Introduction

The purpose of having Integration Services (SSIS) within SQL Server features is to provide a flexible, robust pipeline that can efficiently perform row-by-row calculations and parse data all in memory.

While the extract and load phases of the pipeline will touch disk (read and write respectively), the transformation itself should process in memory. If transformations spill to disk (for example with large sort operations), you will see a big performance degradation.

For this reason, it is advised to construct your packages to partition and filter data so that all transformations fit in memory.

Most of the best practices of SSIS services are related to package design.

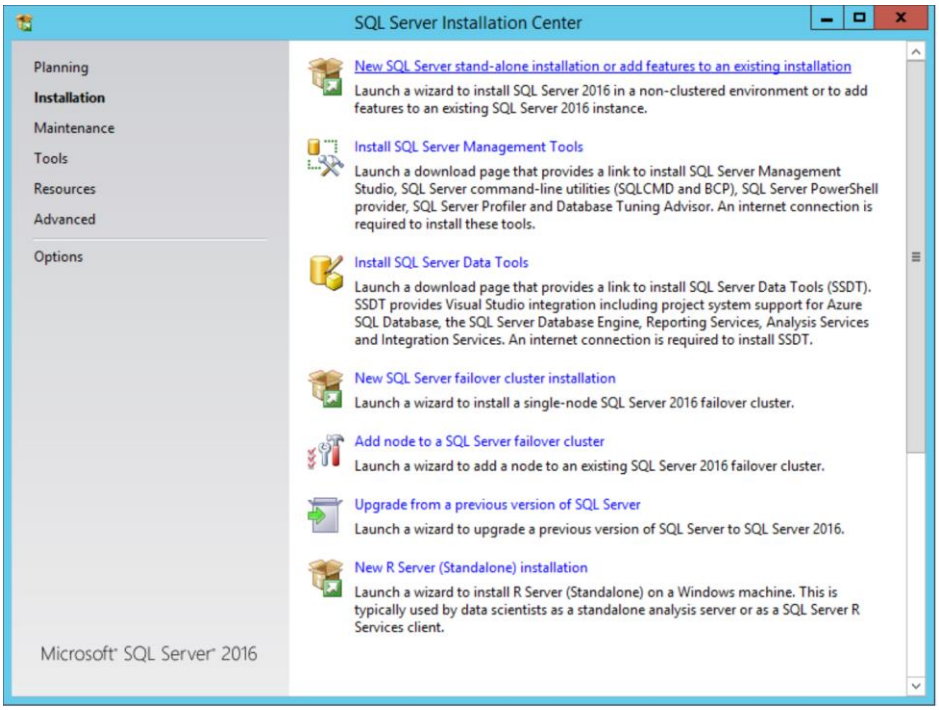
DBAs have little setting to change to optimized SSIS environment.

Following documentation will highlight SSIS 2016 installation steps and SSIS catalogue SSISDB settings for optimization.

