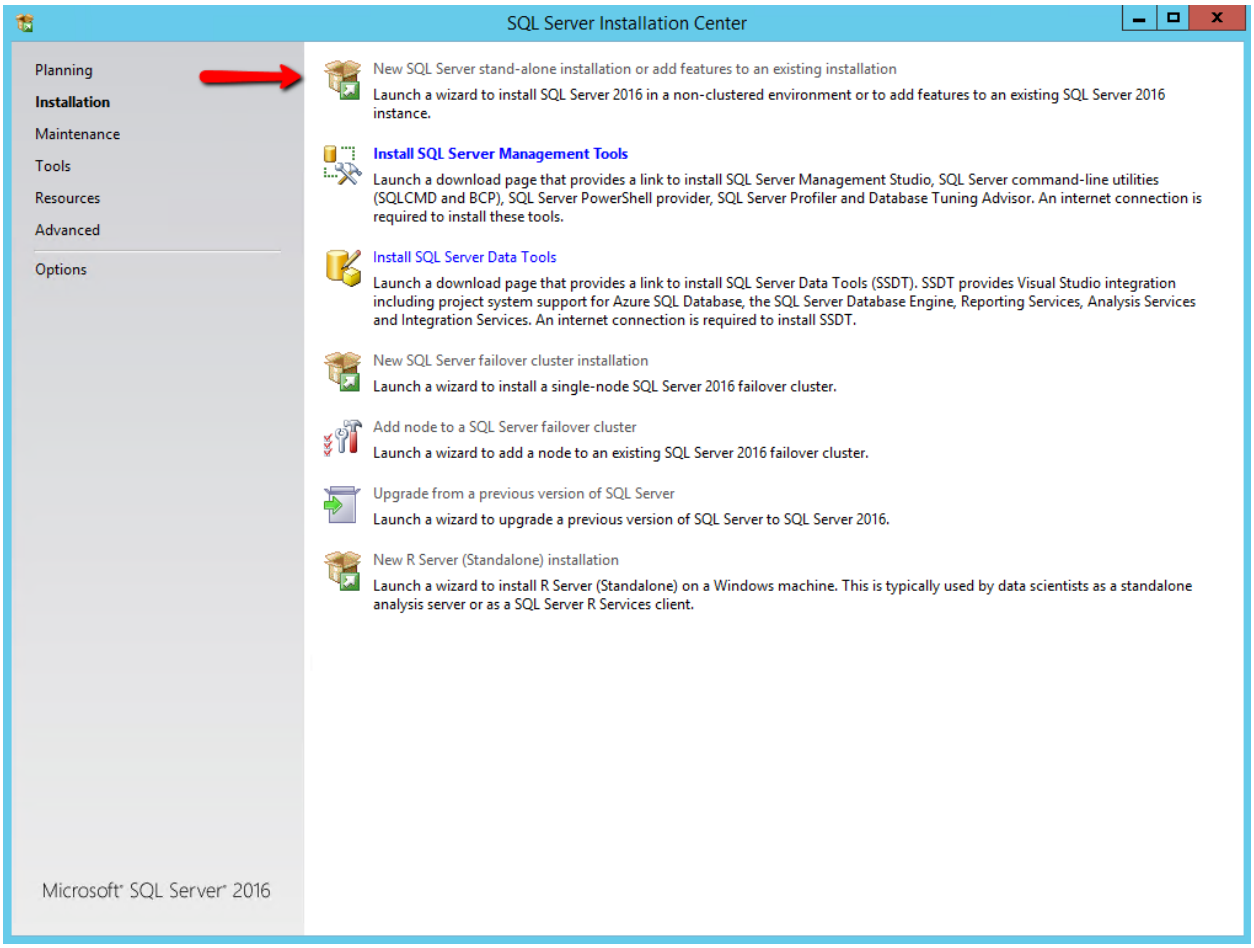
Installing and Configuring SQL Server

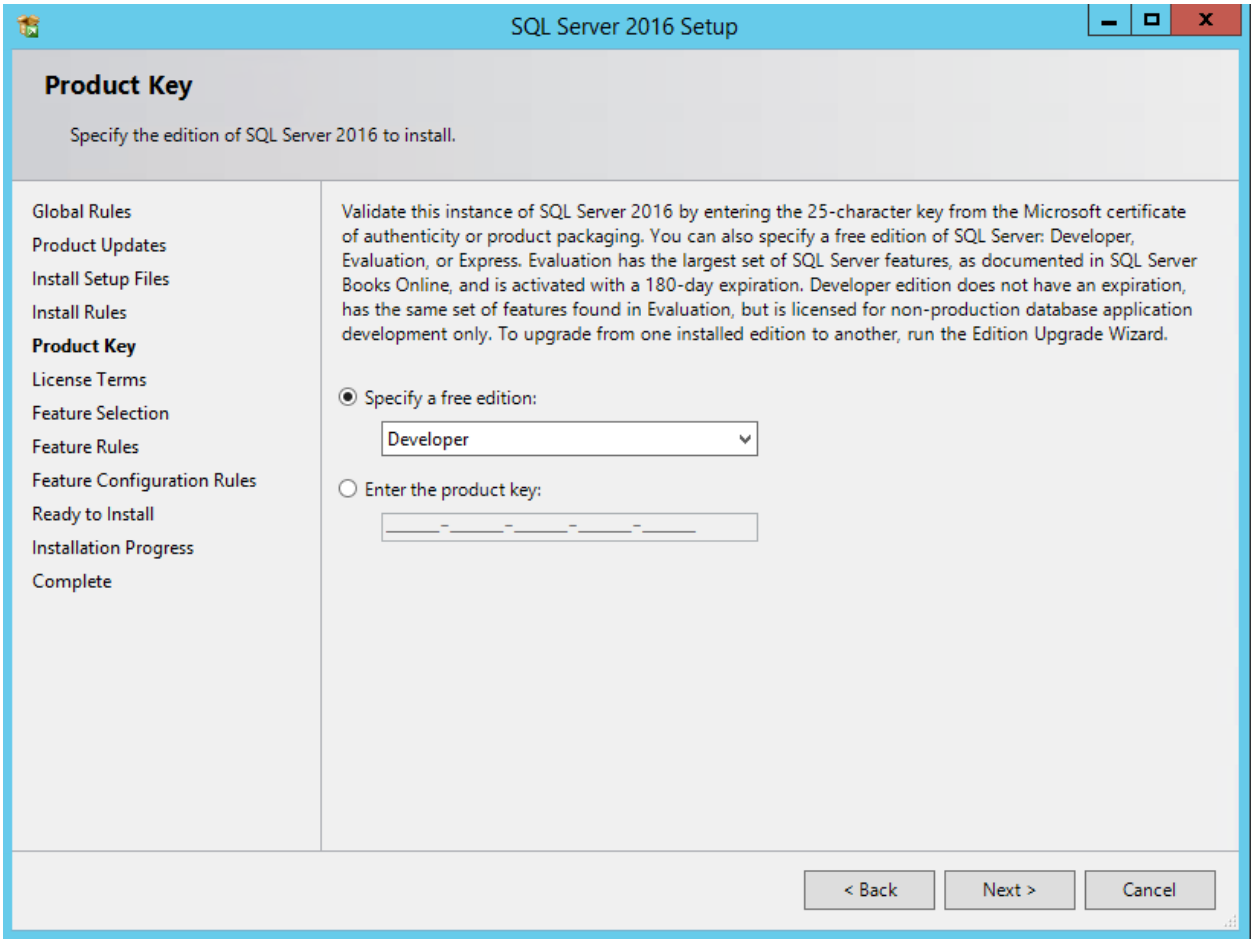
Overview: This document outlines best practices for installing and configuring a new installation of SQL Server.

