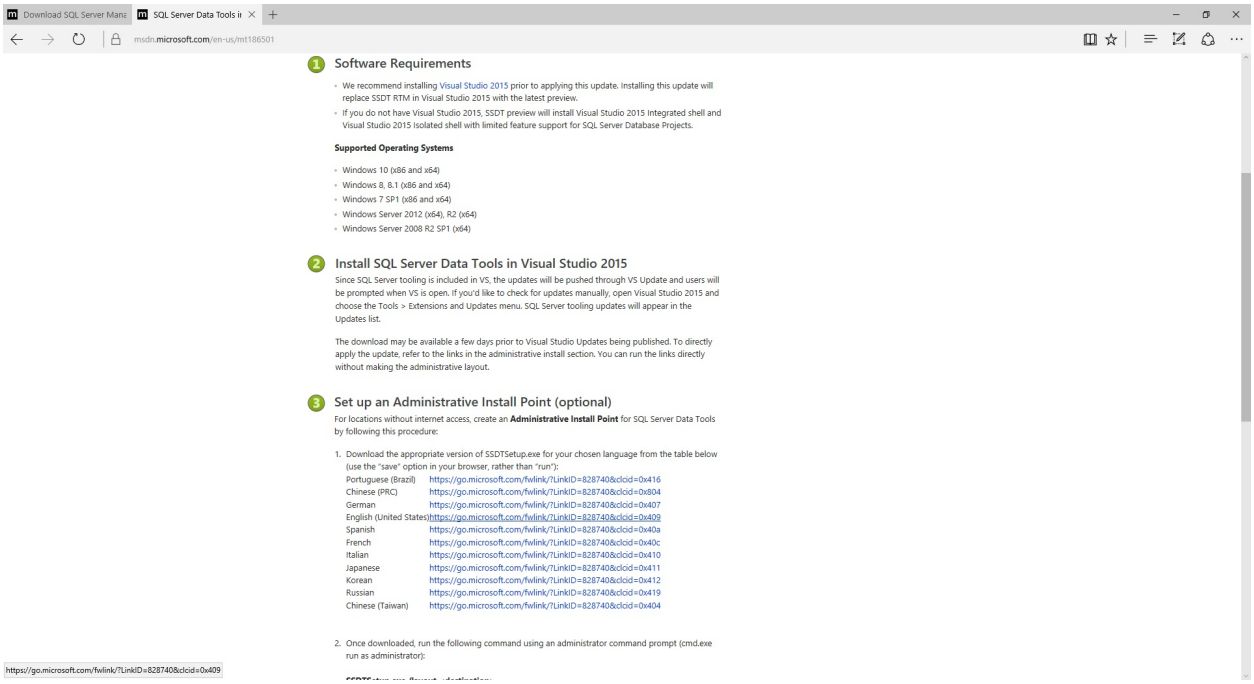


5. Once the installation is completed, you will see the following screen.
6. Since we haven't installed SSDT, don't launch Visual Studio yet, close this window. We'll install SQL Server Data Tools first. Return to the browser window and click **`Download SQL Server Data Tools for Visual Studio 2015`**.

WOW! eBook
www.wowebook.org



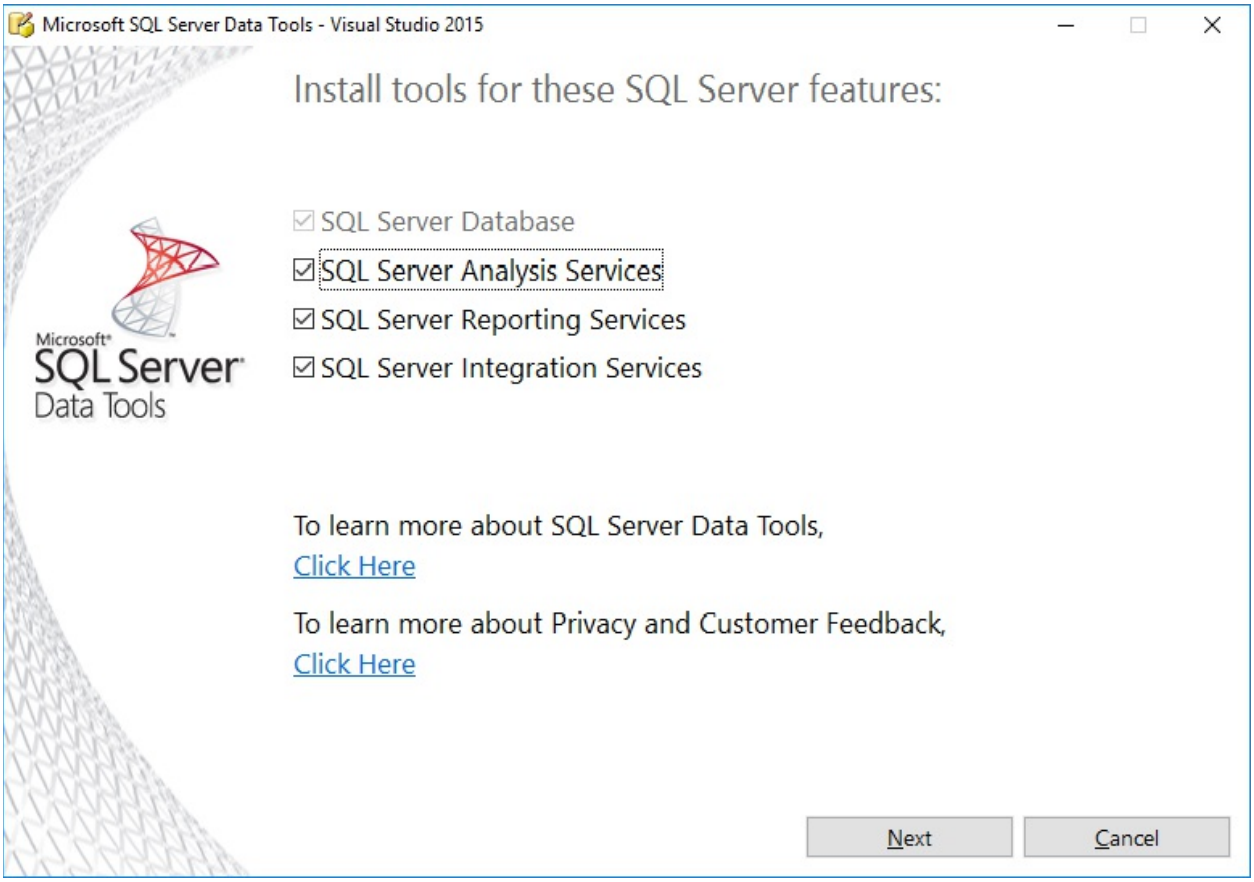
7. This will direct you to the SSDT download screen shown as follows:



8. From the browser page that opens, choose **`English (United States)`**. SSDT setup executable file download starts. Since it's a small file, it takes only a few seconds to download. Once the download completes, click **`run`** or double-click on the newly downloaded file to start SSDT

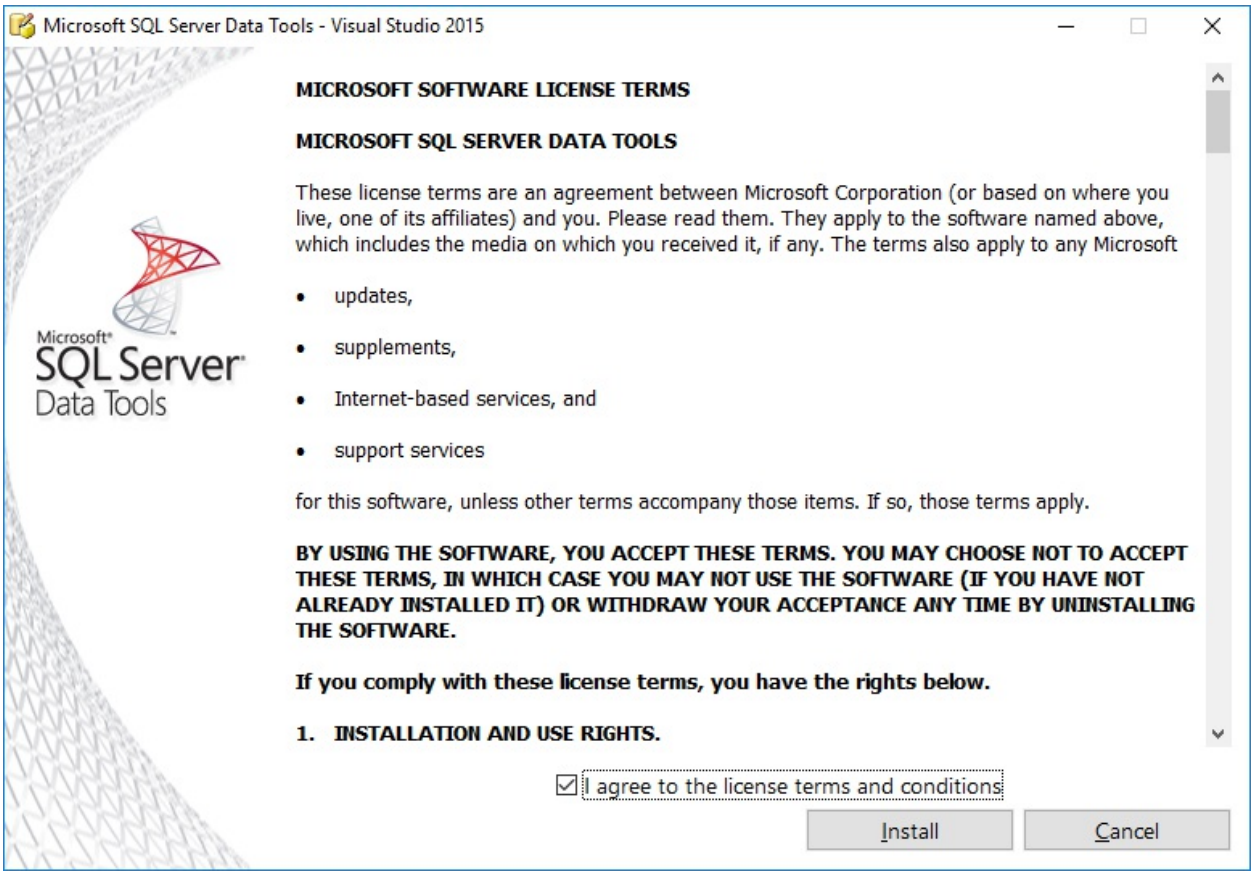
WOW! eBook
www.wowebook.org

installation.



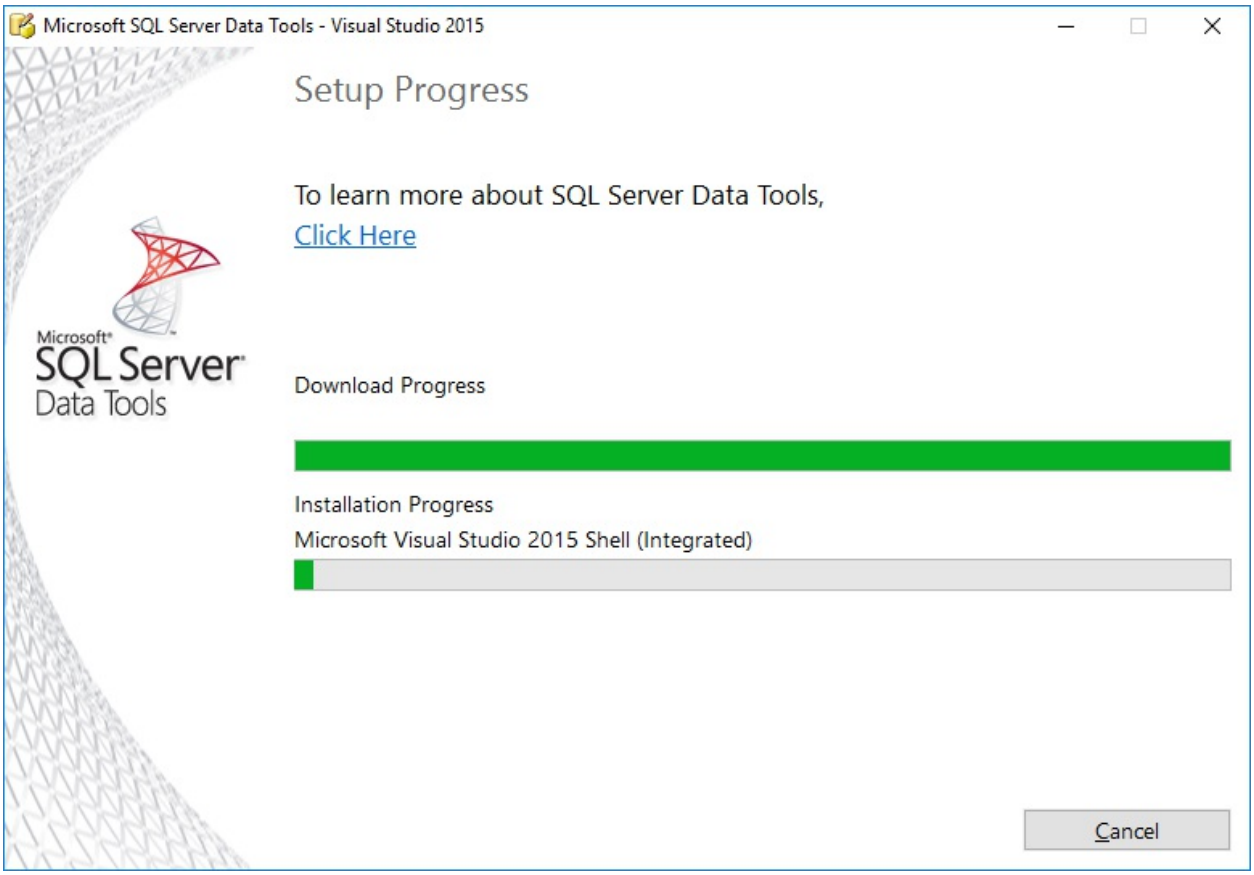
9. Accept the defaults and click **`Next`** to proceed to the next step.

WOW! eBook
www.wowebook.org



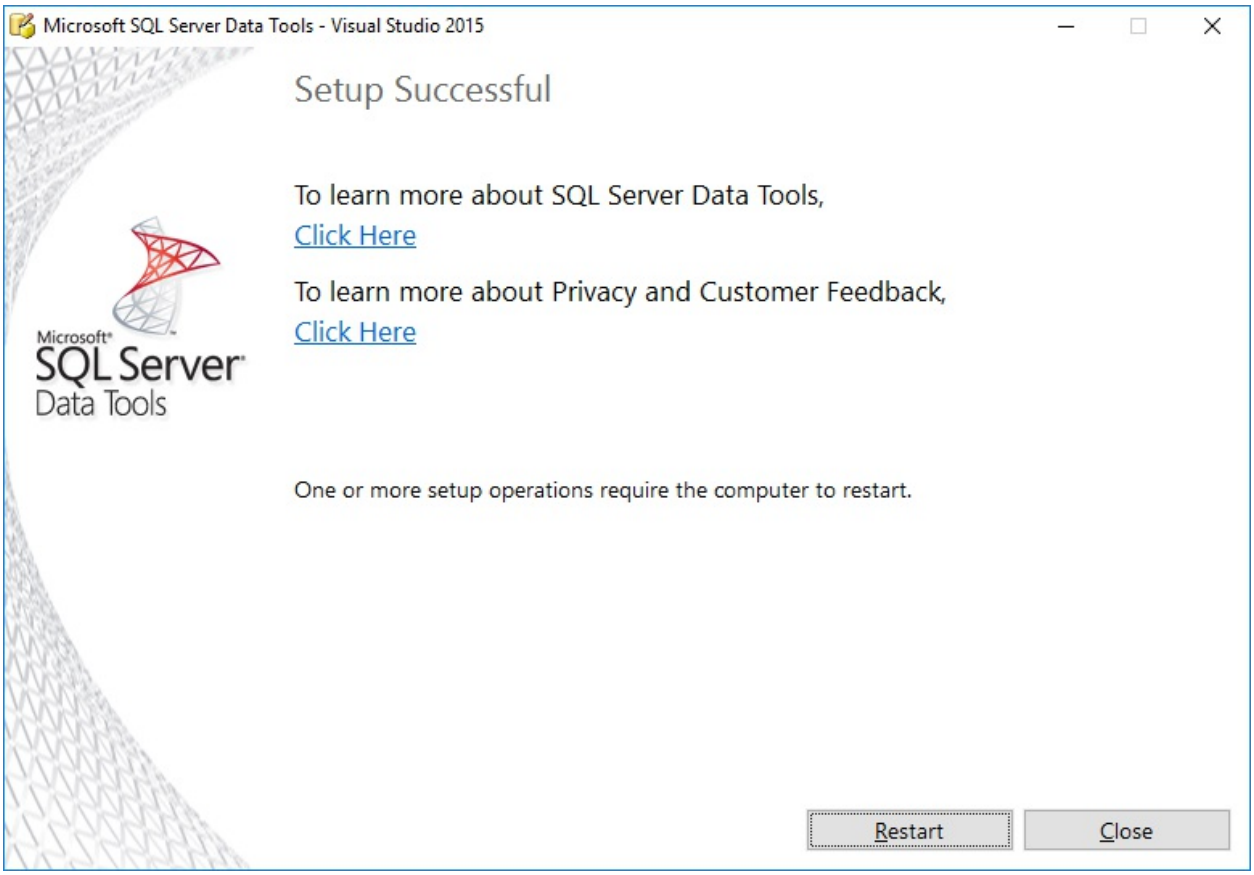
10. Accept the license agreement by checking the **`I agree to the license terms and conditions`** and click **`Install`**.

WOW! eBook
www.wowebook.org



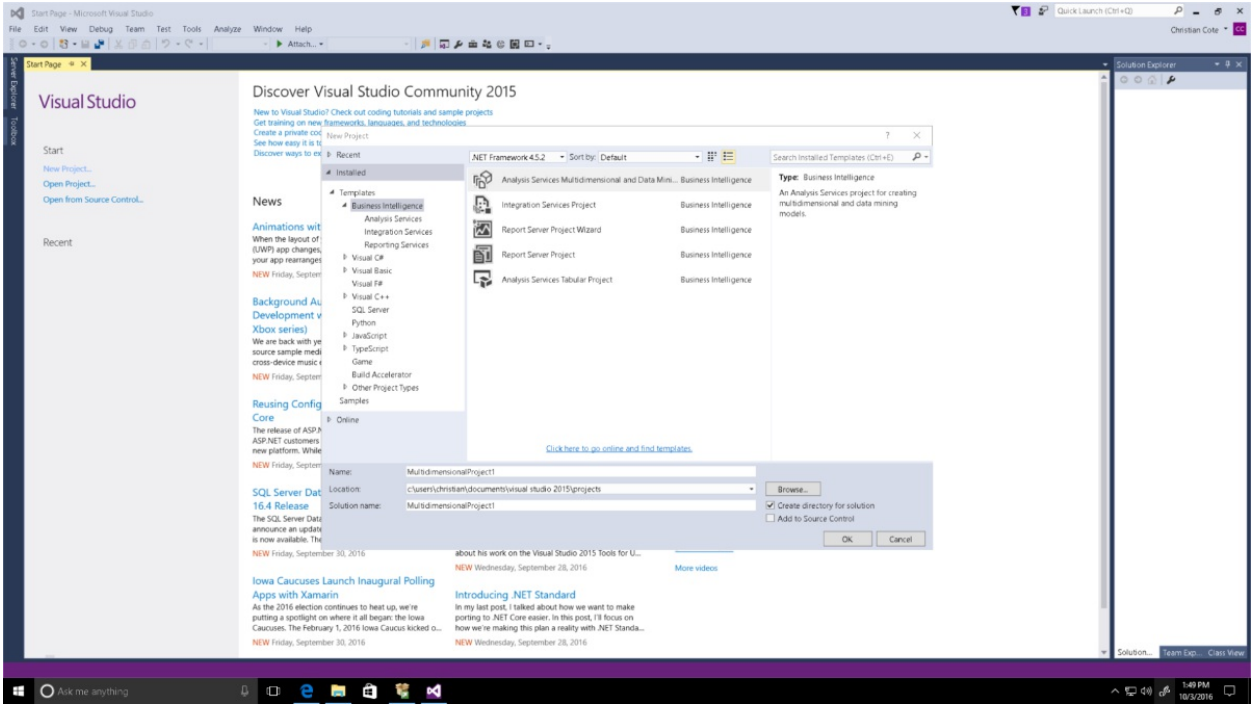
11. The SSDT installer will download the necessary files and proceed to the installation.

WOW! eBook
www.wowebook.org



12. Once the installation completes, you might have to restart your computer. If that's the case, restart it and once that's done, look for SQL Server Data Tools in your start menu and launch it.

WOW! eBook
www.wowebook.org



13. From the **`File`** menu, select **`New Project`**. Once the New Project window appears, you will see **`Business Intelligence`** in the project templates.
14. Close SSDT, we're done with it for now.

WOW! eBook
www.wowebook.org

Test SQL Server connectivity

SQL Server Management Studio (SSMS) has been installed in this chapter in the recipe *SQL Server Management Studio*. We'll now test if we're able to connect to our local instance..

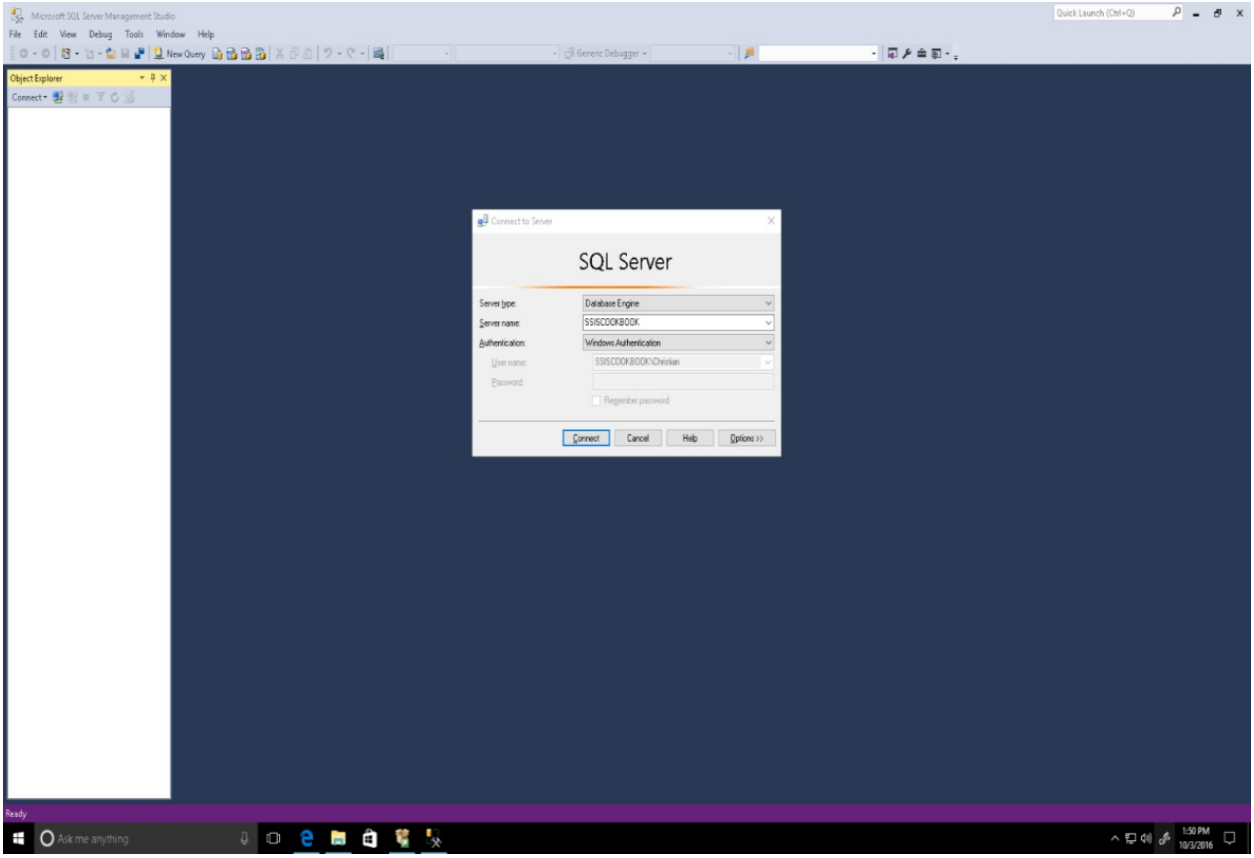
Getting ready

This recipe assumes that you have successfully installed SQL Server 2016 Developer edition as well as SQL Server Management Studio (SSMS).

How to do it...

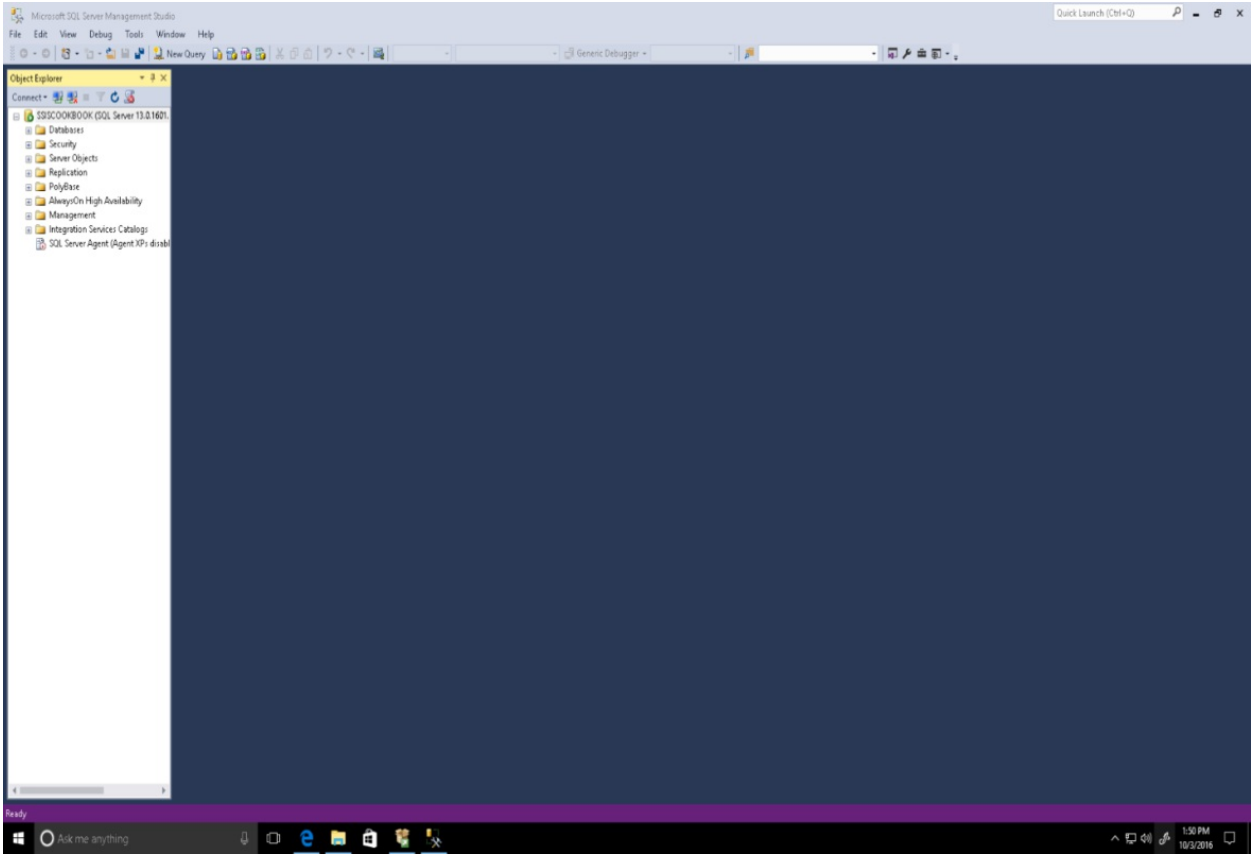
1. Look for SQL Server Management Studio in your start menu and launch it
2. Once the application opens, you should see your PC name in the **`Server Name`** field. Click on **`Connect`**. SSMS will now connect to your local SQL Server instance.

WOW! eBook
www.wowebook.org



3. You are now able to connect to your local SQL Server Instance, we're now ready to work!

WOW! eBook
www.wowebook.org



WOW! eBook
www.wowebook.org

Chapter 2. What Is New in SSIS 2016

This chapter will cover the following recipes:

- Create SSIS Catalog
- Custom logging
- Incremental package deployment
- Azure connectors installation
- Multiple version support
- Error column name
- Control Flow templates

WOW! eBook
www.wowebook.org

Introduction

The 2016 release of SQL Server Integration Services is a major revision of the software. But, instead of being a complete re-write of the product, it's more an evolution of the product. Here is the SSIS timeline since its beginning in SQL Server 7.0 (1998).

In the early years of SQL Server, Microsoft introduced a tool to help developers and database administrators (DBA) to interact with the data: DTS - Data Transformation Services. The tool was very primitive compared to SSIS and it was mostly relying on ActiveX and TSQL to transform the data. SSIS V1.0 (2005) appears in 2005. The tool was a game changer in the ETL world at the time. It was a professional and (pretty much) reliable tool for 2005. 2008/2008R2 versions were much the same as 2005 in a sense that they didn't add much functionalities but they made the tool more scalable.

In 2012, Microsoft enhanced SSIS in many ways. They rewrote the package XML to ease source control integration and make package code easier to read. They also greatly enhanced the way packages are deployed by using a SSIS cataog in SQL Server. Having the catalog in SQL Server gives us execution reports and many views that allows gives us access to metadata or metaprocess information's in our projects.

Version 2014 didn't have anything for SSIS. Version 2016 bring other set of features as you will see in the remaining of this chapter. We now also have the possibility to integrate with Big Data as we'll talk in some later sections of the book.

WOW! eBook
www.wowebook.org

Creating SSIS Catalog

This section will walk you through the various steps to create a SSIS catalog in SSMS. As mentioned before, the SSIS catalog contains information's about the package components and their execution. As we will see later in the book, SSIS projects are deployed into this catalog. It can be easily queried for custom reports as well allows us to create SSIS executions using TSQL. The latter is very useful for on-demand executions of SSIS packages.

SSIS versions prior 2012 did not have these capabilities since the catalog appeared with 2012. It is still possible to bypass the deployment to SSIS catalog by using a special mode: the "Package deployment model". This is mostly used for backward compatibility with previous SSIS frameworks.

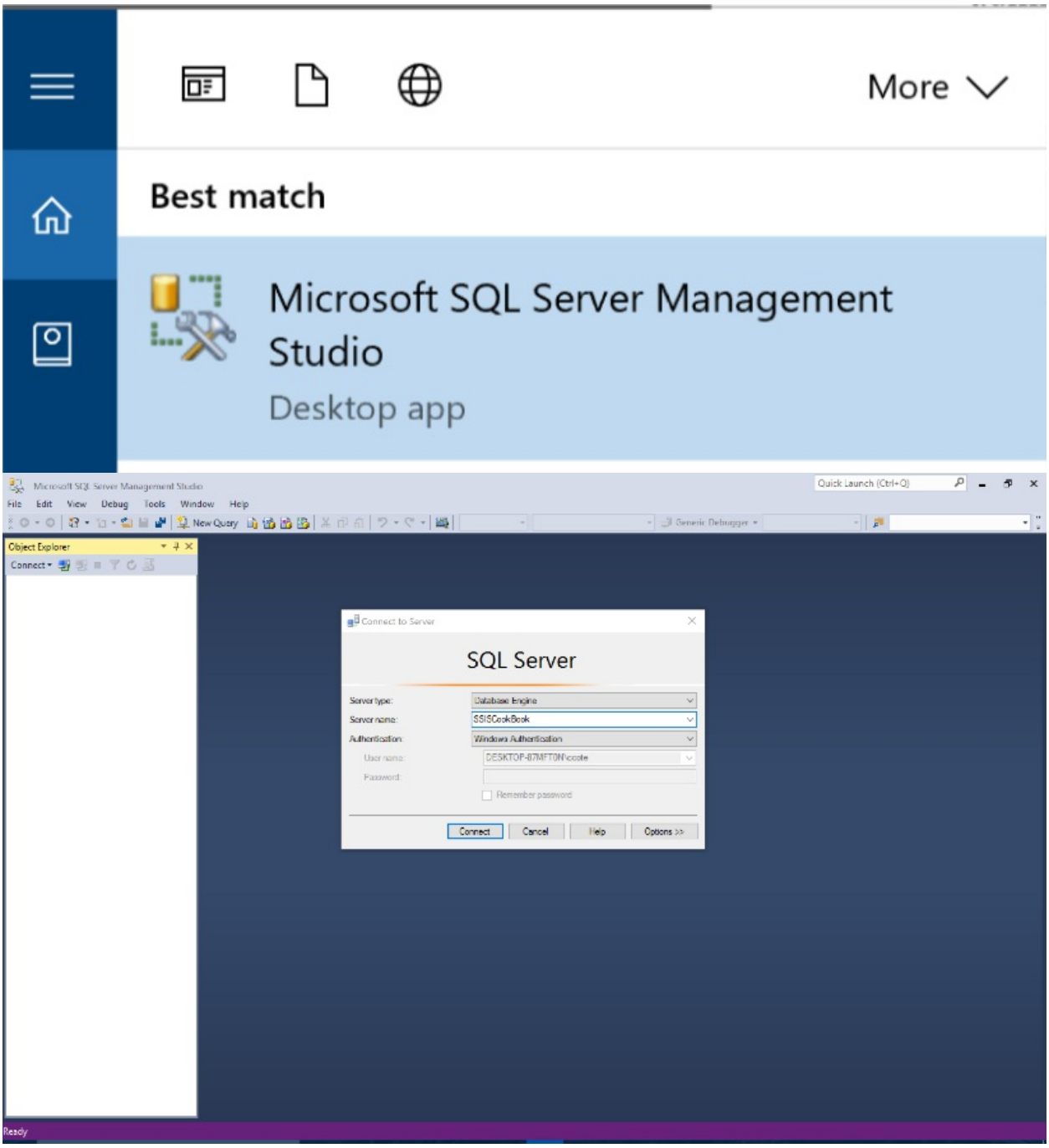
Getting ready

This section assumes you have already installed SQL Server Management Studio.

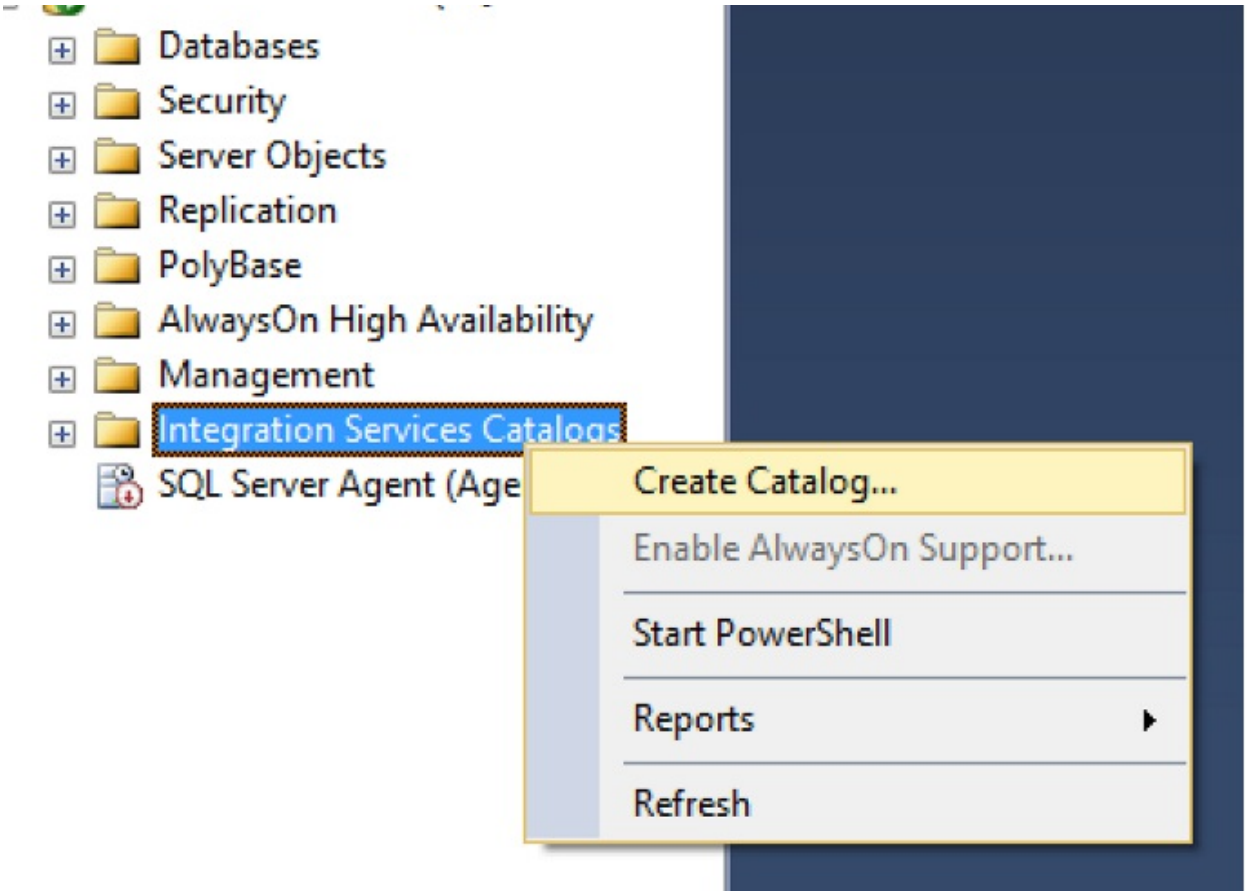
How to do it...

