APELON

BECAUSE TERMINOLOGY MATTERS
75 Sgt William B Terry Dr, Suite 2005, Hingham, MA 02043

+1 (203) 431-2530
www.APELON.com
www.ApelonDTS.org

DTS 4: MySQL 8 Installation Guide

Table of Contents

A. Introduction

This document will provide guidelines for installing and configuring MySQL 8 for use with the Apelon DTS server. This document covers a Windows installation in the most detail, but the configuration settings specified here are applicable to both Windows and Linux.

This guide will require that DTS be installed on a system per the DTS Installation Guide – up through the "**Preparing Your Knowledgebase"** step. At that point, this guide covers the MySQL 8 specific steps for preparing the knowledgebase. After completing this guide, you will return to the primary DTS documentation to continue the configuration.

B. Requirements

To utilize MySQL 8 as your database for a DTS server, you must have MySQL Community Server 8.0. Currently, this is the only version that is officially supported. DTS also requires a 64-bit installation of MySQL 8.

C. Configuration Quick Start

The following is a 1 page summary of all configuration settings that need to be made to a MySQL 8 server to configure it appropriately for DTS. If you already have MySQL 8 installed and are familiar with MySQL 8, set the following options and execute the queries below. For detailed instructions on configuring MySQL 8, skip this page and read the rest of this installation guide.

C.1 my.ini / my.cnf file

```
[mysqld]
#Customize for your installation
datadir="C:/ProgramData/MySQL/MySQL Server 8.0/data/"

character-set-server=utf8mb4
collation-server=utf8mb4_general_ci
default-storage-engine=INNODB
lower_case_table_names=1
max_allowed_packet=100M
log_bin_trust_function_creators=1
local_infile=1
#Customize for your hardware
innodb_buffer_pool_size=2000M
innodb_flush_log_at_trx_commit=2
innodb_redo_log_capacity=512M
```

C.2 User / Database creation

DTS uses the default username and password of “dts4”. However, you can specify your own database credentials that are most suitable for your environment. You can use the following example, replacing the brackets [] and their values with ones you prefer for your setup.

```
create database dts4;
create user '[dts4user]'@'%' identified by '[dts4password]';
grant all privileges on dts4.* to [dts4user];
create user '[dts4user]'@'localhost' identified by
'[dts4password]';
grant all privileges on dts4.* to [dts4user]@localhost;
flush privileges;
commit;
```

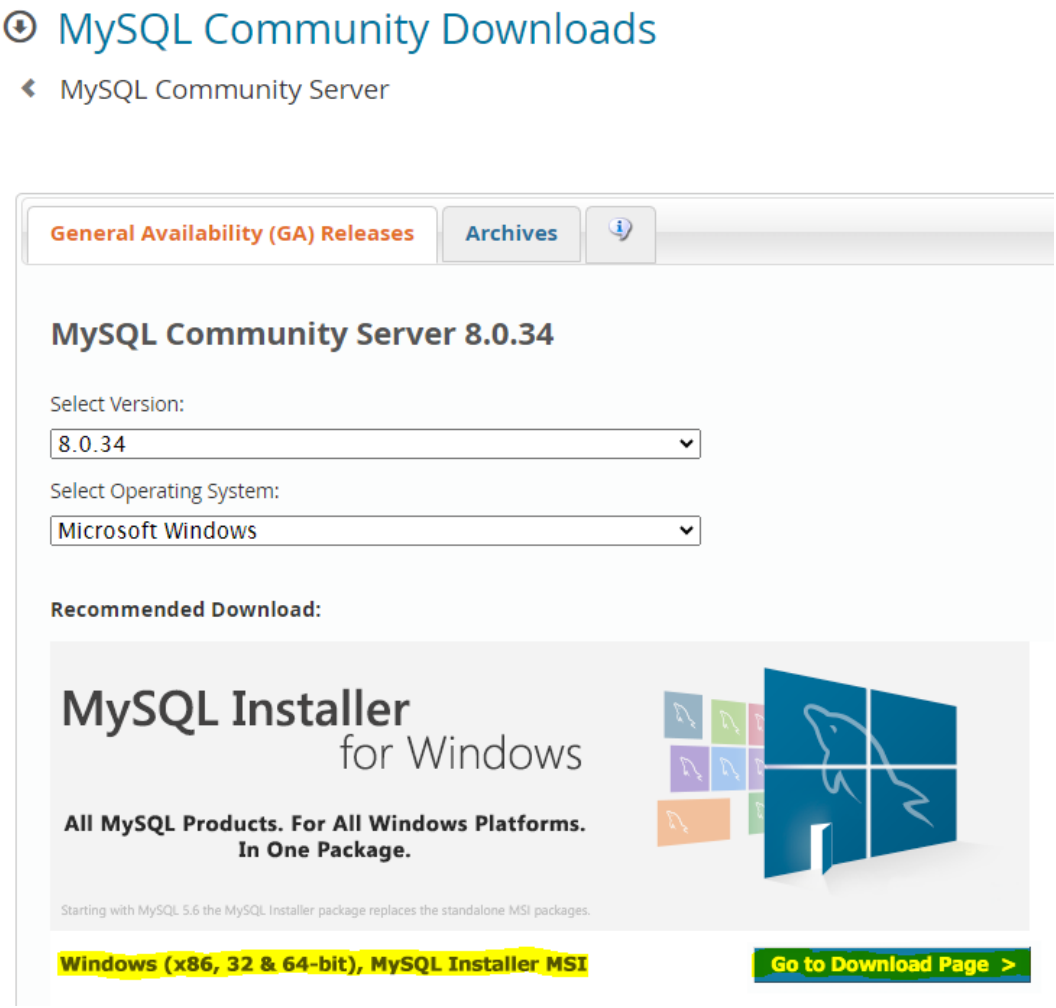
C.3 Tablespace Creation

Follow the steps listed below in Creating and Populating the DTS Schema and Tablespace if you already have MySQL 8 installed and configured for DTS.