I. Installation



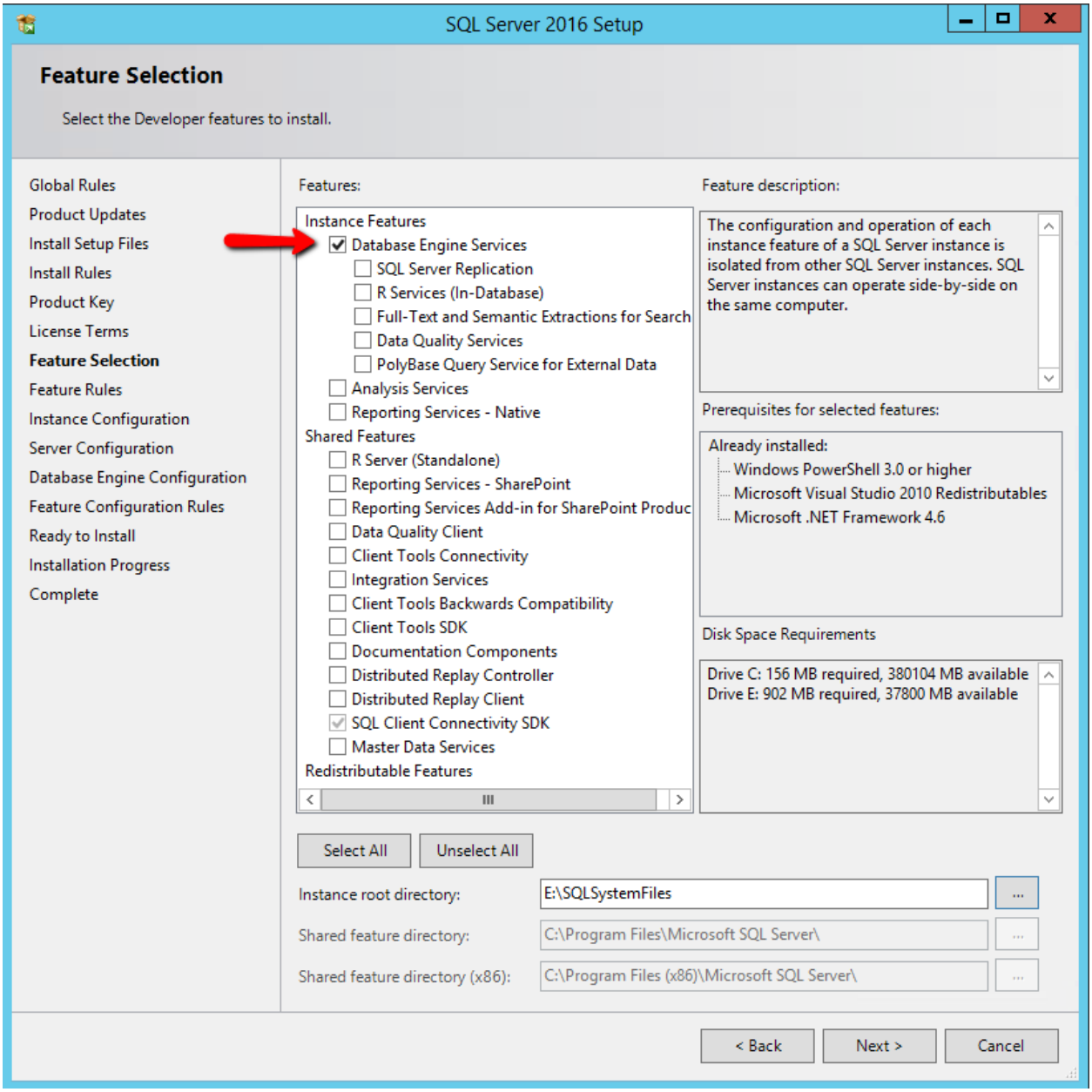
For a new installation of SQL Server, we want to select **New SQL Server stand-alone installation or add features to existing installation**.

II. Select an Edition



After selecting the type of installation, we will see information about rules, updates, and setup files. We can review this information and select default values when prompted. Next, we'll select the edition of SQL Server to install from the dropdown on the **Product Key** page.

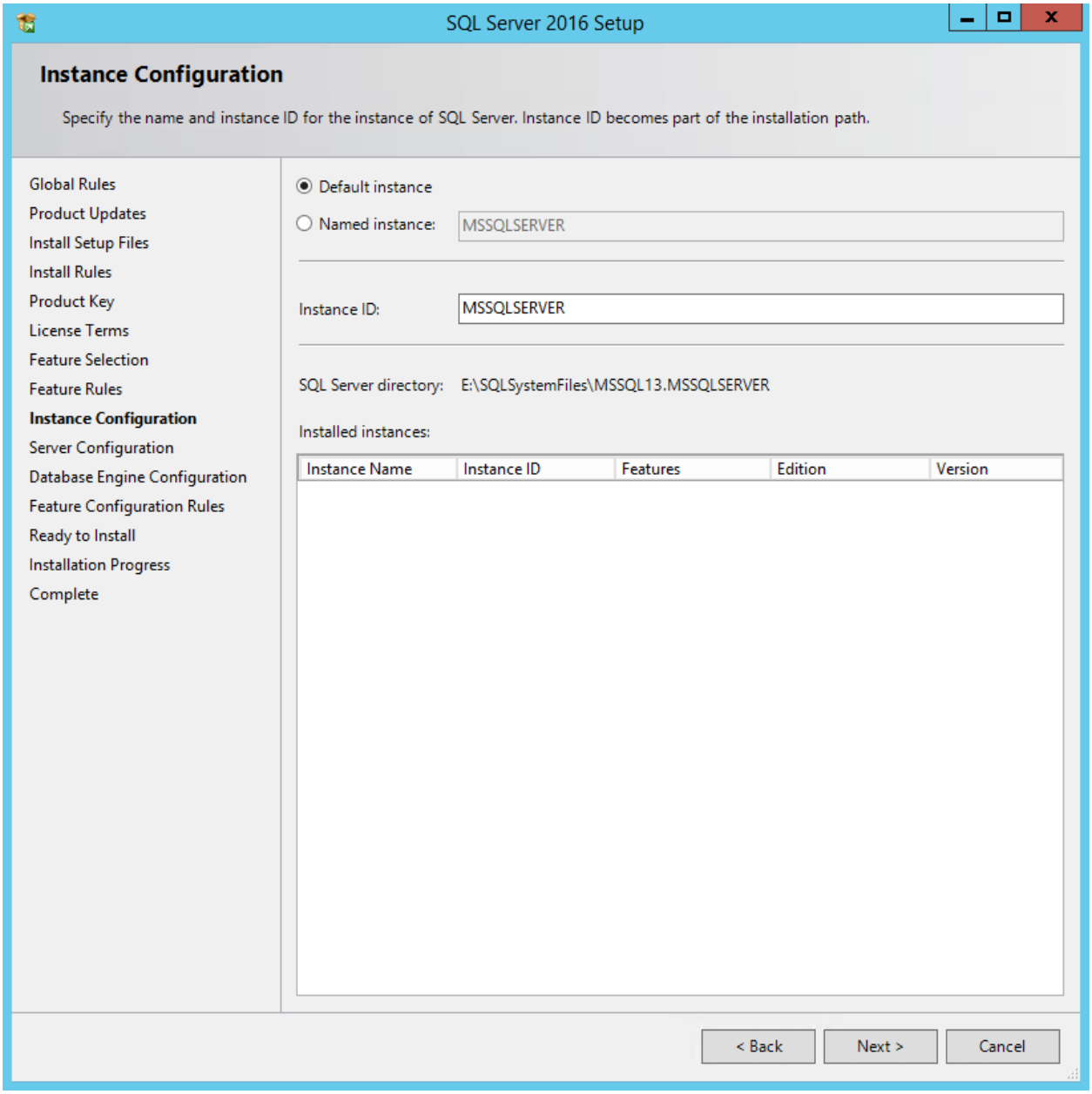
III. Feature Selection



From the **Feature Selection** page, we're able to select which SQL Server tools to install. We only want to install those that will be utilized. We can always install a feature later, if it becomes needed. Select **Database Engine Services**. It provides the core service for storing, processing, and securing data.