1. We'll first create the SSIS catalog in SQL Server Management Studio (SSMS). Open SSMS and connect to your local instance.

WOW! eBook
www.wowebook.org



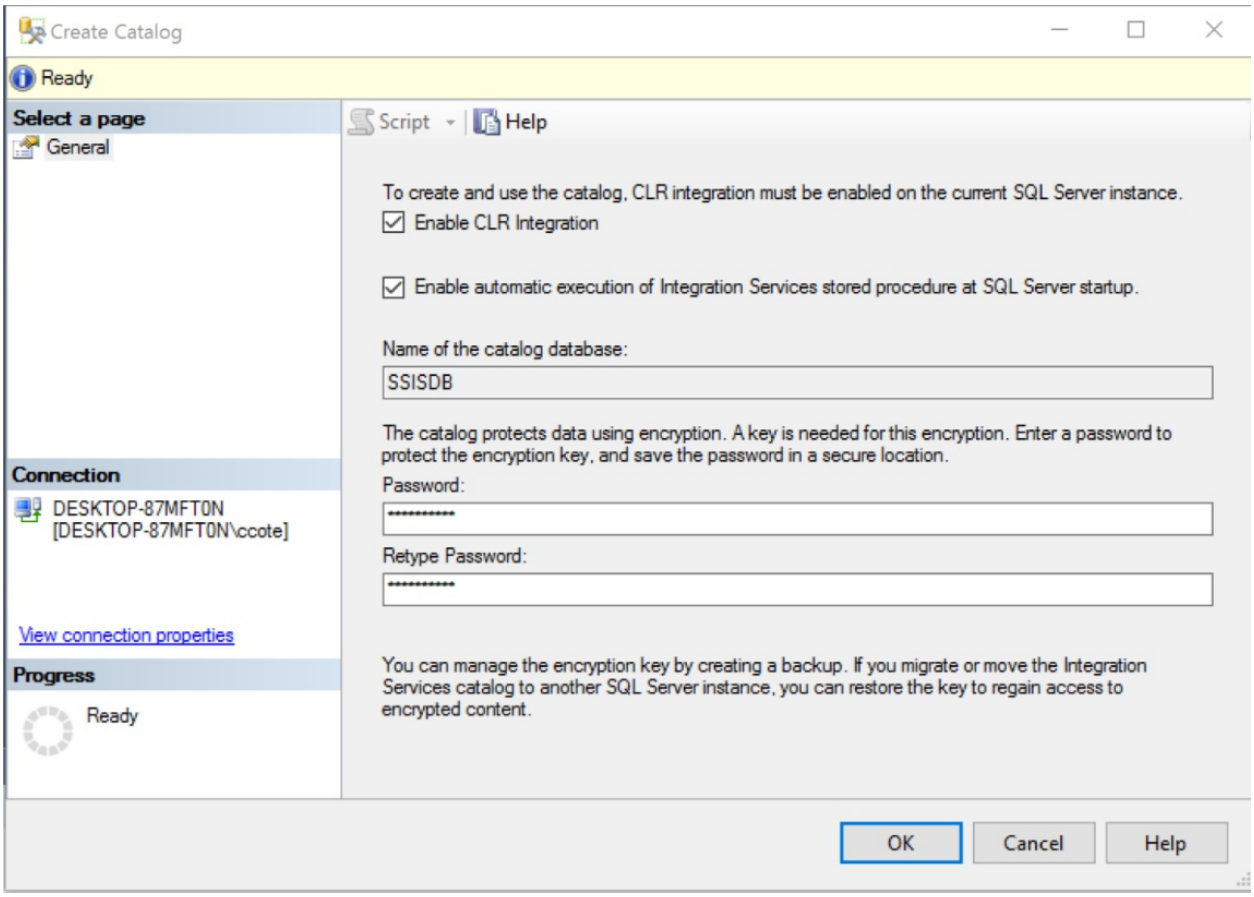
2. Look for the **`Integration Services Catalogs`** folder in object explorer. Right click on it and select the option **`Create Catalog...`**



Note

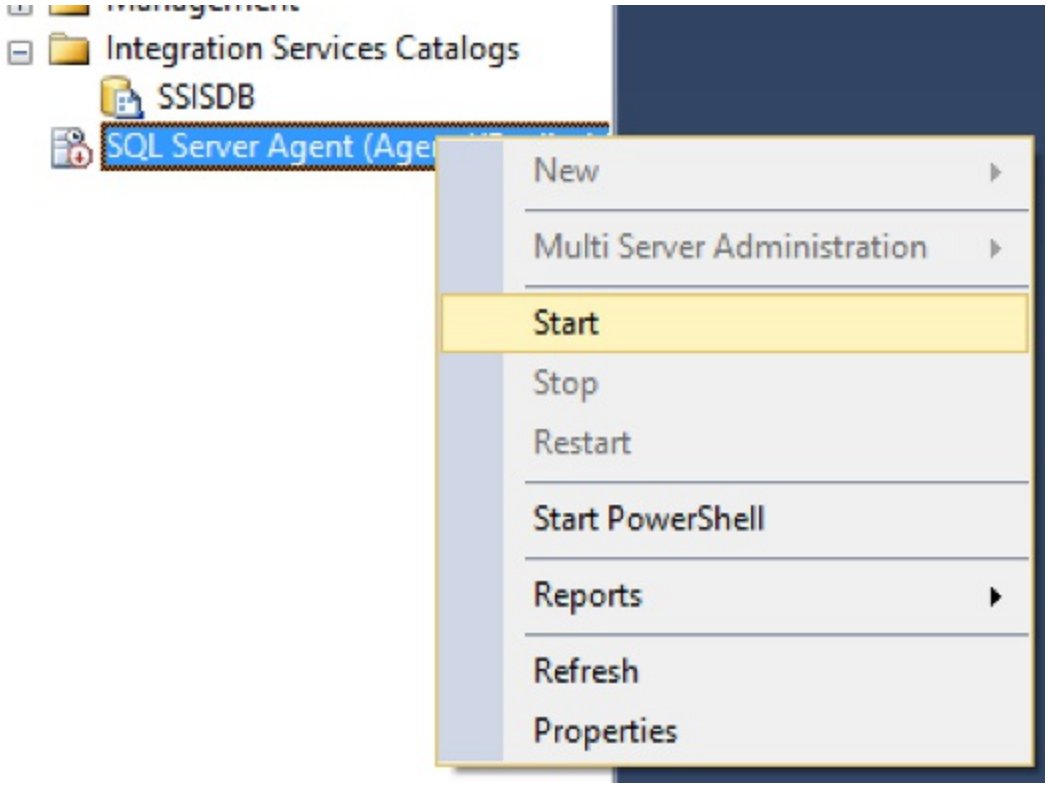
If **`the Create Catalog...`** option is not available (disabled - greyed out), it means that a catalog has already been created. Even if the folder is called **`Integration Services Catalogs`** - plural, only one integration services catalog can be created by SQL Server instance. The **`Create Catalog`** dialog box appears.

WOW! eBook
www.wowebook.org

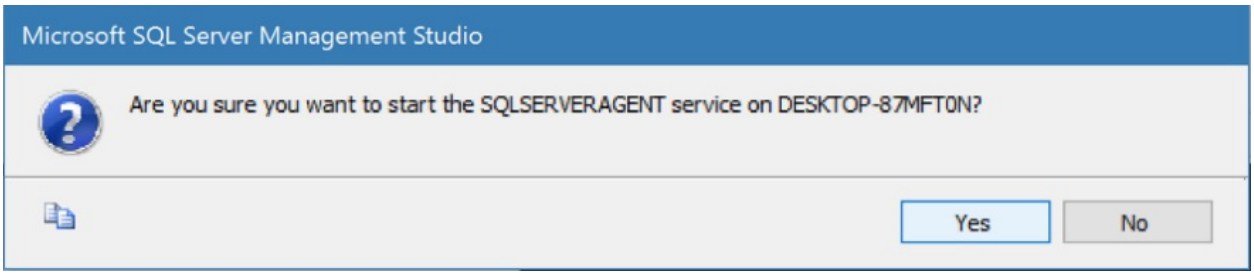


3. SQL Server CLR must be enabled to be able to create a catalog. It's also a good practice to check the **`Enable automatic execution of Integration stored procedures at SQL Server startup`**. This creates a job that clean up the SSIS calalog tables. To enable the job, SQL Server agent must be enabled and started. To enable the SQL Server Agent, right click on **`SQL Server Agent`** and select "Start" from the contextual menu that appears.

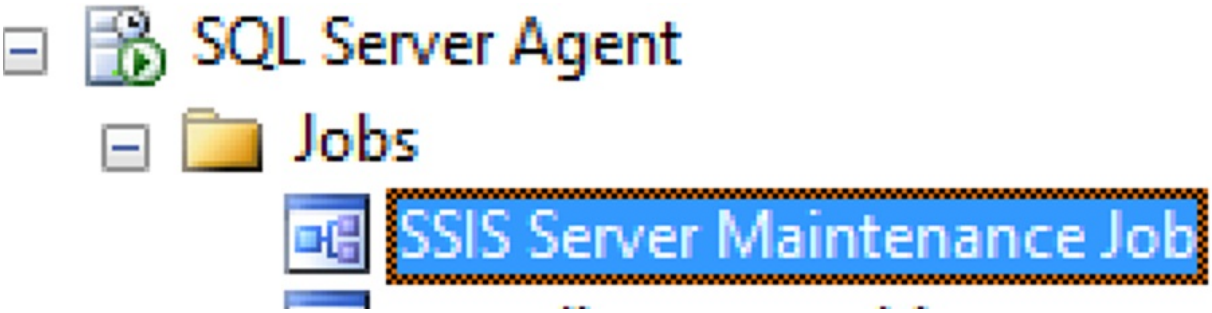
WOW! eBook
www.wowebook.org



4. A confirmation screen appears to confirm that we want to start the agent.

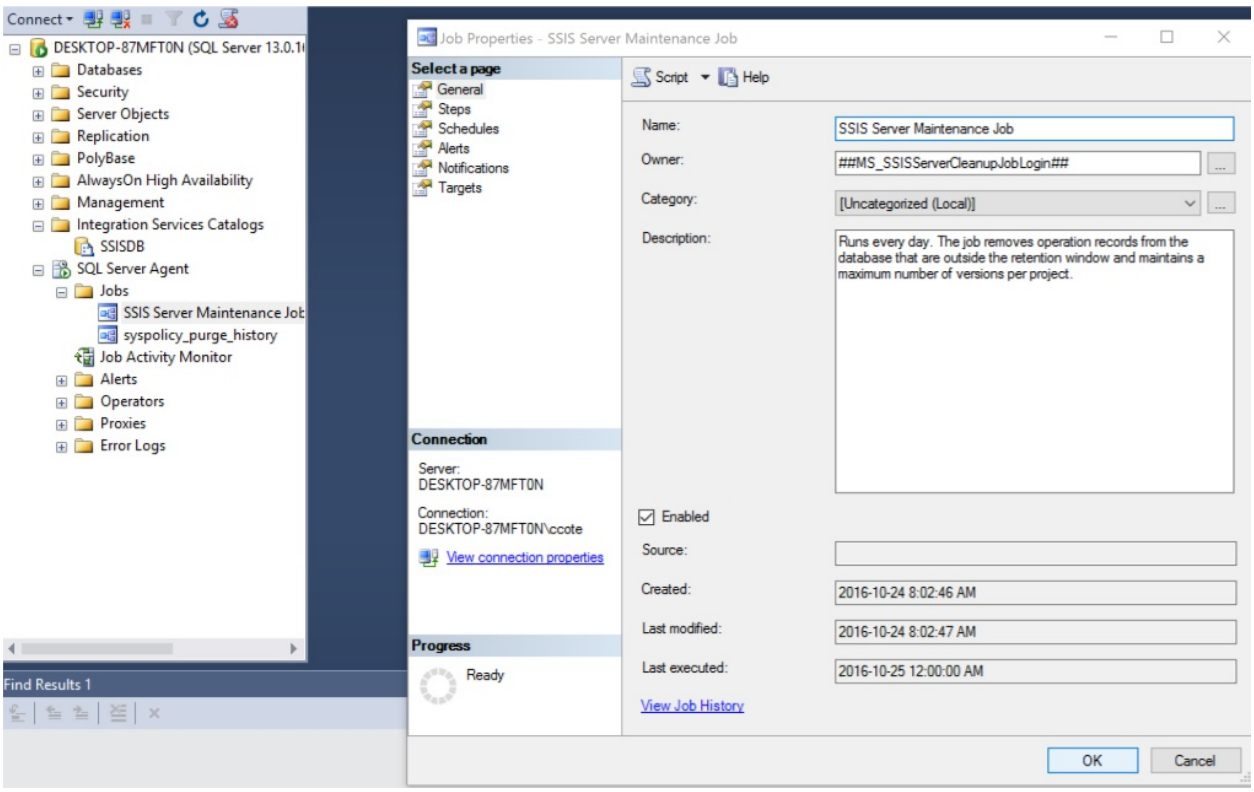


5. Once the agent has successfully started, we can see in the job folder a job called **`SSIS Server Maintenance Job`**



WOW! eBook
www.wowebook.org

6. Double clicking on it, you can see that this job is running every day to clean up the SSIS Catalog based on the retention window.

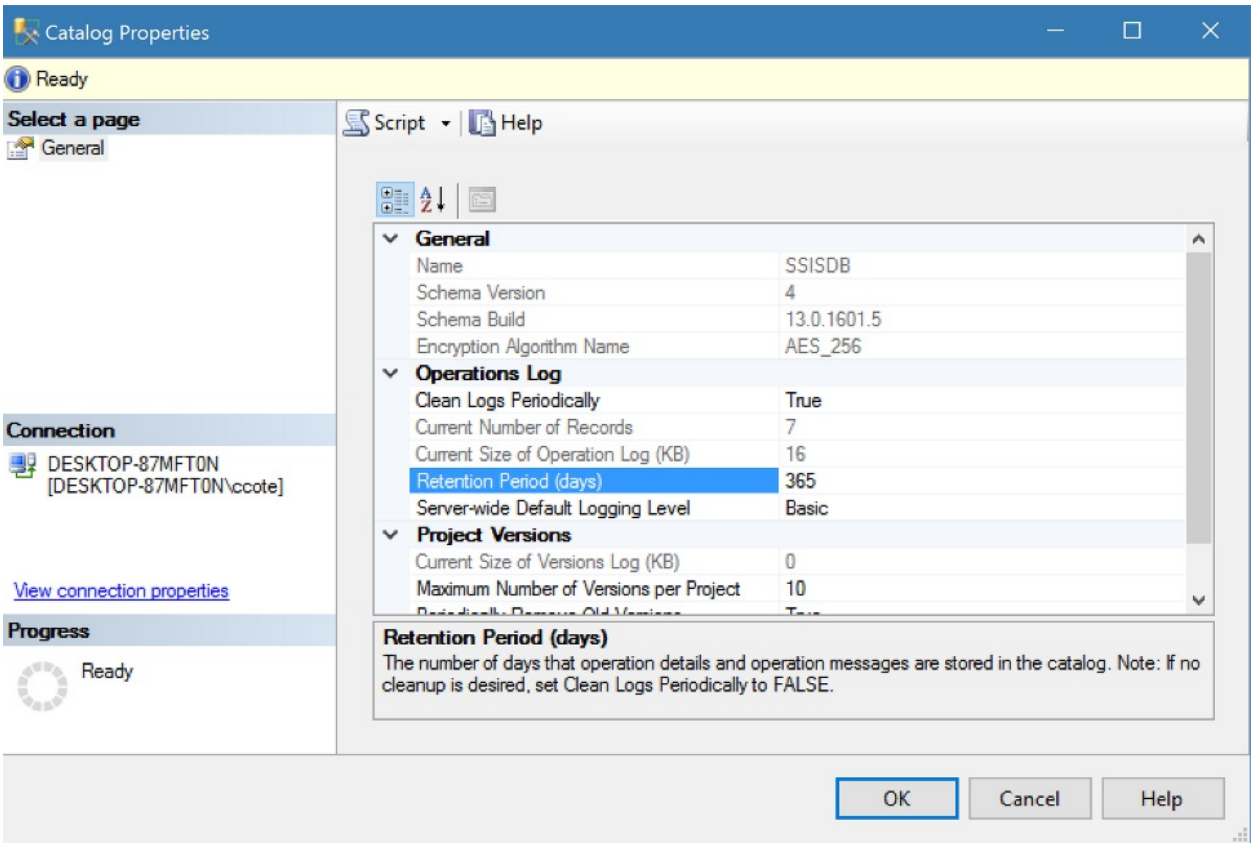


By default, the retention window is set to 365 days. We don't have to change it for the recipes we're going to implement from this book. But you should adjust the retention window setting to ensure that the catalog don't get filled with too many execution logs.

One of the benefits of this job is to execution clean up log entries in the catalog. As all executions are logged (we'll talk about logging in a later section), the catalog tables can fill up fast.

7. To manage the retention window, in SSMS object explorer, right-click on the catalog (SSISDB) and select properties from the drop down menu. The following window appears:

WOW! eBook
www.wowebook.org



8. You can now modify the **`Retention Period (days)`** property as shown above. Notice that you also have the possibility to stop the jobs clean logs jobs schedule by setting the **`Clean Logs Periodically`** to False.

WOW! eBook
www.wowebook.org

Custom logging

This section will talk about various logging and how we can customize logging to suit our need in terms of logging information. The reason why we need logging is because we want to retrieve some information's on our package executions. Here are some examples of logging info we might be interested to get:

- How much time it took to execute a specific package;
- How many rows have been transferred from one transform to another in our data flows;
- What were the warnings or errors that were issued by the package execution;
- The new value that have been assigned to a variable in a package;
- Etc.

All the topics listed here will be discussed in the next sections of the book. For now, we'll focus on the customized logging levels.

There are various ways that we can log package execution information in SSIS. In versions prior to 2012 (or if we opt for a Package Deployment Mode instead of the default one that is Project Deployment Mode), the only way to enable logging was to enable it in each package. If we forgot to enable it in one package, the latter would not log anything. The default Project Deployment mode can also log using package logging but it's better to use SSIS catalog logging, since it's integrated with projects once deployed and SSIS built in execution reports will use it to display package execution information's.

We'll talk about the various deployment models in next chapter, Chapter 3, *Key components of a modern ETL solution*

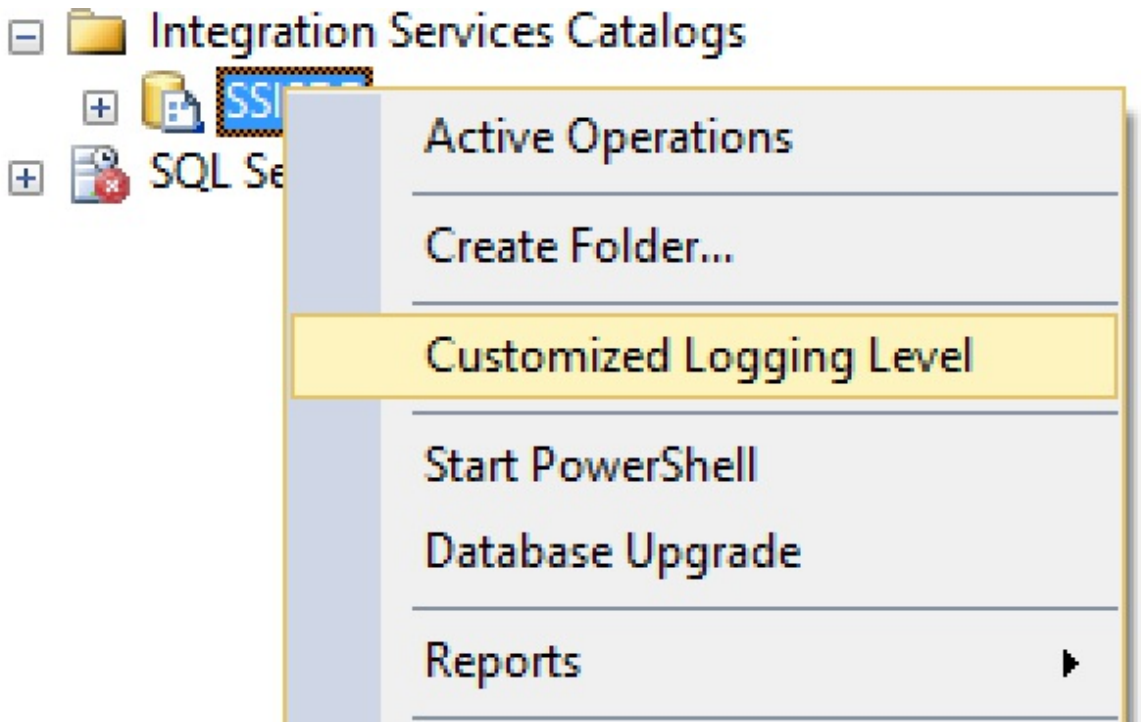
Getting ready

WOW! eBook
www.wowebook.org

This section required you have already installed SQL Server Management Studio, SQL Server Data Tools and created an SSIS catalog.

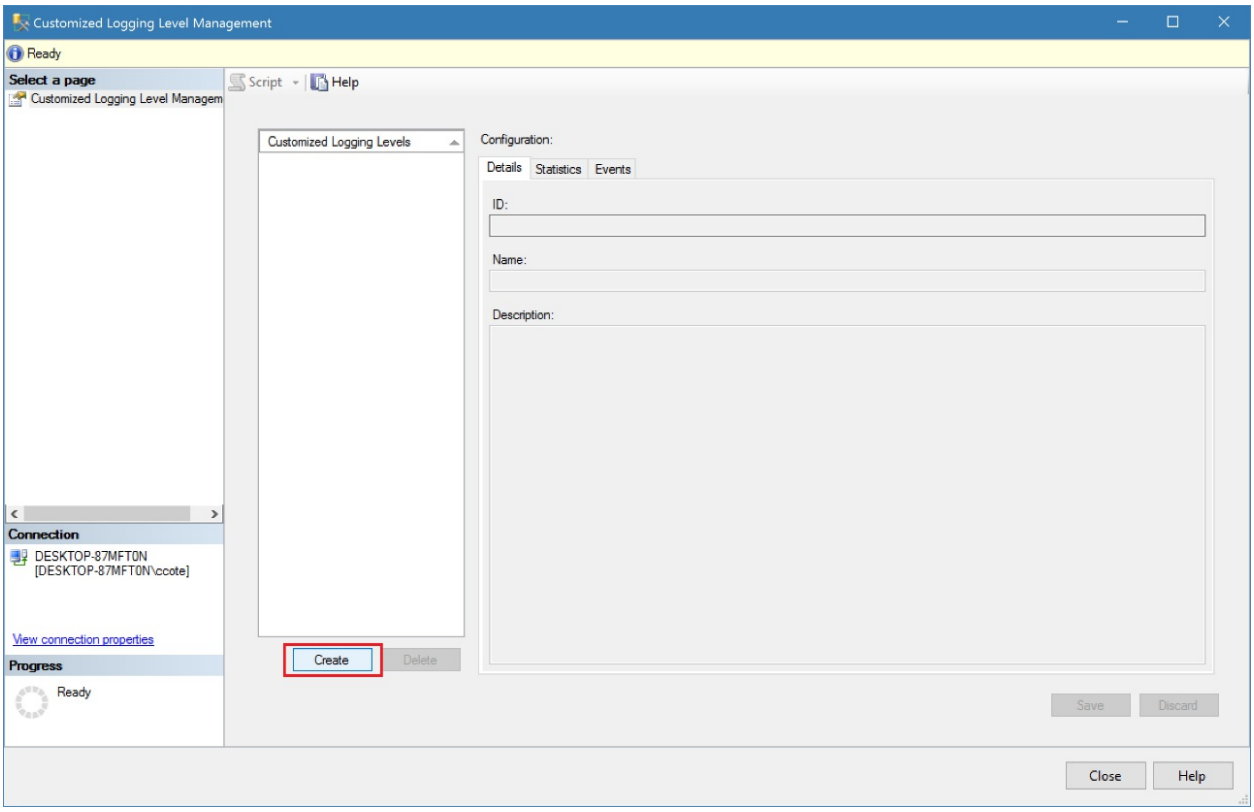
How to do it...

1. We'll start SQL Server Management Studio and connect to our local instance. We'll then expand the Integration Services Catalogs



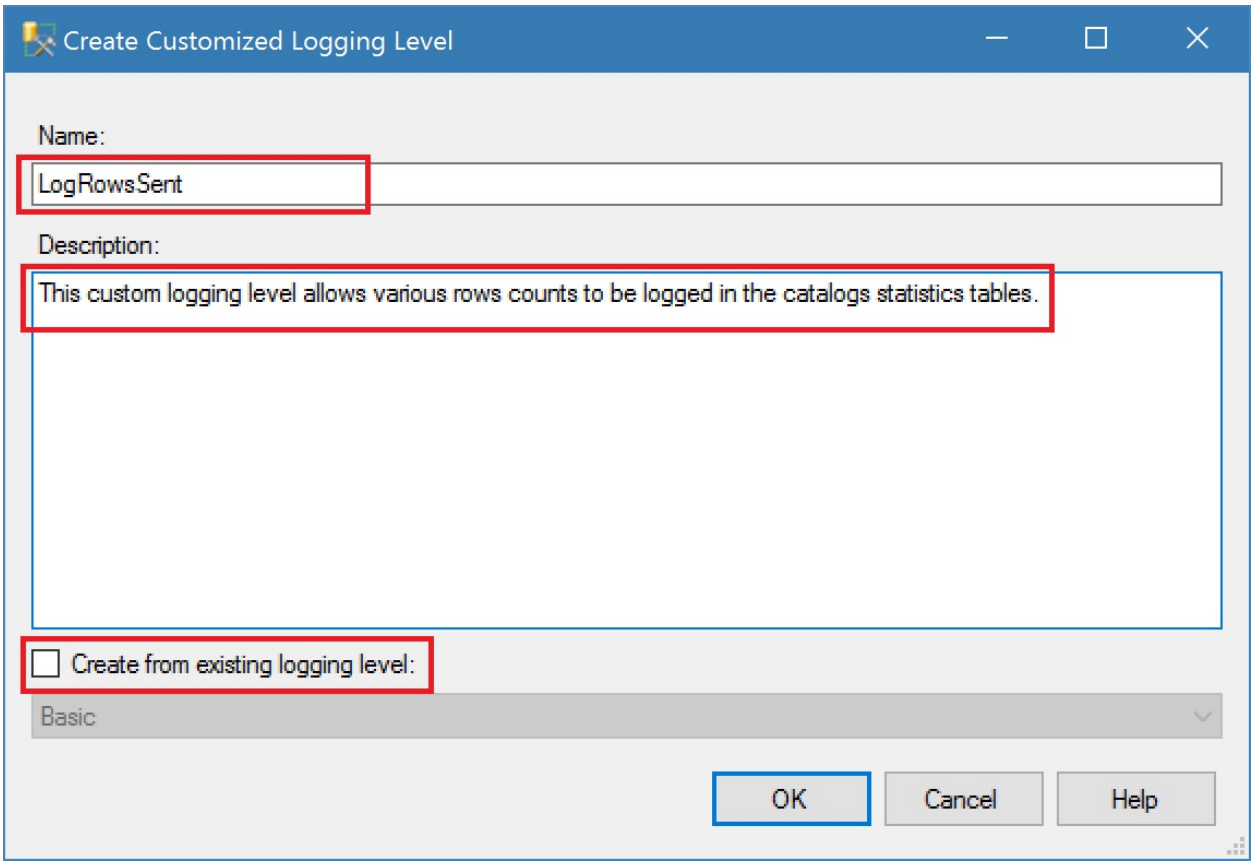
2. The following screen appears, click on the `Create` button to create a custom logging level.

WOW! eBook
www.wowebook.org



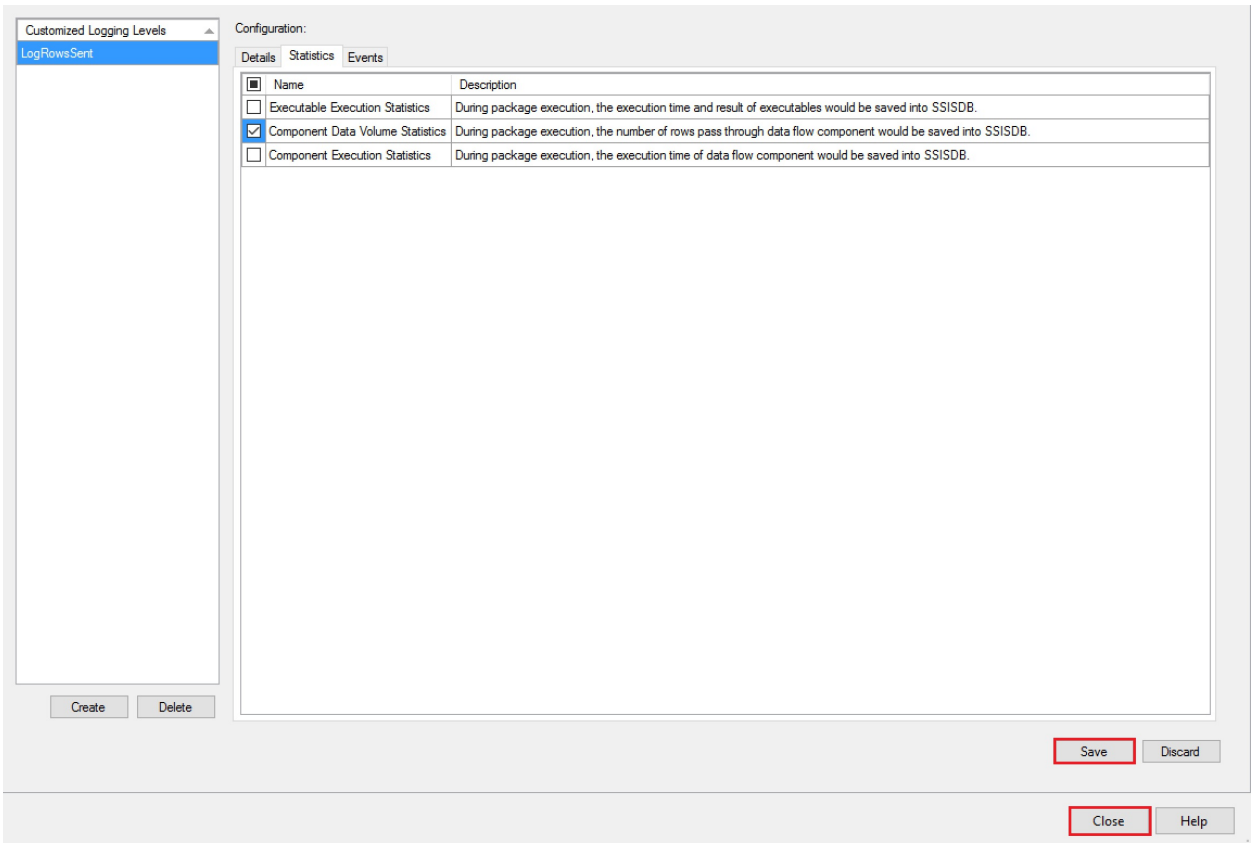
3. The **`Create Customized Logging Level`** screen appears. Set the various properties as shown in the following screenshot but leave the **`Create from existing logging level`** unchecked.

WOW! eBook
www.wowebook.org



4. In the configuration screen, select **`Component Data Volume Statistics`**, click **`Save`** and then **`Close`** as shown in the following screenshot.

WOW! eBook
www.wowebook.org



That's it for now! We have created our custom logging level.

How it works...

Using logging while executing our SSIS packages creates entries in SSIS catalog table called `[execution_data_statistics]` in the `[internal]` schema. This table is meant to record data movement statistics in SSIS packages.

We never use tables in the internal schema as they are seen as system tables. That's the reason we use the view `[catalog].[execution_data_statistics]` instead.

There's more...

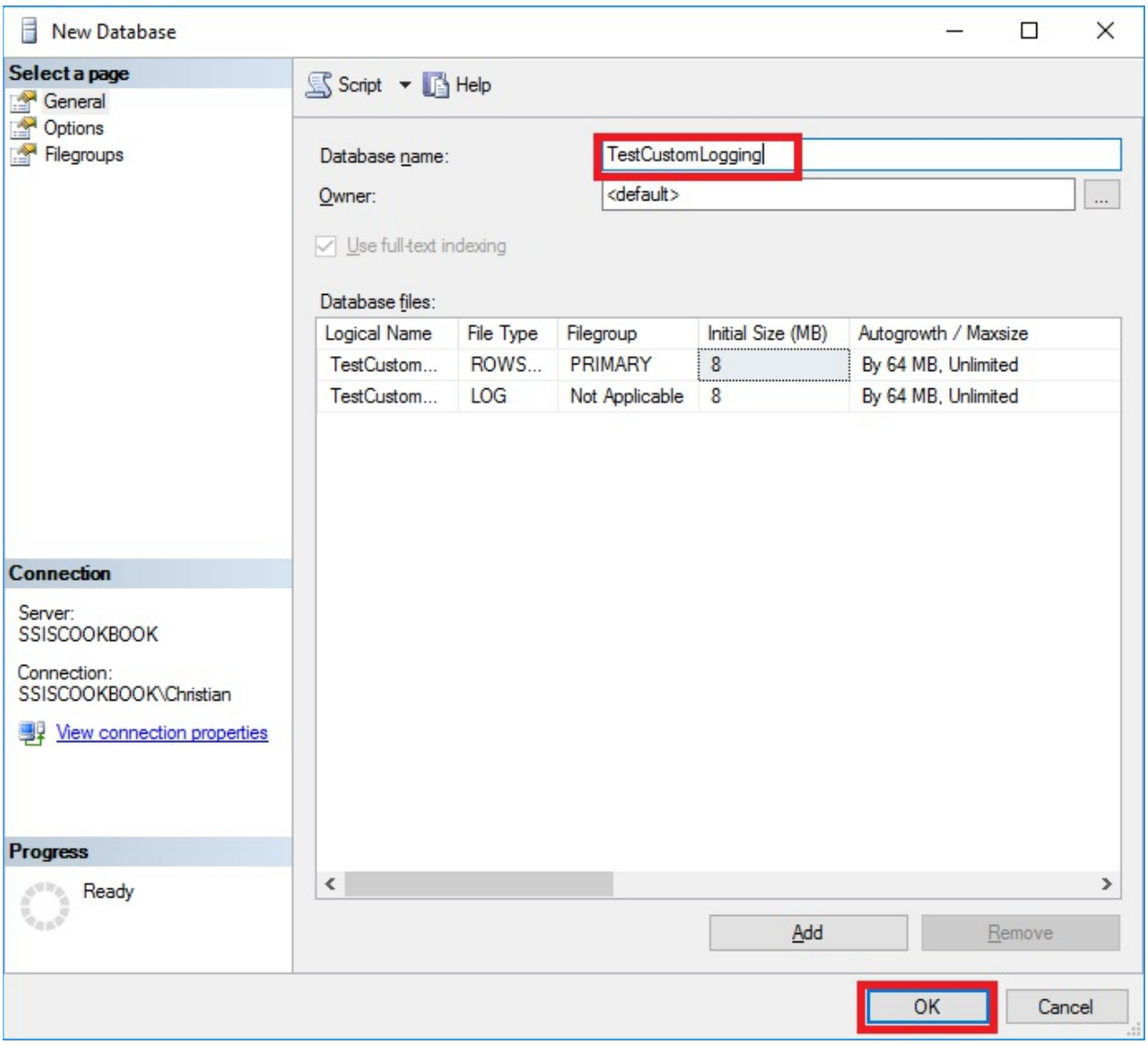
Creating our logging level in SSMS will allow us to use it when our SSIS project will be deployed in the SSIS catalog. We'll do that in the next recipe,

WOW! eBook
www.wowebook.org

Azure Tasks and Transforms; we'll create a simple SSIS package that will use this logging level when we'll execute it.

Create a database

This recipe, like many others in this book requires a database to be created. It can be done by using SQL Server Management Studio. Start SSMS, connect to your local instance and right-click on the database folder. From the contextual menu, choose **`New Database...`**. The following window appears.