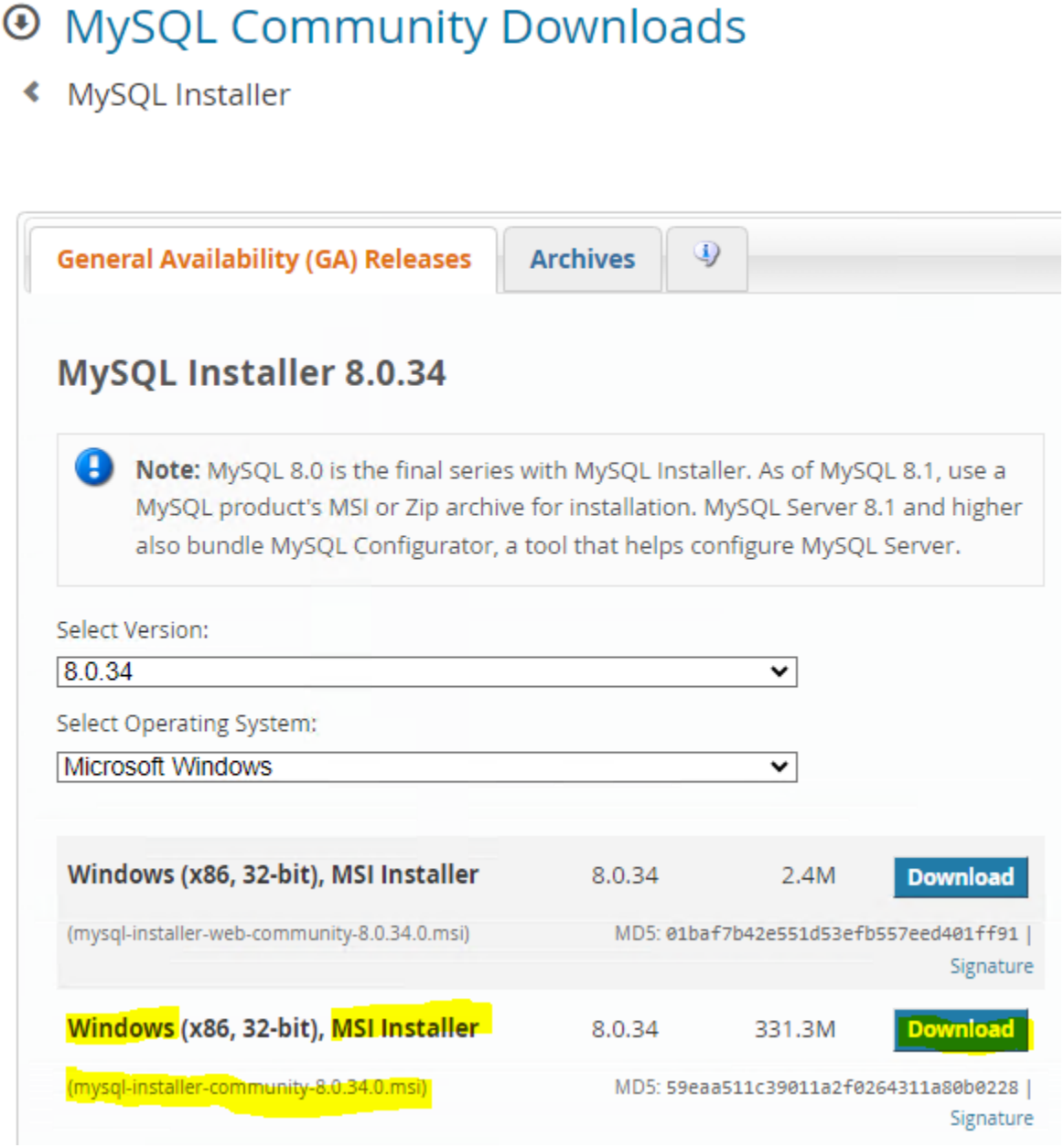
D. Acquire and Install MySQL

D.1 Windows Installation

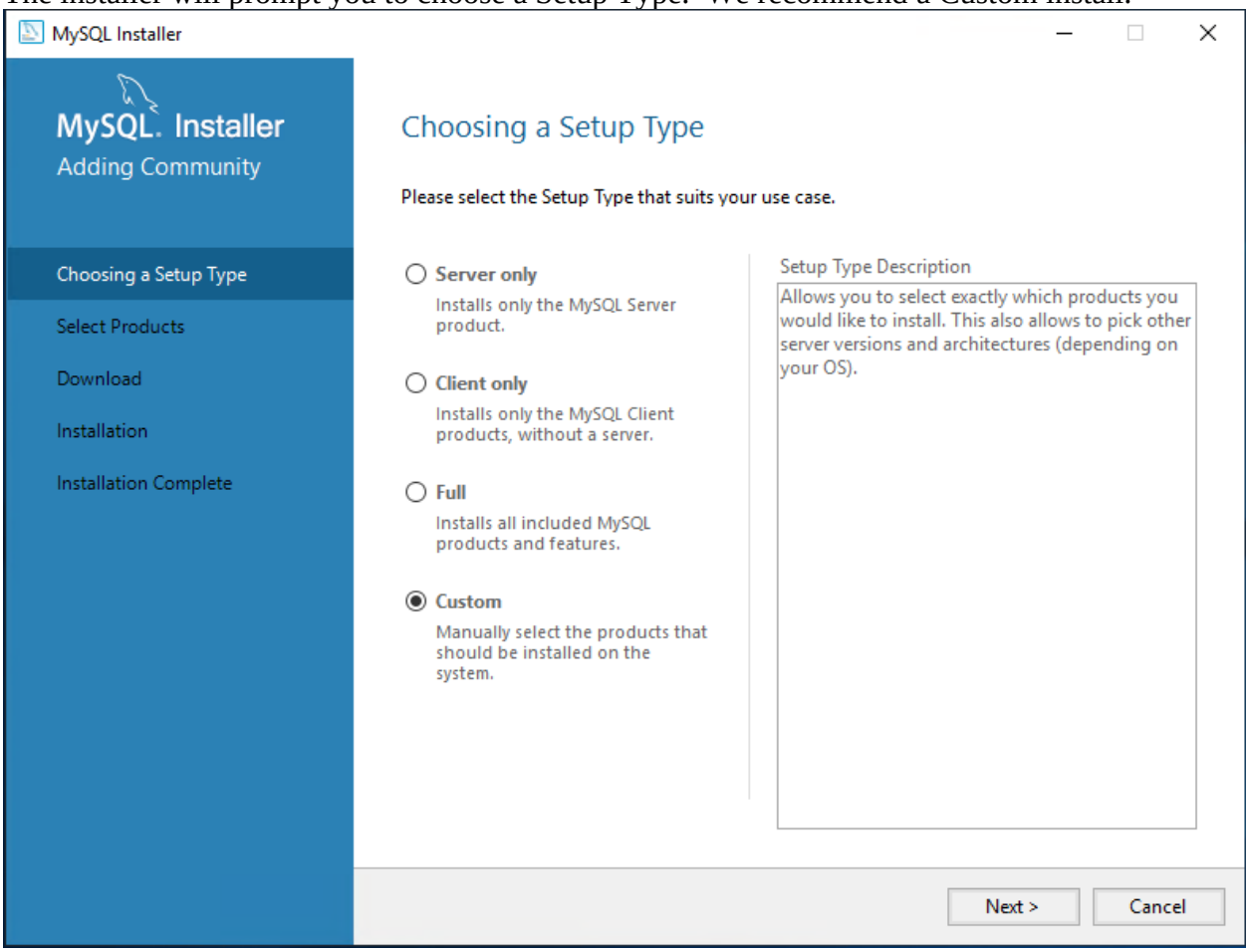
On Windows, the preferred method of installation is to use the single package MySQL Community installer from: https://dev.mysql.com/downloads/mysql/
Select Version 8.0.34 or earlier (DTS 4.8.0 was confirmed against versions 8.0.34, 8.0.33 and 8.0.32).