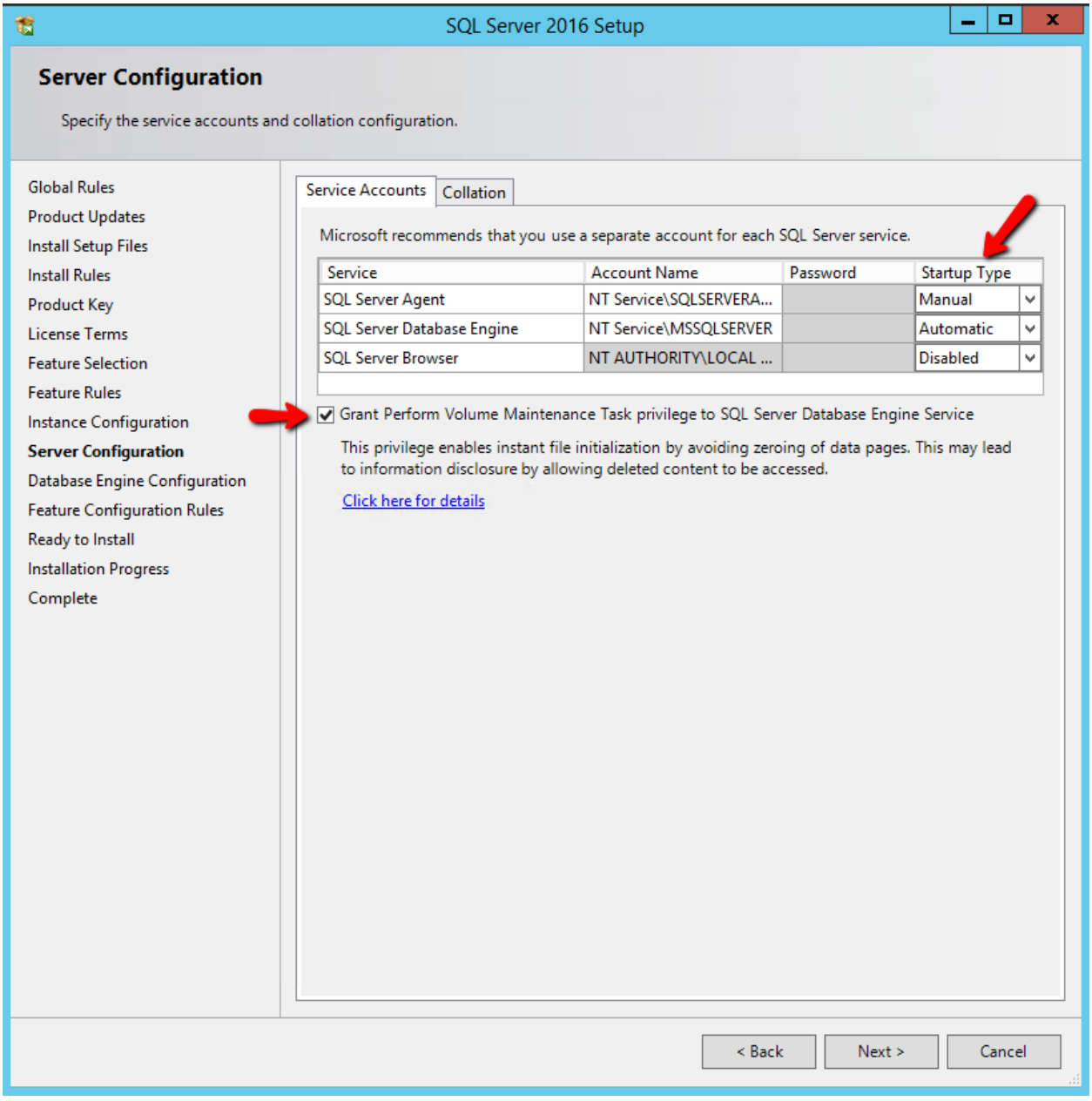
You can also specify the root directory for the instance. If possible, you want to install the root directory on a drive other than the C drive. SQL Server will try to will consume as much memory as it is able to, so it could crowd out the operating system, not allowing it to run.

IV. Choosing Instances



If you're only installing one instance on a server, you should choose **Default instance**. You would use a **Named instance** if you plan to have multiple instances installed.