Fill the **`Database name`** and click **`OK`** as we don't need to change the **`Owner`**

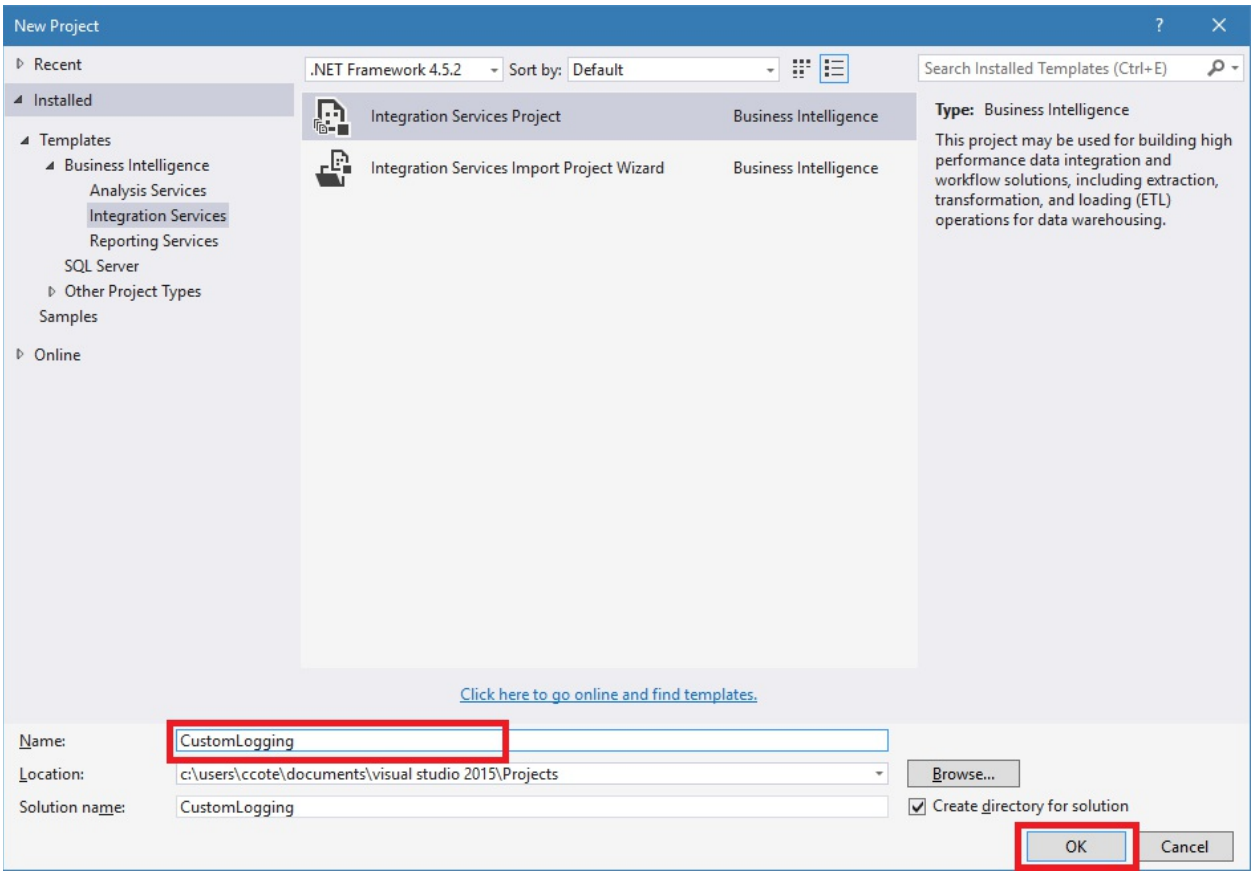
WOW! eBook
www.wowebook.org

and **`Database files`** information. You should now see the database under the Databases folder in SSMS. We're now ready to create a SSIS package that'll use it.

Create a simple project

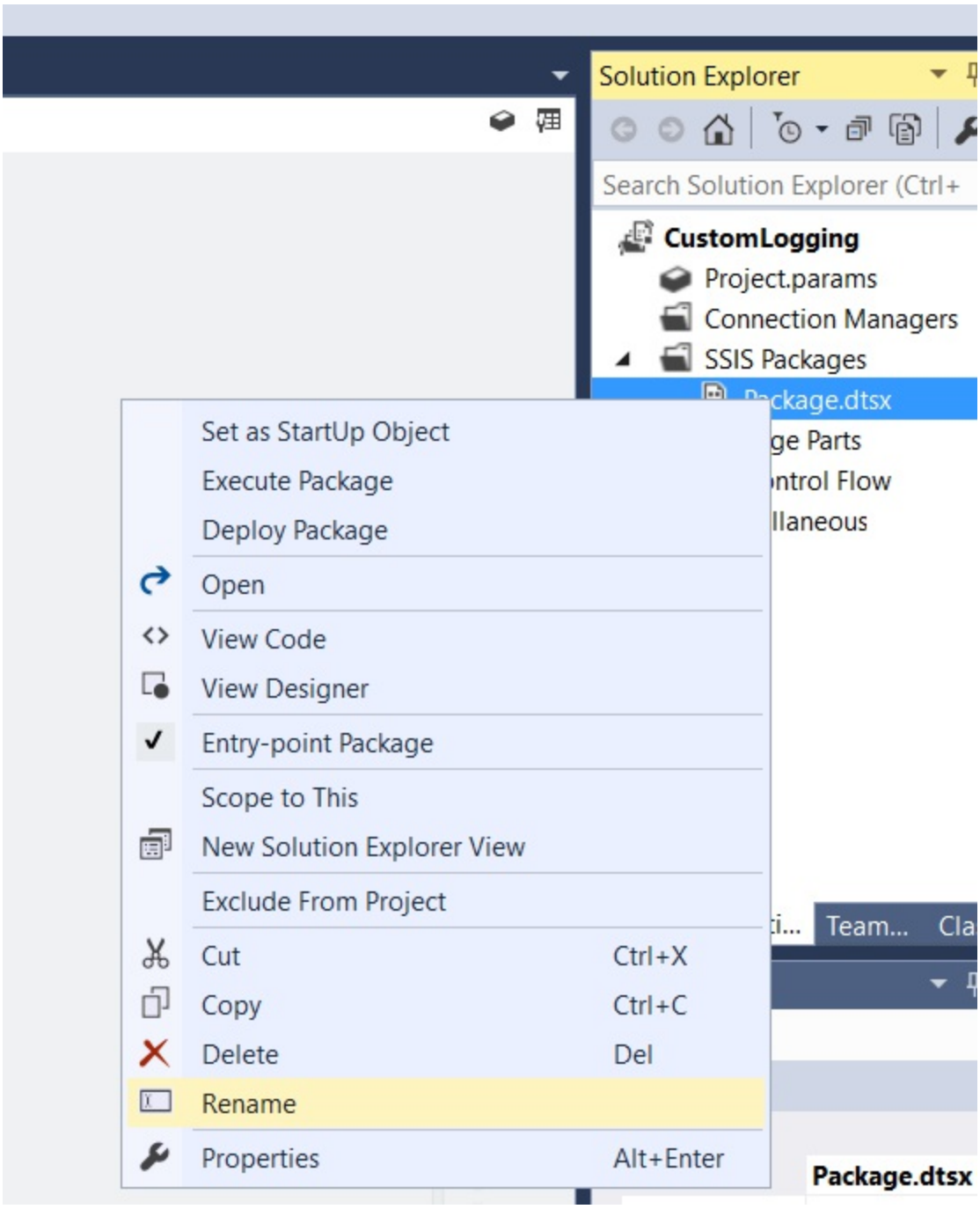
This recipe will show you how to create a SSIS project to be able to use the custom logging level that we just created in the previous part of the recipe.

First, start SQL Server Data Tools and create a new Integration Services project: Fileà NewàIntegration Services Project. The following screenshot suggest how the project can be named.



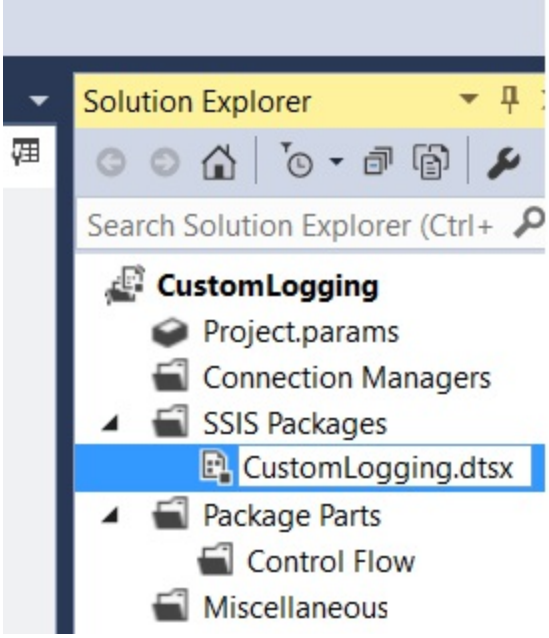
Once the project has been created, we should see a package called `Package.dtsx`. We'll right-click on it and select rename shown in the following screenshot.

WOW! eBook
www.wowebook.org



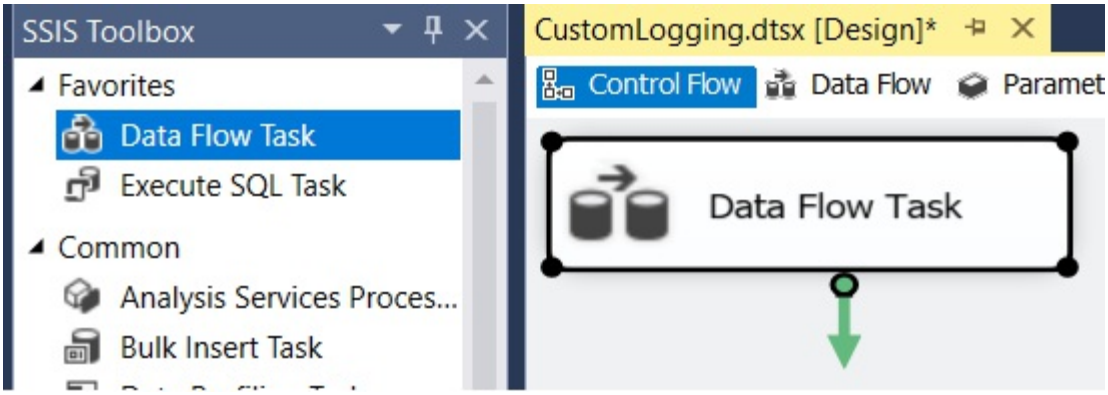
We'll name it `CustomLogging.dtsx`. The solution should now look like the next screenshot.

WOW! eBook
www.wowebook.org



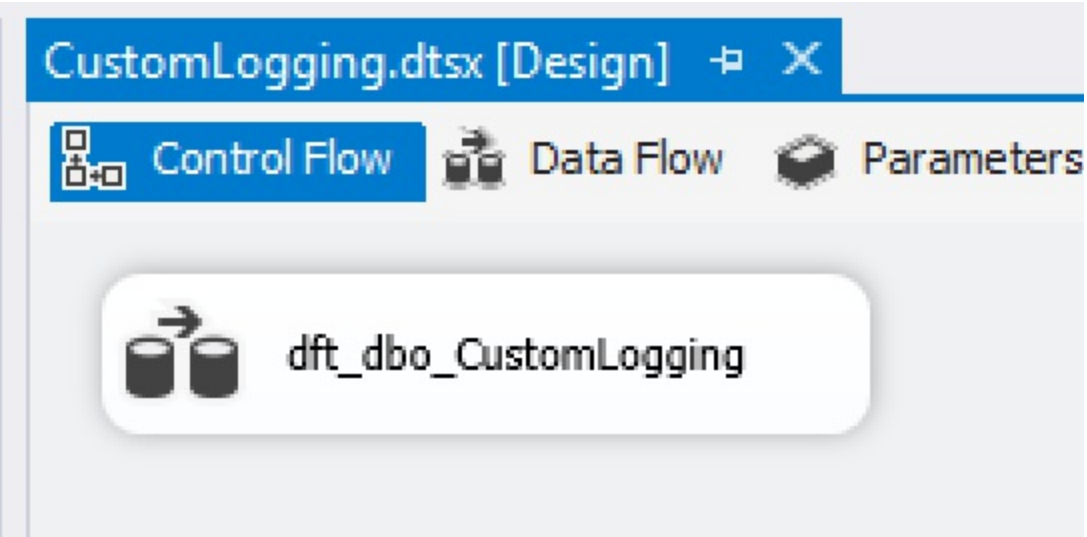
We will now add a Data Flow task in our package. A Data Flow task is a container that will allow us to do data transformations. Its toolbox has a rich set of data transformation tools.

From the SSIS Toolbox, drag a Data Flow task to the package's control flow as shown in the following screenshot.



Rename the DataFlow task `dft_dbo_CustomLogging` as shown in the following screenshot.

WOW! eBook
www.wowebook.org



In the next chapter, we'll talk about the way we name our SSIS tasks and transforms

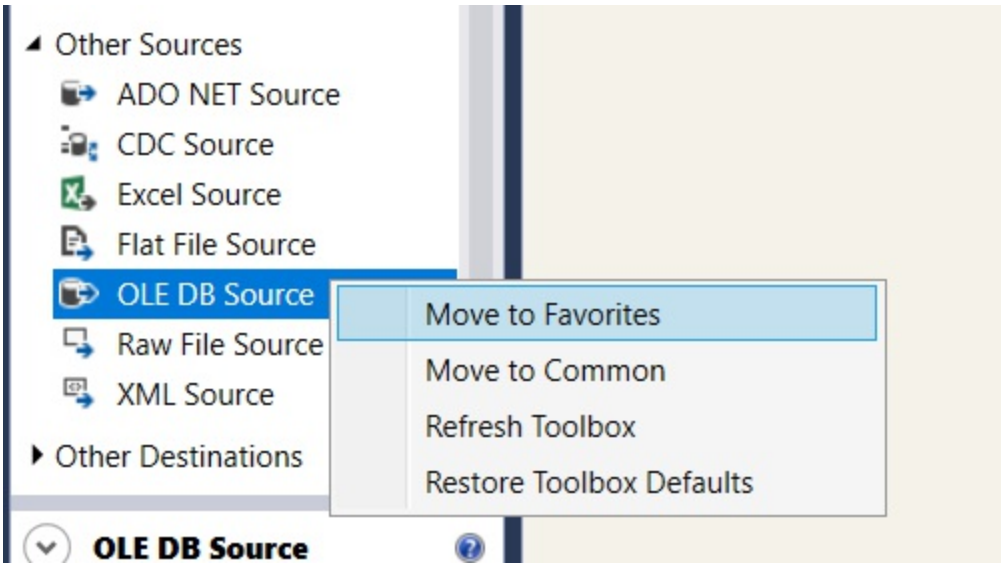
In the next few steps, we'll start customizing our SSIS toolbox. Double-click on the `dft_dbo_CustomLogging` dataflow task to go into it.

Throughout the book, we'll often customize the SSIS toolbox to suit our needs. You'll notice that the toolbox has sections like: Favorites, Common, Azure (will be covered in the next recipe), Other Sources, Other Transforms and Other Destinations.

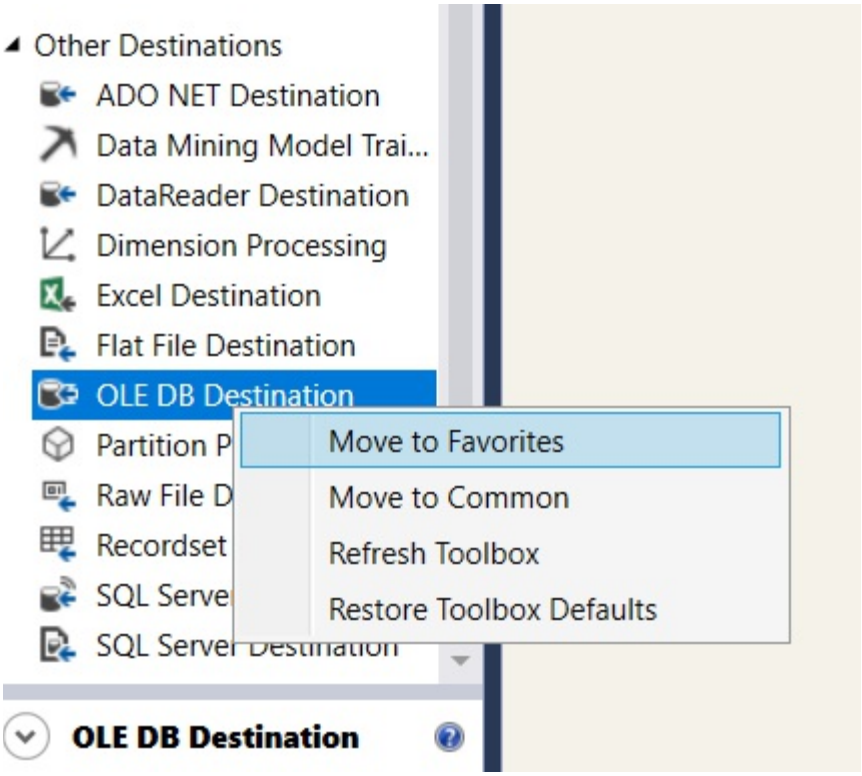
For now, we remove the source and destination assistants and add Ole DB source and destination to the favorites transforms.

5. In the SSIS Toolbox, scroll down to **`Other Sources`** and right-click on **`OLE DB Source`**. From the contextual menu that appears, select **`Move to Favorites`** as shown in the following screenshot.

WOW! eBook
www.wowebook.org



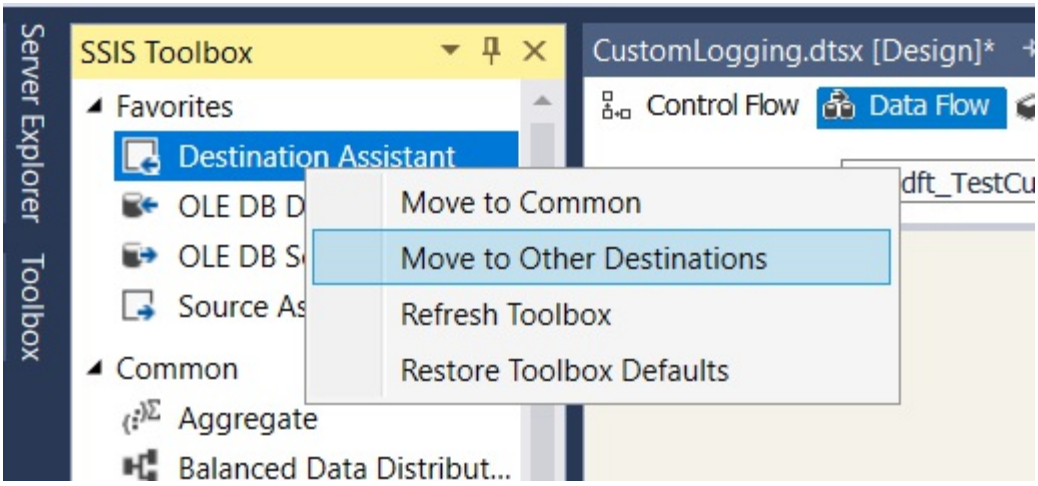
6. Next, scroll down again to **`Other Destinations`**, right-click on **`OLE DB Destination`** and select **`Move to Favorites`** as shown in the following screenshot.



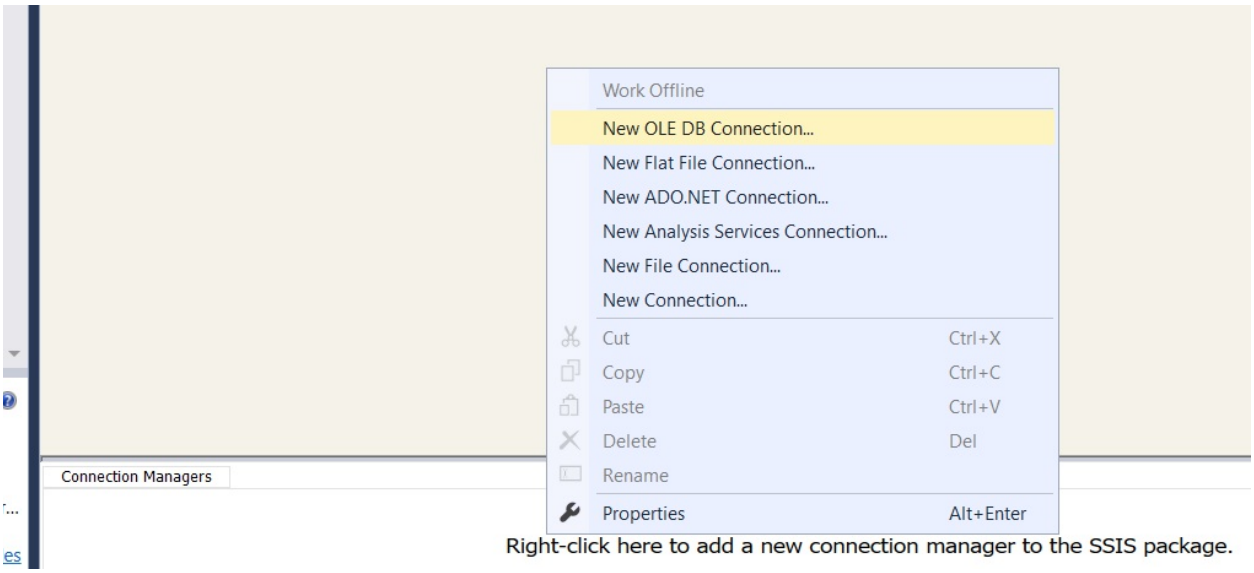
7. Now scroll up to the **`Favorites`** group at the top. We'll remove the source and destination assistants from the **`Favorites`**. As the following screenshot demonstrates, right-click on the **`Destination Assistant`** and

WOW! eBook
www.wowebook.org

select **`Move to Other Destinations`**. Repeat same process for the Source Assis

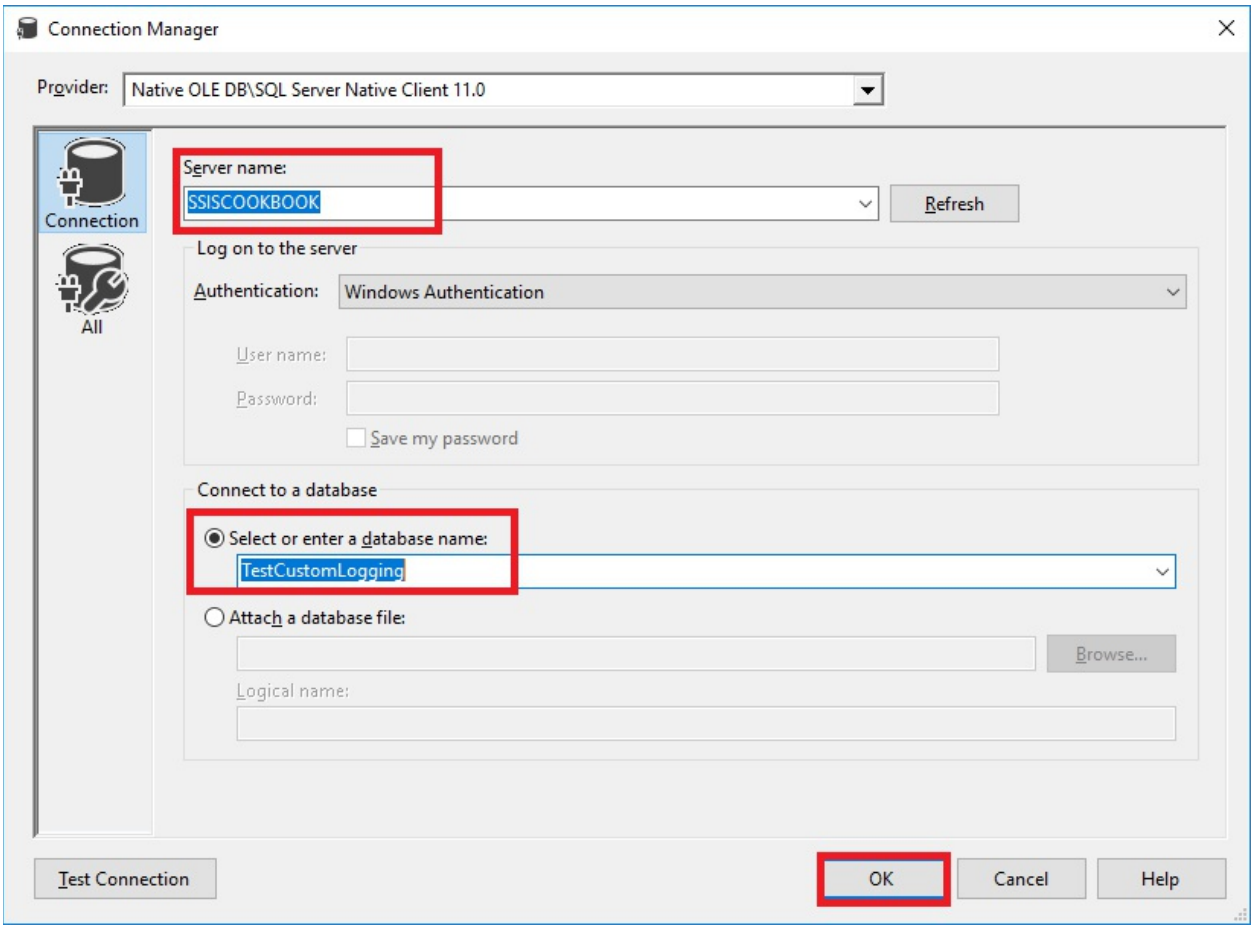


8. Repeat same process for the Source Assistant. We'll move it to the Other Sources group.
9. Now, we're ready to create a connection manager. We'll use it to read or insert data into the TestCustomLogging database that we created earlier. As shown in the following screenshot, right-click in the **`Connection Managers`** area and select **`New OLE DB Connection`** from then menu that appears.



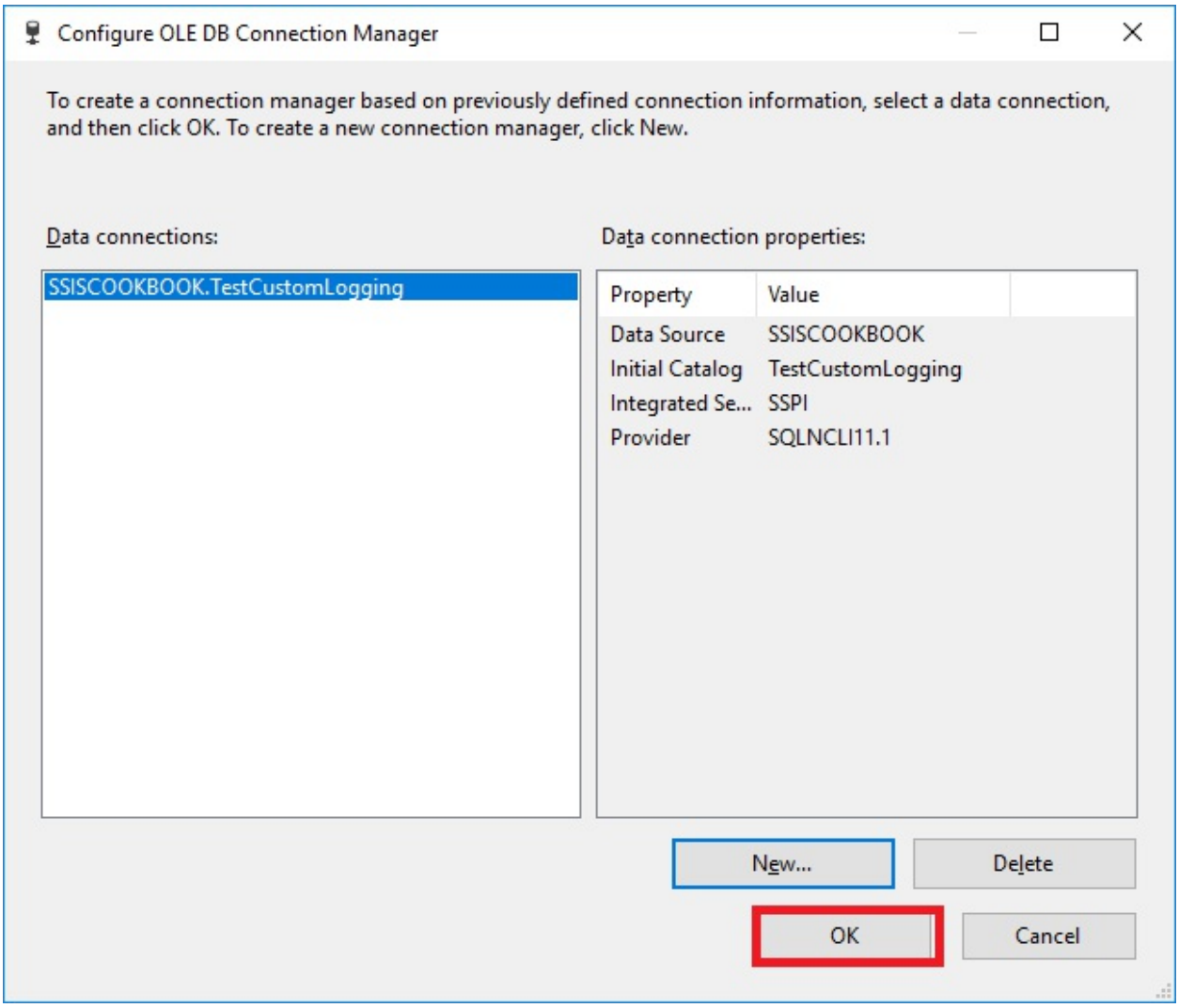
10. Click on New in the Configure OLE DB Connection window that appears and the following screen will appear. Set the **`Server Name`** to the

name of your machine or the named instance you might have chosen when you set up SQL Server. Select TestCustomlogging as the **`Database Name`**. The following screenshots shows the two properties set up for my PC. Click ok **`OK`** once finished.

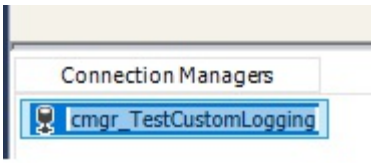


11. You're back on click **`OK`** in **`the Configure OLE DB Connection Manager`** window as shown in the following screenshot.

WOW! eBook
www.wowebook.org



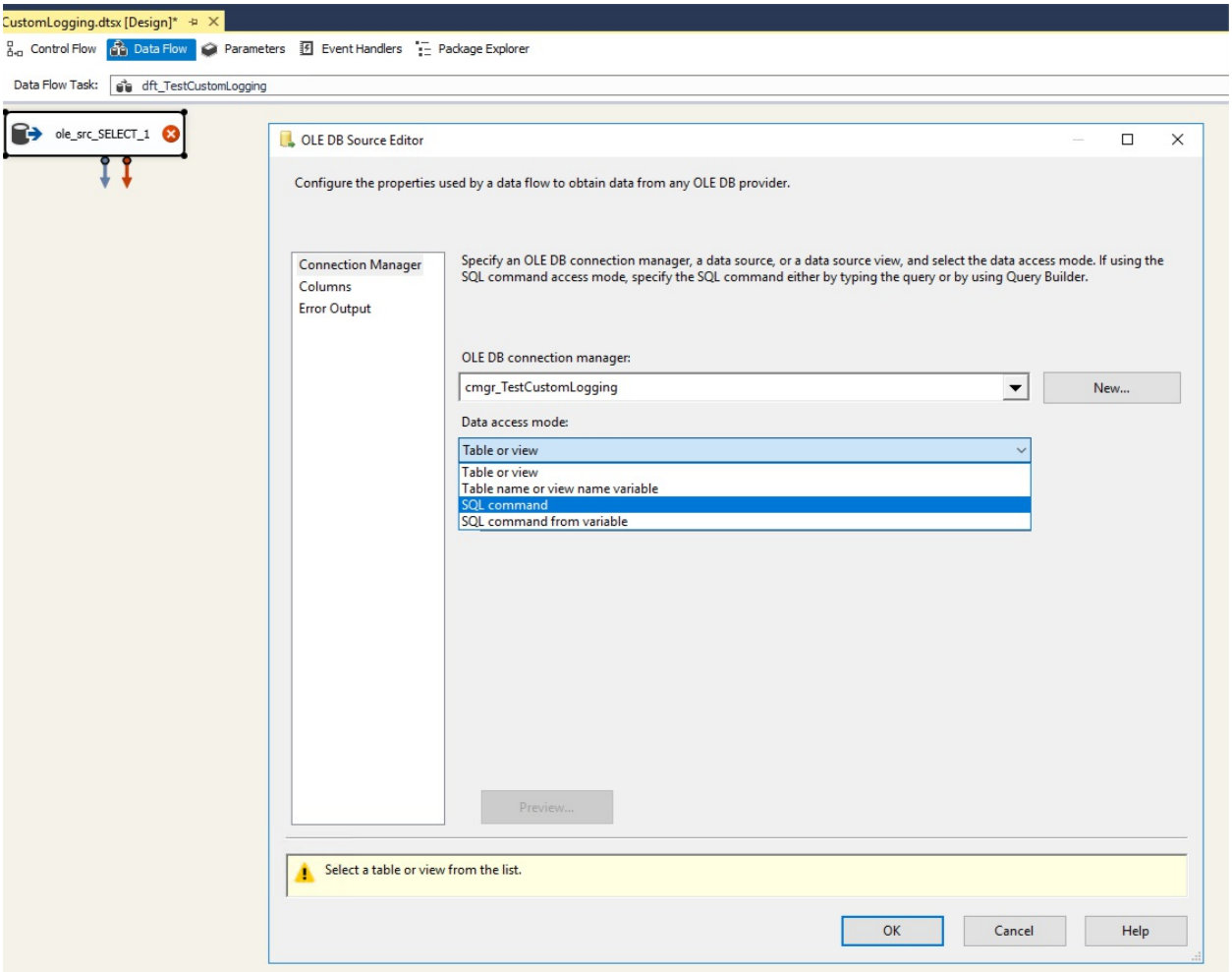
12. We'll now rename our connection manager. Select the newly created connection manager (SSISCOOKBOOK TestCustomLogging) to cmgr_TestCustomLogging as shown in the following screenshot.



13. Now, from the favorite section of the SSIS toolbox, drag and drop an OLEDB source on the surface of the dataflow. Now, as shown on the following screenshot, rename it to **`ole_src_SELECT_1`**. Double-click on it to get to the **`OLE DB Source Editor`** window. Set the **`OLE DB connection manager`** to **`cmgr_TestCustomLogging`** as demonstrated in

WOW! eBook
www.wowebook.org

the screenshot. Set the **Data access mode** to **SQL command**.