Choose the Windows 64-bit MySQL Installer MSI (mysql-installer-community-8.0.xx.0.msi) not the web-community.



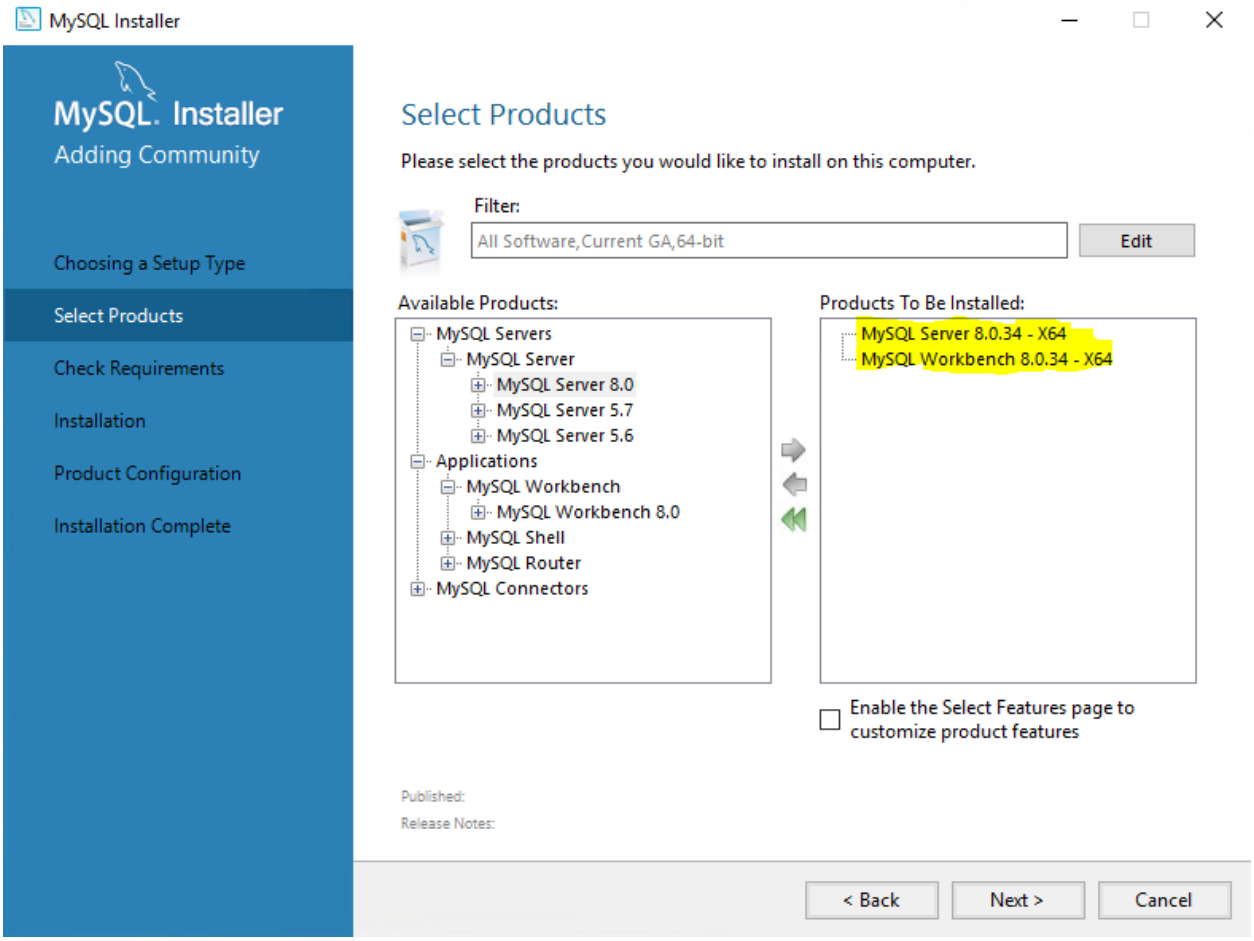
The installer will prompt you to choose a Setup Type. We recommend a Custom install.