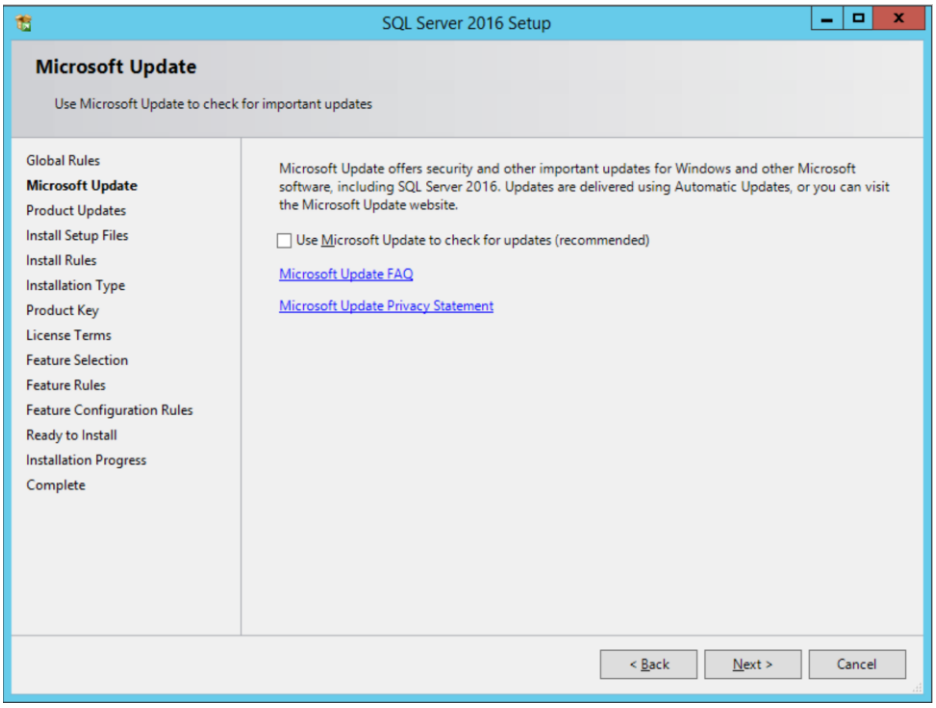
SSIS 2016 Installation

SSIS installation starts with SQL Server media. Following are steps to be followed to install SSIS 2016.

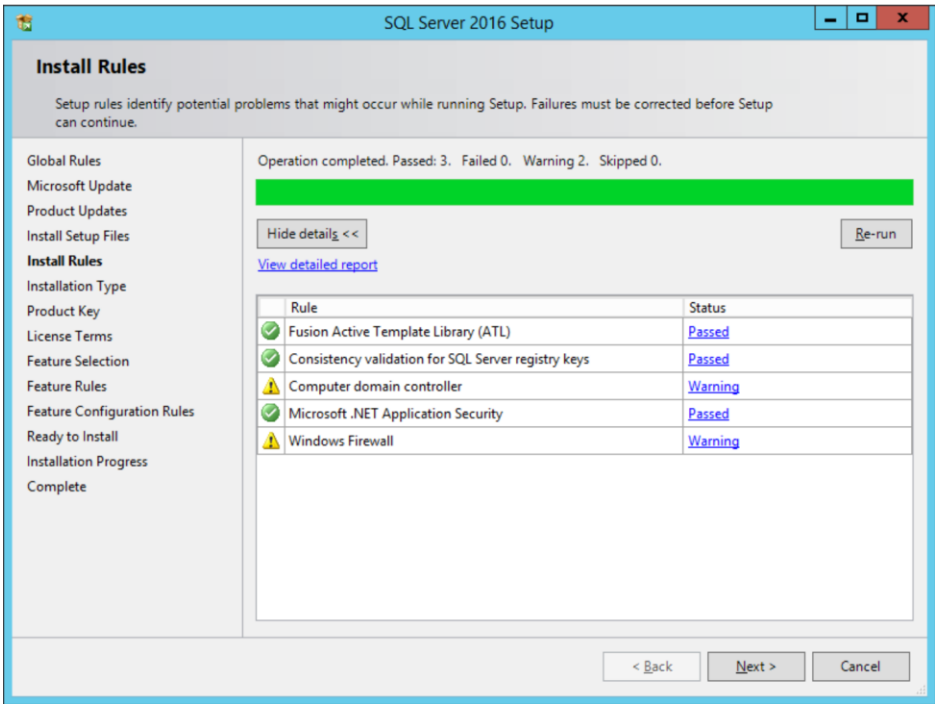
Step 1- Start SQL Server Installation media