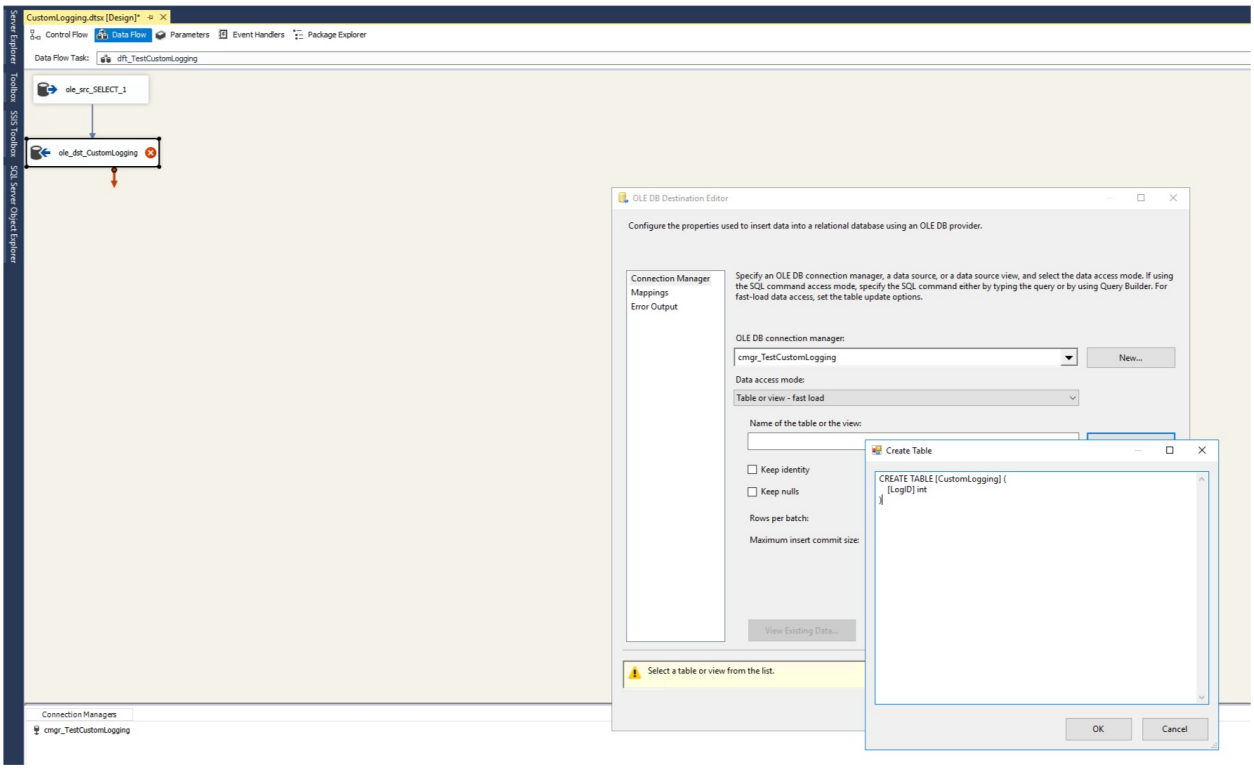


14. Now, enter the following SQL command in the command text and click on **OK**:

```
SELECT 1 AS LogID
UNION ALL
SELECT 2 AS LogID
```

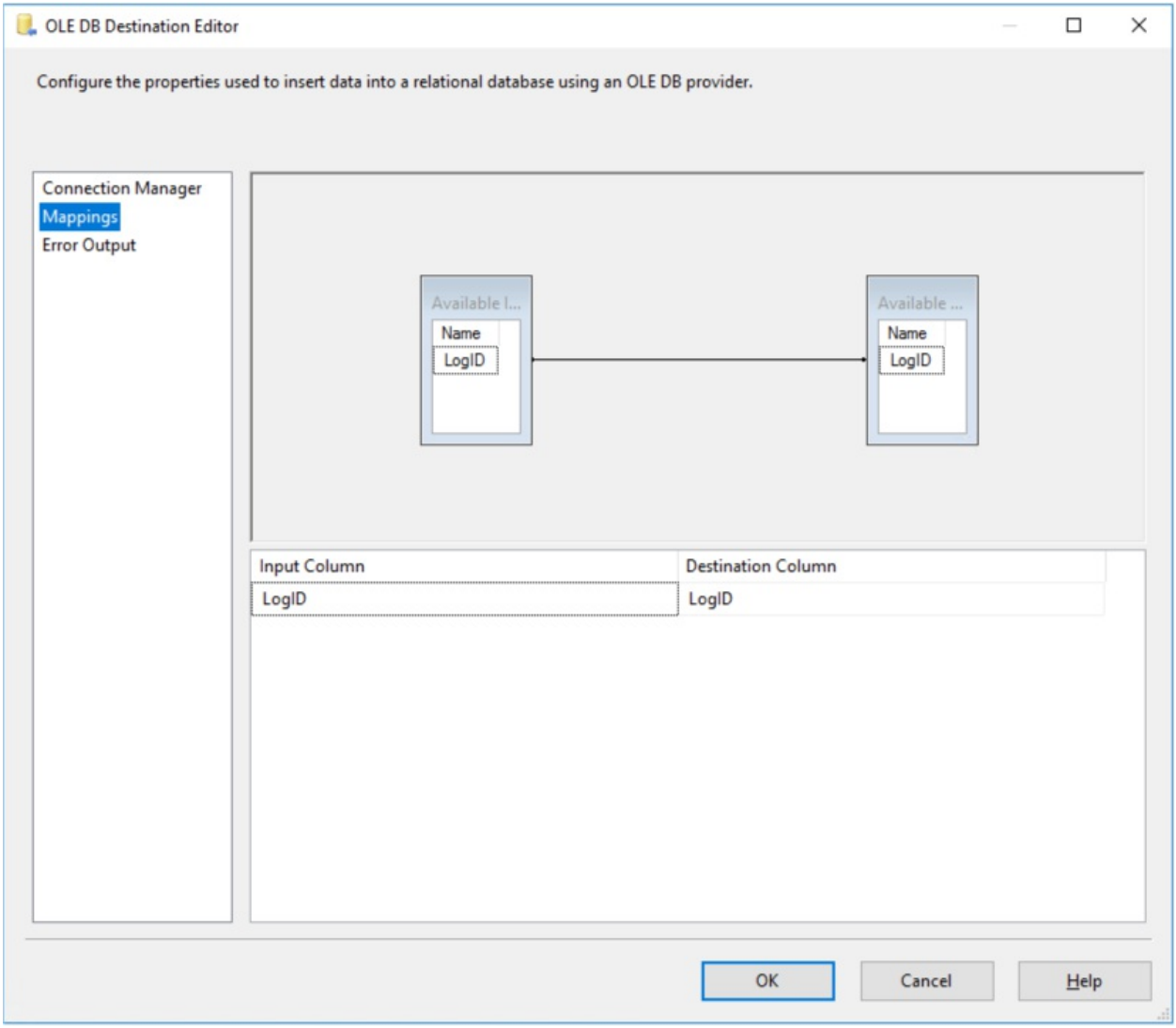
15. Again, from the Favorites section of the SSIS Toolbox, drag an OLE DB Destination in the dataflow. Connect the source to the destination and rename the OLE_DB Destination **ole_dst_dbo_CustomLogging**. Double-click on it, assign the **OLEDB connection manager** property and click ok the **New** button at the right of the Name of the table or the view property. The Create Table window appears. Modify the command as shown in the following screen shot and click on **OK**.

WOW! eBook
www.wowebook.org



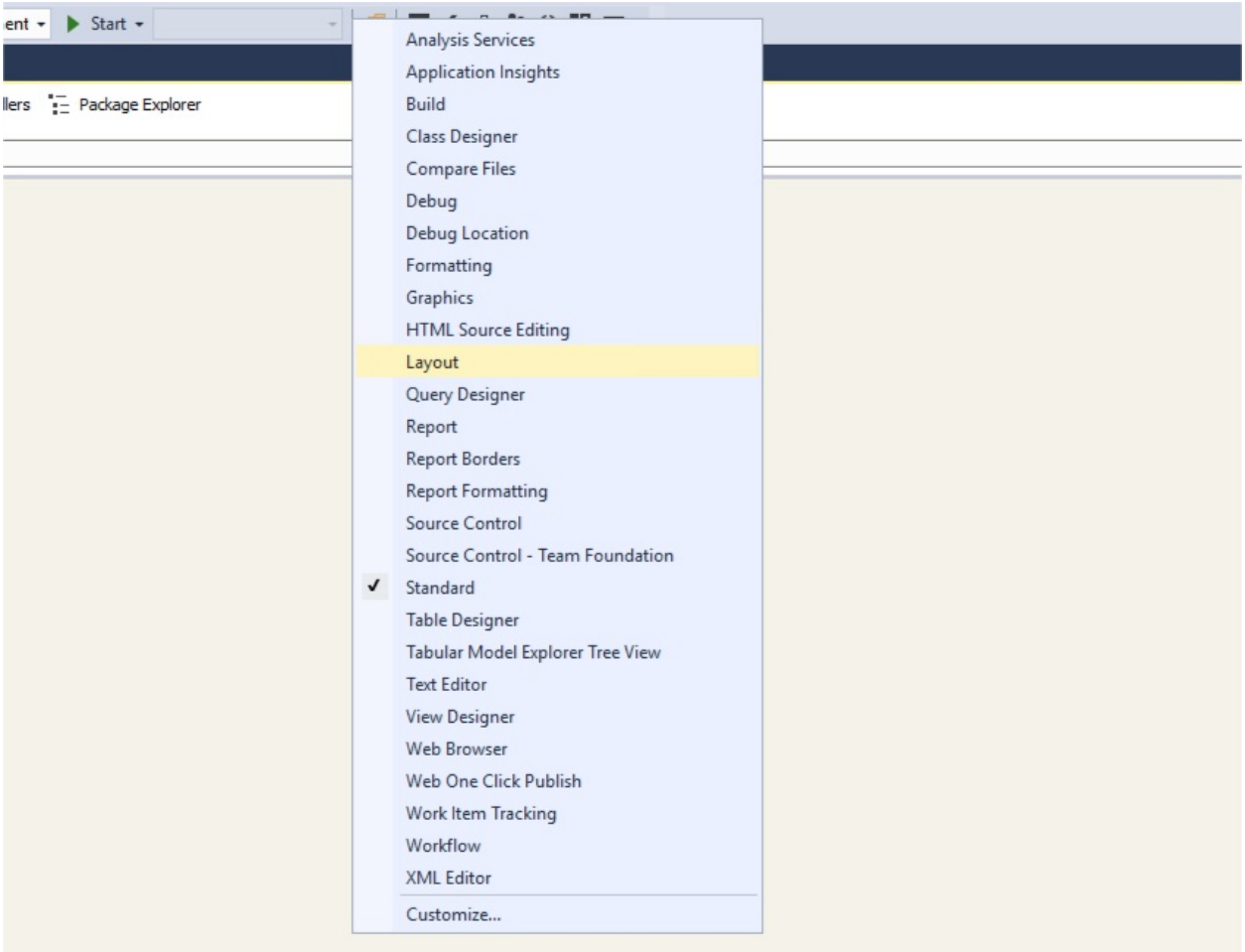
16. Click on **`Mappings`** in the list at the top left of the **`OLE DB Destination Editor`**. You should get the same screen as in the following screenshot.

WOW! eBook
www.wowebook.org



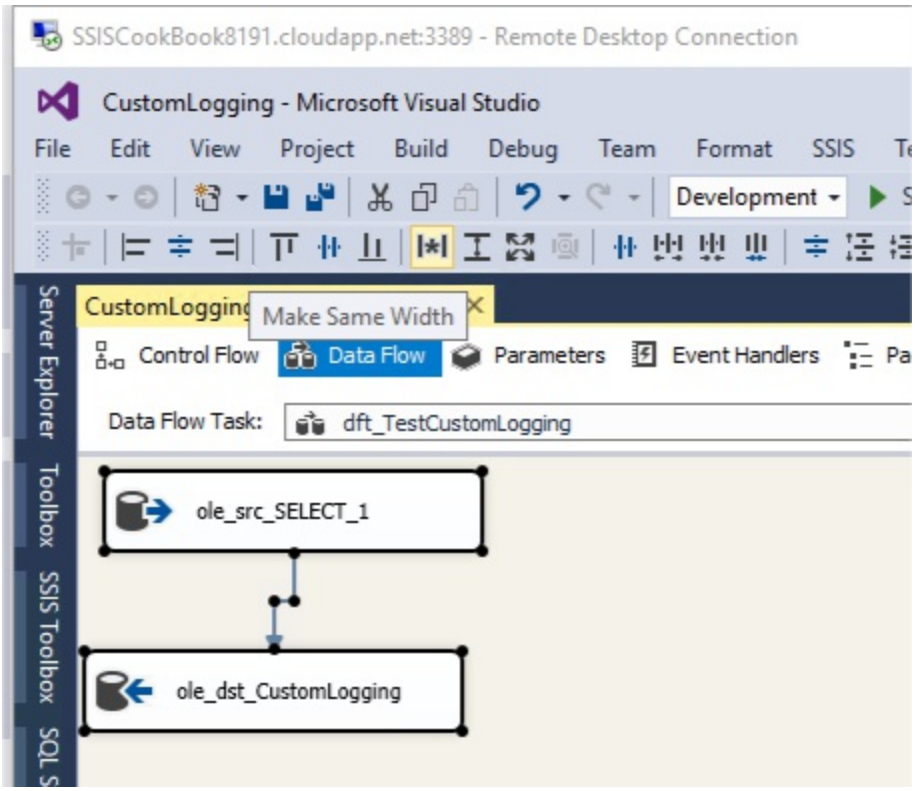
17. Click on **OK** to close the **OLE DB Destination Editor**.
18. Now, we'll bring the layout toolbar. We'll use it throughout of this book to format our package objects properly. Right-click anywhere in an empty section of the quick access toolbar and select **Layout** as highlighted in the following screenshot.

WOW! eBook
www.wowebook.org

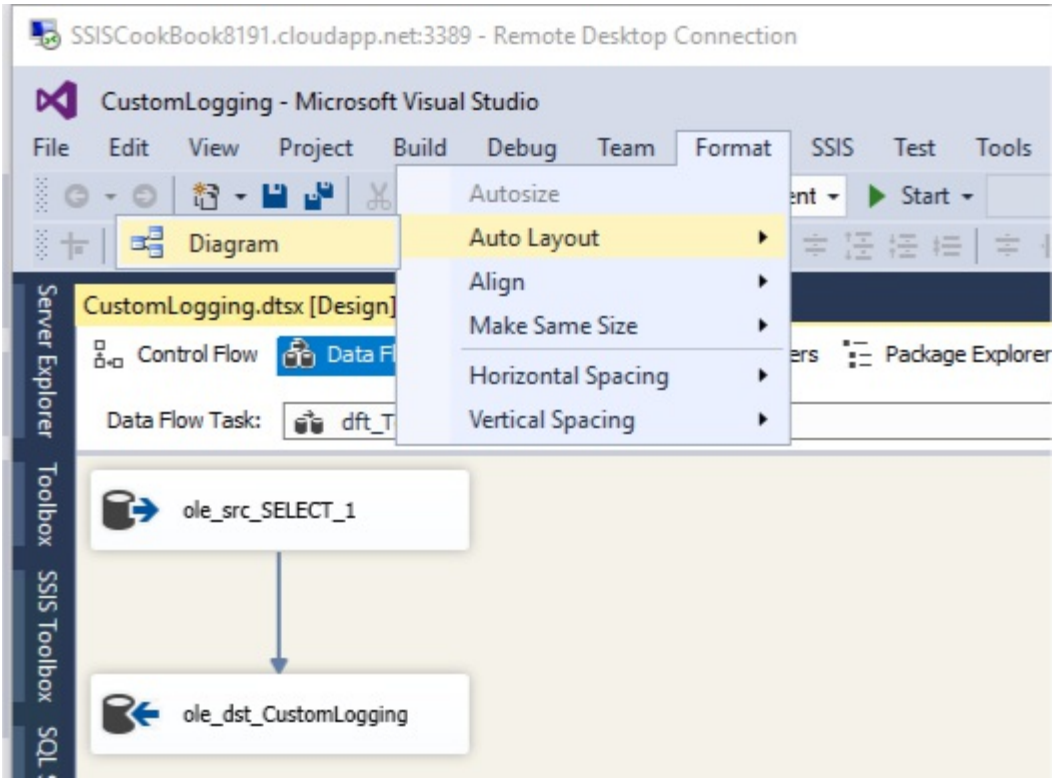


19. Now, from the Edit menu, select Select All of press Ctrl and A to select the entire dataflow content. In the Layout toolbar, click on the [*], Make Same Width button as sown in the following screenshot.

WOW! eBook
www.wowebook.org

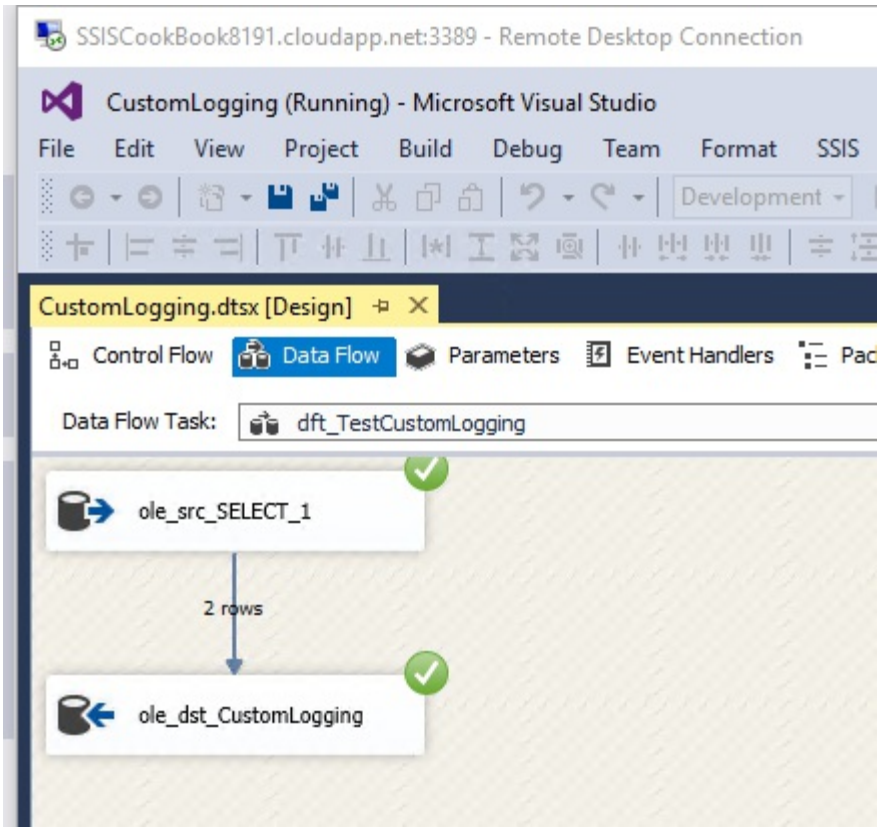


20. From the menu, select Auto LayoutàDiagram. This will format the data flow objects as shown in the following screenshot.



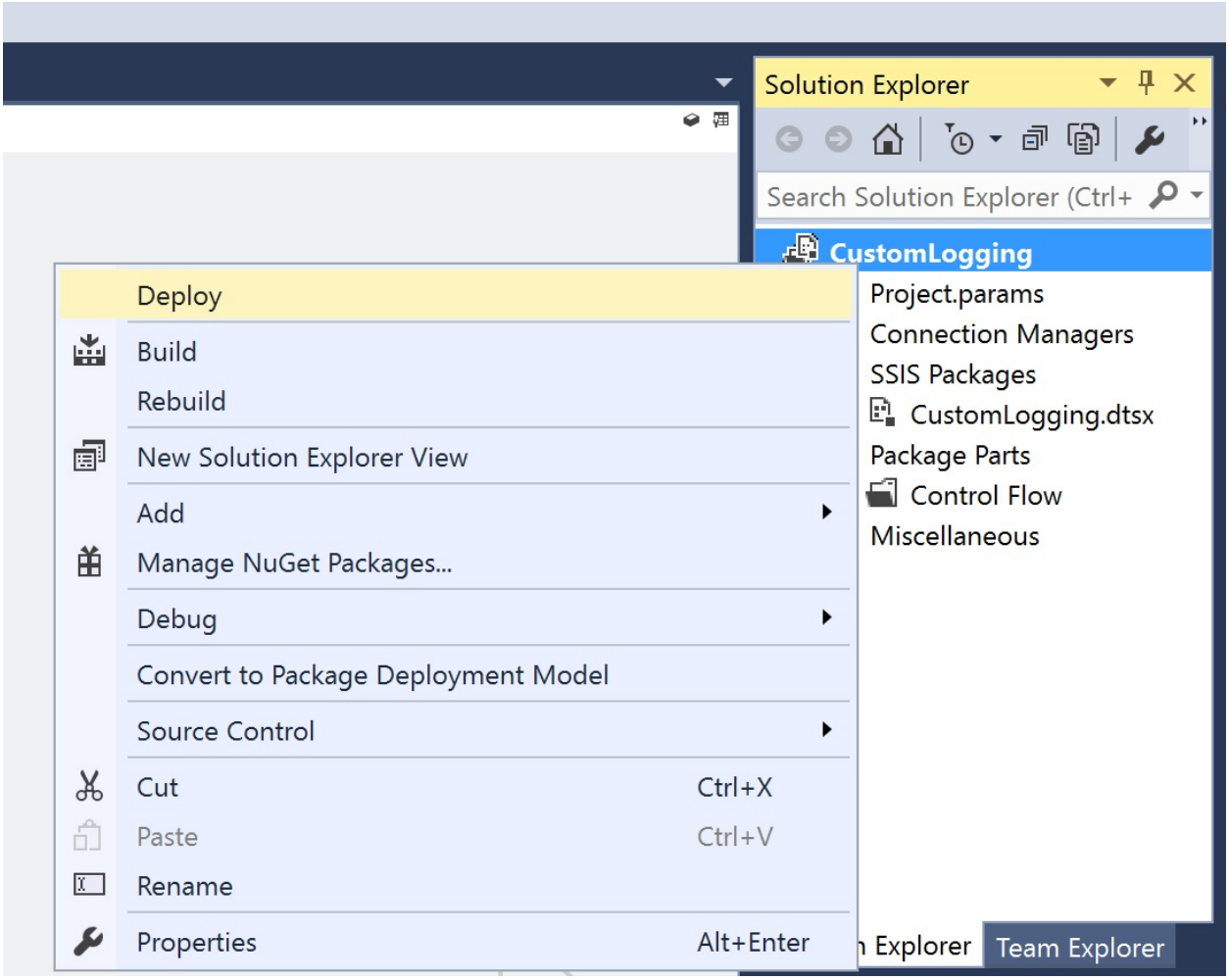
WOW! eBook
www.wowebook.org

21. Now, right-click anywhere in the data flow task background and select Execute Task from the contextual menu that appears. The data flow should execute successfully as follows.



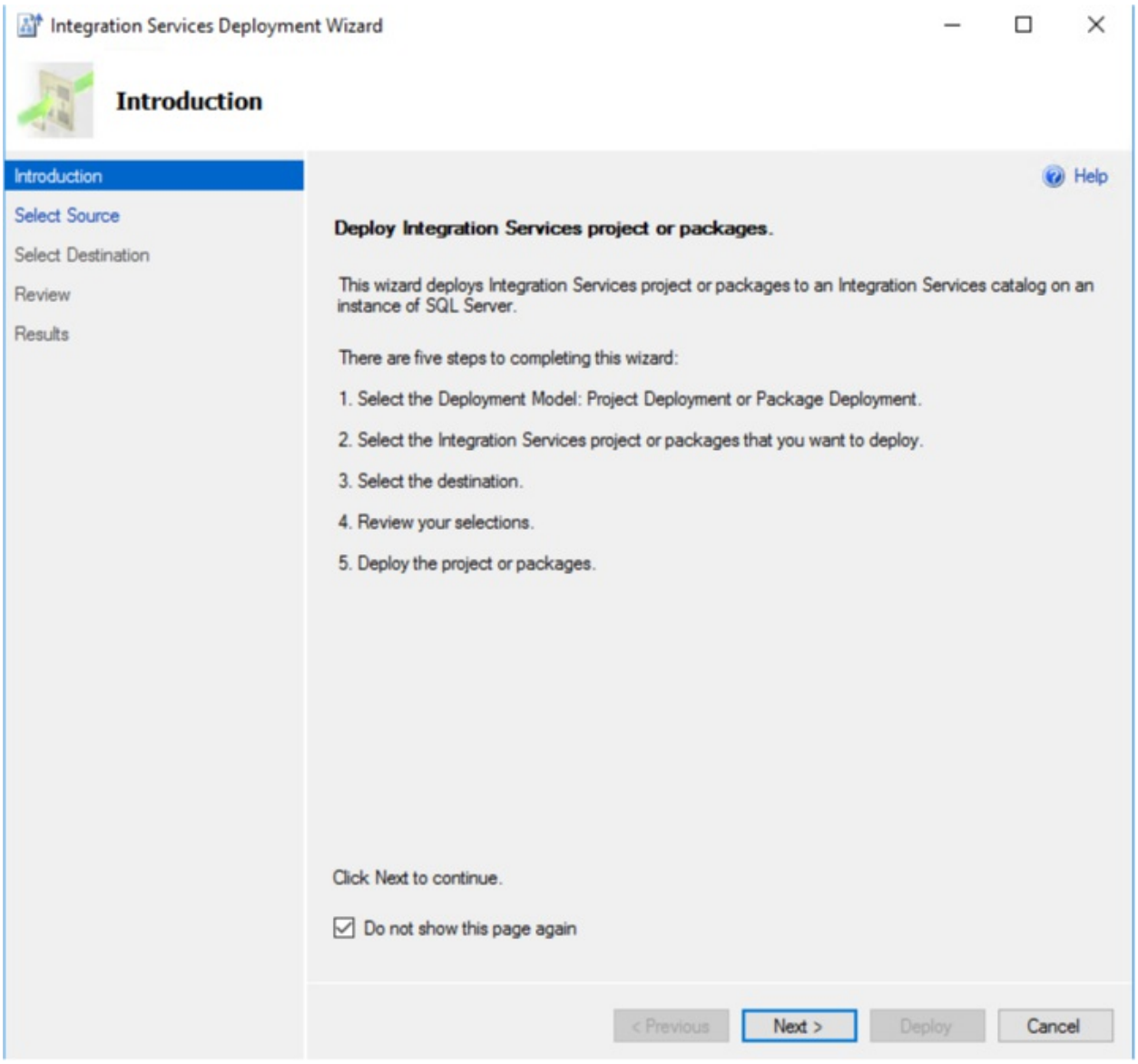
22. Now, we're ready to deploy our project to the SSIS Catalog that we created at the beginning of the chapter, under the recipe, *Creating SSIS Catalog*. Right on the CustomLogging project in the Solution Explorer at the top right of SSDT and hit Deploy. The project deployment wizard starts as shown in the following screenshot.

WOW! eBook
www.wowebook.org



23. The first page explains the step performed by the wizard. Check the **`Do not show this page again as in the following screenshot`** if you want to skip this step in future deployments of the project or individual packages as we'll see later in this chapter.

WOW! eBook
www.wowebook.org



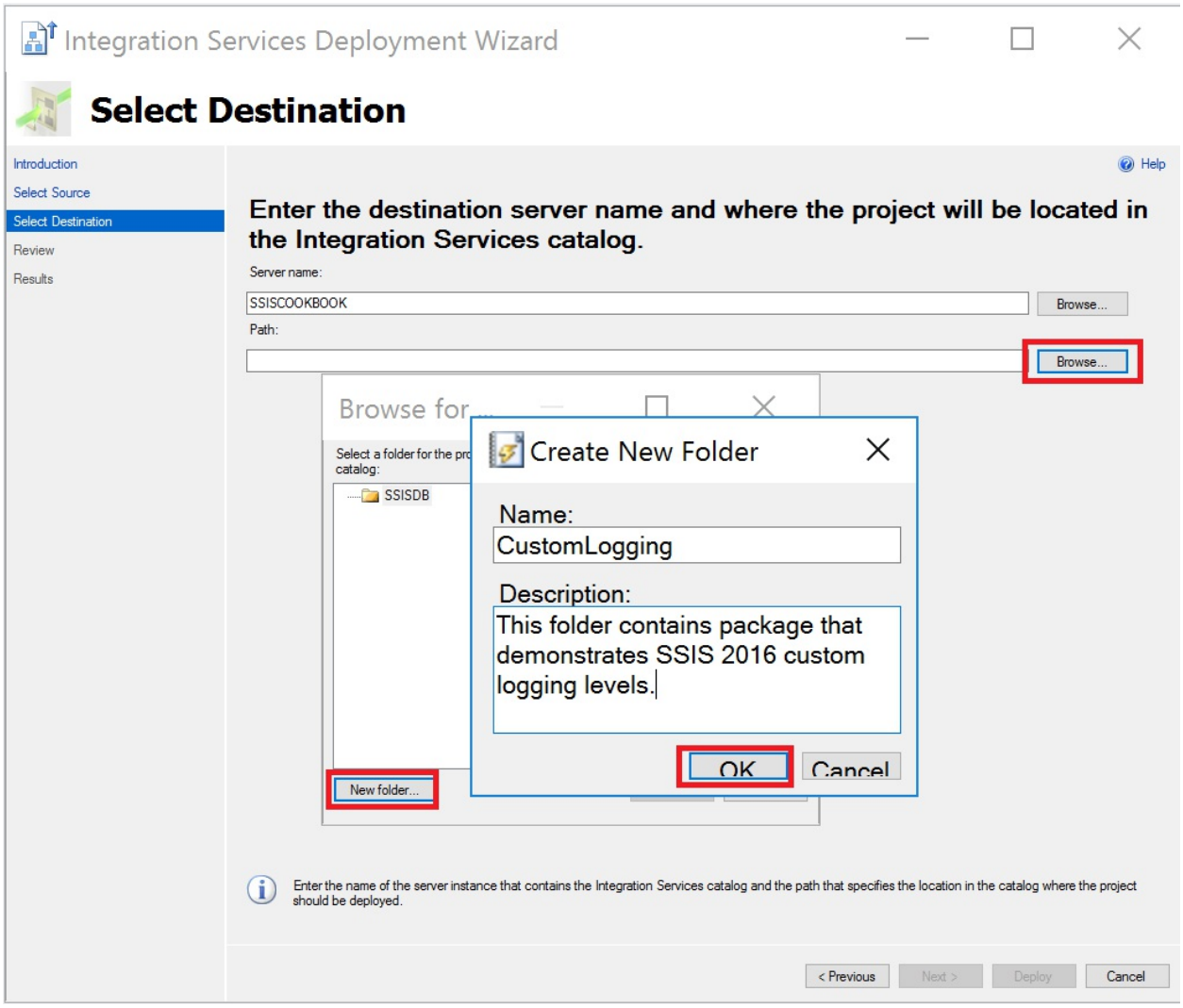
24. The wizard is now asking for a SSIS catalog and folder. Since we don't have any folder yet in the catalog, we'll create one called CustomLogging. As shown in the following screens:

1. Select the **`Server name`** by clicking on the **`Browse`** button. Select your machine name from the list and click on OK.

2. The **`Path`** property specifies where the project will be deployed in the SSIS catalog. To assign a path to the project deployment, click on **`Browse`** at the right of it. The **`Browse for Folder or Project`** window appears. We're going to create a folder for our project. Click on **`New folder`**. From the

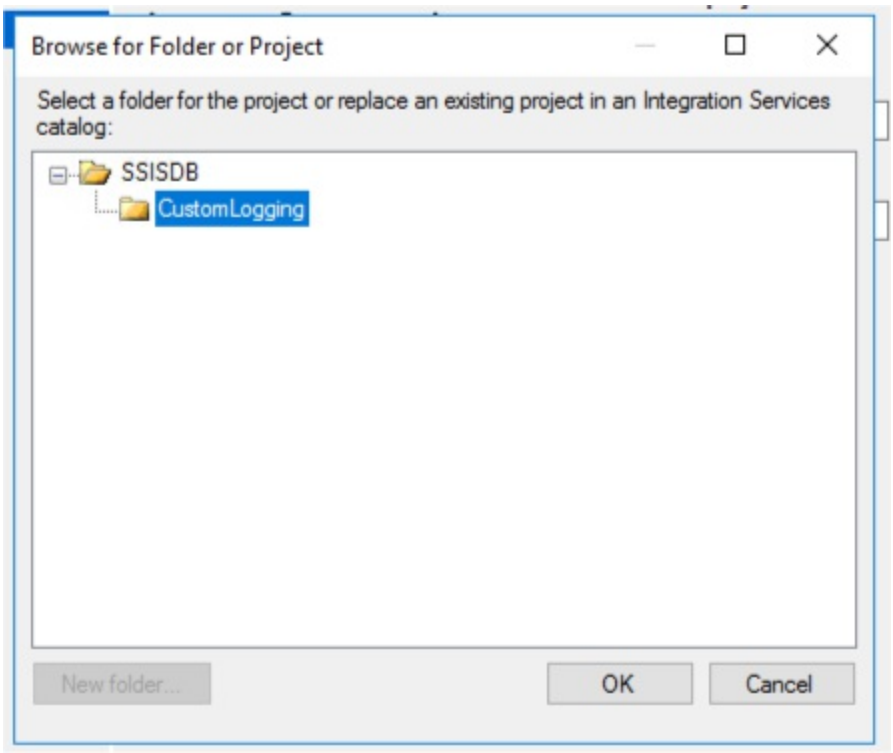
WOW! eBook
www.wowebook.org

`Create New Folder` window that appears, fill the text boxes as shown in the following screenshot. **`Click`** OK to save and close the window.



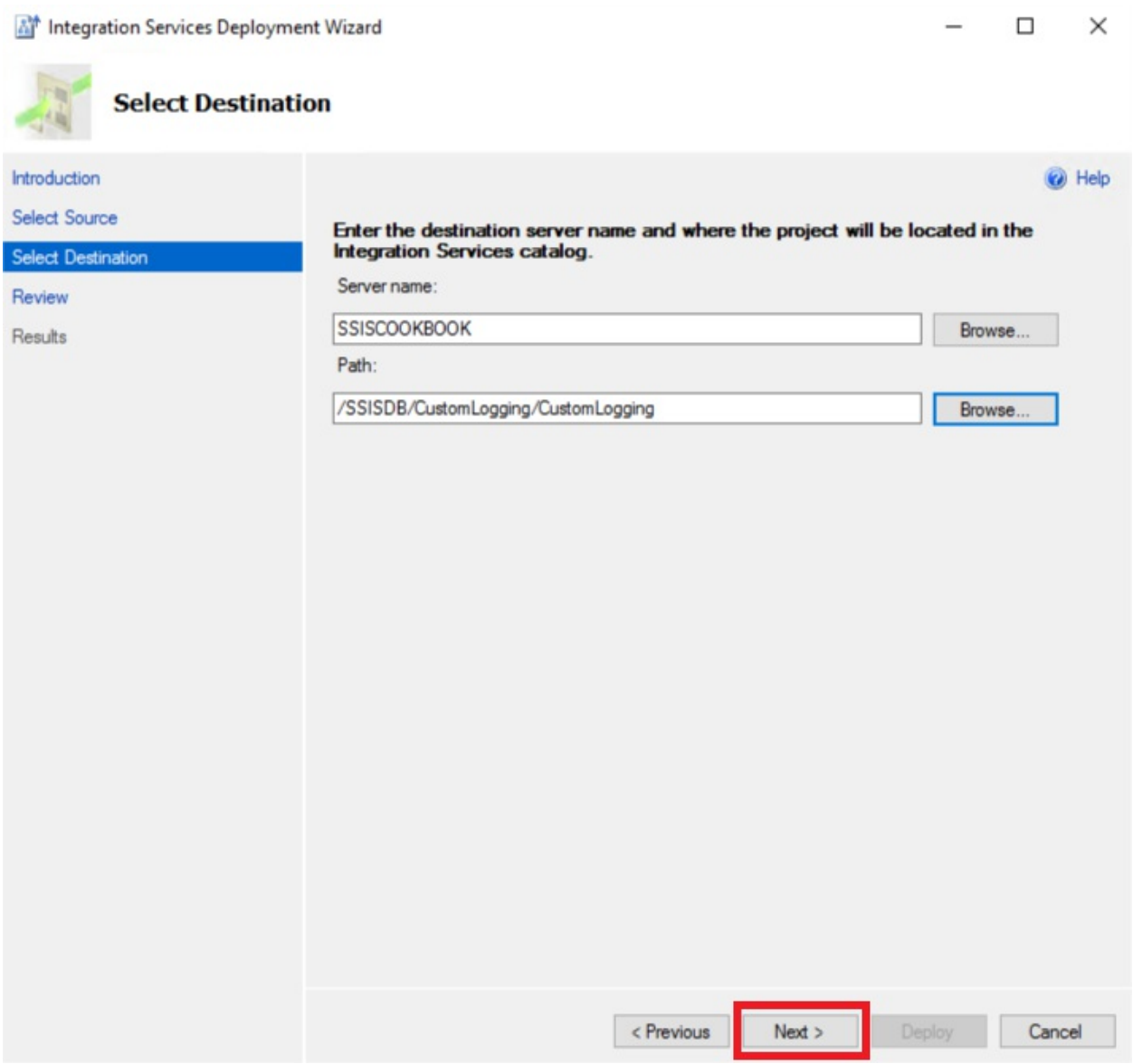
25. Your screen will look like the following screenshot. Click **`OK`** again to close the **`Browse for Folder or Project`** window.