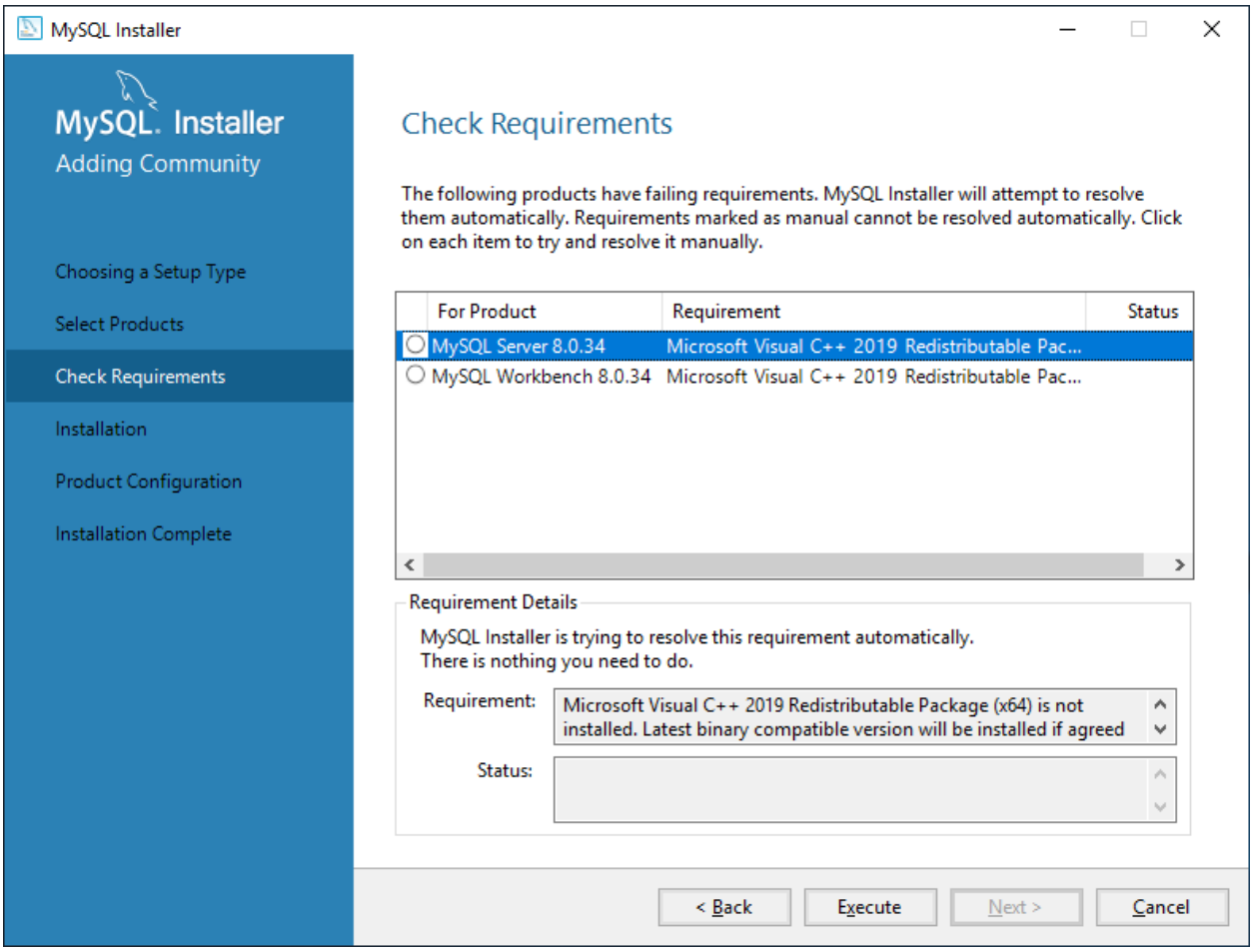
After choosing a Custom Setup, please install the following components:

- Expand MySQL Servers
  - Expand MySQL Server
    - Expand MySQL Server 8.0
      - Add “MySQL Server 8.0.xx – X64”
- Expand Applications
  - Expand MySQL Workbench
    - Expand MySQL Workbench 8.0
      - Add “MySQL Workbench 8.0.xx – X64”

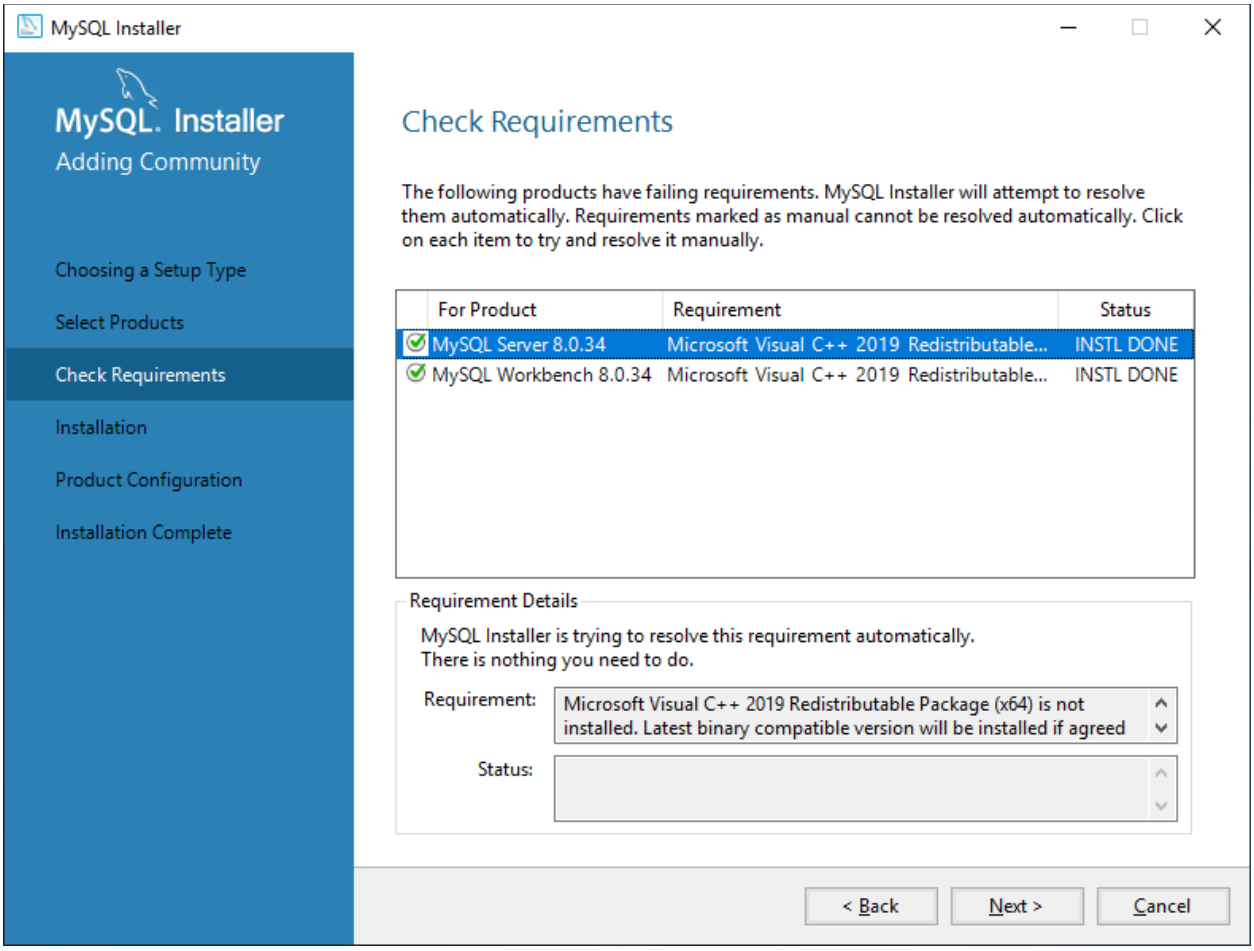
Click [Next>]