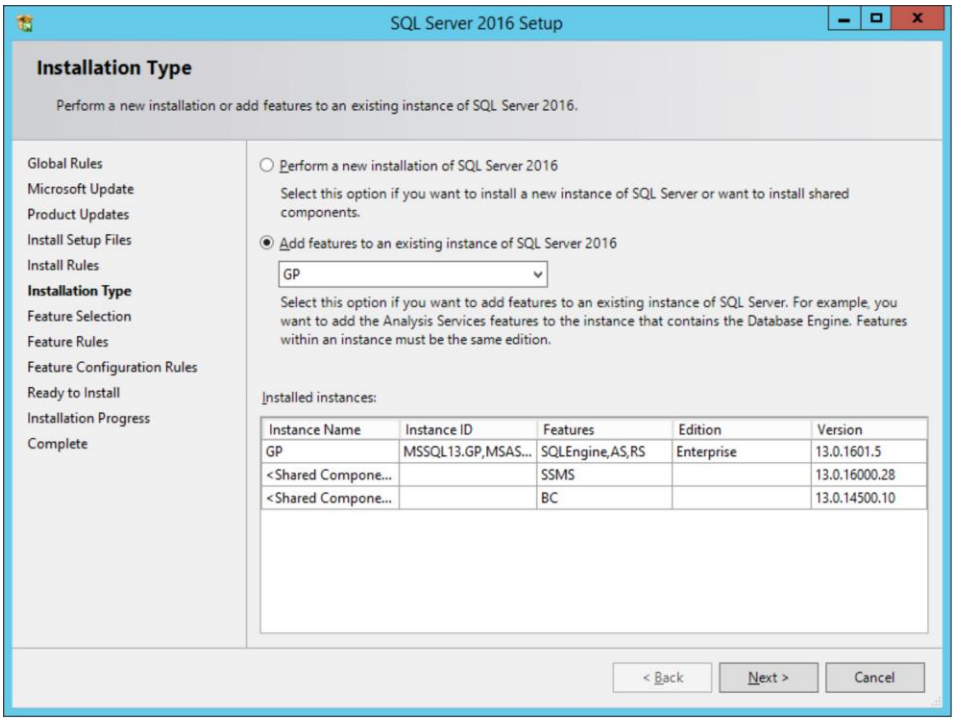
Step 2- Confirm update of the product if you have access to internet



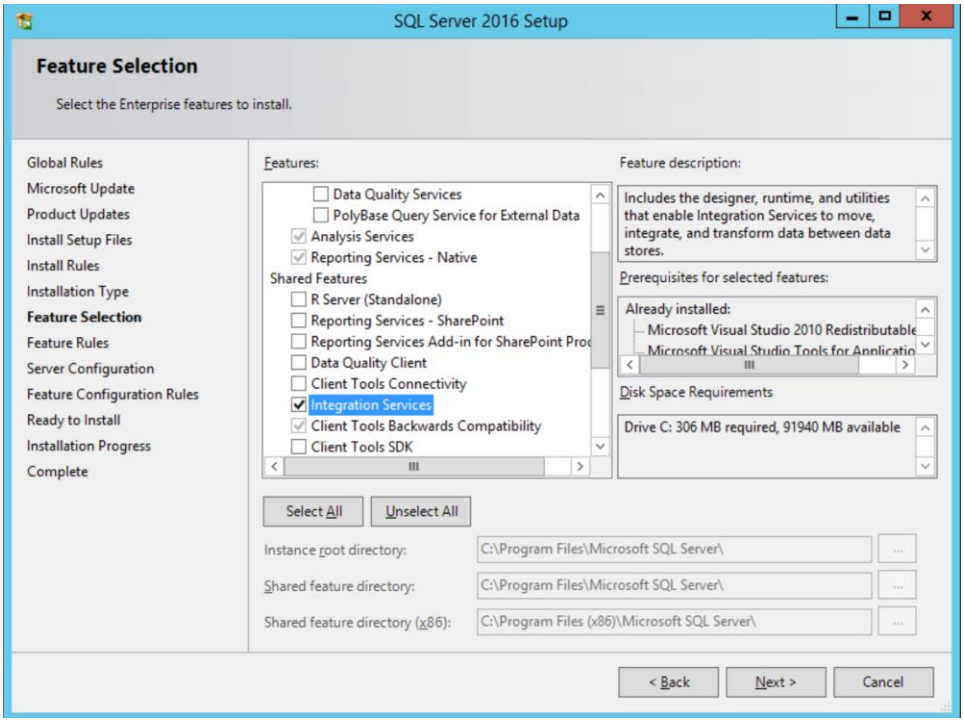
Step 3- Validation of installation rules