WOW! eBook
www.wowebook.org



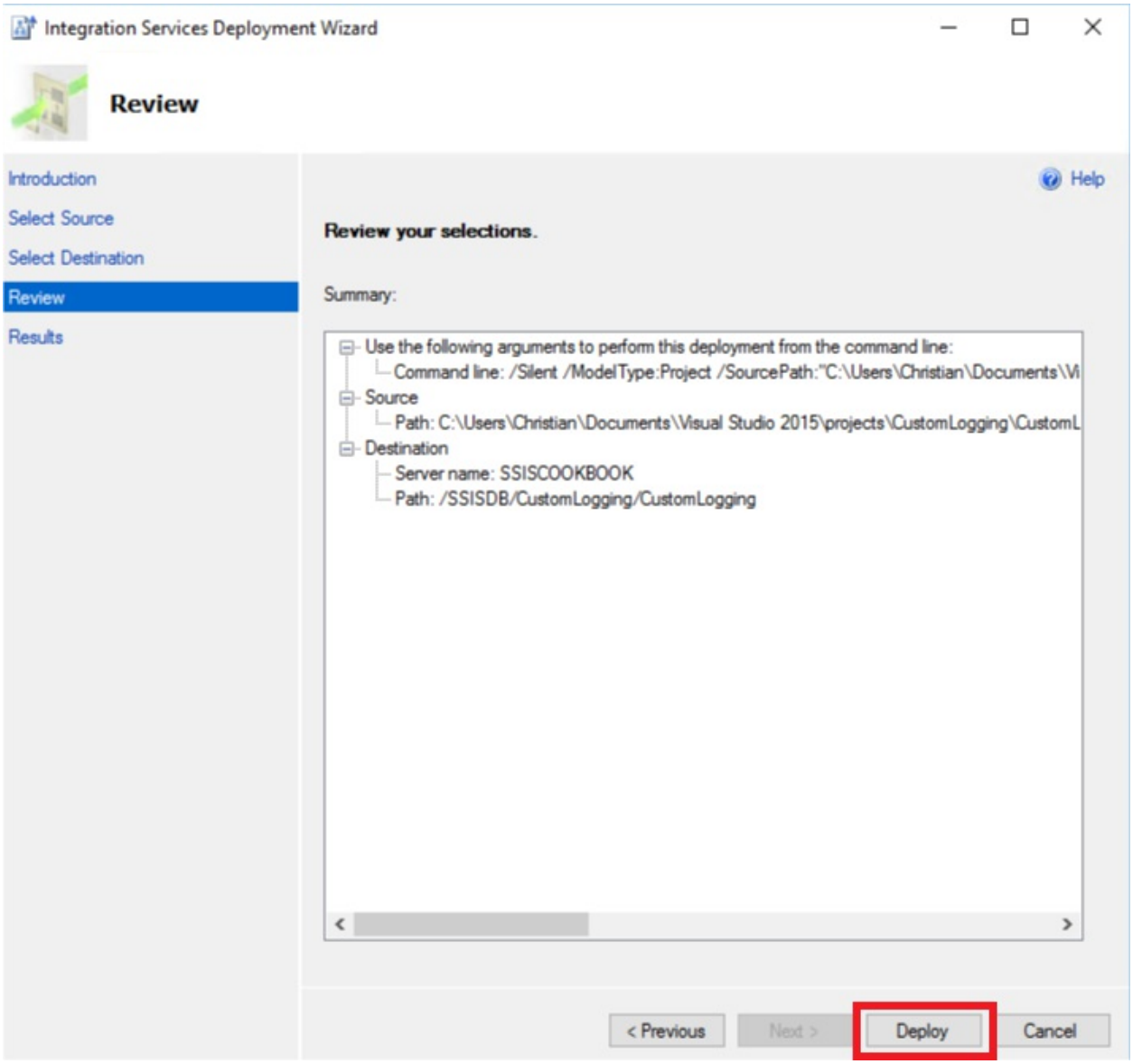
26. Back on the **`Integration Services Deployment Wizard`**, click on the **`Next`** button to go to the next deployment step. You should have a window similar to the the following screenshot.

WOW! eBook
www.wowebook.org



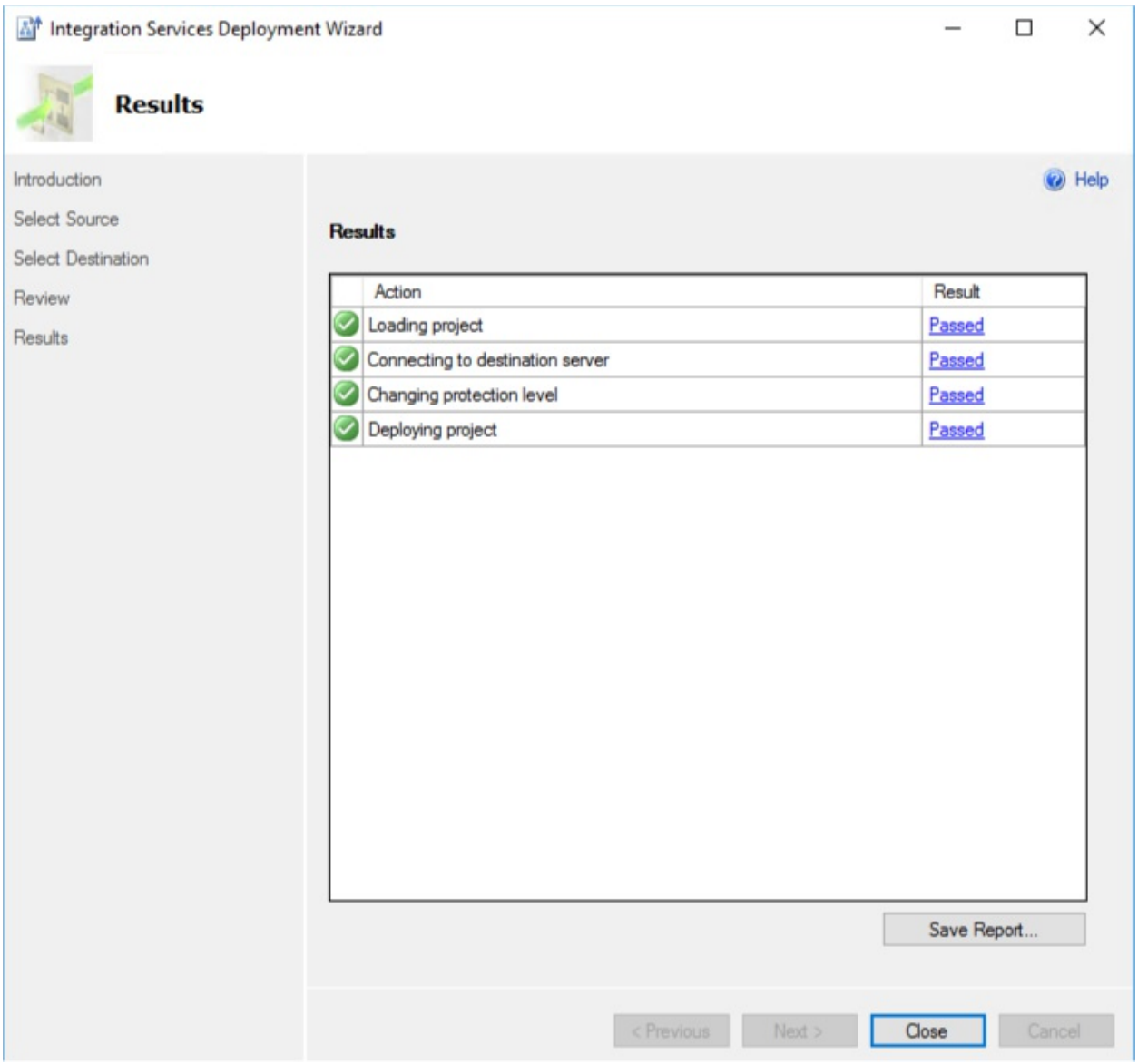
27. Last step, click on Deploy as shown in the following screenshot.

WOW! eBook
www.wowebook.org



28. As the following screenshot shows, the project is deployed in the SSIS catalog. If this fails, click on the **`Report`** button to investigate on the error details. Whenever deployment errors occur, you can click on previous to make a correction. Click **`Close`** to terminate the deployment wizard.

WOW! eBook
www.wowebook.org

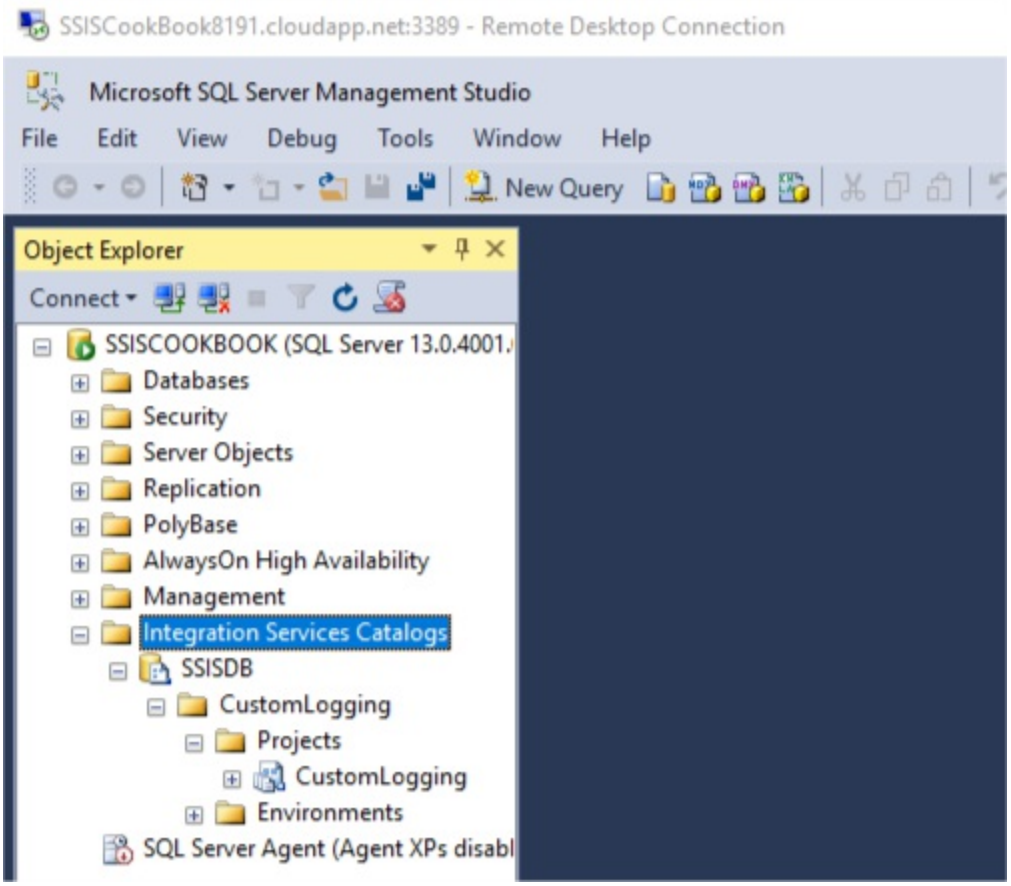


Testing the Custom Logging level

This part of the recipe will guide you through the steps to execute our sample package on the server (SSIS catalog on the local machine). We will use the custom logging level that we created previously in the recipe.

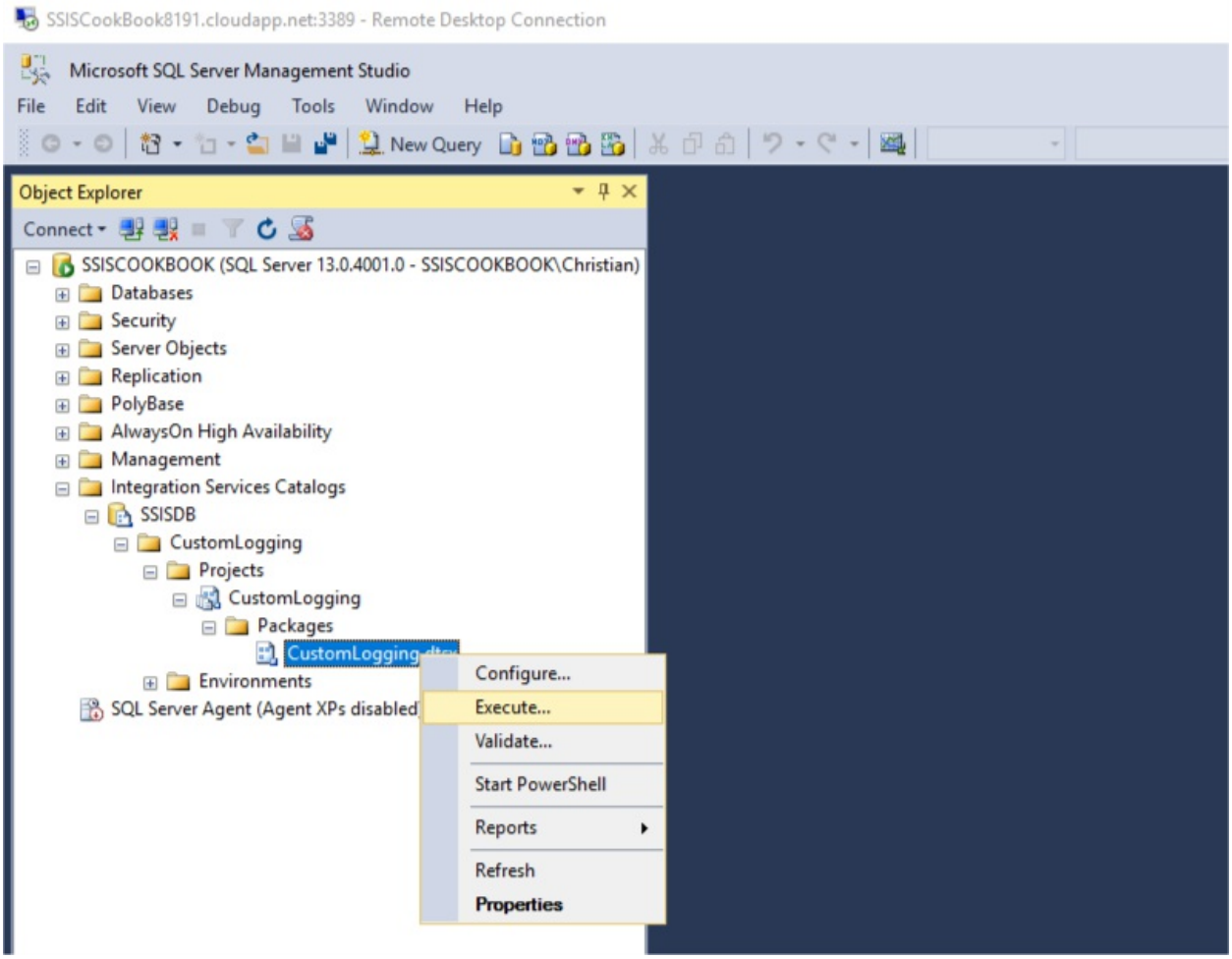
29. Open SSMS and expand the Integration Services Catalog as shown in the following screenshot. Expand the CustomLogging folder and navigate to the package as in the following screenshot.

WOW! eBook
www.wowebook.org



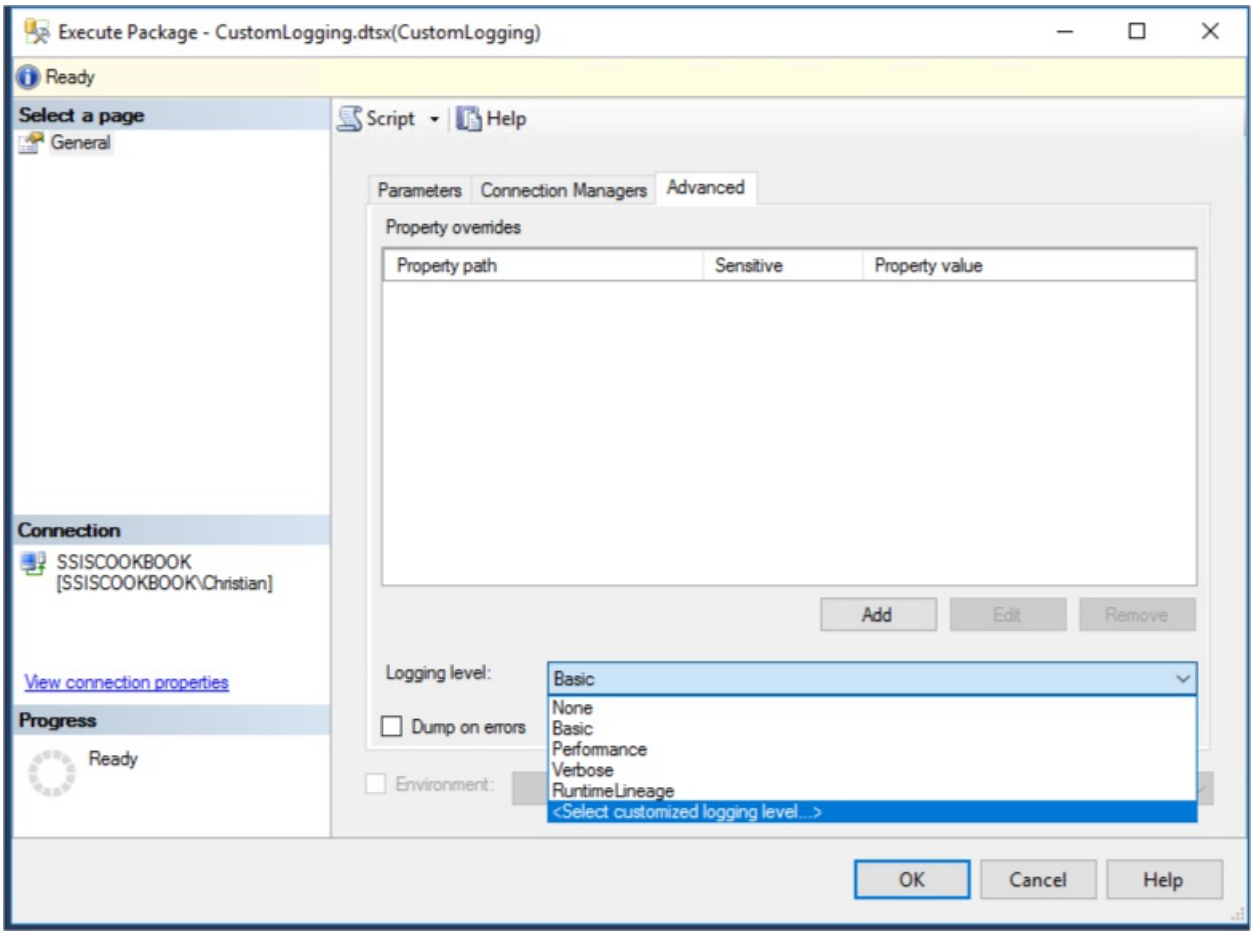
30. When you expand the SSISDB catalog, you'll notice that there is a folder called CustomLogging; it has been created previously in this recipe when we deployed the SSIS project

WOW! eBook
www.wowebook.org



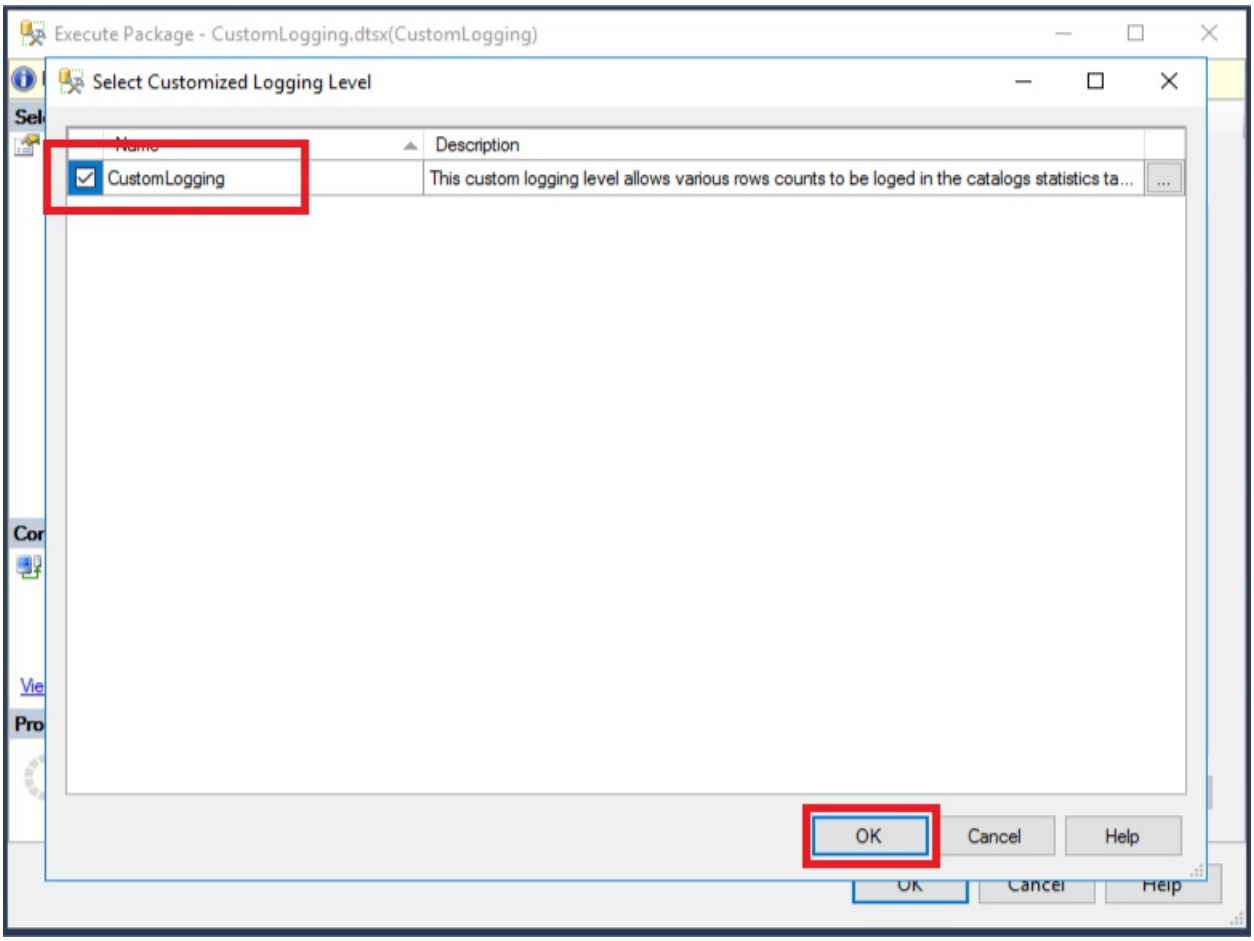
31. Click on the drop down list near the **`Logging level`** property and select **`< Select customized logging level... >`** as shown in the following screenshot.

WOW! eBook
www.wowebook.org



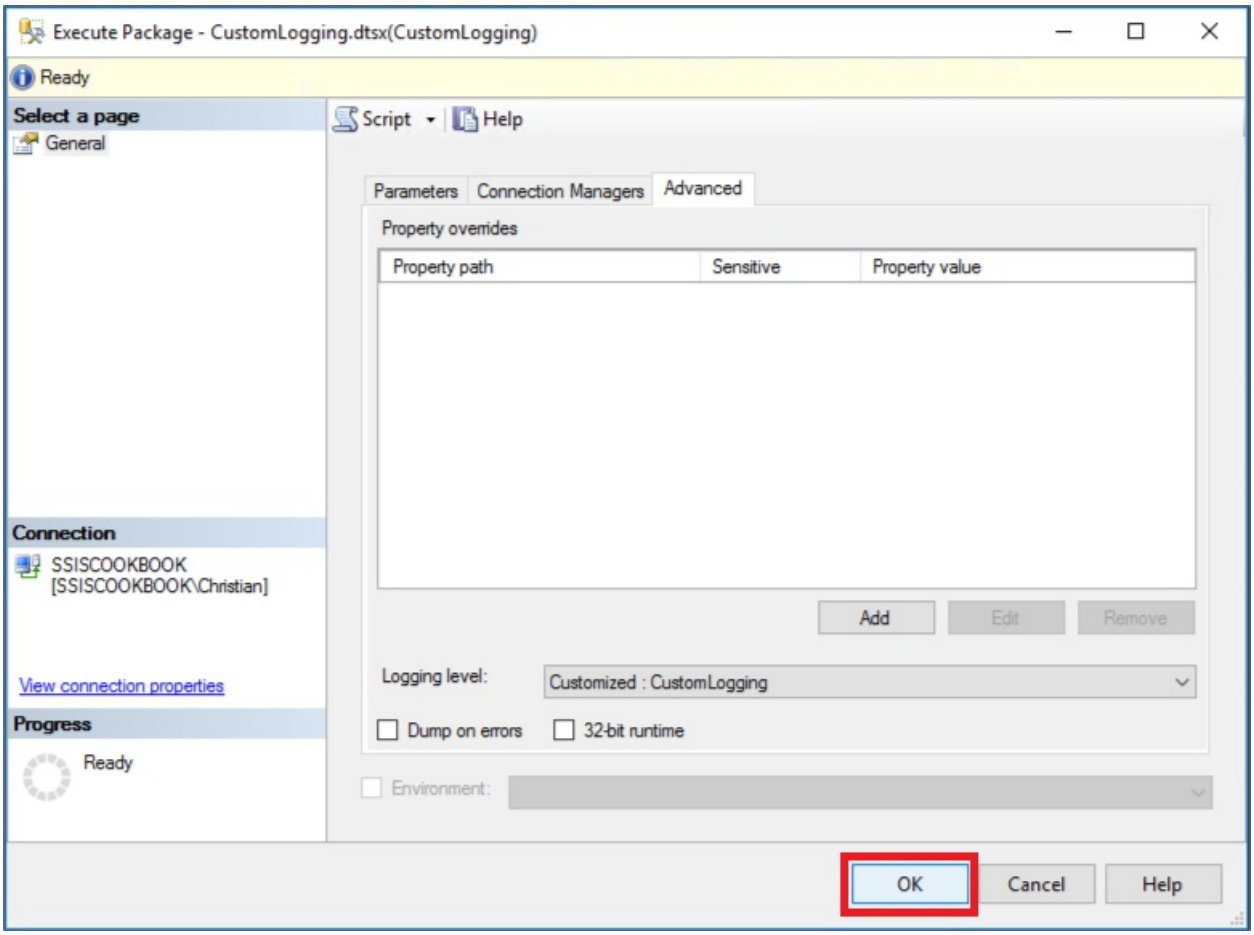
32. Make sure that our custom logging level **`CustomLogging`** is selected and click **`OK`** as shown in the following image.

WOW! eBook
www.wowebook.org

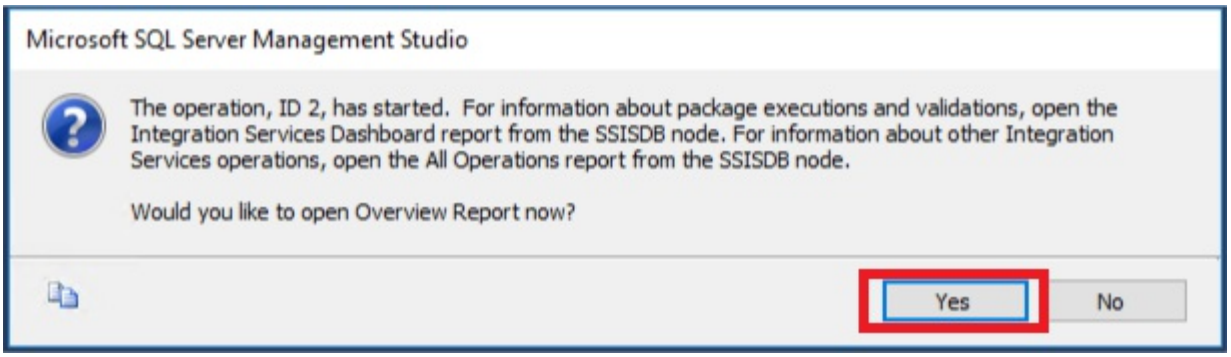


33. Make sure that the **`Logging level`** property is set to Customized: **`CustomLogging`** and click on **`OK`** to close the window and start package execution as in the following screenshot.

WOW! eBook
www.wowebook.org



34. The following screen appears, click on **`Yes`** to see the execution report.



35. The execution report tells us that the package executed properly as shown in the following screenshot.

WOW! eBook
www.wowebook.org

Overview - 11/30...PM - SSISCOOKBOOK

SQL Server

Overview

on SSISCOOKBOOK at 11/30/2016 8:57:14 PM

This report provides an overview of the package tasks and parameters, including execution or validation information.

View Messages

View Performance

Execution Information

Operation ID	2
Package	CustomLogging\CustomLogging\CustomLogging.dtsx
Environment	-
Status	Succeeded
Machine	SSISCOOKBOOK

Duration (sec)	6.125
Start Time	11/30/2016 8:56:34 PM
End Time	11/30/2016 8:56:40 PM
Caller	SSISCOOKBOOK\Christian

Execution Overview

Filter: Result: All; (3 more)

Result	Duration (sec)	Package Name	Task Name	Execution Path

Parameters Used

Name	Value	Data Type
CALLER_INFO		String
CUSTOMIZED_LOGGING_LEVEL	CustomLogging	String
DUMP_EVENT_CODE	0	String
DUMP_ON_ERROR	False	Boolean
DUMP_ON_EVENT	False	Boolean
LOGGING_LEVEL	100	Int32
SYNCHRONIZED	False	Boolean

Property Overrides

Property Path	Property Value

1 / 1

36. Click on New Query in SSMS and type the following query. Click on Execute or F5 to execute the query. You should see that the package read and inserted two (2) rows.

WOW! eBook
www.wowebook.org

GetRowCounts.sql -...OK\Christian (145))

```sql
USE SSISDB
GO

DECLARE @RowcountExecution_id bigint = (SELECT MAX(execution_id) FROM [catalog] . [execution_data_statistics]);


SELECT
 @RowcountExecution_id AS SSISExecutionID
 , [package_name]
 , [dataflow_path_id_string]
 , [source_component_name]
 , [destination_component_name]
 , SUM( [rows_sent]) as [Rows]
FROM [catalog].[execution_data_statistics]
WHERE execution_id = @RowCountExecution_id
GROUP BY [package_name]
       , [dataflow_path_id_string]
       , [source_component_name]
       , [destination_component_name]
```

100 %

Results | Messages

	SSISExecutionID	package_name	dataflow_path_id_string	source_component_name	destination_component_name	Rows
1	3	CustomLogging.dtsx	Paths[ole_src_SELECT_1.OLE DB Source Output]	ole_src_SELECT_1	ole_dst_CustomLogging	2

See also

We'll talk more about logging in the next chapter where we'll talk about an SSIS framework. In the meantime, this recipe introduced you to logging in SSIS but there's much more to be covered later in the book.

WOW! eBook
www.wowebook.org

Azure Tasks and Transforms

This section will guide you on how to install Azure Feature Pack that in turn, will install Azure control flow task and data flow components. Azure ecosystem is becoming predominant in Microsoft ecosystem and SSIS has not been left over in the past few years.

The Azure Feature pack is not a SSIS 2016 specific feature. It's also available for SSIS version 2012 and 2014. It's worth mentioning that it appeared in July 2015, a few months before SSIS 2016 release.

Getting ready

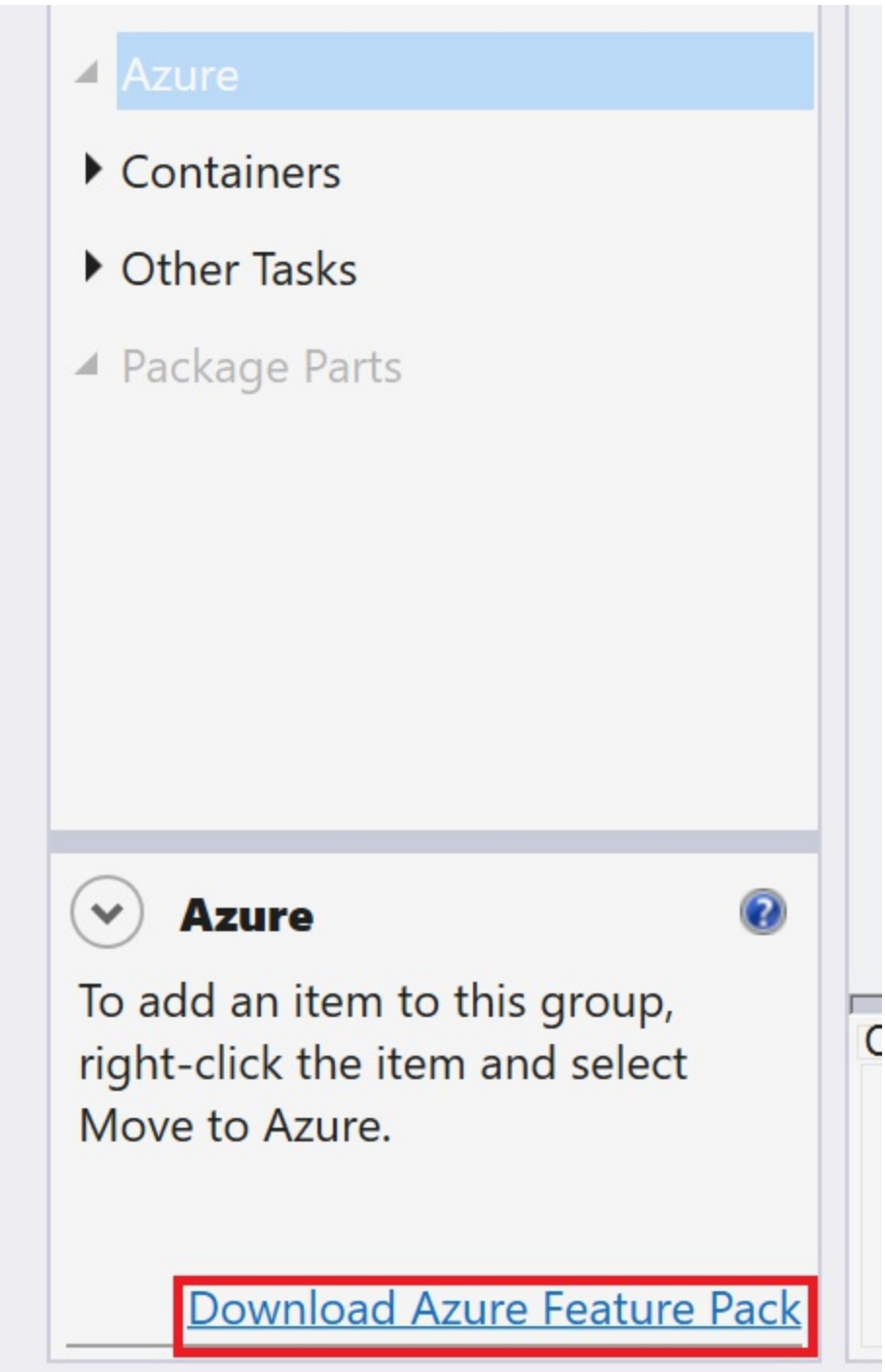
This section assumes that you have installed SQL Server Data Tools 2015.

How to do it

We'll start SQL Server Data Tools, and open the CustomLogging project if not already done.

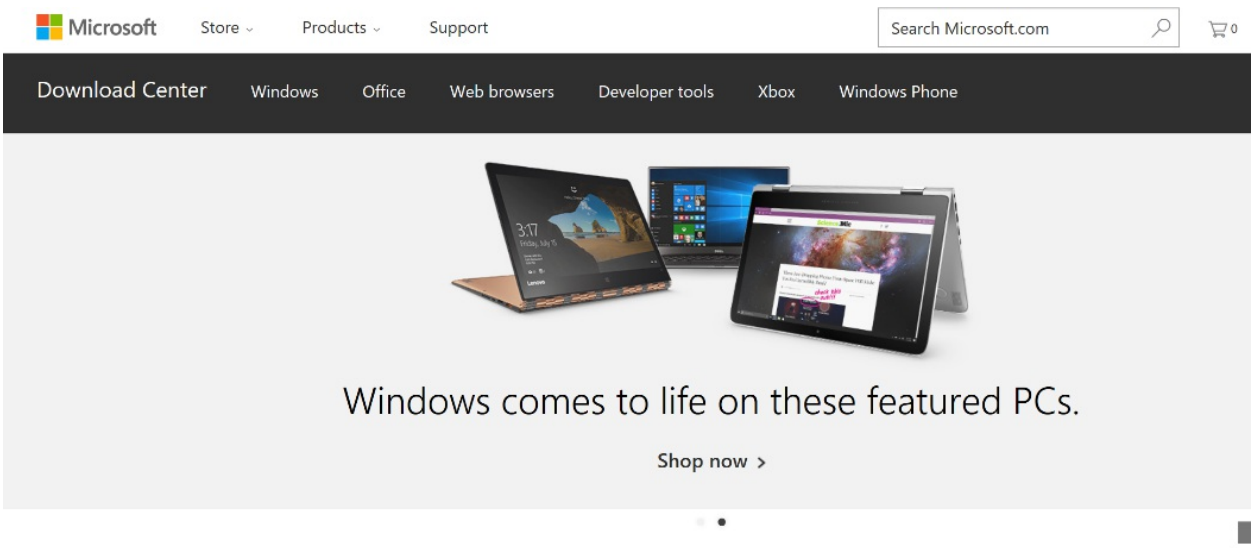
1. In the SSIS Toolbox, scroll to the **`Azure`** group. Since the Azure tools are not installed with SSDT, the **`Azure`** group is disabled in the toolbox. The tools must be downloaded using a separate installer. Click on **`Azure`** group to expand it and click on **`Download Azure Feature Pack`** as shown in the following screenshot.