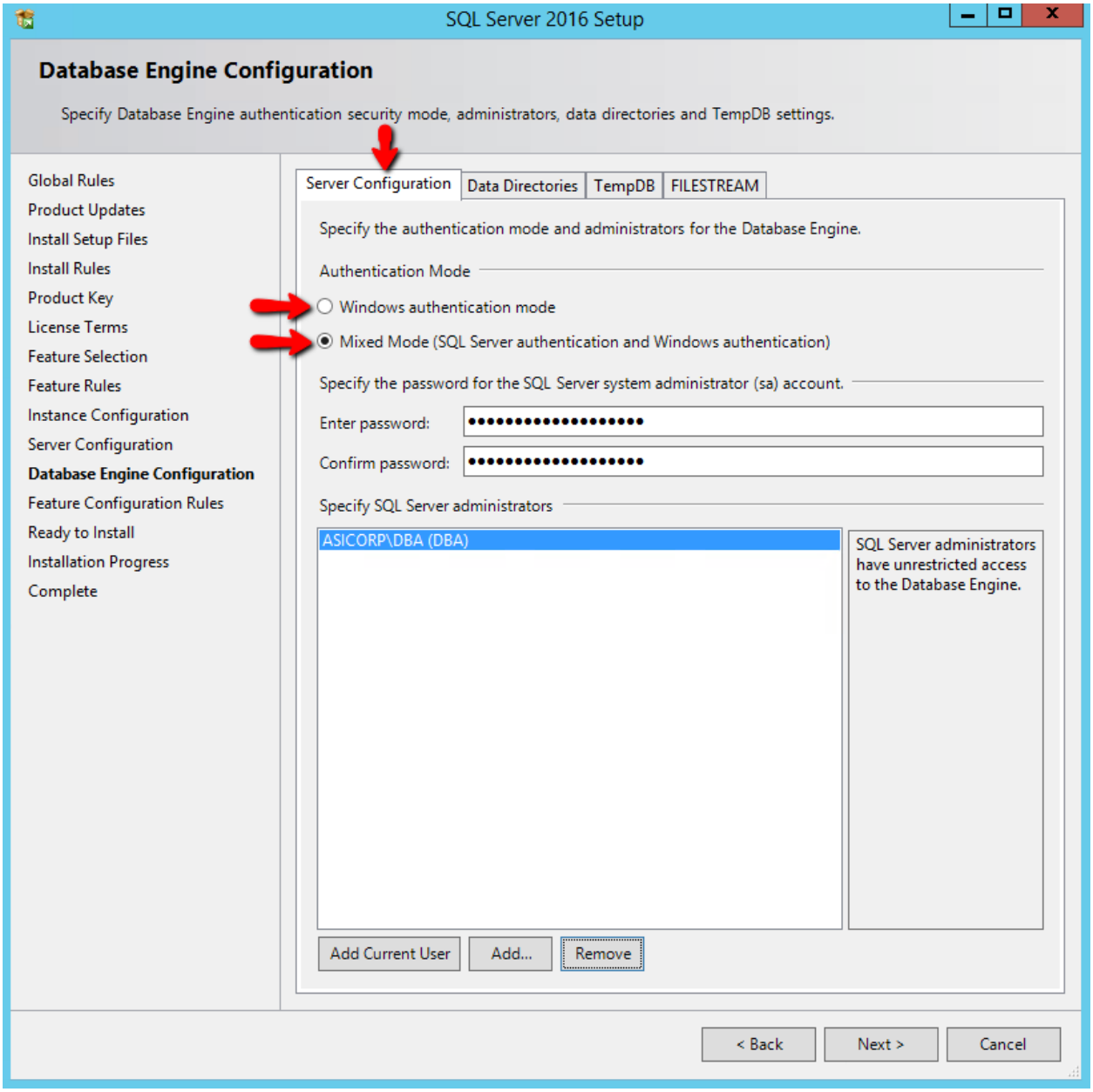
V. Server Configuration



Select the Services to run when the server is running. If we're going to use Agent Jobs frequently, **SQL Server Agent** should be set to **Automatic**.

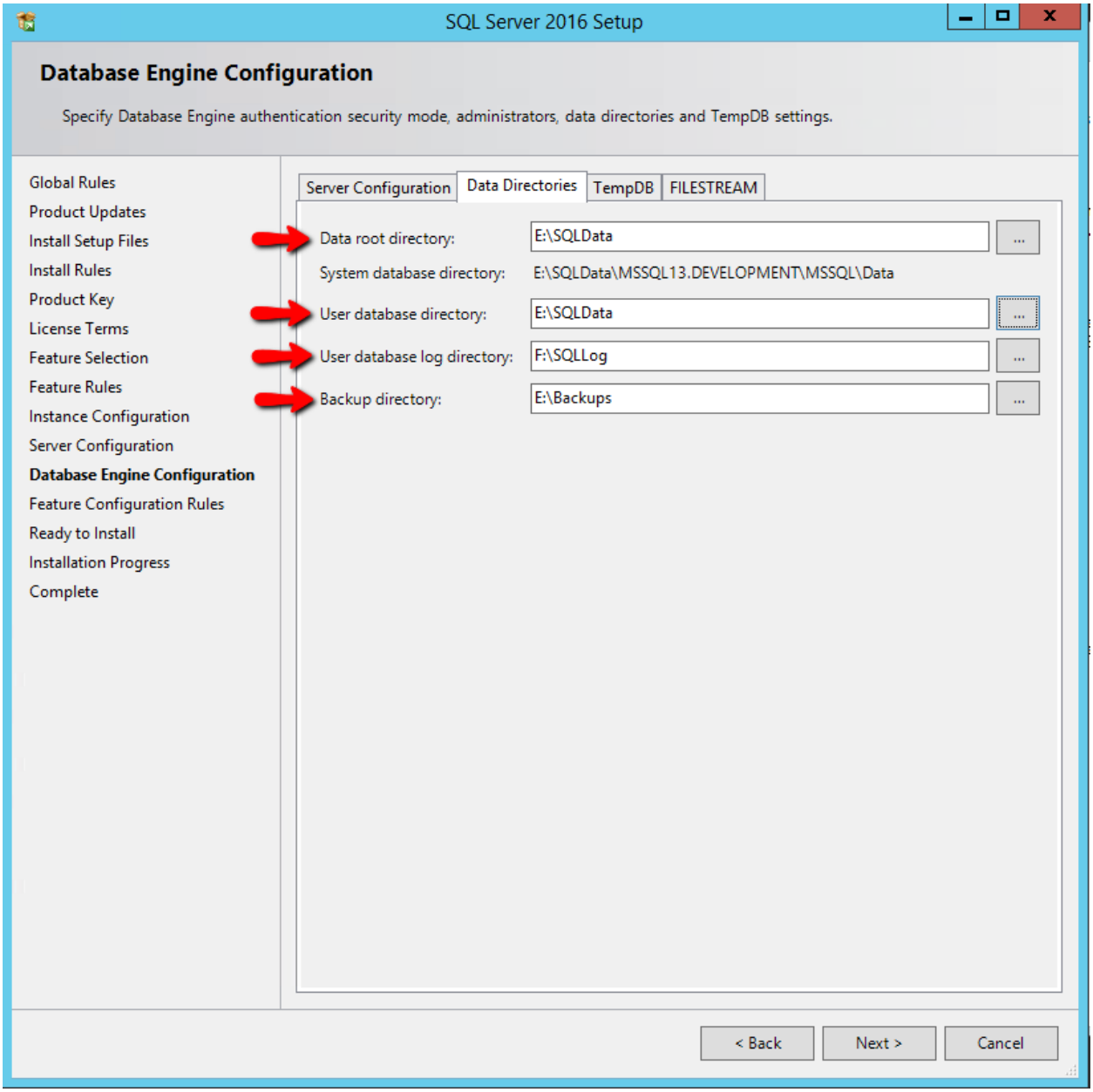
Grant Perform Volume Maintenance Task privilege to SQL Server Database Engine Service will allow SQL Server to immediately initialize new bytes, without it having to zero out deleted record before writing new data.

VI. Database Engine Configuration – Server Configuration



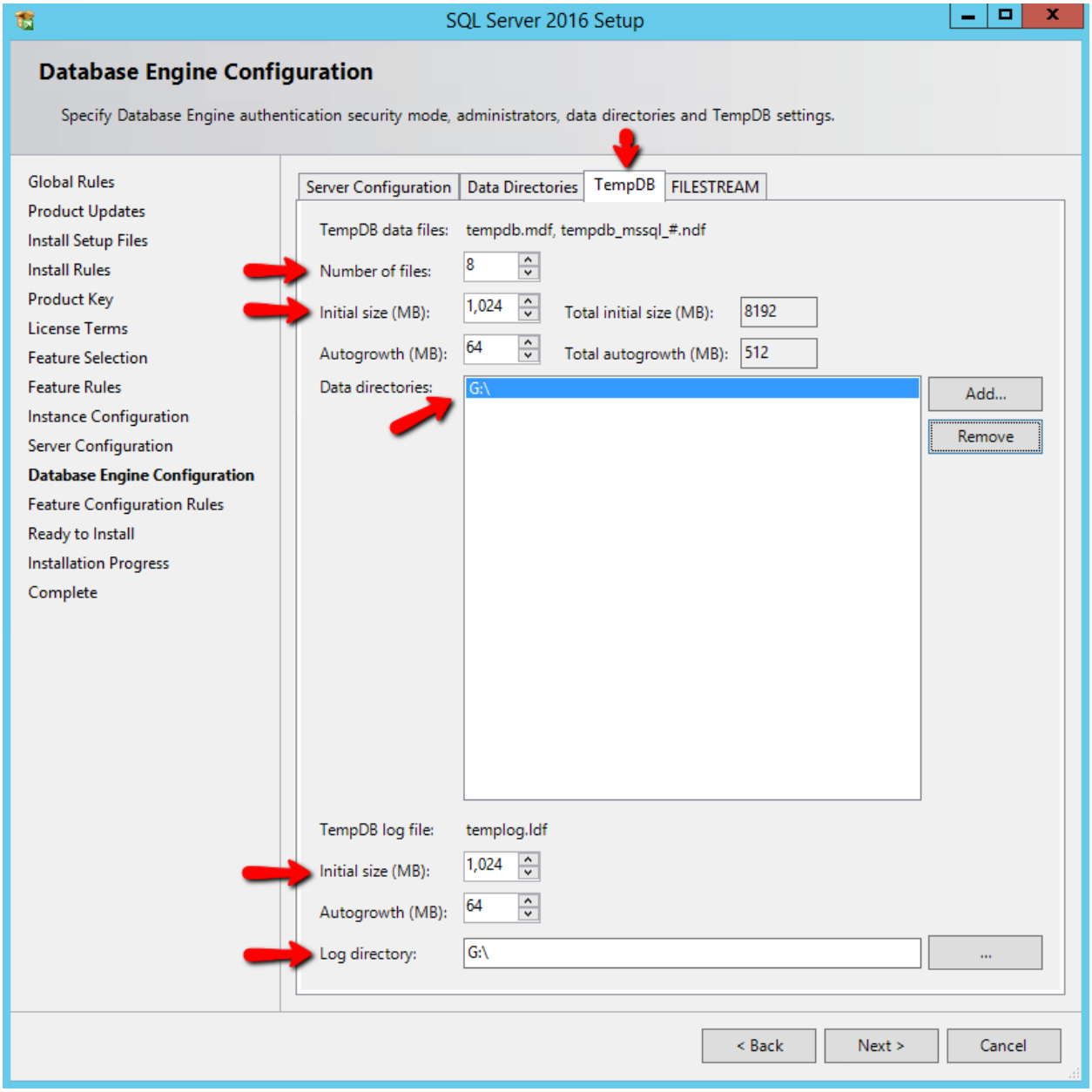
Under **Database Engine Configuration**, we're able to select the **Authentication Mode**. If selecting **Mixed Mode**, we need to set the SQL Server administrator to a DBA account.