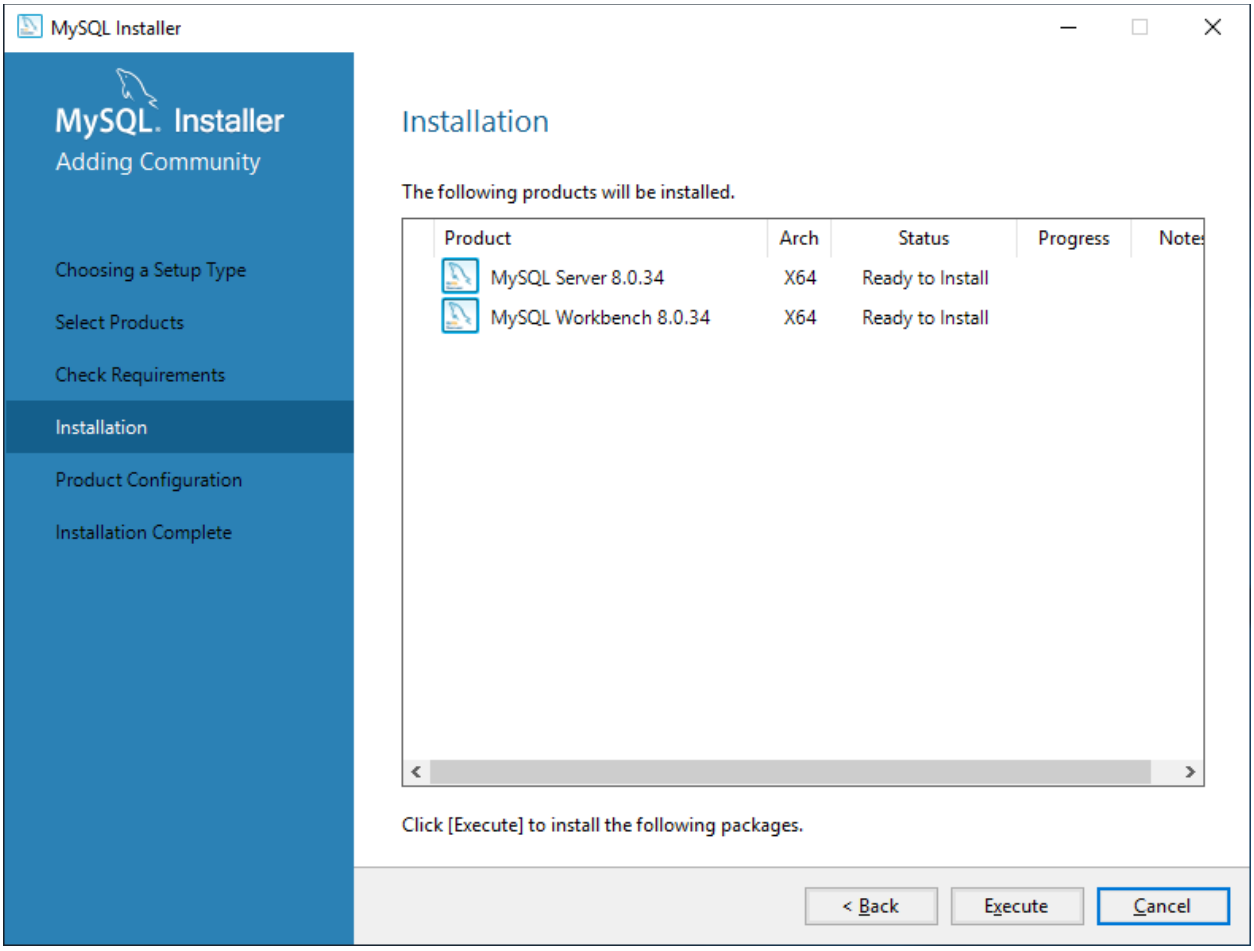
The Window's installer has prerequisite requirements. The installer will prompt you with what prerequisites are required to install if necessary.
Select [Execute] to begin the installation process of any prerequisite requirements.