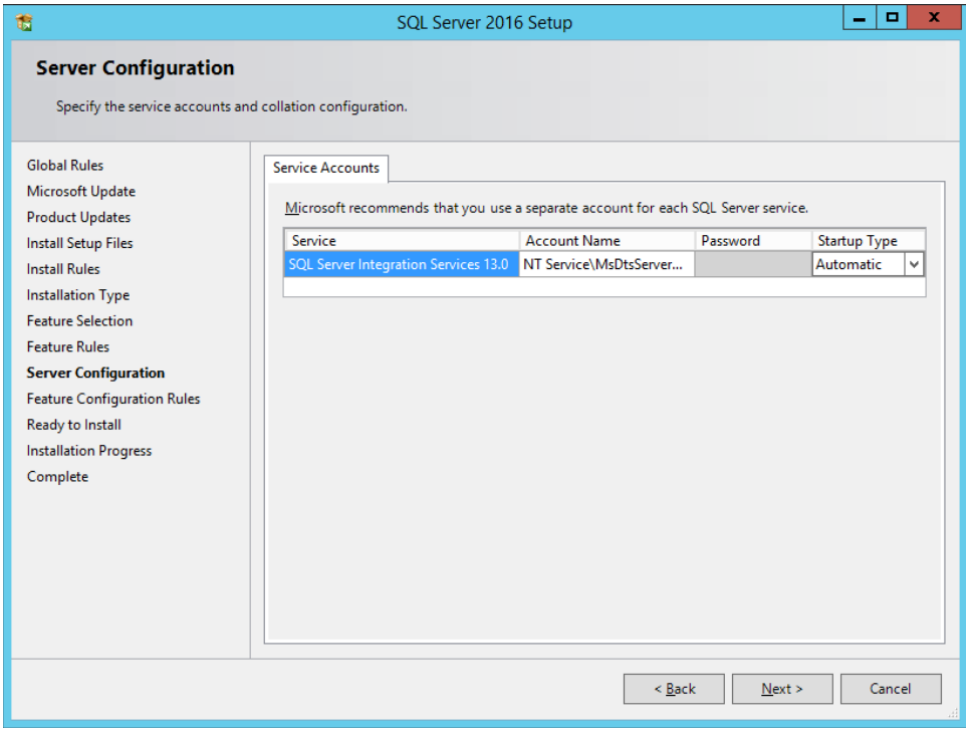
Step 4- Add features to old installation or create a new installation



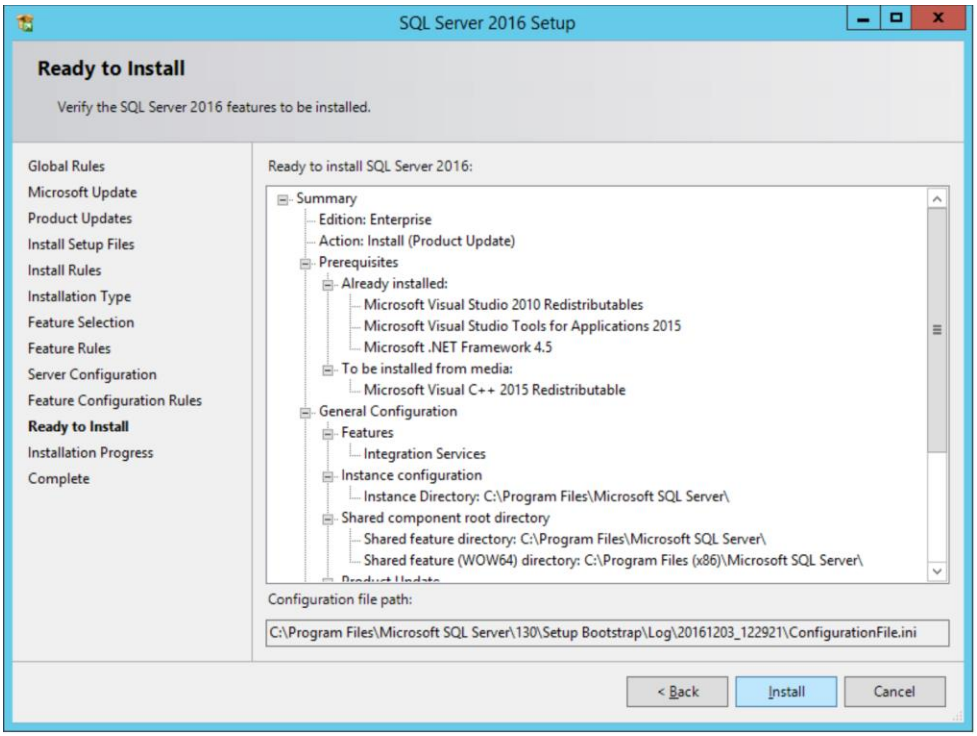
Step 5- Select SQL Server Integration service and shared components