VII. Database Engine Configuration – Data Directories



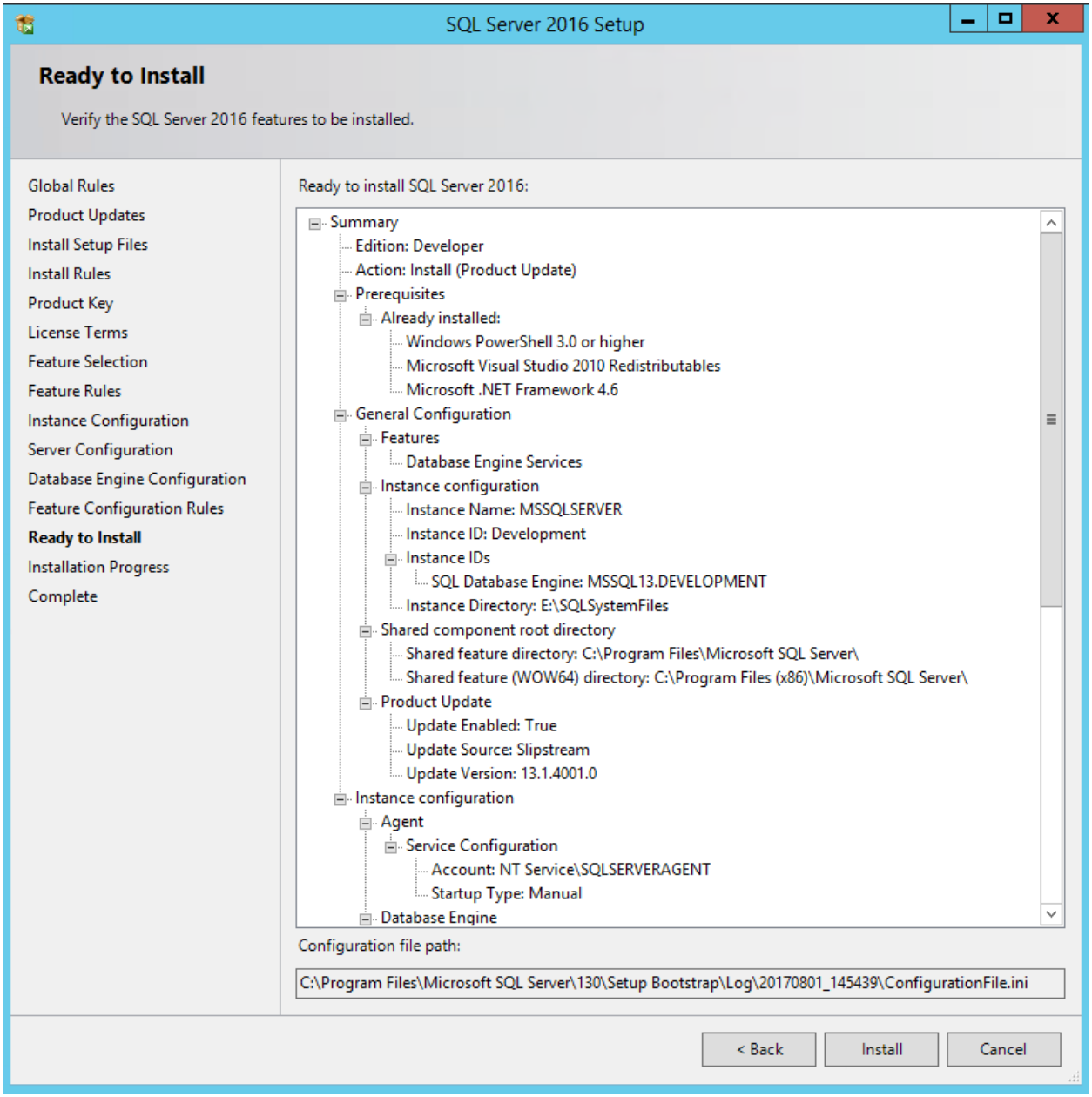
Under the **Data Directories**, we can specify the file paths for data, user database directory, user database log directory, and backup directory. It is best practice to put data, log, and backup files all on separate drives.