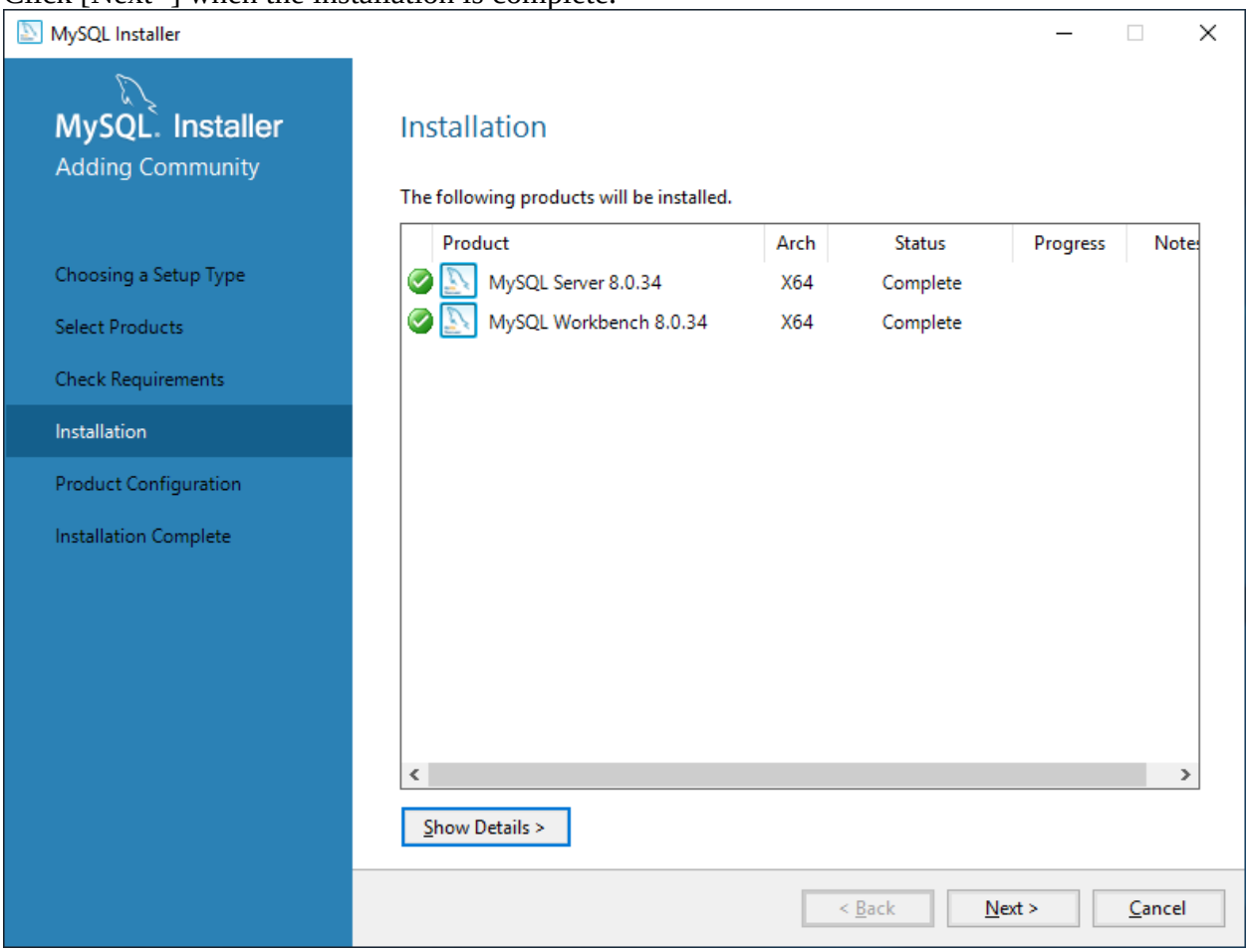
After prerequisite requirements have been installed successfully, click [Next>].



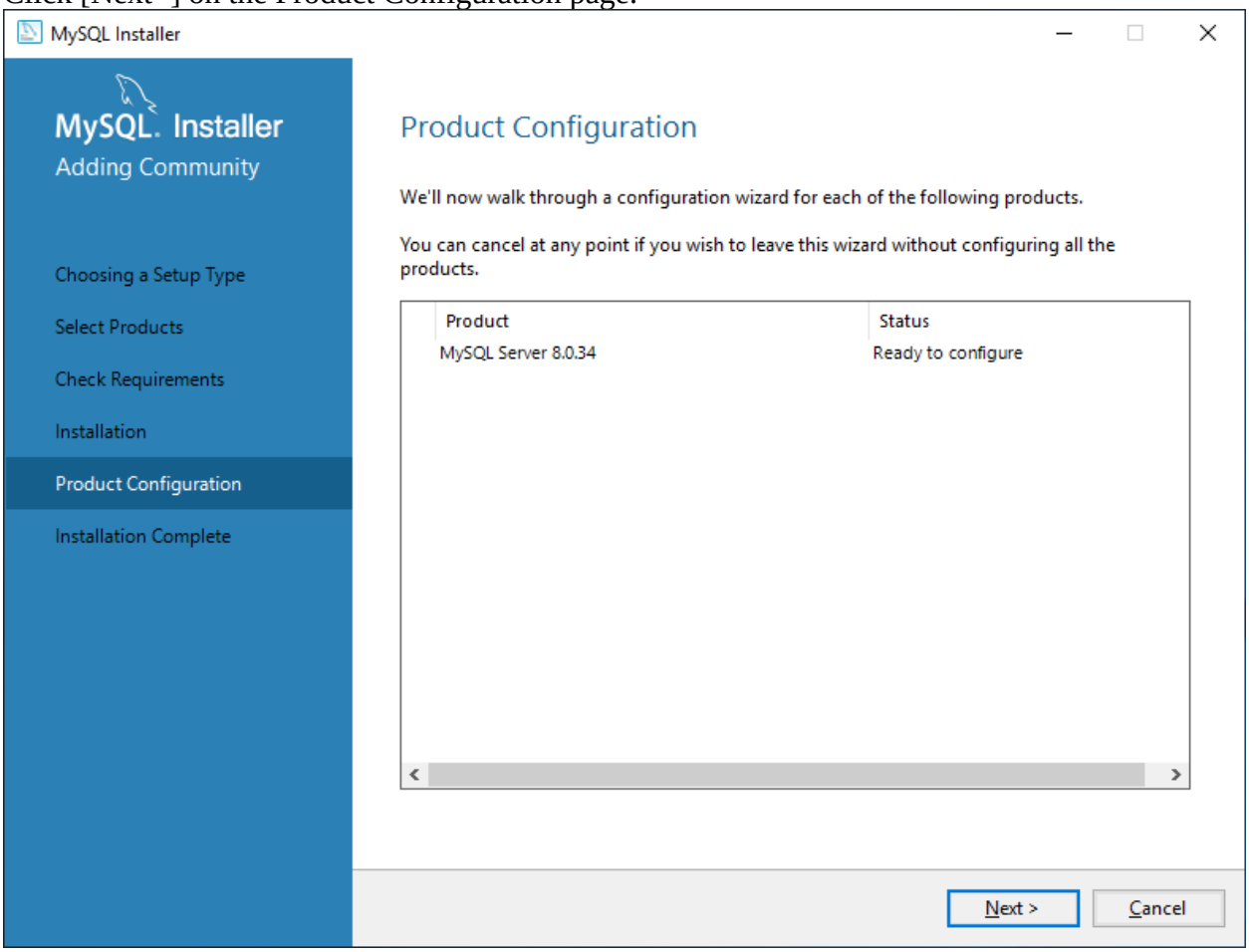
Click [Execute] on the Installation page.