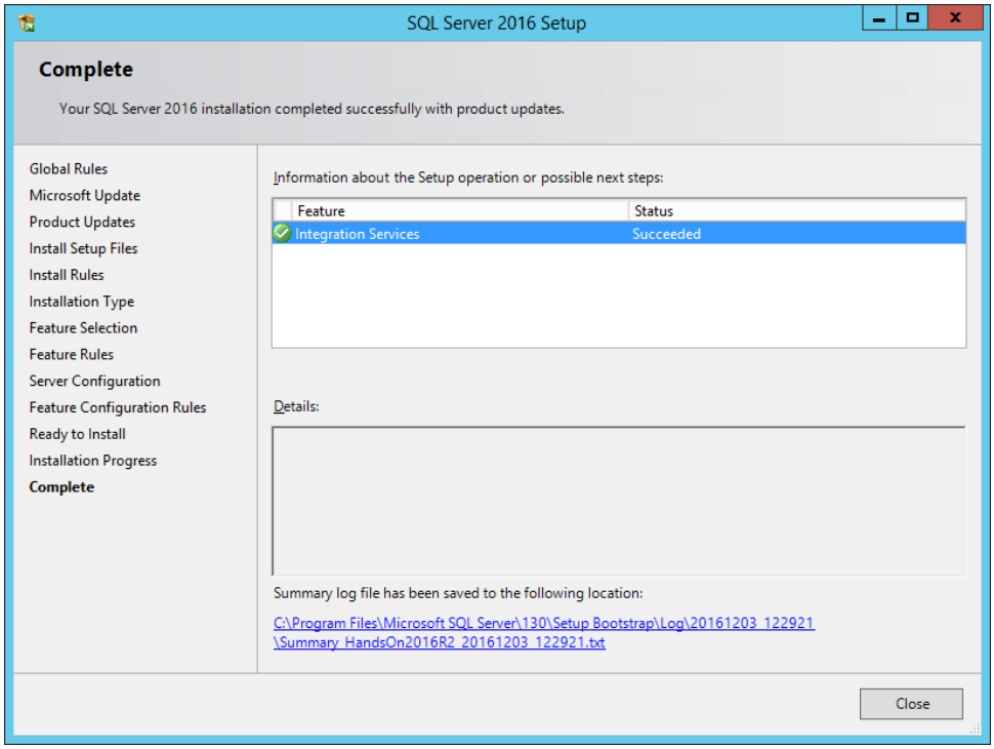
Step 6- Change default service account with a domain account pre-staged before



Step 7- Confirm your installation and generate configuration.ini file



Step 8- Installation completion



POST INSTALLATION CONFIGURATIONS

SSIS 2016 SERVICE CONFIGURATION

Typically, you do not have to make any changes to ssis configuration file, nor do you have to change the file's default location. However, you will have to modify the configuration file if your packages are stored in a named instance or a remote instance of Database Engine, or in multiple instances of the Database Engine. (Reference)

However, it could be important to do that if your packages are stored in a named instance or a remote instance of Database Engine, or in multiple instances of the Database Engine.