VIII. Database Engine Configuration – TempDB



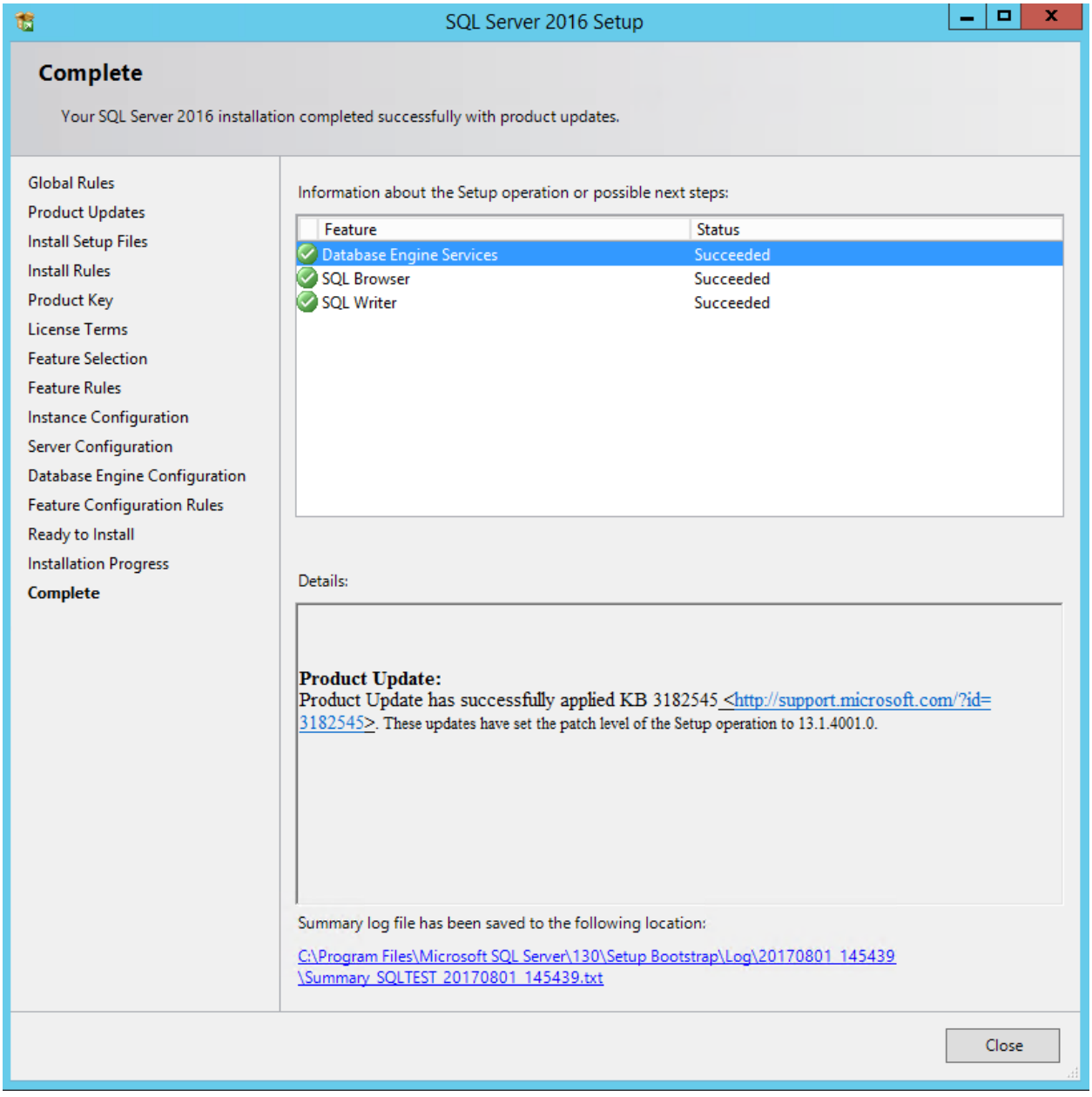
It is best practice to set the number of files to the number of cores you have. Also, by default, the initial size (MB) is set to 8 (MB). SQL Server allows us to increase the initial size to 1 GB. The temp database should also be placed on its own drive.

IX. Install



Next, you can review all of your settings, and then choose **Install**.

X. Installation Complete



SQL Server should now be installed and ready for specific configurations, such as setting up maintenance jobs and performance tracking stored procedures.

Configuring SQL Server

I. Set Trace Flag 1118 (for SQL Server 2014 and prior)

A.

```
DBCC TRACEON (1118, -1);
GO
DBCC TRACESTATUS(1118, -1);
GO
```

B. **SQL Server Configuration Manager** → **SQL Server Services** → right click on **SQL Server** → **Properties** → on the **Start Parameters** tab, type "-T1118" → click **Add**

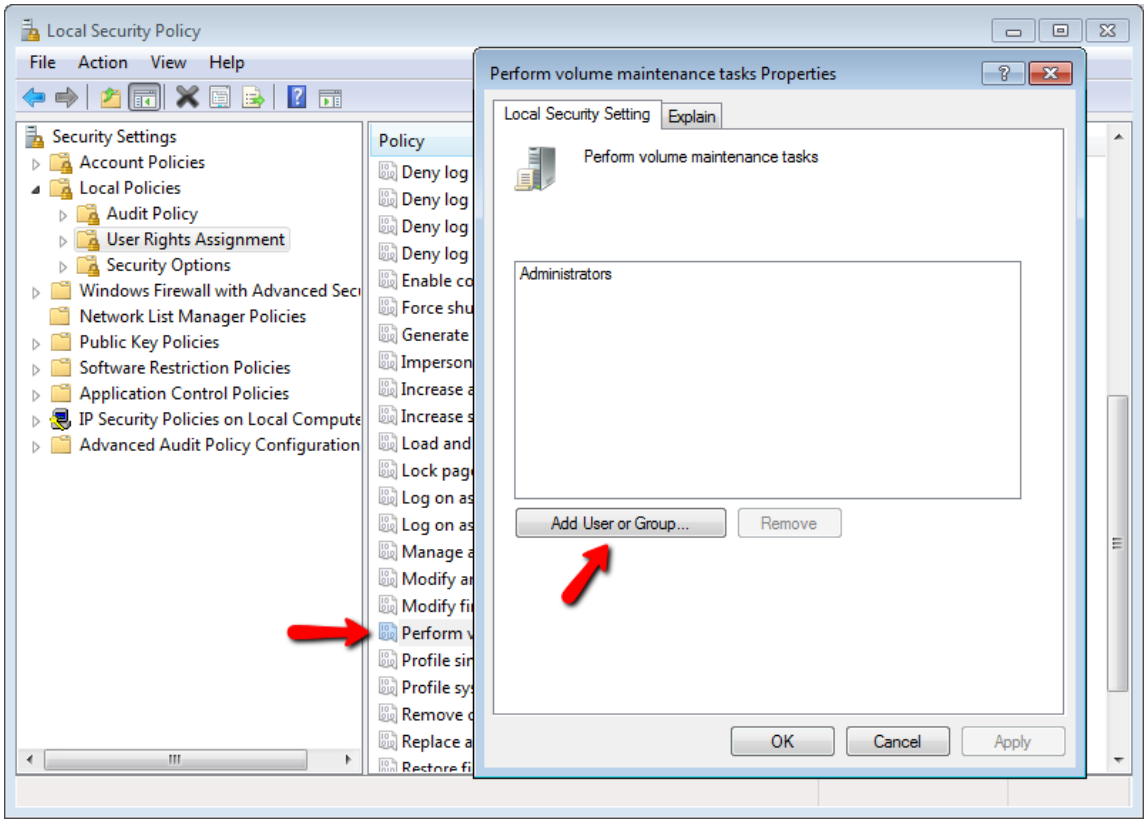
C. Trace Flag 1118 is used to allocate a Uniform Extent instead of Mixed Extents to minimize contention in extent allocation. If this trace flag is enabled, then the first 8 data pages for tables were also Uniform Extents rather than Mixed Extents.

In SQL Server 2016, this uniform extent allocation is the default behavior, and we can change this behavior if required by using an ALTER DATABASE command.

1. Uniform extents are owned by a single object; all eight pages in the extent can only be used by the owning object.
2. Mixed extents are shared by up to eight objects. Each of the eight pages in the extent can be owned by a different object.