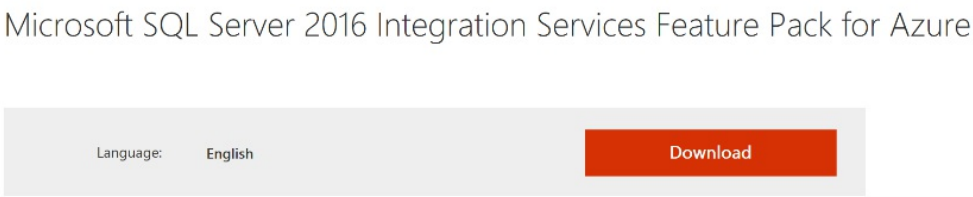
WOW! eBook
www.wowebook.org



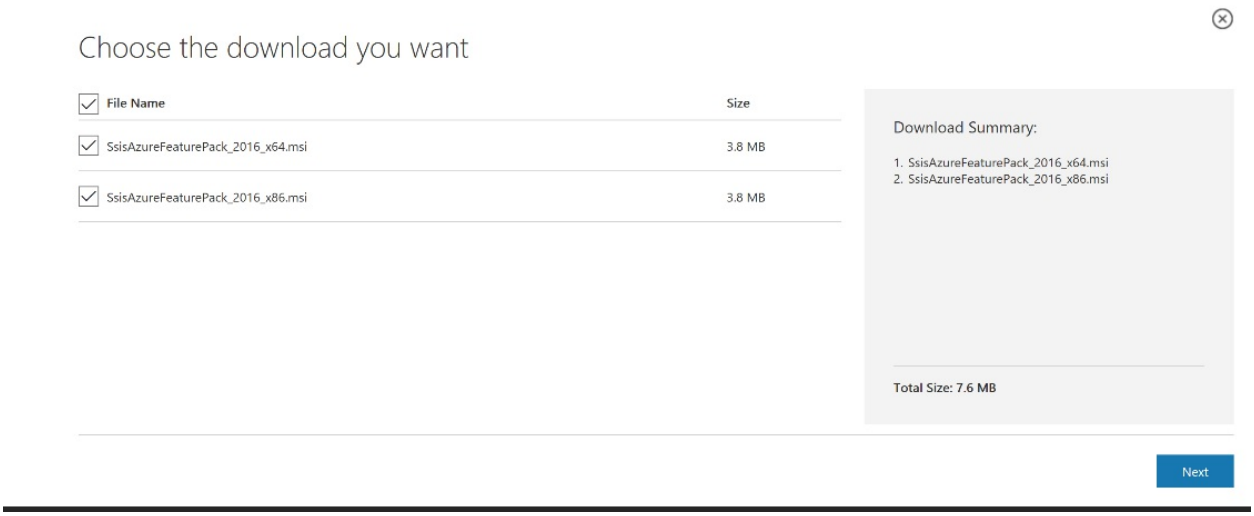
2. Your default browser opens and the **`Microsoft SQL Server 2016 Integration Services Feature Pack for Azure`** opens. Click on **`Download`** as shown in the following screenshot.

WOW! eBook
www.wowebook.org



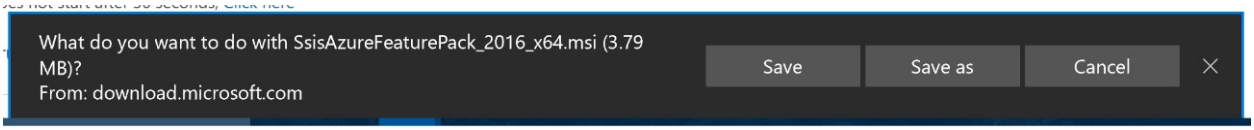


3. From the popup that appears, select both 32 and 64-bit version. The 32-bit version is necessary for SSIS package development since SSDT is a 32-bit program. Click **`Next`** as shown in the following screenshot.

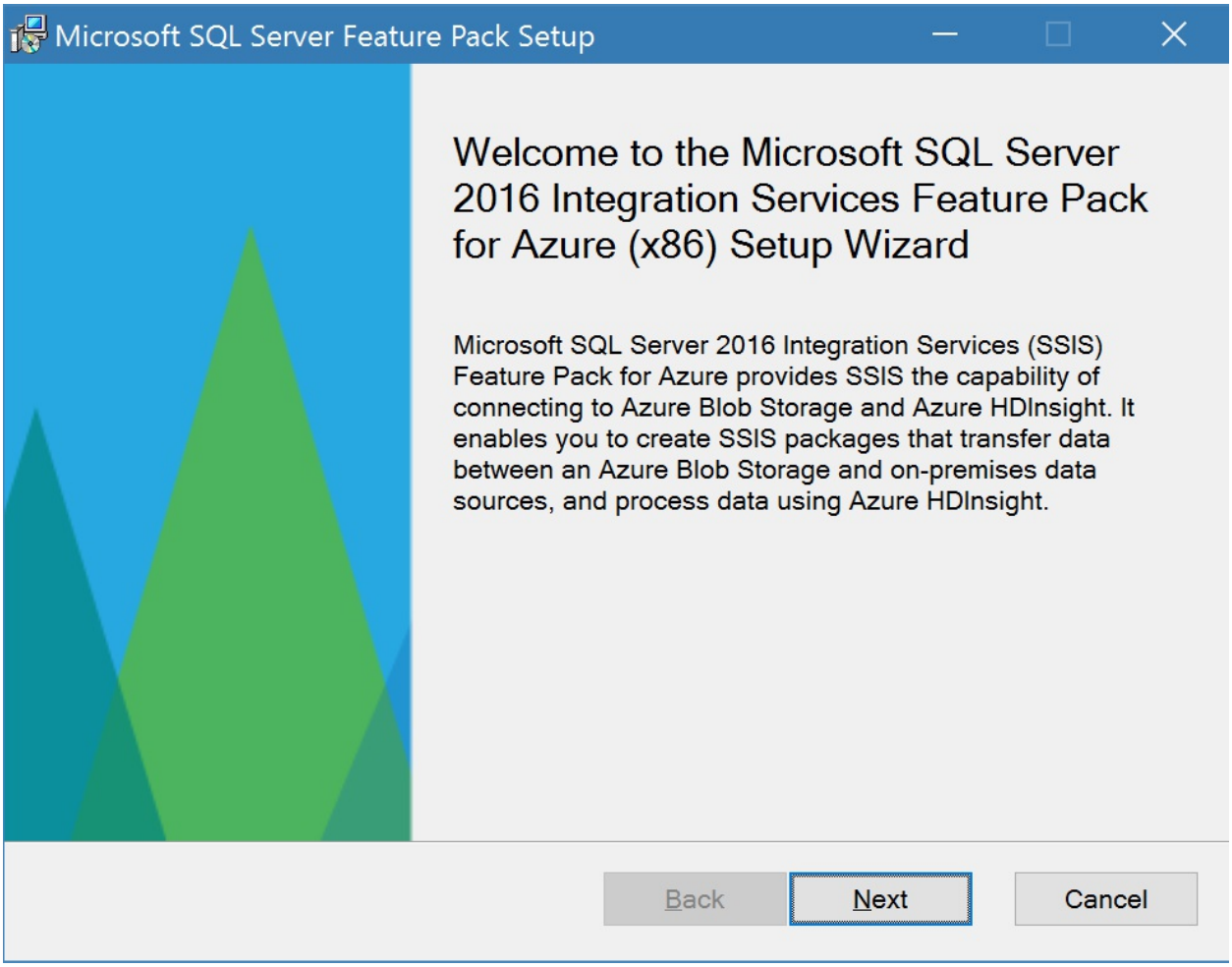


4. As shown in the following screenshot, the files are downloaded.

WOW! eBook
www.wowebook.org

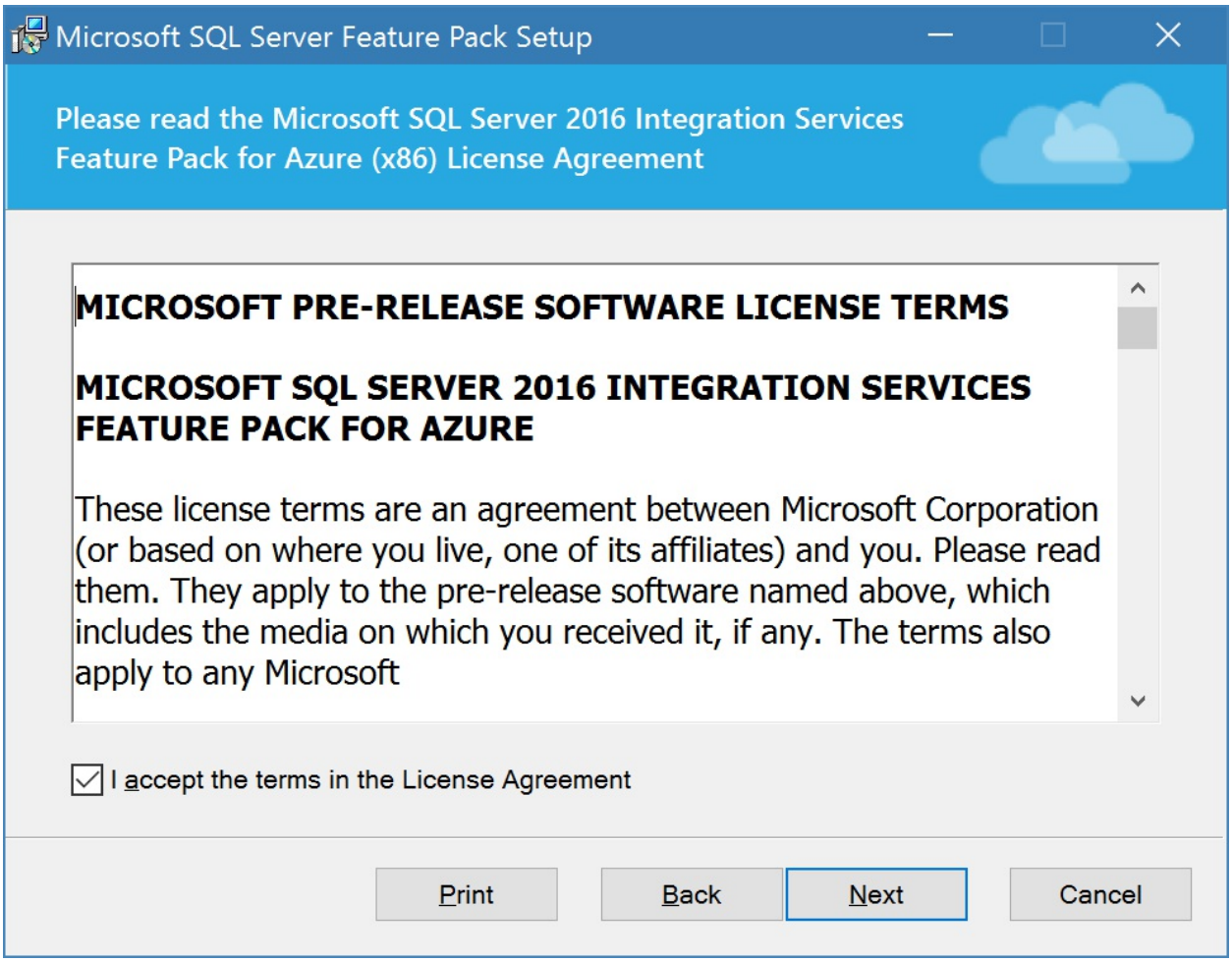


5. Once the download completes, run on one the installers downloaded. The following screen appears. In that case, the 32-bit (**`x86`**) version is being installed. Click **`Next`** to start the installation process.



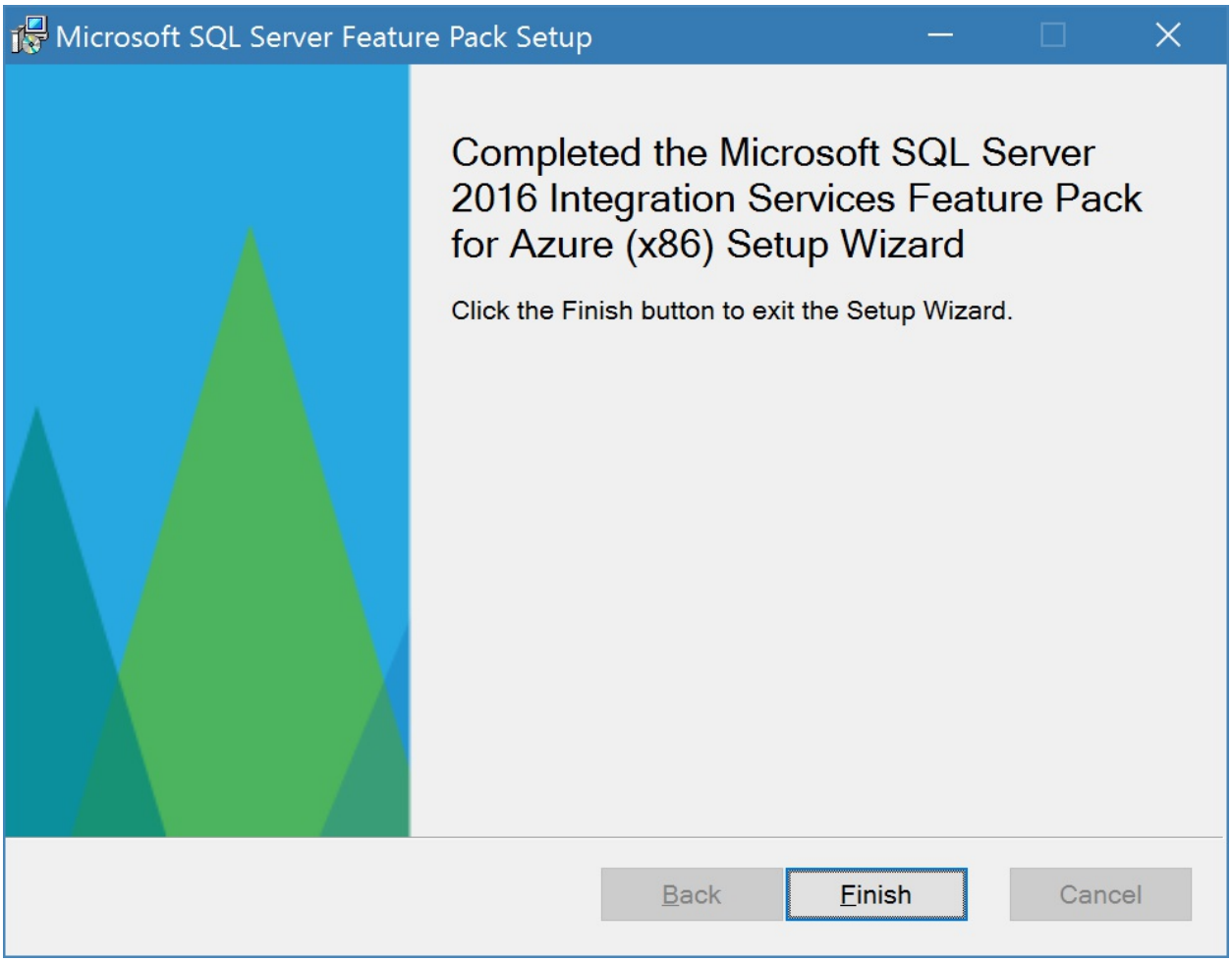
6. As shown in the following screenshot, check the box near **`I accept the terms in the License Agreement`** and click **`Next`**. The installation starts.

WOW! eBook
www.wowebook.org



7. The following screen appears once the installation is completed. Click Finish to close the screen.

WOW! eBook
www.wowebook.org



8. Install the other feature pack you downloaded using previous recipe steps, *Custom Logging*.
9. If SSDT is opened, close it. Start SSDT again and open the `CustomLogging` project we created in another recipe. In the Azure group in the SSIS toolbox, you should now see the Azure tasks as in the following screenshot.

See also

This book contains a dedicated chapter on Azure tasks and transforms, Chapter 9, *Azure Data Integration*. They will be explained in more details with useful examples.

WOW! eBook
www.wowebook.org

Incremental package deployment

Prior to SSIS 2012, packages needed to be deployed one by one. We were usually download all packages from source control software's like Team Foundation Server (TFS), Visual Source Safe, SVN, etc. Once downloaded, packages were moved to their destination. At that time, the person who deployed the packages had the choice to overwrite or skip existing packages. Usually, they overwrote all the packages since they were using source control.

For those who didn't use source control, they had all the necessary flexibility to deploy what needed to be deploy. Usually, they were keeping a backup somewhere on a file share of all packages. The reason why they chose what to deploy was mainly because they had doubts about the consistency of the packages in the file share. They were simply not sure of the state of the packages because they were using a manual process to maintain their solution. Source control software's help a lot with it. We have the possibility to compare versions between package committed in the source control, when and who pushed the package among other benefits.

Arrive SSIS 2012 with Project Deployment. The only way to deploy a package was to deploy the entire project, and thus, all packages in it. If source control was in place to manage package code, this was barely an issue. But, not everybody is using source control and Microsoft recognized it. With SSIS 2016, we can now deploy part of your project packages. You're not forced to deploy the entire project.

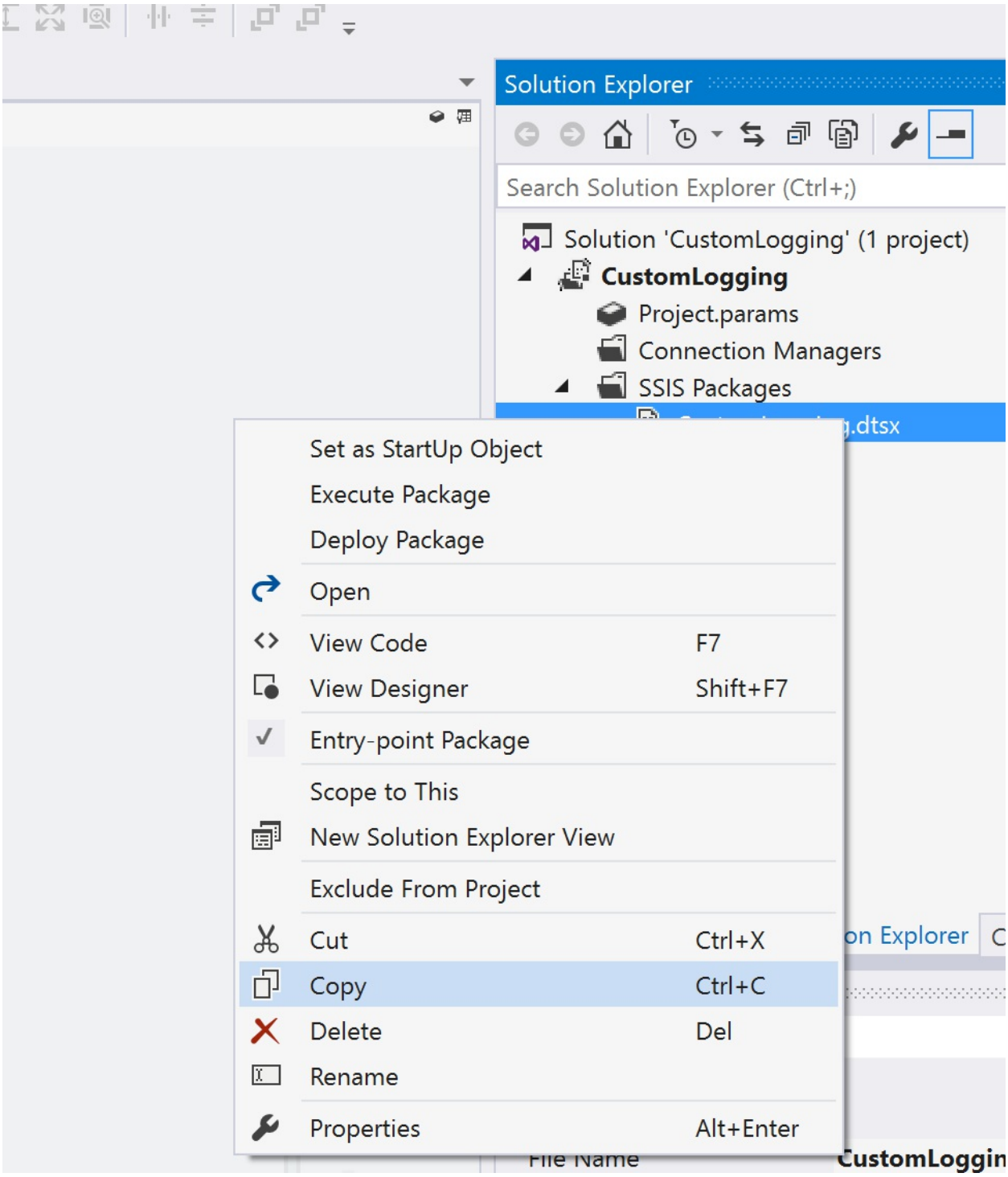
Getting ready

This recipe assumes that you have created and deployed the CustomLogging recipe.

How to do it...

WOW! eBook
www.wowebook.org

1. With the **`CustomLogging`** project opened, right-click on the **`CustomLogging.dtsx`** package and select **`Copy`** as shown in the screenshot below.

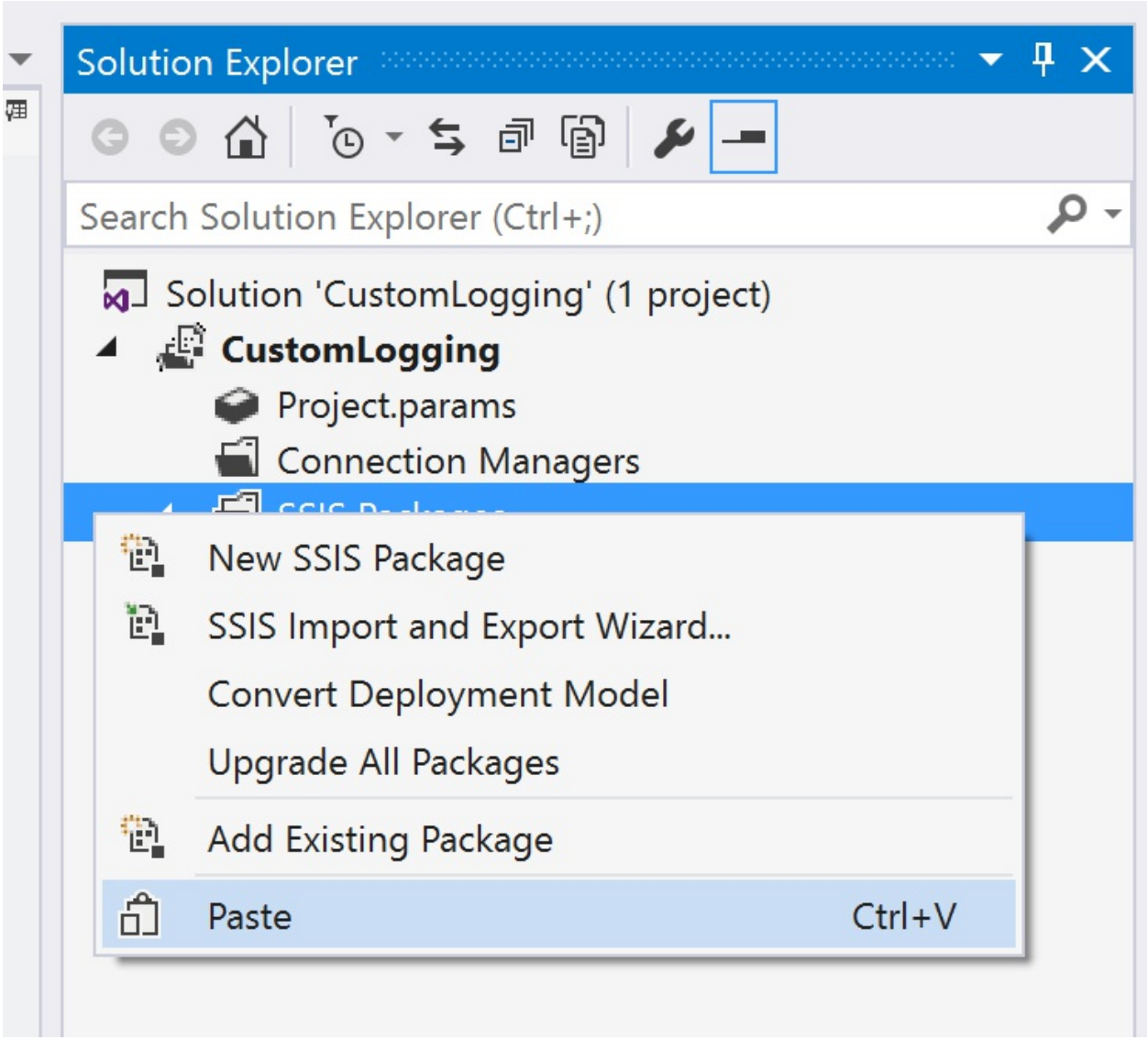


2. Now, right-click on the SSIS Packages folder and select **`Paste`** as shown

WOW! eBook
www.wowebook.org

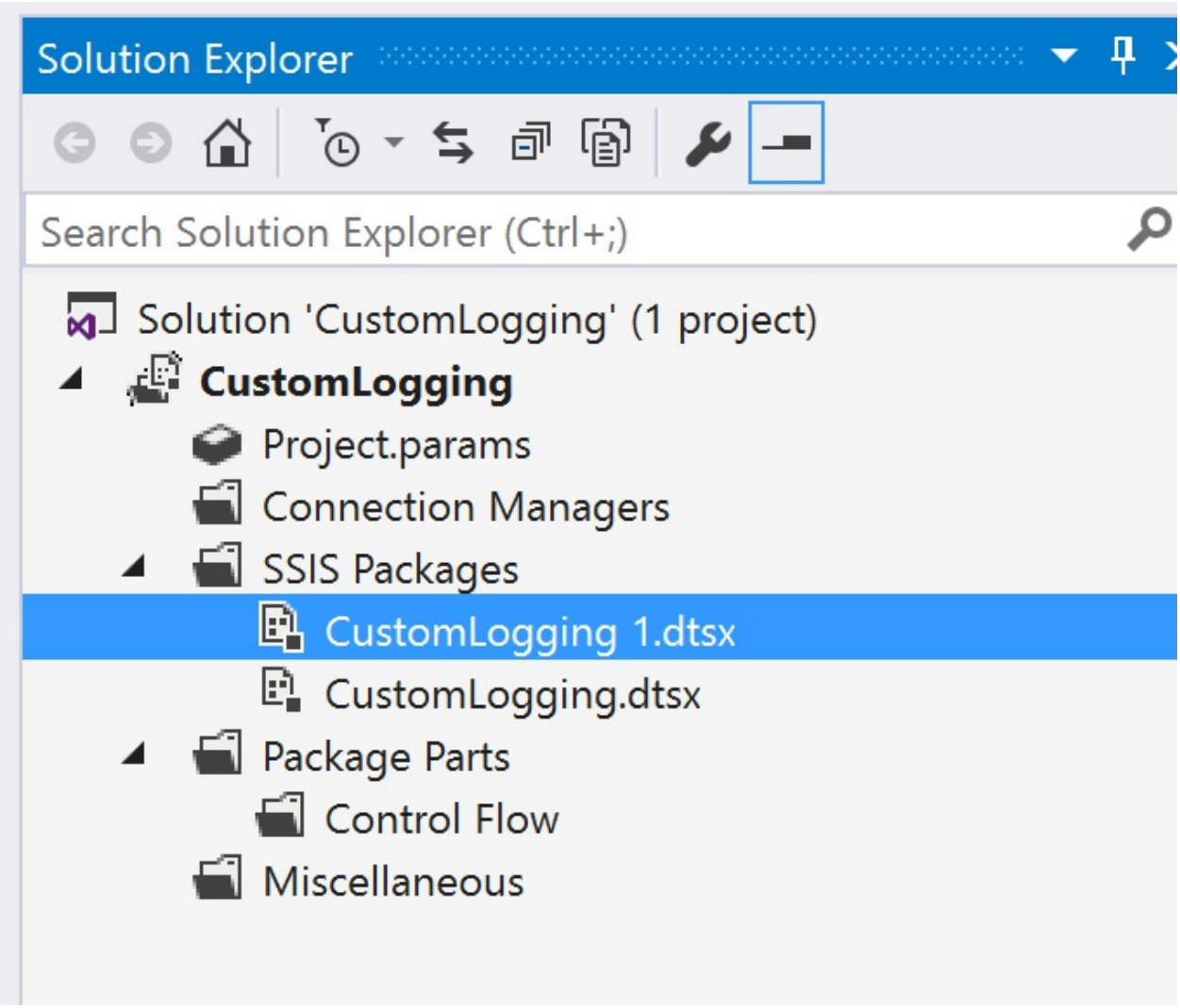
in the following screenshot.

.



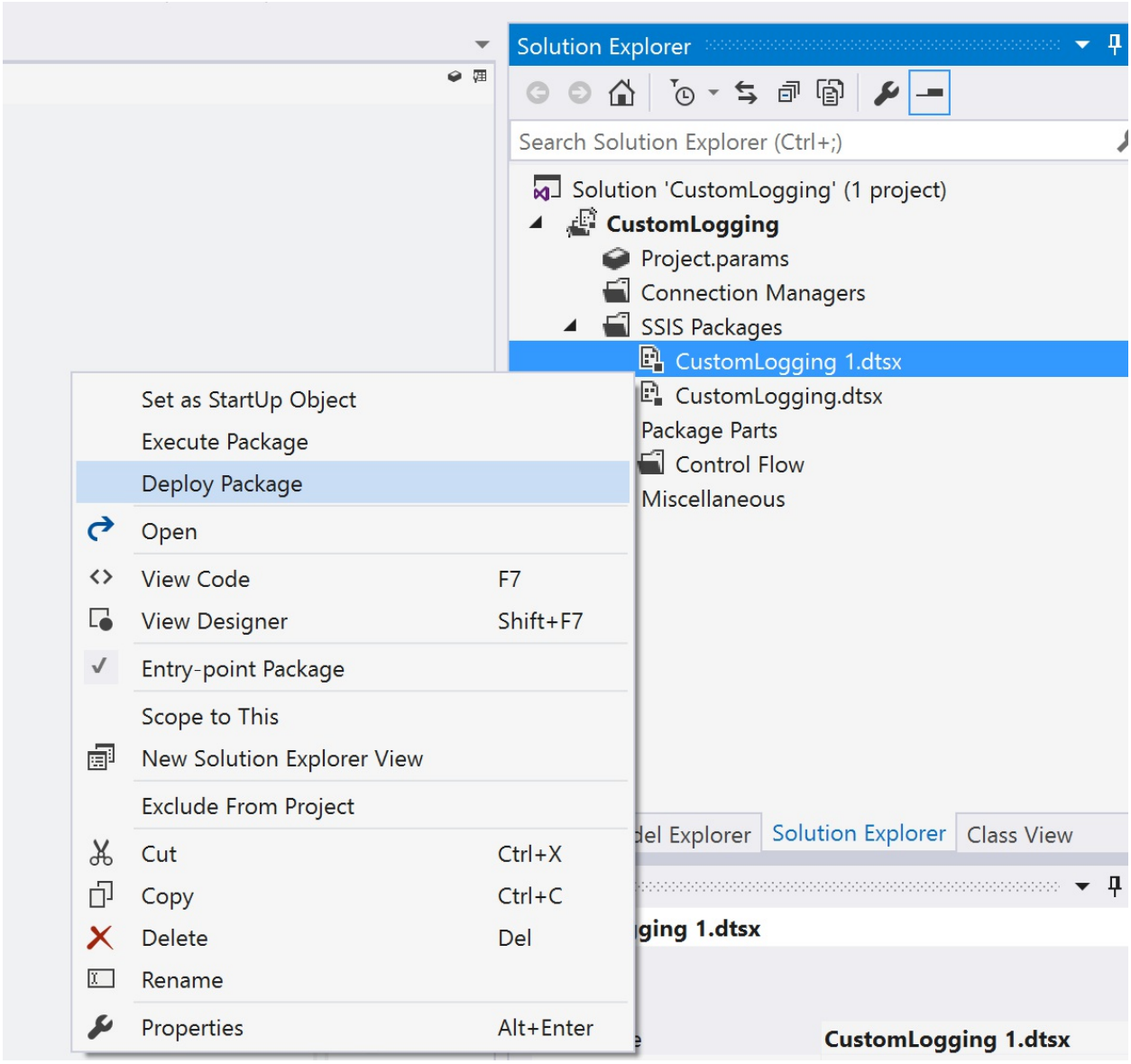
3. A copy of the package is created and named **`CustomLogging 1.dtsx`** as shown in the following screenshot.

WOW! eBook
www.wowebook.org



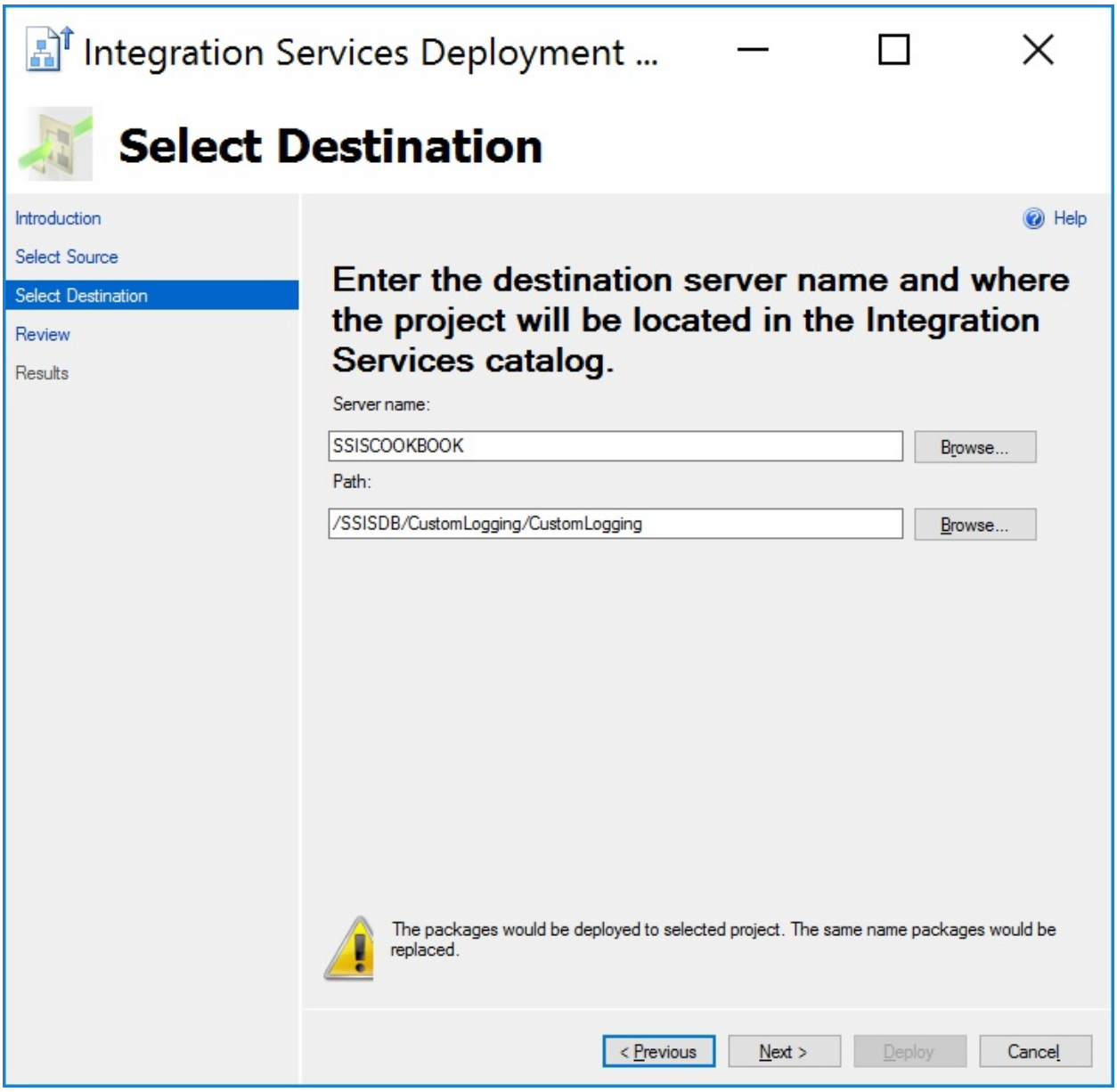
4. Right-click on the newly created package, **`CustomLogging 1.dtsx`** and **`select Deploy Package`** from the menu as shown in the screen as follows.

WOW! eBook
www.wowebook.org



5. As shown in the following screenshot, the **`Server name`** is already selected since we've deployed the project there before. It can be changed if necessary. The **`Path`** is where the deployed package will be located; `SSISDB/CustomLogging/CustomLogging` folder in the SSIS catalog. Click **`Next`** to advance to the next step.