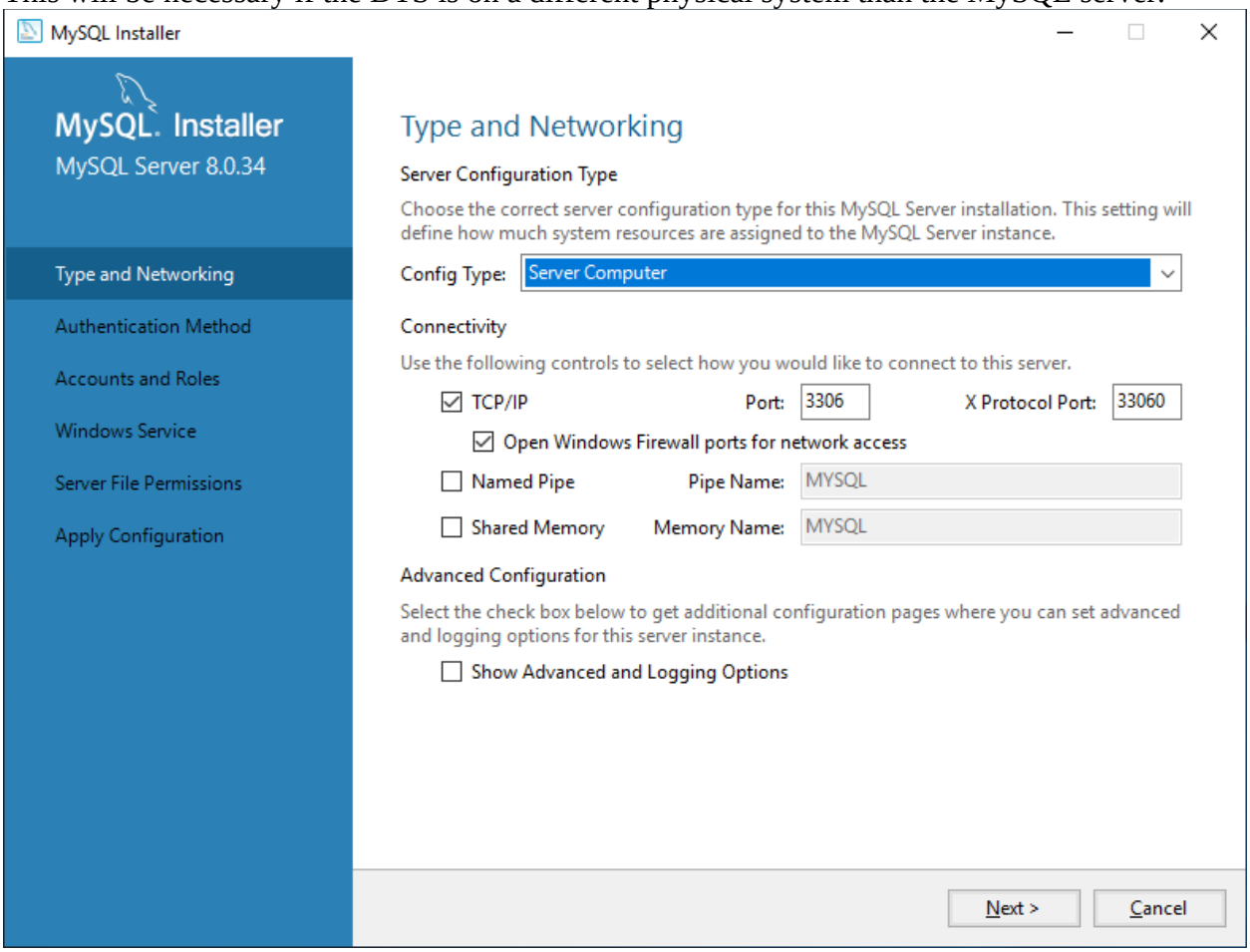
Click [Next>] when the installation is complete.