II. Enable Instant File Initialization for Service Account (for SQL Server 2014 and prior)

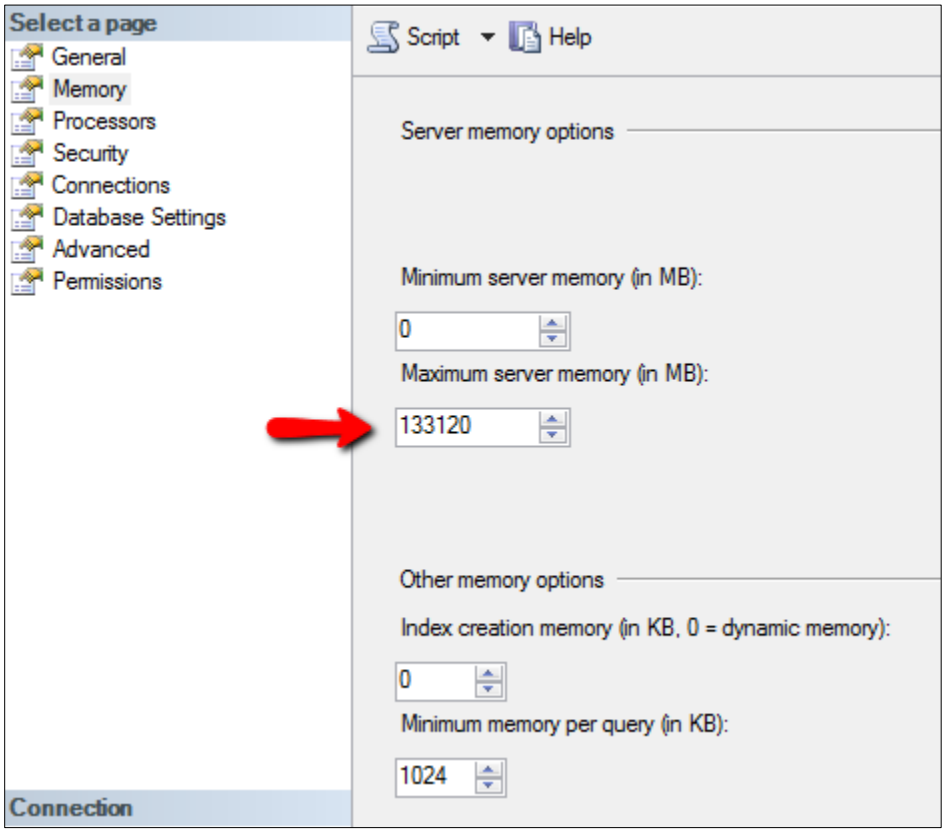
A. From the start menu → **Local Security Settings** → **Local Policies** →**User Rights Assignment** → **Performance Volume Tasks** → Add your SQL Server Service Account, and Click **OK** → Restart SQL Server Service



B. Up until SQL Server 2016 there was not an installation option for Instant file Initialization. Without Instant File Initialization, SQL Server first fills the space it needs with zeros. In many cases, writing zeros across the disk space before using that space is unnecessary. Instant file initialization (IFI) allows SQL Server to skip the zero-writing step and begin using the allocated space immediately for data files.

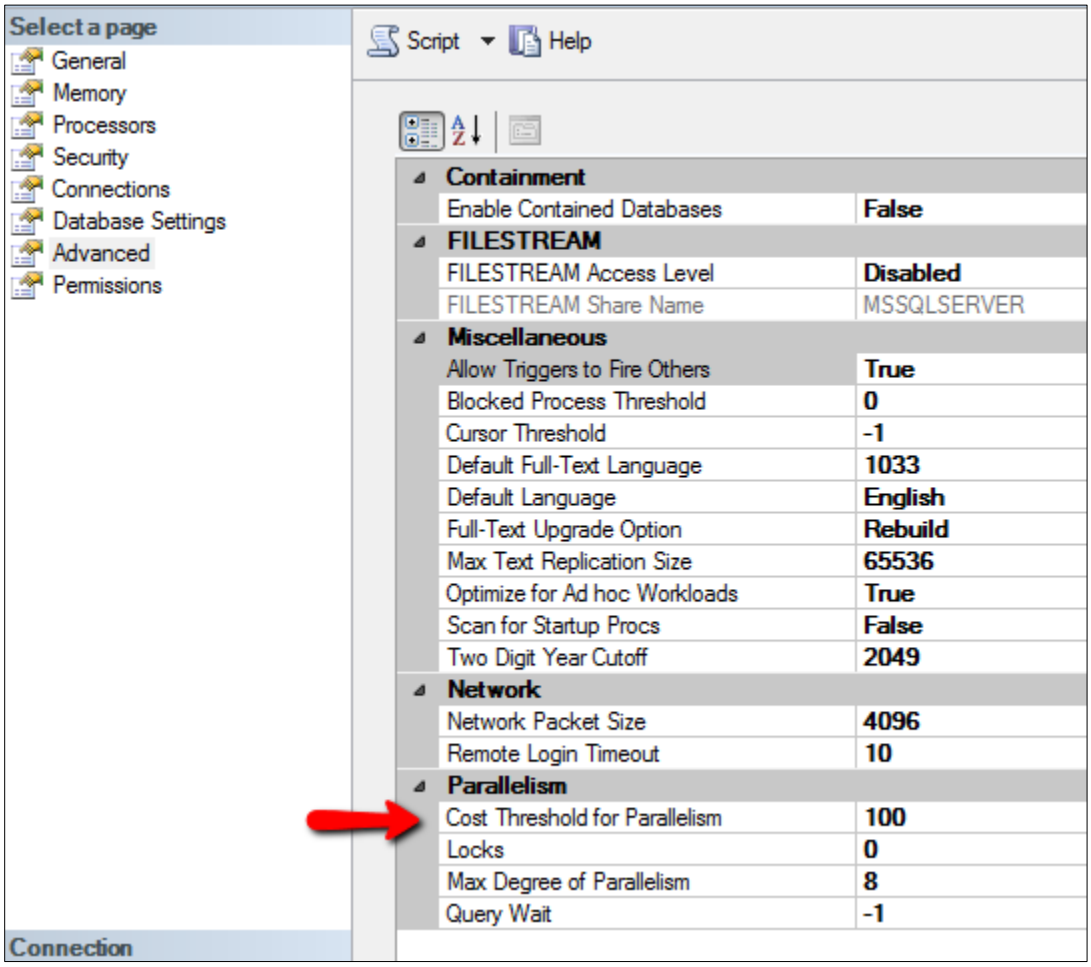
III. Set Max Memory to 90% of system memory to be sure there is enough memory left for the Operating System

A. From **Object Explorer** right click on the server name → **Properties** → **Memory** → Adjust **Maximum server memory (in MB)** to be 90% of system memory