WOW! eBook
www.wowebook.org

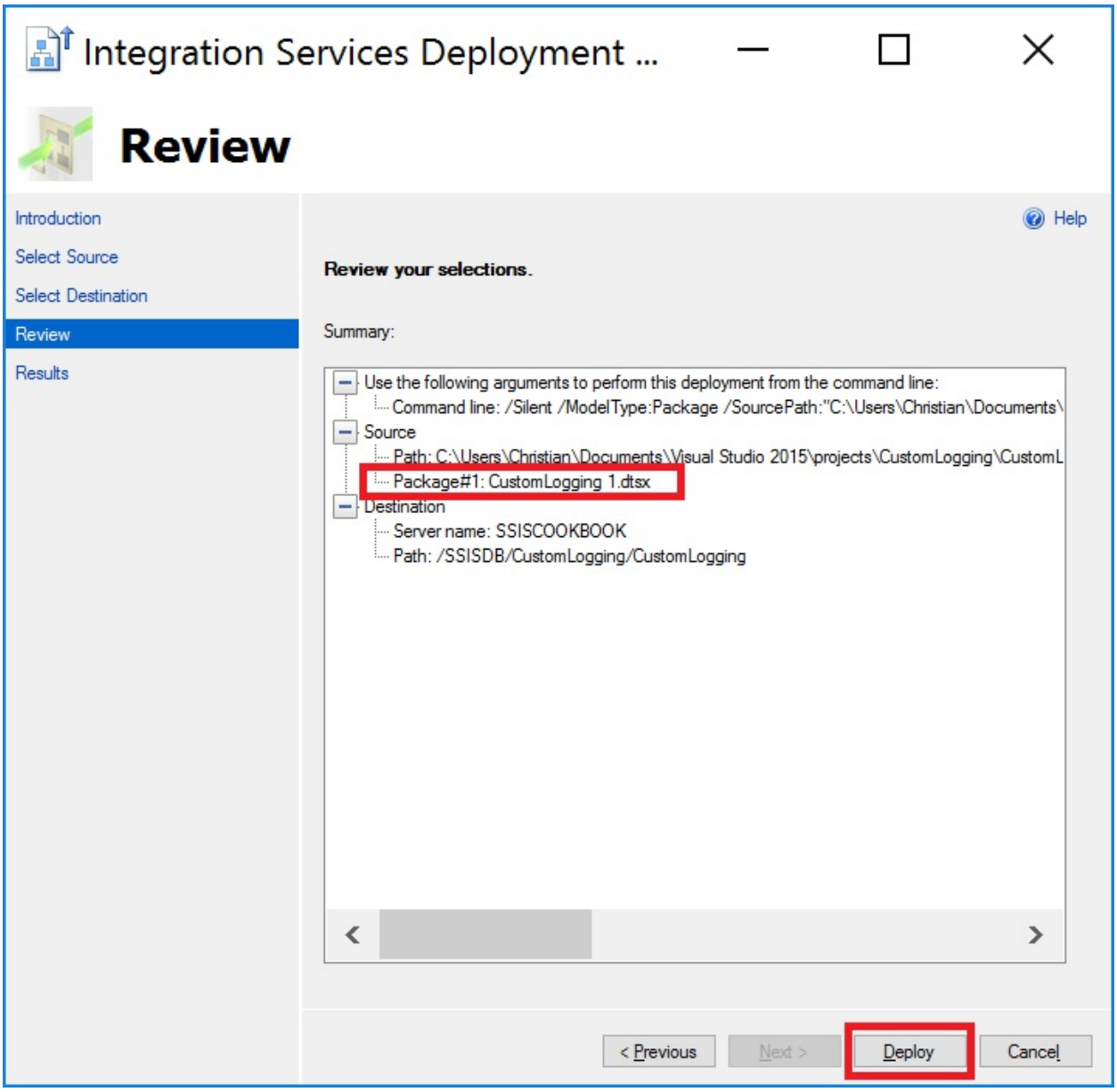


6. Now we are at the Review screen as shown in the following screenshot.

Note

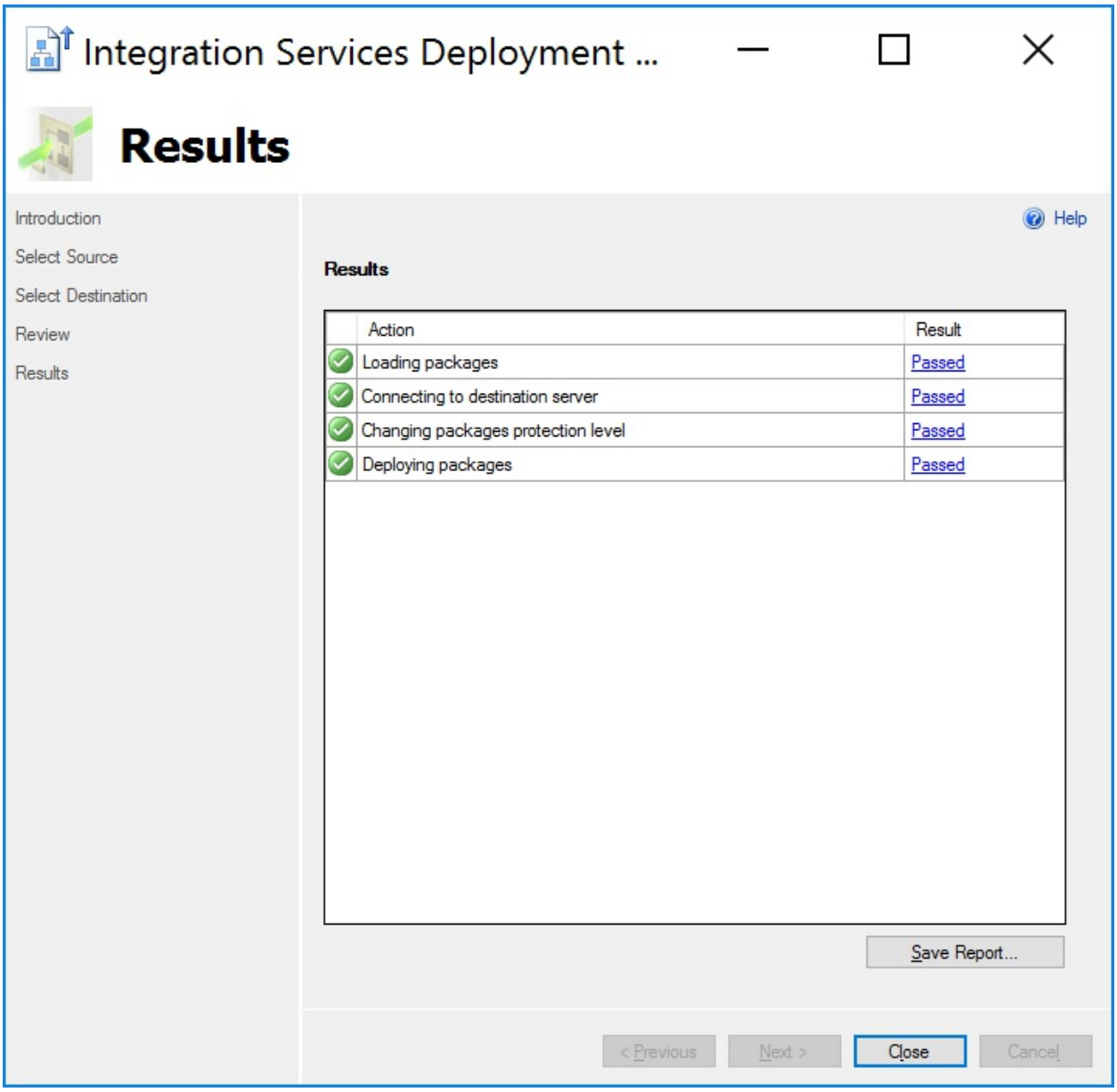
Notice that the package to be deployed is **`CustomLogging 1.dtsx`**. It is tagged as **`Package#1`** since we might have selected multiple packages to deploy. Click **`Deploy`** to proceed to the deployment.

WOW! eBook
www.wowebook.org



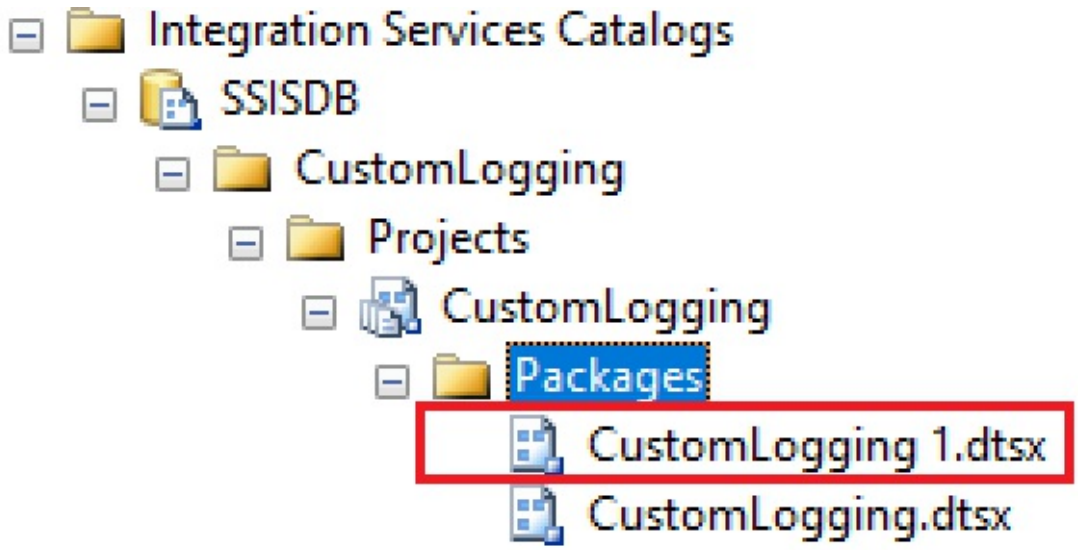
7. If everything goes well, you should get a screen like the following screenshot meaning that the package has been successfully deployed. Click on **`Close`** to get rid of it.

WOW! eBook
www.wowebook.org



8. Now, go to SSMS, open the SSIS catalog as shown in the following screenshot. You will see that the package has been deployed. We won't execute it for now as we only wanted to demonstrate how single package deployment works.

WOW! eBook
www.wowebook.org



There's more...

The capability to deploy individual packages is possible with SSIS 2016 only. If we have tried to deploy to a SSIS catalog of a previous version of SSIS (2012-2014), the deployment would have failed. So, to reinforce the point here, SSDT can show you the option that you can deploy individual packages but you'll need SSIS 2016 to be able to use it.

WOW! eBook
www.wowebook.org

Multiple version support

Since its inception, SSIS designer never supported backward compatibility. For example, if you developed a package in SSIS 2014 and tried to deploy it in a SSIS 2012 catalog, you would not be able to do it. Or worse, if you opened a package developed with SSIS 2012 with a SSDT that was used with SSIS 2014, the package was upgraded. When another developer tried to open it with SSDT used for SSIS 2012, he/she was not able to do it. The package was upgraded, period.

With SSIS 2016, SSDT had the following enhancements:

- Backward compatibility to prior SSIS version down to SSIS 2012;
- Unified SSDT: as we'll see later in the book, SSDT can be used for BI components development as well as database development. Prior SSDT for Visual Studio 2015, it was confusing whether we were using SSDT-BI for BI development or SSDT-SQL for database development.

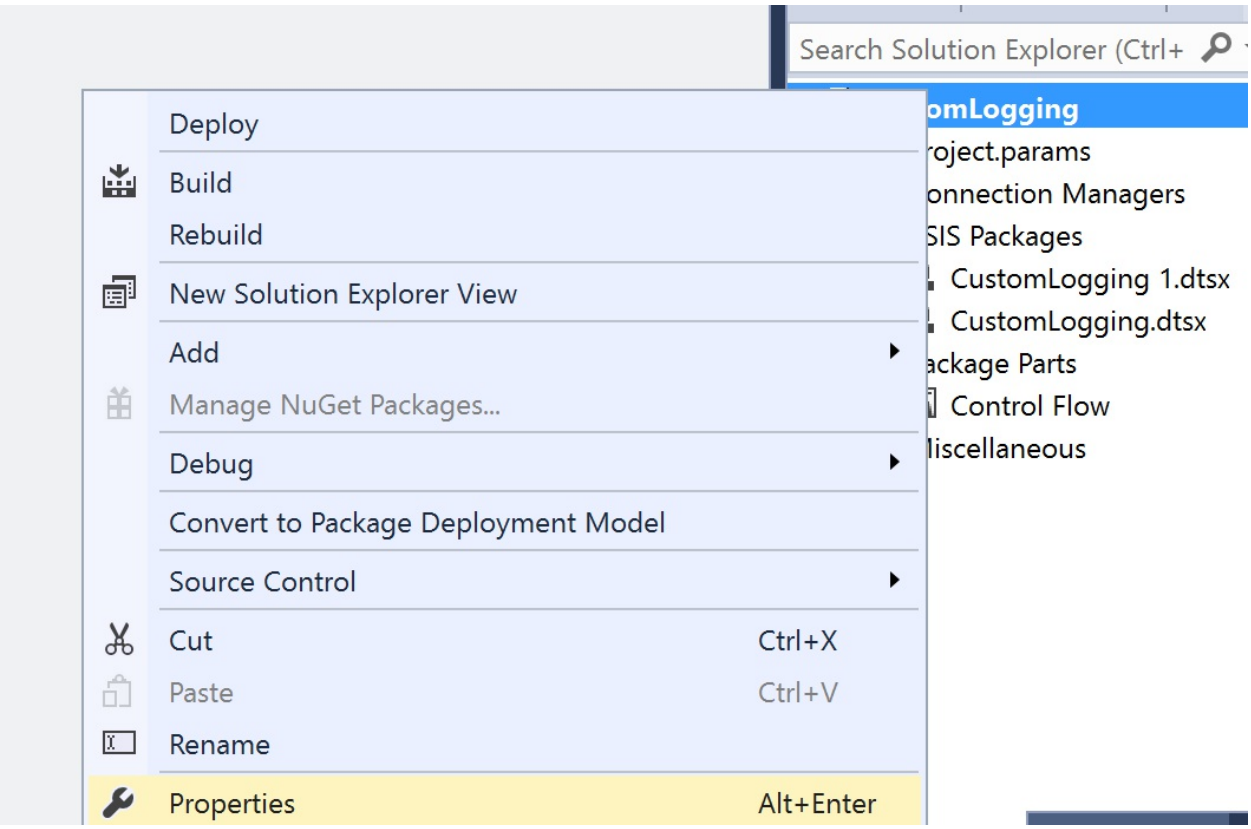
Getting Ready

This recipe assumes that you have created a SSIS project in SSDT. We'll be using the CustomLogging SSIS project developed in a previous recipe in this chapter.

How to do it...

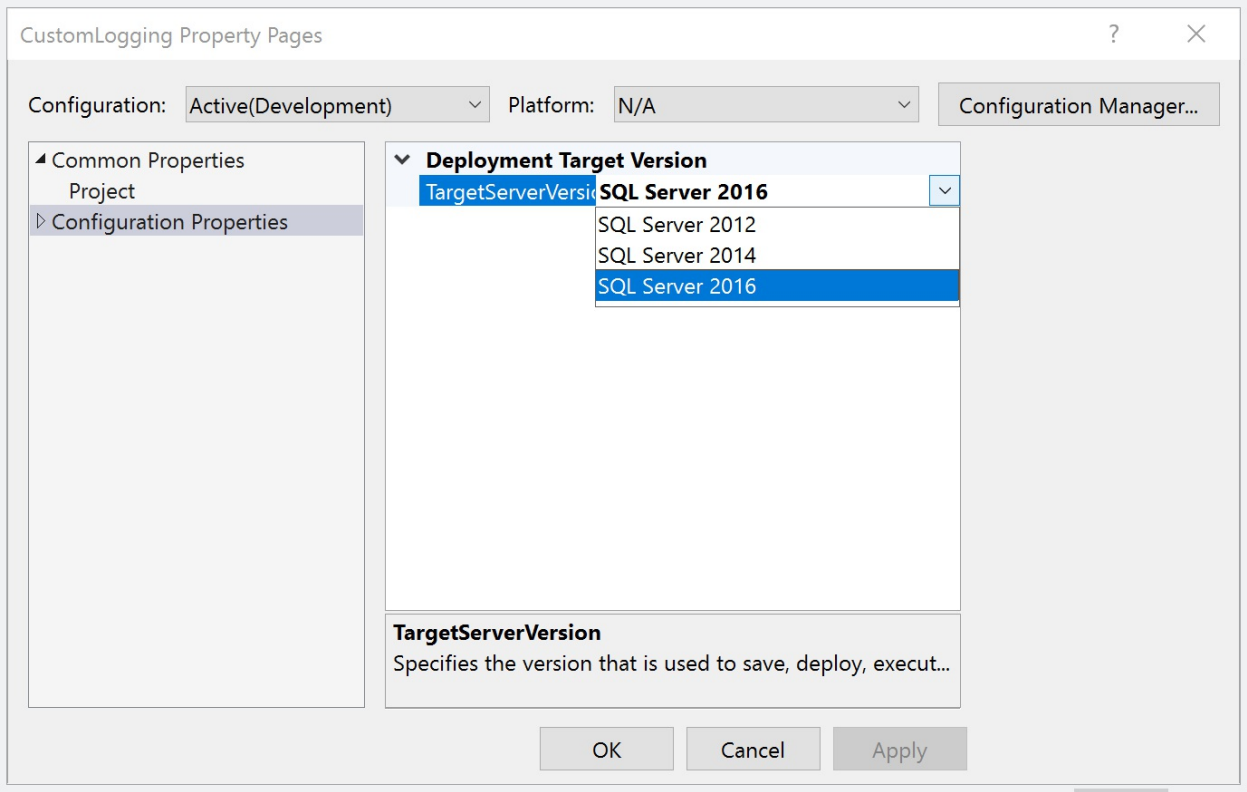
1. With SSDT and CustomLogging SSIS project opened, right-click on the project and select Properties as shown in the screenshot below.

WOW! eBook
www.wowebook.org



2. The **`CustomLoggingProperty Pages`** dialog box opens. Select Configuration Properties and you'll notice a drop down list near the property **`TargetServerVersion`**. In our case, the selected version is **`SQL Server 2016`**. You'll notice that you can select prior versions as shown in the following screenshot. Click on **`OK`** to close this window.

WOW! eBook
www.wowebook.org



There is more...

This simple recipe shows you how easy we can now use SSDT for Visual Studio 2015 with prior SSIS versions. If you open an existing SSIS project that is developed in SQL Server 2012 or 2014 SSDT version, the designer (SSDT) will be smart enough to select the right target version (2012 or 2014) without upgrading your project or packages.

The default Target Server version is SQL Server 2016. If you created your SSIS project without setting this property and try to deploy it on a prior version that 2016, the deployment will fail. Also, if you change this property from e.g. SQL Server 2016 down to SQL Server 2014 and your package uses new SSIS functionality, the project won't build and therefore you will not be able to deploy it.

WOW! eBook
www.wowebook.org

Error Column Name

This recipe will show you a new neat feature of SSIS which the error column name. We could achieve something similar before SSIS 2016 but it involves using a script component and it is difficult to reuse this kind of transform. Although we can copy and paste a script component from one package to another, every time we do so, we must change the input columns and recompile the script. It's not very difficult but it's a tedious task. The error column name native implementation in SSIS 2016 is very welcome.

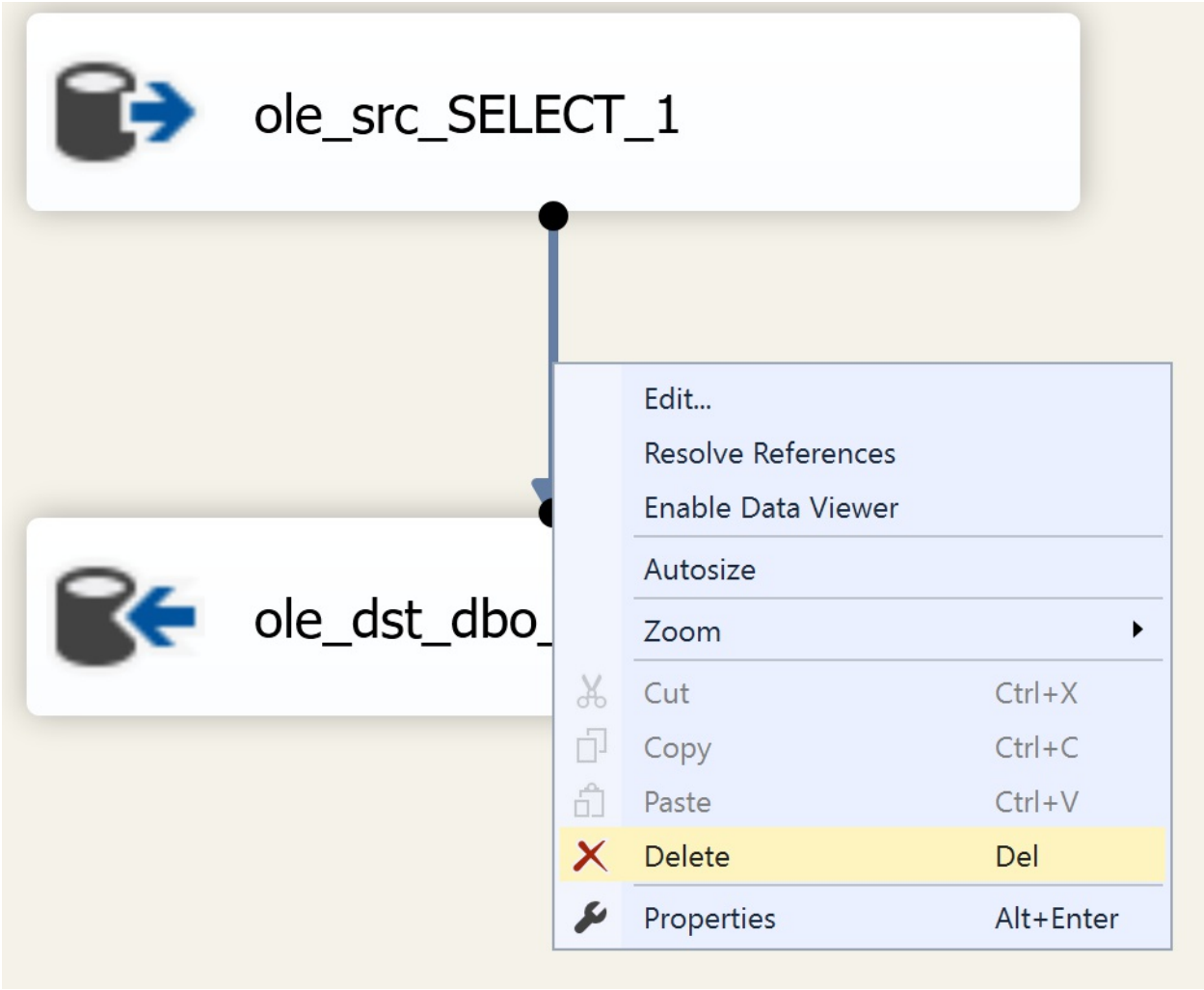
Getting ready

This recipe assumes that you have created the CustomLogging project and package from previous recipe.

How to do it...

1. In the CustomLogging.dstx package, navigate to the `dft_dbo_CustomLogging` by double-clicking on the dataflow task. Click on the path (blue arrow) between the **`ole_src_SELECT_1`** and the **`ole_dst_dbo_CustomLogging`** transform to select it. Right-click on it and select **`Delete`** to delete it.

WOW! eBook
www.wowebook.org



2. Drag and drop a Derived Column transform from the SSIS Toolbox onto the data flow. Link it to the ole_src_SELECT_1 and double-click on it to open the Derived Column Transformation. As shown in the following screenshot, do the following:

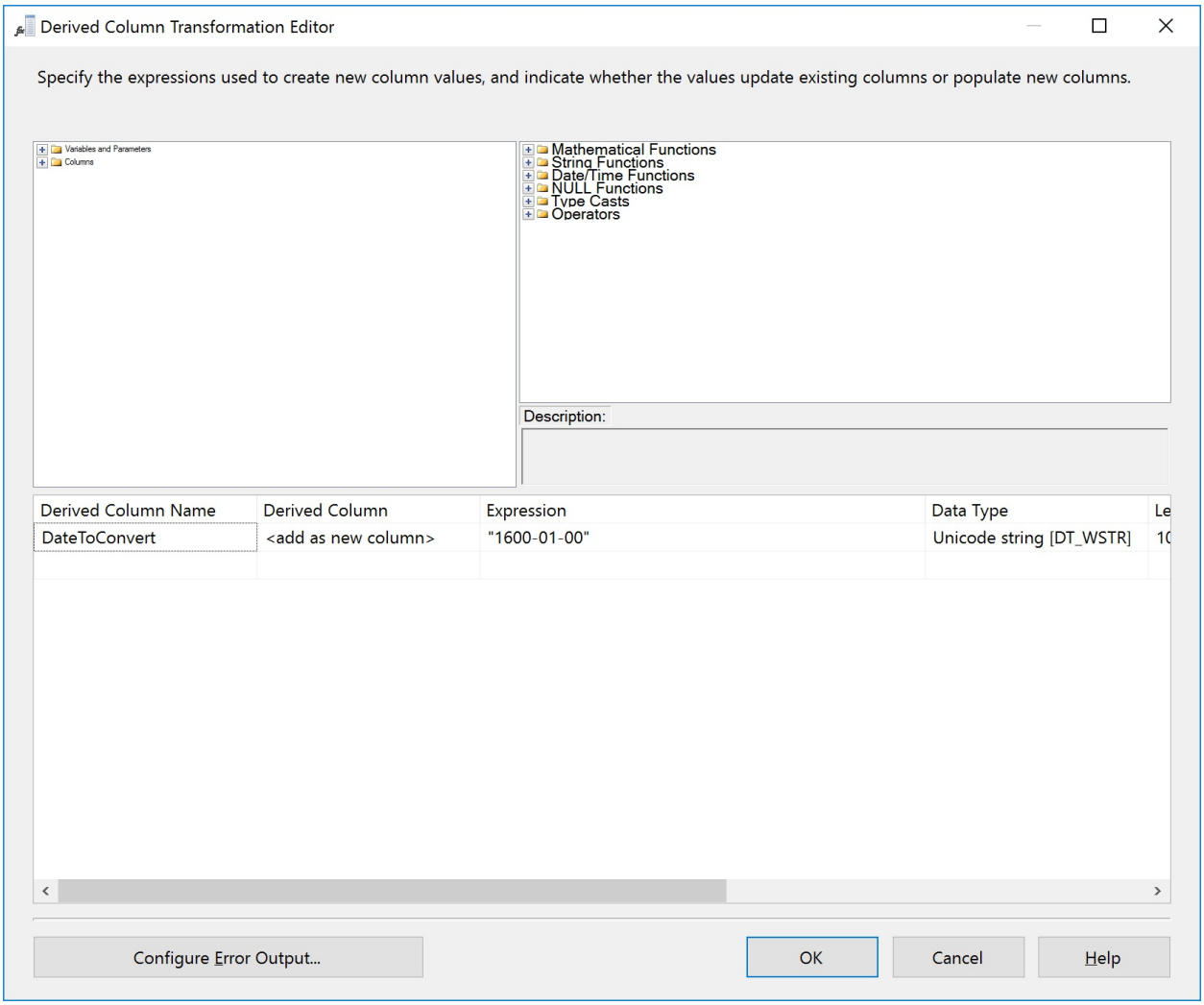
1. **Derived Column Name**: **DateToConvert**

2. Derived Column: leave it as **<Add as new column>**

3. Expression: Type "1600-01-00". This is not a valid date, it will cause an error and that's precisely what we want.

4. Click on OK when finished

WOW! eBook
www.wowebook.org



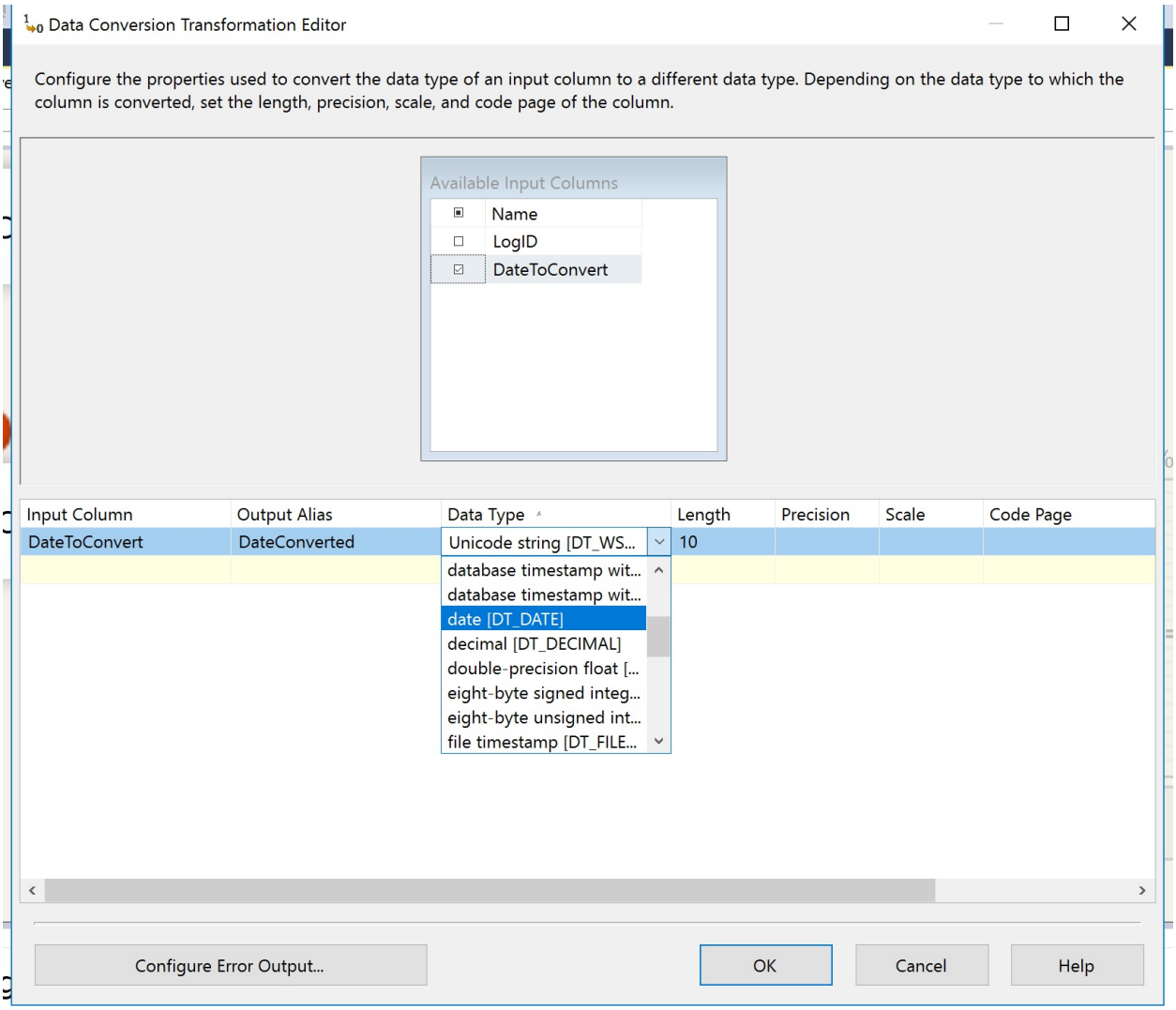
3. Rename the Derived Column der_AddDate.
4. Now, drag and drop a Data Conversion transform onto the dataflow task. Attach it to the der_AddDate derived column created above and double-click on it to open the **`Data Conversion Transformation Editor`**. Enter the values as shown in the following screenshot.

1. **`Input Column`**: Check the column **`DateToConvert`** in the Available Input Columns.

2. **`Output Alias`**: Change it to **`DateConverted`**.

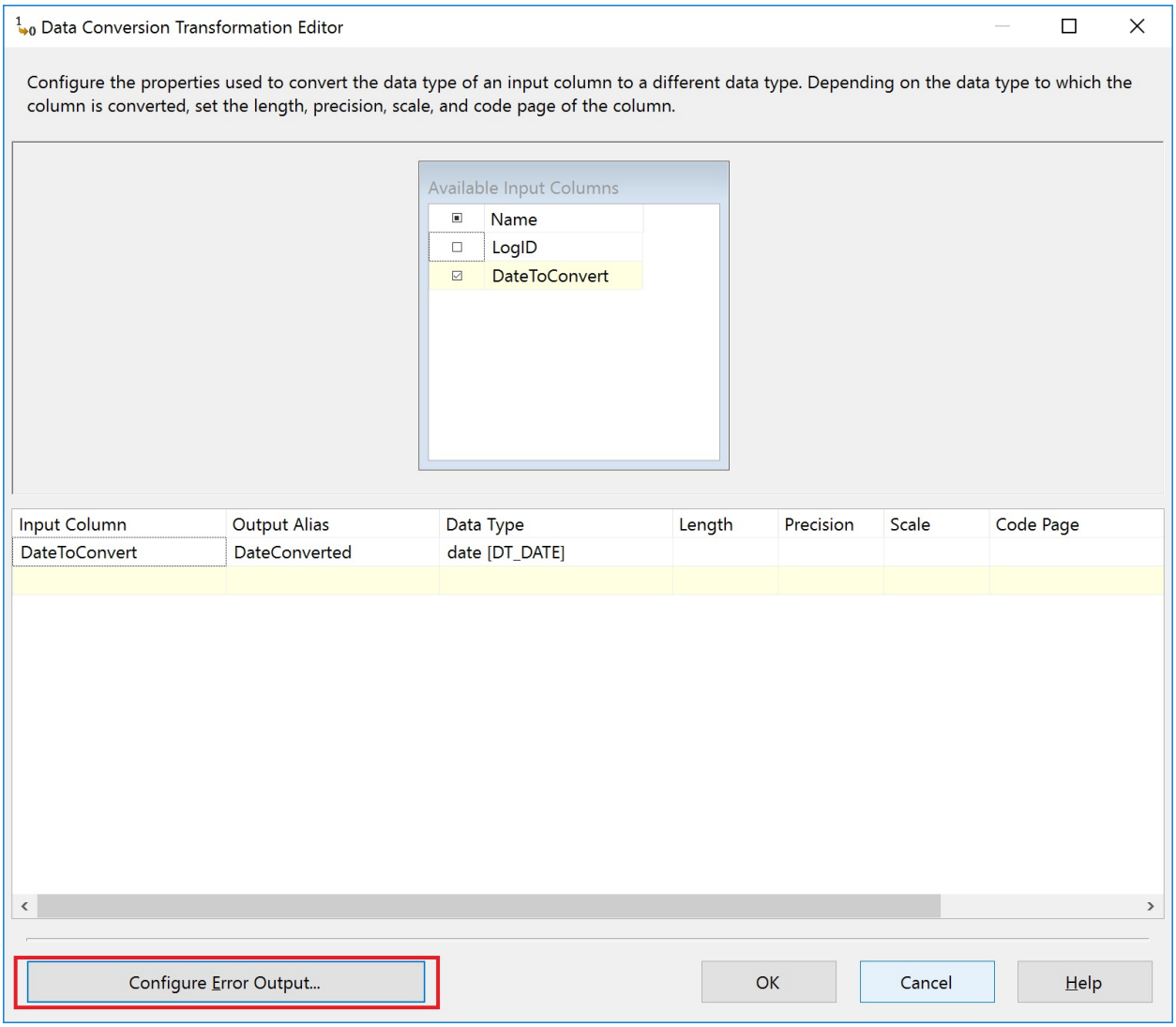
3. **`Data Type`**: select **`date [DT_DATE]`** from the drop down list.

WOW! eBook
www.wowebook.org



5. The **`Data Conversion Transformation Editor`** should now look like the following screenshot. Click on the **`Configure Error Output...`** button.

WOW! eBook
www.wowebook.org



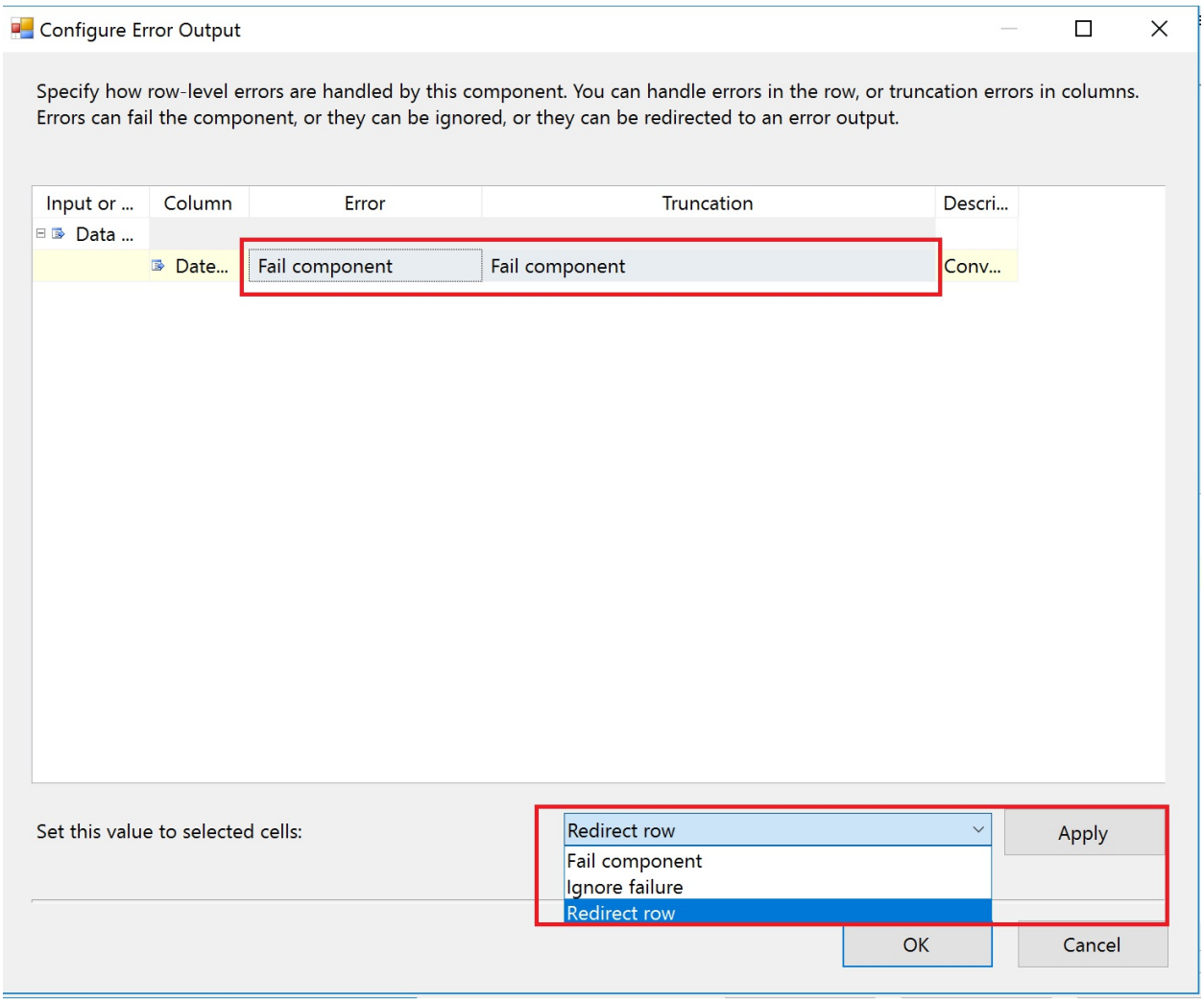
6. You will get a screen like the following screenshot. By default, the **`Error`** and **`Truncation`** errors will fail the component.