By default, the name for this configuration file is MsDtsSrvr.ini.xml, and the file is located in the folder, %ProgramFiles%\Microsoft SQL Server\130\DTS\Binn.

The file structure is described below:

```
<?xml version="1.0" encoding="utf-8"?>
<DtsServiceConfiguration xmlns:xsd="http://www.w3.org/2001/XMLSchema" xmlns:xsi="http://www.w3.org/2001/
  <StopExecutingPackagesOnShutdown>true</StopExecutingPackagesOnShutdown>
  <TopLevelFolders>
    <Folder xsi:type="SqlServerFolder">
      <Name>MSDB</Name>
      <ServerName>.</ServerName>
    </Folder>
    <Folder xsi:type="FileSystemFolder">
      <Name>File System</Name>
      <StorePath>..\Packages</StorePath>
    </Folder>
  </TopLevelFolders>
</DtsServiceConfiguration>
```

In case your packages are stored in MSDB, change your MsDtsSrvr.ini.xml as documented below:

```
<?xml version="1.0" encoding="utf-8"?>
<DtsServiceConfiguration xmlns:xsd="http://www.w3.org/2001/XMLSchema" xmlns:xsi="http://www.w3.org/2001/
  <StopExecutingPackagesOnShutdown>true</StopExecutingPackagesOnShutdown>
  <TopLevelFolders>
    <Folder xsi:type="SqlServerFolder">
      <Name>MSDB</Name>
      <ServerName>ServerName\InstanceName</ServerName>
    </Folder>
    <Folder xsi:type="FileSystemFolder">
      <Name>File System</Name>
      <StorePath>..\Packages</StorePath>
    </Folder>
  </TopLevelFolders>
</DtsServiceConfiguration>
```

PS: Recall database name and instance name in the graphic above.

Using SSISDB catalogue

In SSIS 2012+, SSISDB catalog plays a critical role while deploying packages to the Integration Services server.

After SQL Server SSIS installation SSISDB catalogue must be installed to be able to use it.

Create SSIDB Catalogue (reference)

1. Open SQL Server Management Studio.
2. Connect to the SQL Server Database Engine.
3. In Object Explorer, expand the server node, right-click the **Integration Services Catalogs** node, and then click **Create Catalog**.
4. Click **Enable CLR Integration**.
5. The catalog uses CLR stored procedures.
6. Click **Enable automatic execution of Integration Services stored procedure at SQL Server startup** to enable the catalog.startup stored procedure to run each time the SSIS server instance is restarted.
7. The stored procedure performs maintenance of the state of operations for the SSISDB catalog. It fixes the status of any packages there were running if and when the SSIS server instance goes down.
8. Enter a password, and then click **Ok**.

   The password protects the database master key that is used for encrypting the catalog data. Save the password in a secure location. It is recommended that you also back up the database master key. For more information, see Back Up a Database Master Key.

SSISDB catalogue CONFIGURATION

You could fine tune your catalogue behavior by changing its properties. You could access the dialog box of this property by right-clicking **SSISDB** in Object Explorer.

The following **SSISDB** catalog properties define how this SQL Server Agent job behaves:

Clean Logs Periodically	The job step for operations cleanup runs when this property is set to True.
Retention Period (days)	Defines the maximum age of allowable operations data (in days). Older data are removed. The minimum value is one day. The maximum value is limited only by the maximum value of the SQL Server **int** data. For information about this data type, see int, bigint, smallint, and tinyint (Transact-SQL)
Periodically Remove Old Versions	The job step for project version cleanup runs when this property is set to True.
Maximum Number of Versions per Project	Defines how many versions of a project are stored in the catalog. Older versions of projects are removed
Encryption Algorithm	• The **Encryption Algorithm** property specifies the type of encryption that is used to encrypt sensitive parameter

	values. You can choose from the following types of encryption AES_256 (default) • AES_192 • AES_128 • DESX • TRIPLE_DES_3KEY • TRIPLE_DES • DES

SSIS catalogue SSISDB configuration could be found here: https://docs.microsoft.com/en-us/sql/integration-services/catalog/ssis-catalog?view=sql-server-2017#Configuration.