IV. Set Cost Threshold for Parallelism to between 50 and 75

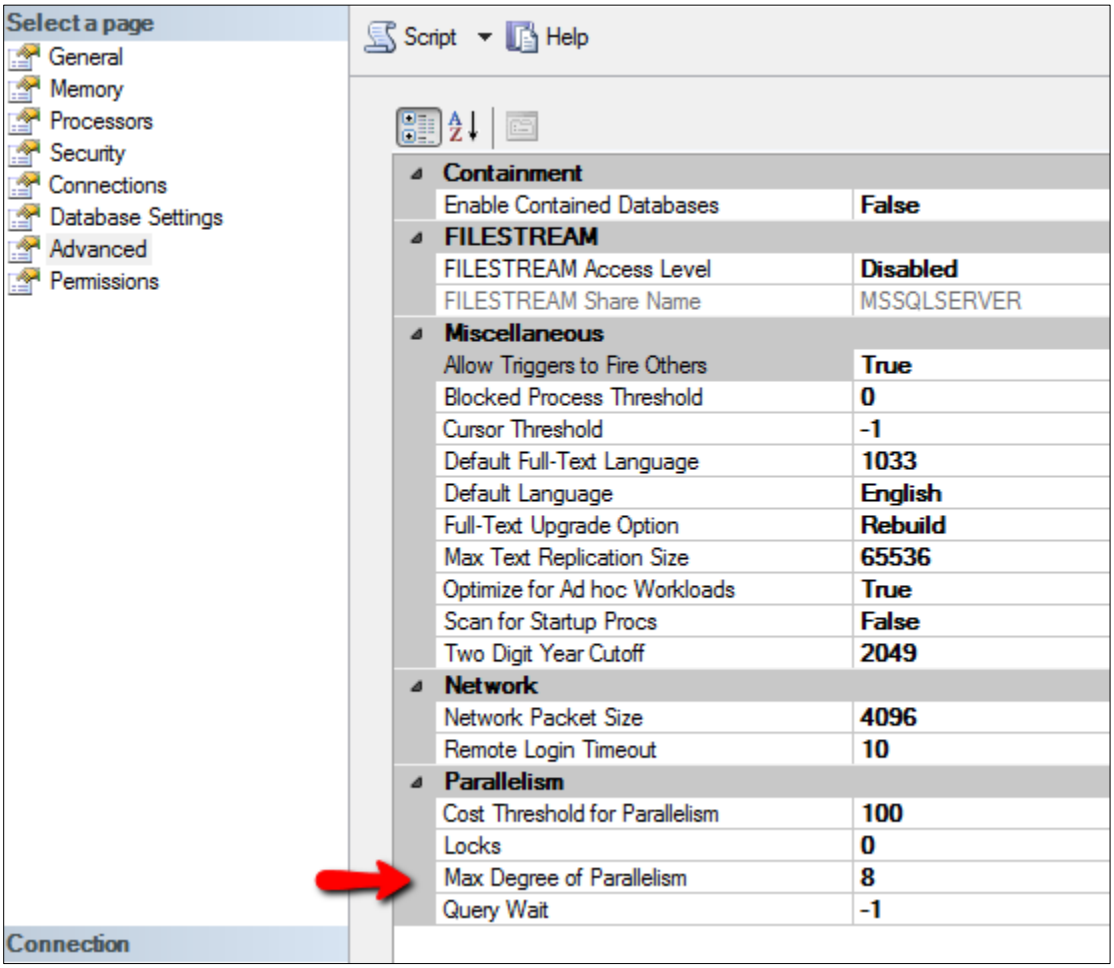
A. From **Object Explorer** right click on the server name → **Properties** → **Advanced** → Set **Cost Threshold for Parallelism** between 50 and 75



Note: If you're setting this on a test or dev server, you'll want to set this to be equal to your production settings, so you can get true performance metrics.

V. Set Max Degree of Parallelism to the number of cores per NUMA node

B. From **Object Explorer** right click on the server name → **Properties** → **Advanced** Set **Max Degree of Parallelism** to the number of cores per NUMA node



VI. Reconfigure Advanced Options

A. Turn on show advanced options

```
sp_configure 'show advanced options', 1
GO
```

B. Check be make sure you won't be changing additional configurations

```
Select name, value, value_in_use
From sys.configurations
Where value <> value_in_use;
GO
```

C. Reconfigure

```
RECONFIGURE
GO
```

D. Setup backup compression On (2012 or newer for Standard Edition)

```
EXEC sp_configure 'backup compression default',1;
RECONFIGURE WITH OVERRIDE;
GO
```

E. Turn On Optimize for Ad Hoc

```
sp_configure 'optimize for ad hoc workloads', 1;
GO
```

F. Enable remote DAC

```
sp_configure 'remote admin connections',1;
GO
```

G. Enable Common Language Runtime for SSIS

```
sp_configure 'clr enabled', 1;
GO
```

H. Reconfigure

```
RECONFIGURE
GO
```

VII. Adjust Model Database Settings

```
Alter Database Model Set Recovery SIMPLE;
GO
ALTER DATABASE Model MODIFY FILE ( NAME = N'modeldev', FILEGROWTH = 256MB );
GO
ALTER DATABASE Model MODIFY FILE ( NAME = N'modellog', FILEGROWTH = 128MB );
GO
```

VIII. Adjust Auto Growth Settings on Master and MSDB

```
USE [master]
GO
ALTER DATABASE [master] MODIFY FILE ( NAME = N'master', FILEGROWTH = 10240KB )
GO
ALTER DATABASE [master] MODIFY FILE ( NAME = N'mastlog', FILEGROWTH = 10240KB )
GO
ALTER DATABASE [msdb] MODIFY FILE ( NAME = N'MSDBData', FILEGROWTH = 10240KB )
GO
ALTER DATABASE [msdb] MODIFY FILE ( NAME = N'MSDBLog', FILEGROWTH = 10240KB )
GO
```

IX. Create admin utility database

X. Install Brent Ozar Scripts – Performance and Health Stored Procedures

A. Install all of the Brent Ozar scripts in the utility database

1. sp_Blitz: Overall Health Check
2. sp_BlitzCache: Find the Most Resource-Intensive Queries
3. sp_BlitzIndex: Tune Your Indexes
4. sp_BlitzFirst: Real-Time Performance Advice
5. sp_BlitzWho: What Queries are Running Now

XI. Install Ola Hallengren Scripts – Maintenance Related Jobs and Stored Procedures

A. Adjust script settings – more information available at ola.hallengren.com

1. SET @CreateJobs = 'Y' if a new installation
2. SET @BackupDirectory to root directory path
3. SET @CleanupTime to desired hours
4. SET @OutputFileDirectory = NULL
5. SET @LogToTable = 'Y'

B. Install in dba_Auditlog database

XII. Install Adam Machanic's Who Is Active – activity tracking

A. Replaces systems stored procedure sp_who

XIII. Configure database mail

A. From **Object Explorer** → **Management** → Right click on **Database Mail** → **Configure Database Mail** → Follow the configuration prompts to complete setup using the values in step B below

1. Account Name: SQL Alert
2. Description: Account to send agent alerts and notifications.
3. Email address: outgoing@domainname.org
4. Display Name: <ServerName> SQL Alert
5. Reply email: donotreply@domainname.org
6. server name: dev-smtp.domain.net

XIV. Setup Agent Operator for DBA Alerts

A. Create Operator

```
USE [msdb]
GO
EXEC msdb.dbo.sp_add_operator @name=N'DBA',
   @enabled=1,
   @pager_days=0,
   @email_address=N'dba@domainname.org'
GO
```

XV. Turn off 'Limit size of job history log' since Ola Hallengren scripts will clean up log records over 30 days old.

A. From **Object Explorer** → **Management** → right click on **SQL Server Agent** → **Properties** → **History** → check **Remove agent history** → set **Older than** to *30* and time to *Day(s)*

XVI. Enable Alert System in SQL Server Agent. Set failsafe operator.

A. Fail-safe operator is notified when a SQL Server Agent cannot access system tables in the msdb database.

B. From **Object Explorer** → **Management** → right click on **SQL Server Agent** → **Properties** → **Alert System** → check **Enable mail profile** → check **Enable fail-safe operator**

XVII. Set Agent Alerts

XVIII. Test Alerts

A. Restart Agent to enable mail

1. Right click on **SQL Server Agent** → **Restart**

B. Create a test error

```
RAISERROR('Alert Test',18,1)WITH LOG;
GO
```

XIX. Create Job Categories

XX. Setup Admin Related Jobs

XXI. Check all schedules

XXII. Check that all notifications are set

```
USE [msdb]
GO

SET NOCOUNT ON;

SELECT 'SQL Agent job(s) without notification operator found:' AS [Message]
    ,j.[name] AS [JobName]
FROM [dbo].[sysjobs] j
LEFT JOIN [dbo].[sysoperators] o ON (j.[notify_email_operator_id] = o.[id])
WHERE j.[enabled] = 1
AND j.[notify_level_email] NOT IN (1, 2, 3)
GO
```

XXIII. Run SP_Blitz and correct findings

```
USE [dba_AuditLog]
Execute sp_Blitz
GO
```