1. Select both columns and from the drop down list near Set this value to select cells: select **`Redirect row`**. Click Apply.

2. You should now see that both column values are now set to **`Redirect row`**.

3. Click **`OK`** to close the editor and rename the transform to dcnv_DateConverted.

WOW! eBook
www.wowebook.org

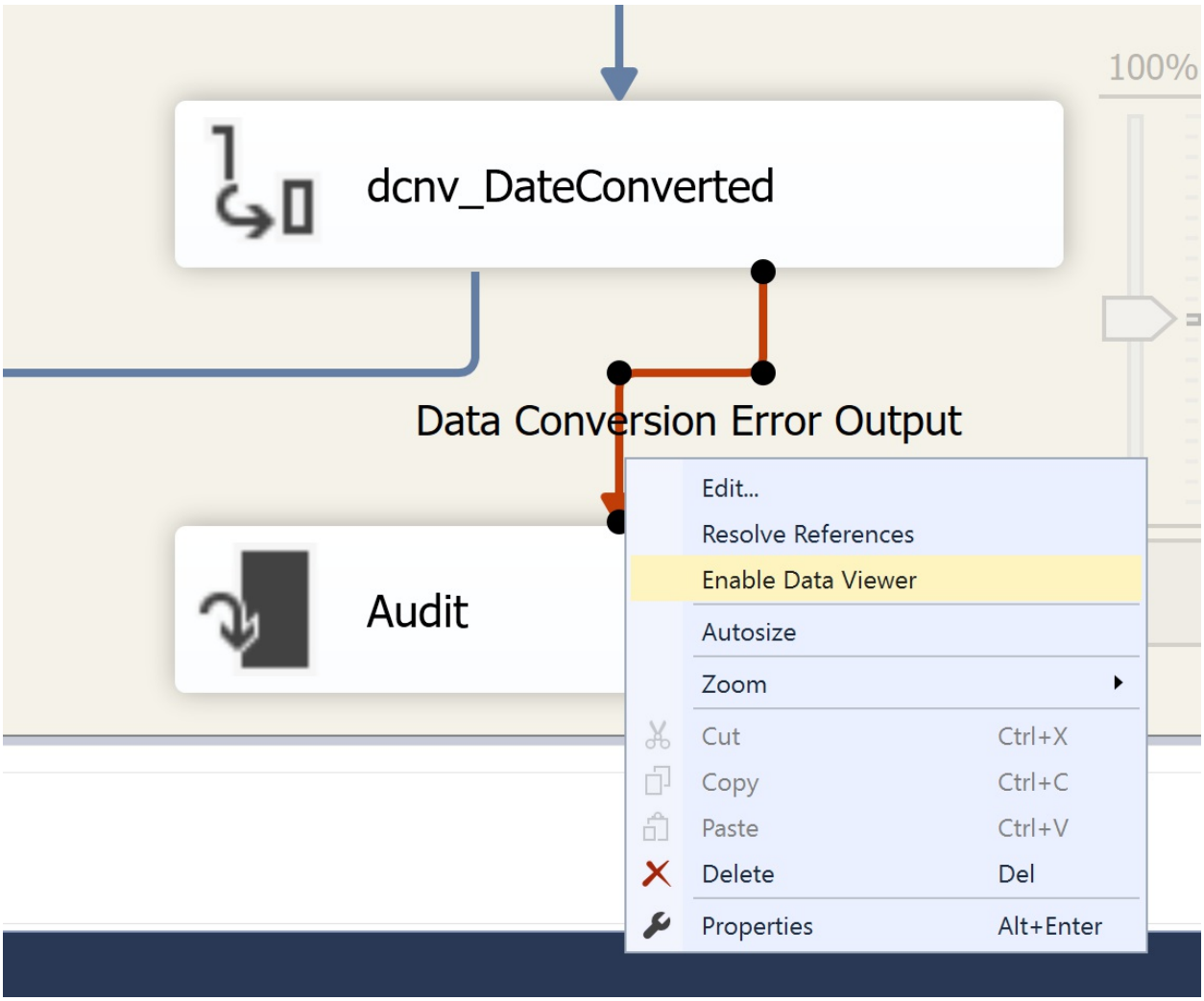


7. Now, bring an Audit transform from Other Transforms onto the dataflow task. The following steps detail what's in the following screenshot.

1. Attach the **`Data Conversion Error Output`** (red path or arrow) to it.

2. Right-click on the Error path and **`select Enable Data Viewer`** from the menu that appears.

WOW! eBook
www.wowebook.org

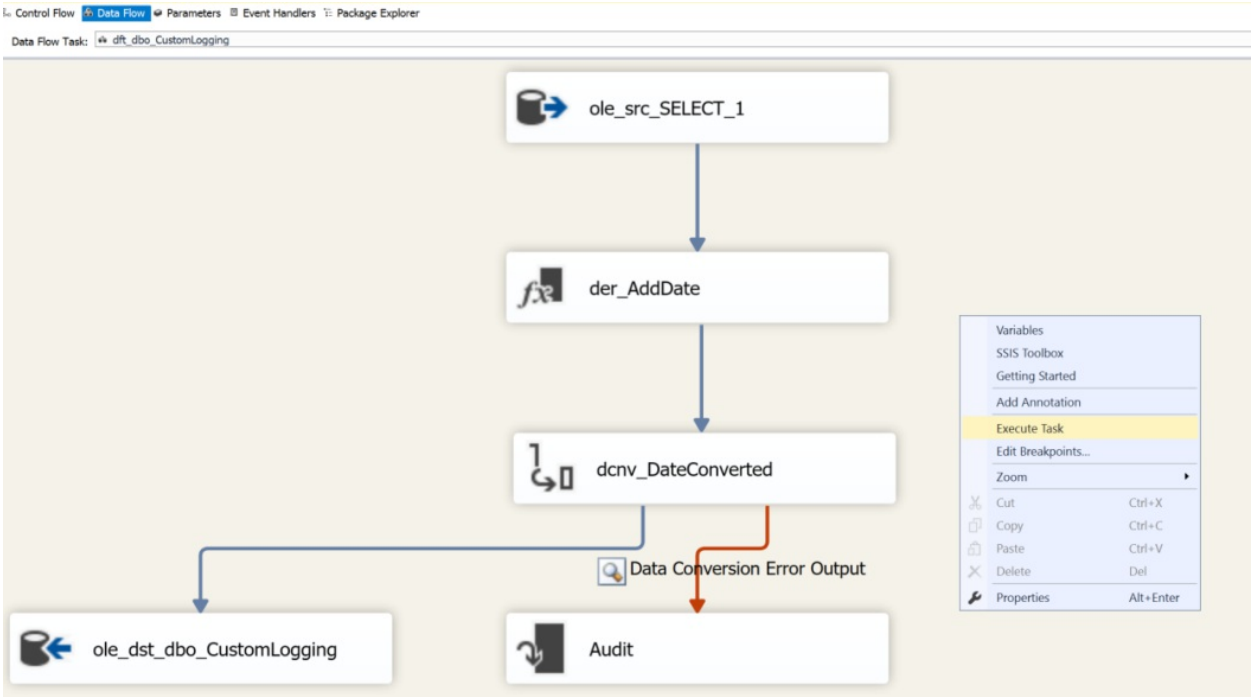


Note

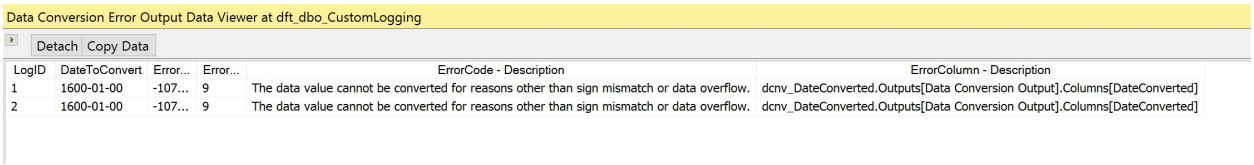
As we did before in the customized logging level recipe, select all the transforms and click on **`Make Same Width`** from the Layout toolbar. From the Format menu, select FormatàAuto Layout à Diagram to format the data flow task objects properly.

8. Your dataflow task should look like the following screenshot. Now, right-click anywhere in the background of the dataflow task and select **`Execute Task`** from the menu.

WOW! eBook
www.wowebook.org



9. You should see a data viewer like the following screenshot.
    1. The **`DateToConvert`** is the column we tried to convert.
    2. The second column is the **`ErrorCode`**, an internal code to SSIS.
    3. The third column is the **`ErrorColumn`** which is a lineage ID that SSIS assign to all columns in the dataflow task.
    4. **`ErrorCode - Description`** column gives us the reason of the error.
    5. **`The ErrorColumn - Description`** gives us the column that failed to be converted (**`DateConverted`**).



Data Conversion Error Output Data Viewer at dft_dbo_CustomLogging

Detach Copy Data

LogID	DateToConvert	Error...	Error...	ErrorCode - Description	ErrorColumn - Description
1	1600-01-00	-107...	9	The data value cannot be converted for reasons other than sign mismatch or data overflow.	dcnv_DateConverted.Outputs[Data Conversion Output].Columns[DateConverted]
2	1600-01-00	-107...	9	The data value cannot be converted for reasons other than sign mismatch or data overflow.	dcnv_DateConverted.Outputs[Data Conversion Output].Columns[DateConverted]

WOW! eBook
www.wowebook.org

Control Flow templates

Control Flow templates are an addition to SSIS 2016 that will surely be promising...in the future. For now, Microsoft put the foundation of something that looks very interesting.

If there's one thing missing with SSIS is the reusability of custom components without doing .Net code. We'll see how to achieve custom task and transforms later in this book using .Net and you'll see that it's tedious to achieve even for something simple. Let's say that we would like to create a truncate table task, that is, a task that we would use to solely erase a table's content. This task would then appear in the SSIS toolbox and be available to all packages in your projects. This recipe is exactly what we will do using Control Flow templates.

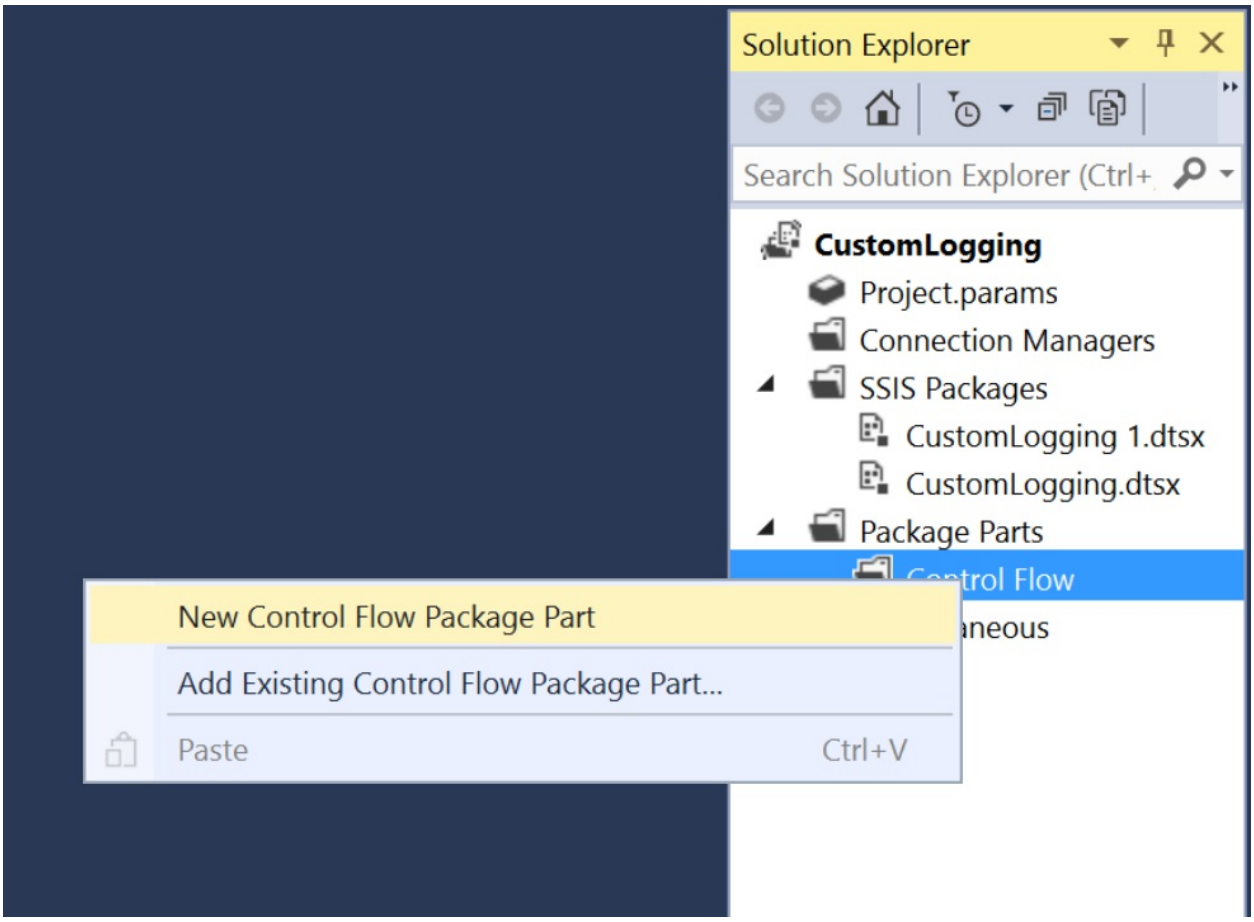
Getting ready

This recipe assumes that you have done all previous recipes in this chapter or you have your own SSIS project open.

How to do it...

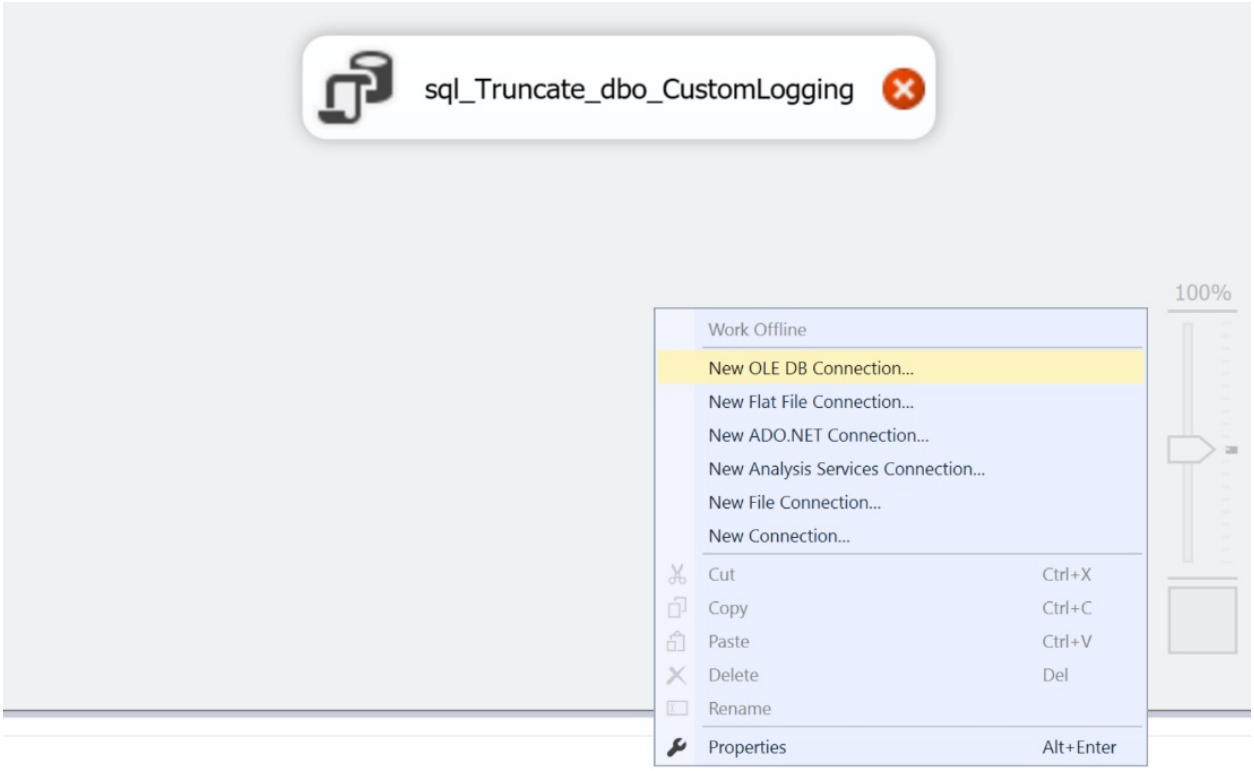
1. In SSDT, with your project open, right-click on the **`Control Flow`** folder in the **`Package Parts`** section of the solution explorer. As shown in the following screenshot, select **`New Control Flow Package Part`** from the contextual menu. Rename it `Chapter2Part.dtsxp`.

WOW! eBook
www.wowebook.org



2. Drag and drop a Execute SQL task on it.
    1. Rename it **`sql_Truncate_dbo_CustomLogging`**.
    2. Click anywhere in the Control Flow. Set its description property to "This is a simple template that is meant to truncate the table `[dbo].[CustomLogging]`".
    3. As shown in the following screenshot, right-click in the Connection Managers area and choose **`New OLE DB Connection...`** from the menu that appears.

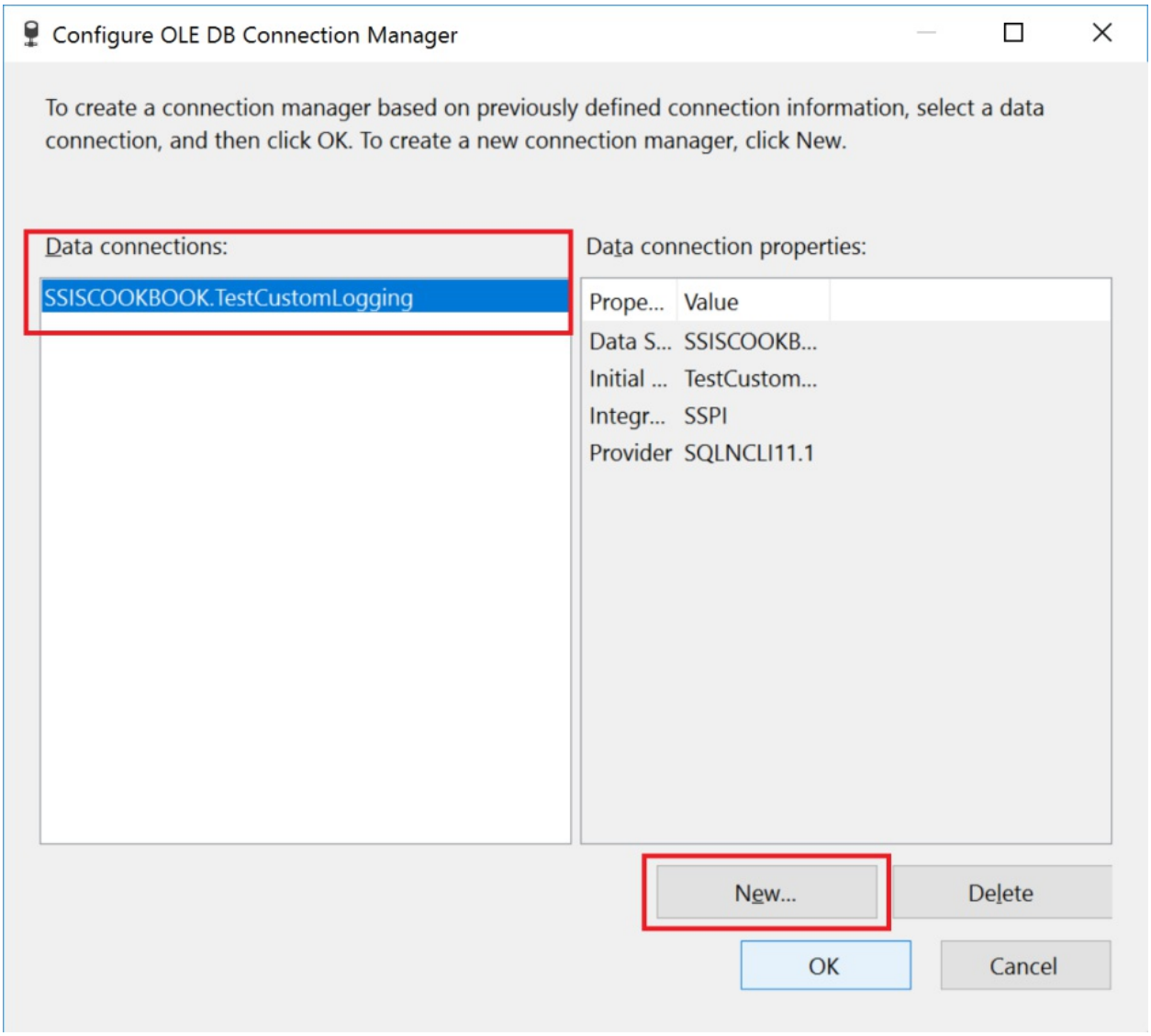
WOW! eBook
www.wowebook.org



Right-click here to add a new connection manager to the SSIS package.

4. As shown in the following screenshot, select the existing connection (**`SSISCOOKBOOK\TestCustomLogging`** in that case) in the top left or create a new one by clicking on **`NEW`**. . . . Click on **`OK`** when done to close the **`Configure OLE DB Connection Manager`** window.

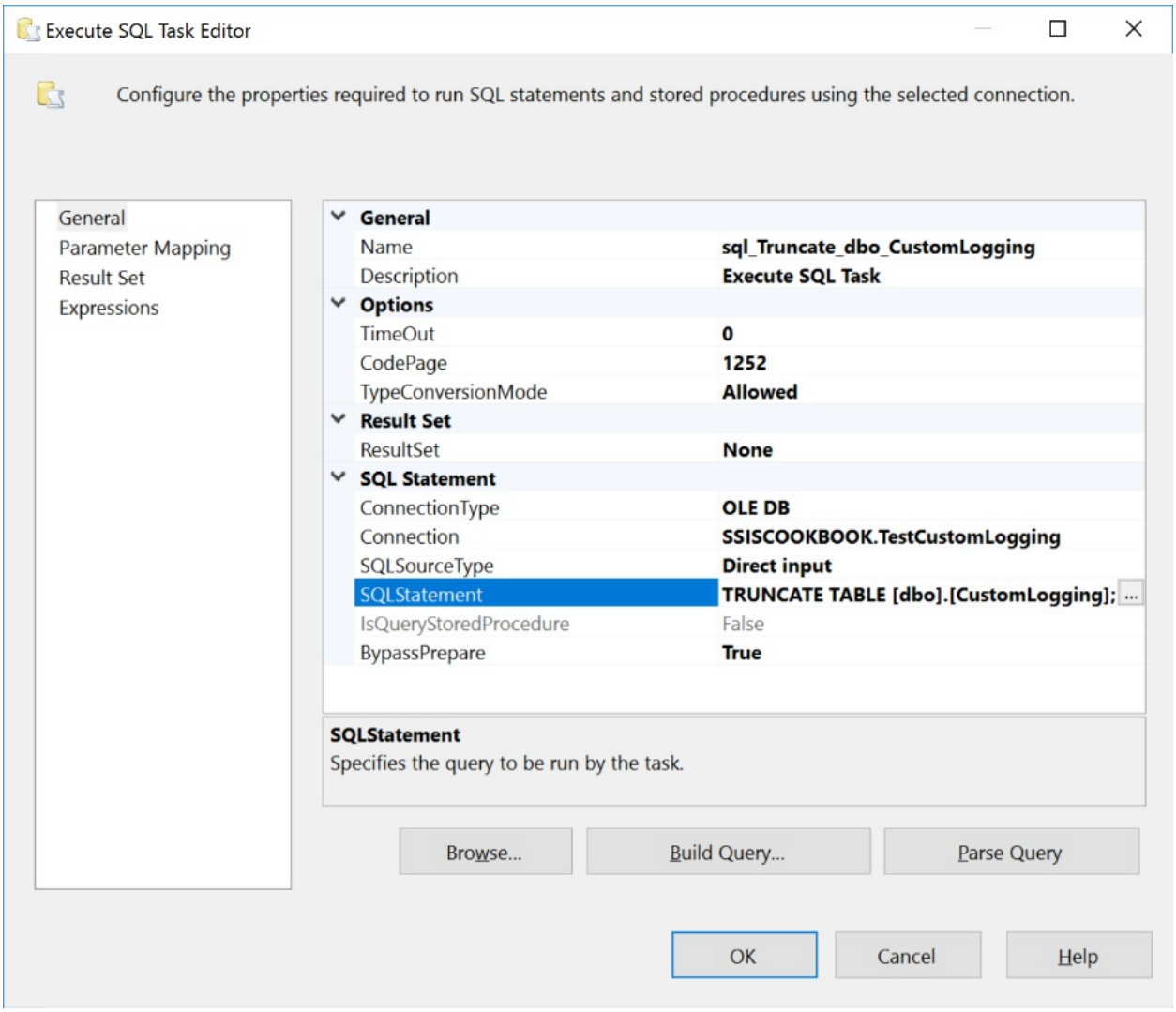
WOW! eBook
www.wowebook.org



5. Double-click on the **`sql_Truncate_dbo_CustomLogging`** task to open the **`Execute SQL Task Editor`**. Set the properties in the SQL Statement section as follow.
   1. **`ConnectionType`**: **`OLE DB`**, the default value
   2. **`Connection`**: set it to the connection manager created in the previous step.
   3. **`SQLSourceType`**: Leave the default value, **`Direct input`**
   4. Type the following SQL DML statement in the **`SQLStatement`** property:

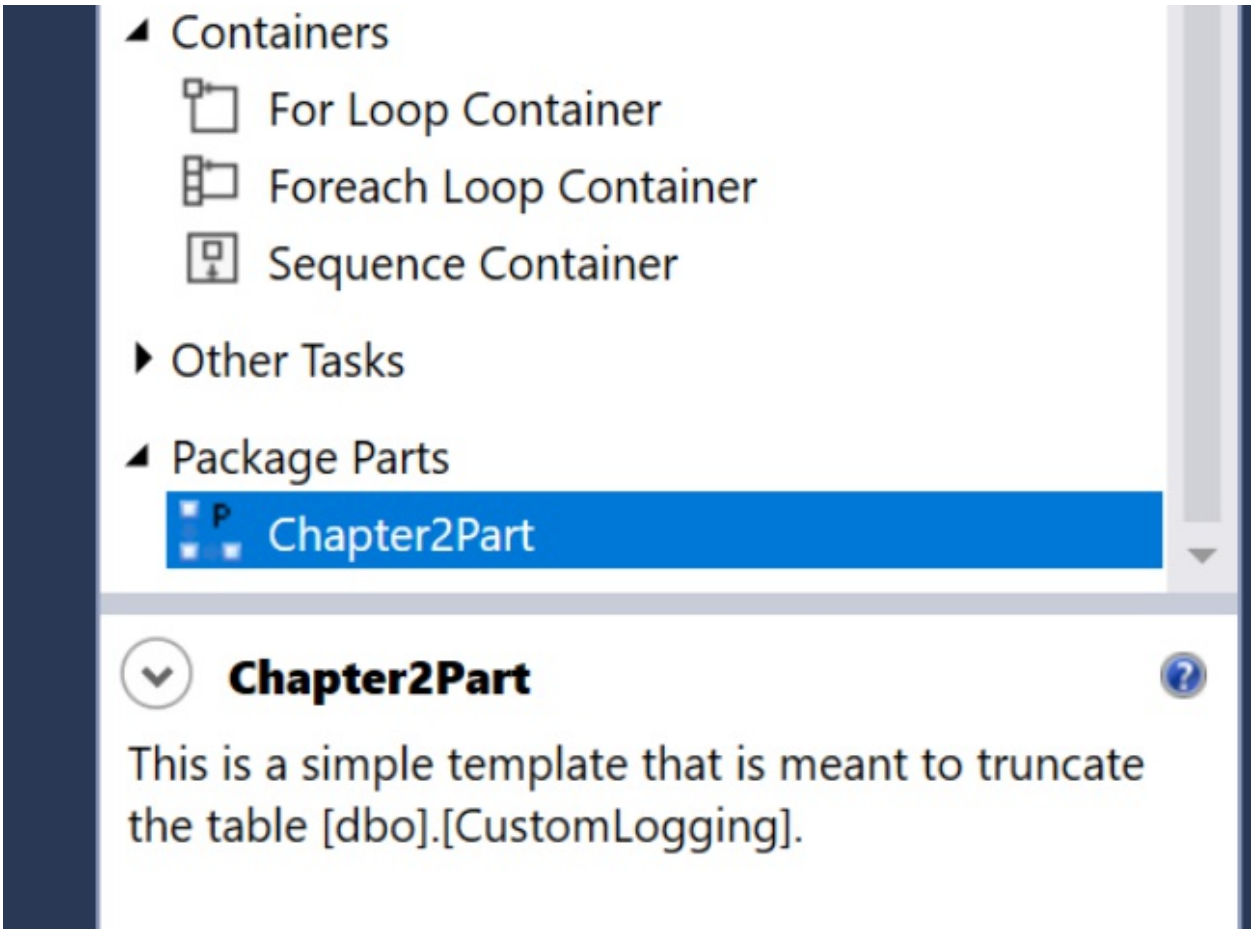
```
TRUNCATE TABLE [dbo].[CustomLogging];
```

WOW! eBook
www.wowebook.org



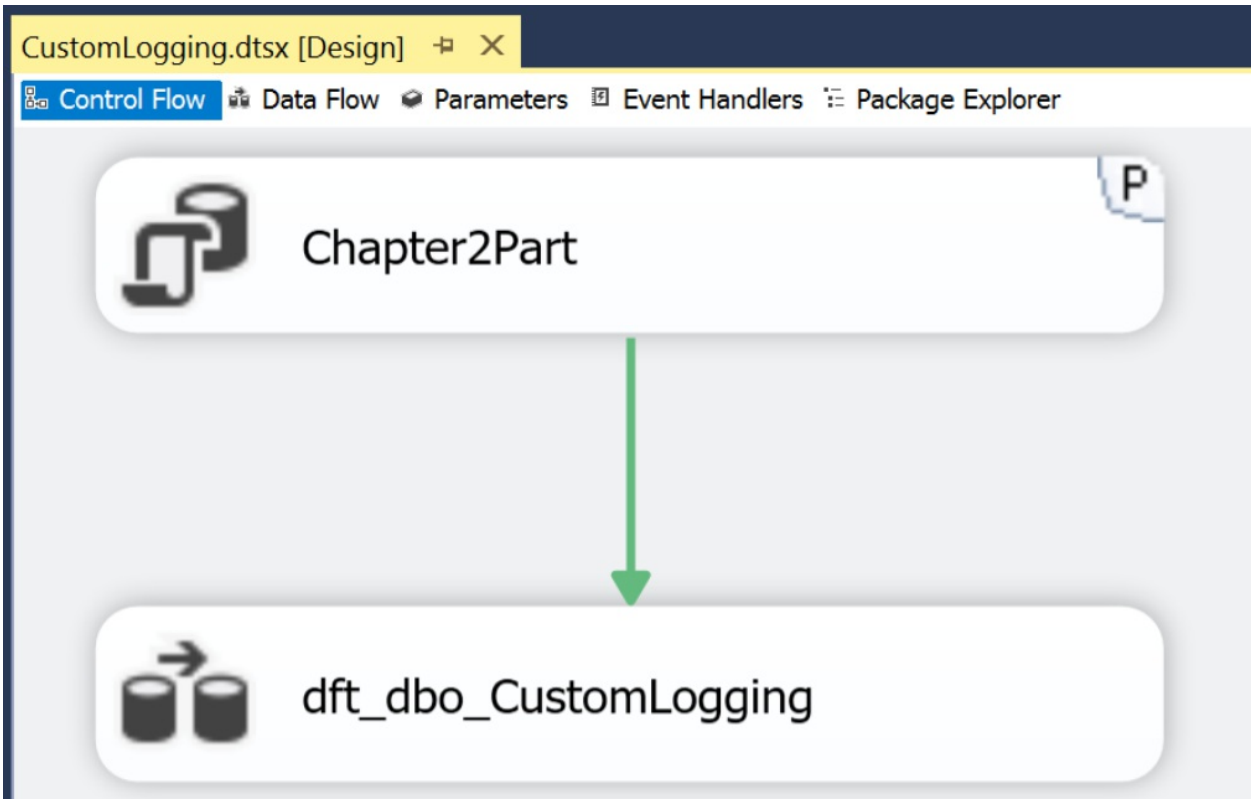
5. Save and close the `Chapter2Part.dtsxp` package part.
6. Now, open the CustomLogging package from the solution explorer. As shown in the following screenshot, there is now a new component in the **`Package Parts`** section of the SSIS toolbox: the **`Chapter2Part`** created in the previous steps. Also, notice the description below the **`Package Parts`** section. This is the one that we assigned at the control flow level of the package part when it has been created.

WOW! eBook
www.wowebook.org

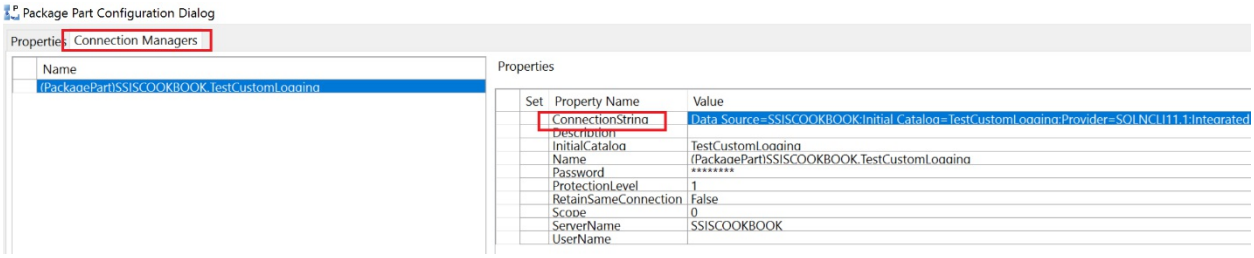


7. Drag and drop a **`Chapter2Part`** onto the control flow of the **`CustomLogging`** package. Now to get the same as in the following screenshot, execute the following steps
    1. Connect it to the **`dft_dbo_CustomLogging`** data flow task.
    2. Select all Control flow components.
    3. In the Layout toolbar, click Make Same Size.
    4. From the Format menu, select Auto LayoutàDiagram to properly align the tasks

WOW! eBook
www.wowebook.org



5. Execute the package to make sure that the Package Part is properly working and stop package execution when done.
6. Double-click on the Chapter2Part to view its properties as shown in the following screenshot. Go to the **`Connection Managers`** Tab. Locate the **`ConnectionString`** property as highlighted in the screenshot. Notice that it has a fixed value. This is the biggest limitation of these parts, there is no way for now to alter any of the property at runtime. We cannot use package configurations or parameters to alter these values dynamically as we can with regular tasks.



WOW! eBook
www.wowebook.org

Appendix 1. Feedback

If you have any feedback on this eBook or are struggling with something we haven't covered, let us know by clicking on this link.

If you have any concerns you can also get in touch with us at customercare@packtpub.com

We will send you the next chapters when they are ready……!

Hope you like the content presented.

WOW! eBook
www.wowebook.org