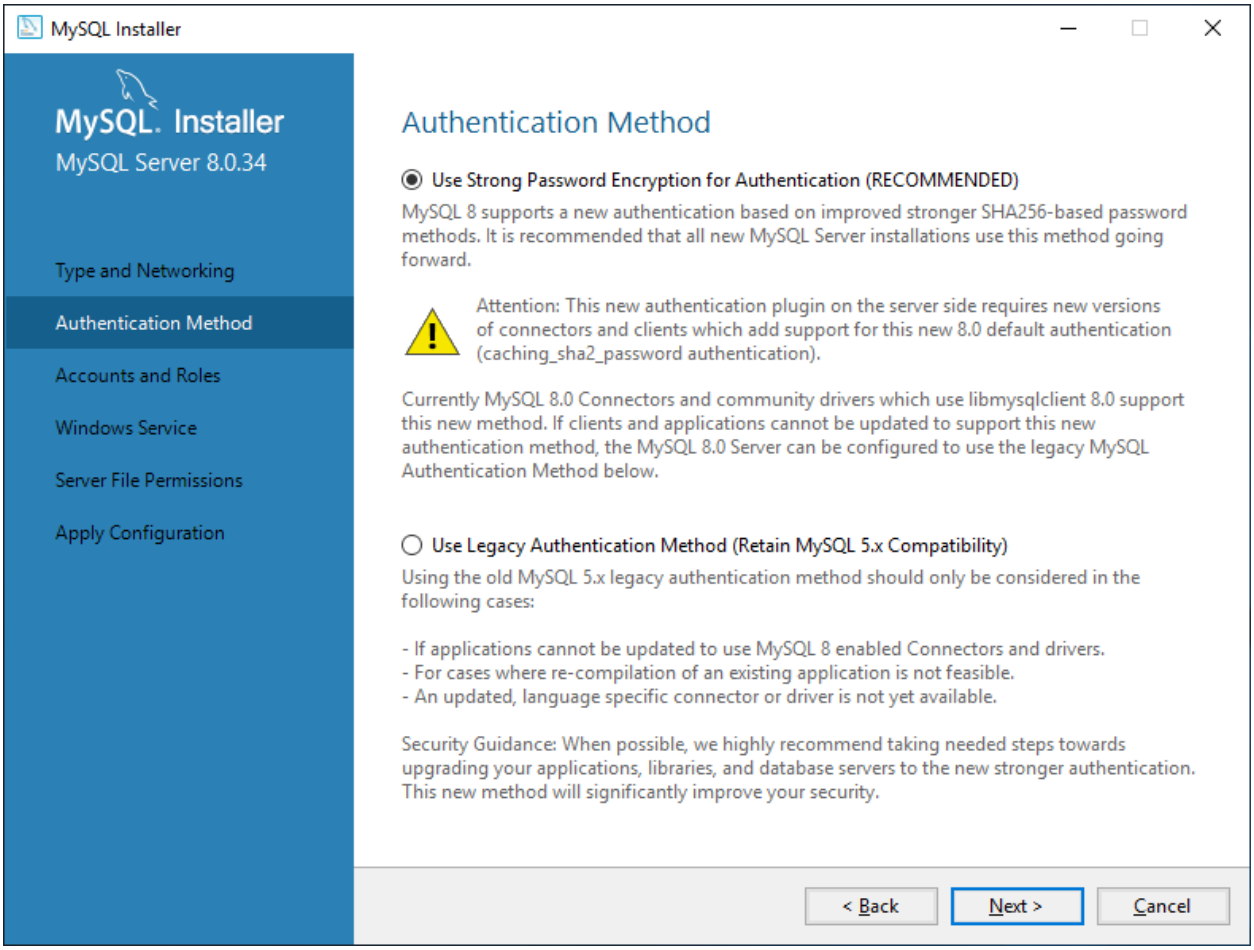
Click [Next>] on the Product Configuration page.



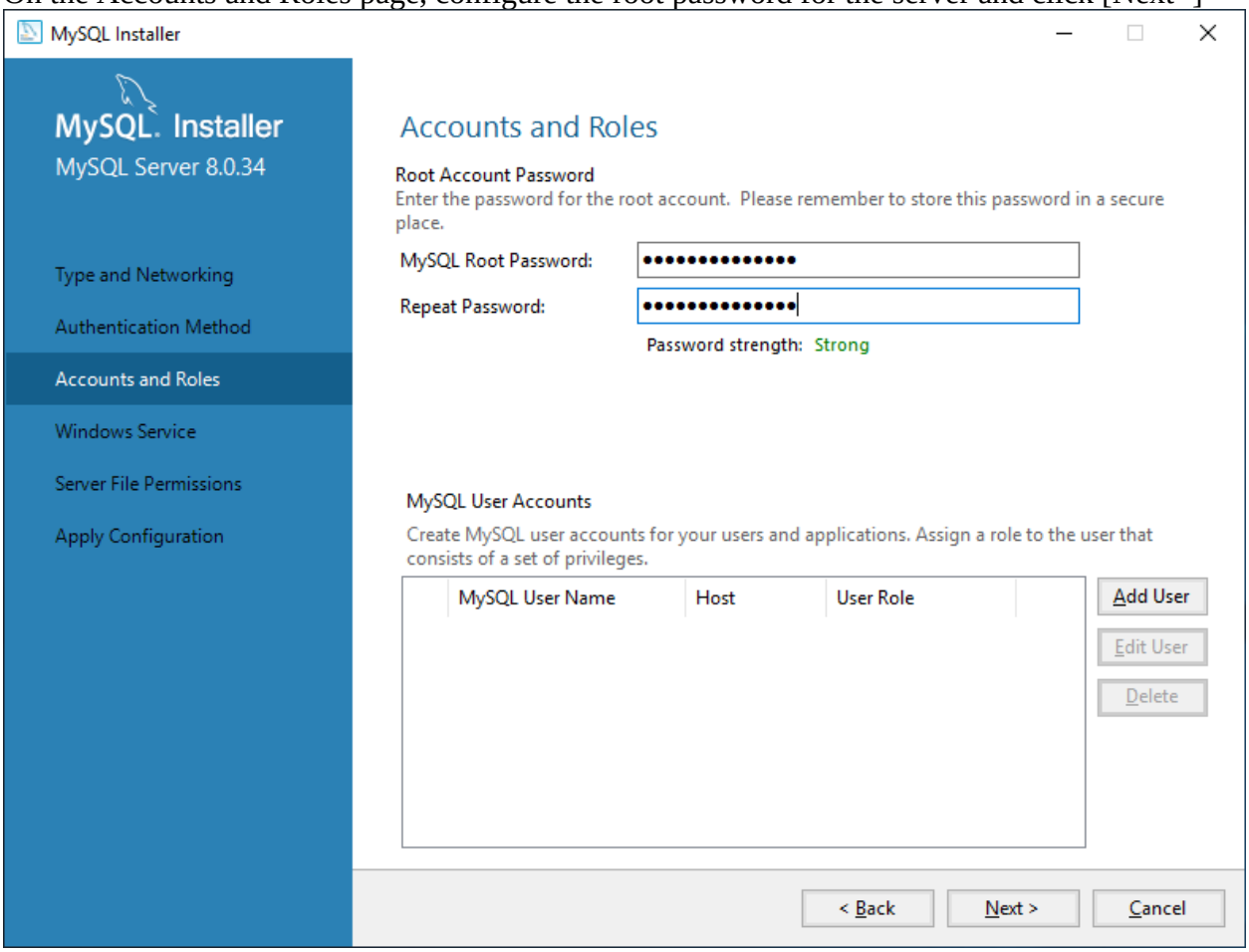
On the Type and Networking pages, the suggested **Config Type** is **Server Computer**. This sets the defaults that MySQL uses for memory usage.
Ensure that **TCP/IP** Networking is enabled with the default **Port** of "3306" and **X Protocol Port** of "33060". Also, select the option to **Open Windows Firewall ports for network access**.
This will be necessary if the DTS is on a different physical system than the MySQL server.



Choose the default Authentication Method of **Use Strong Password Encryption for Authentication**.

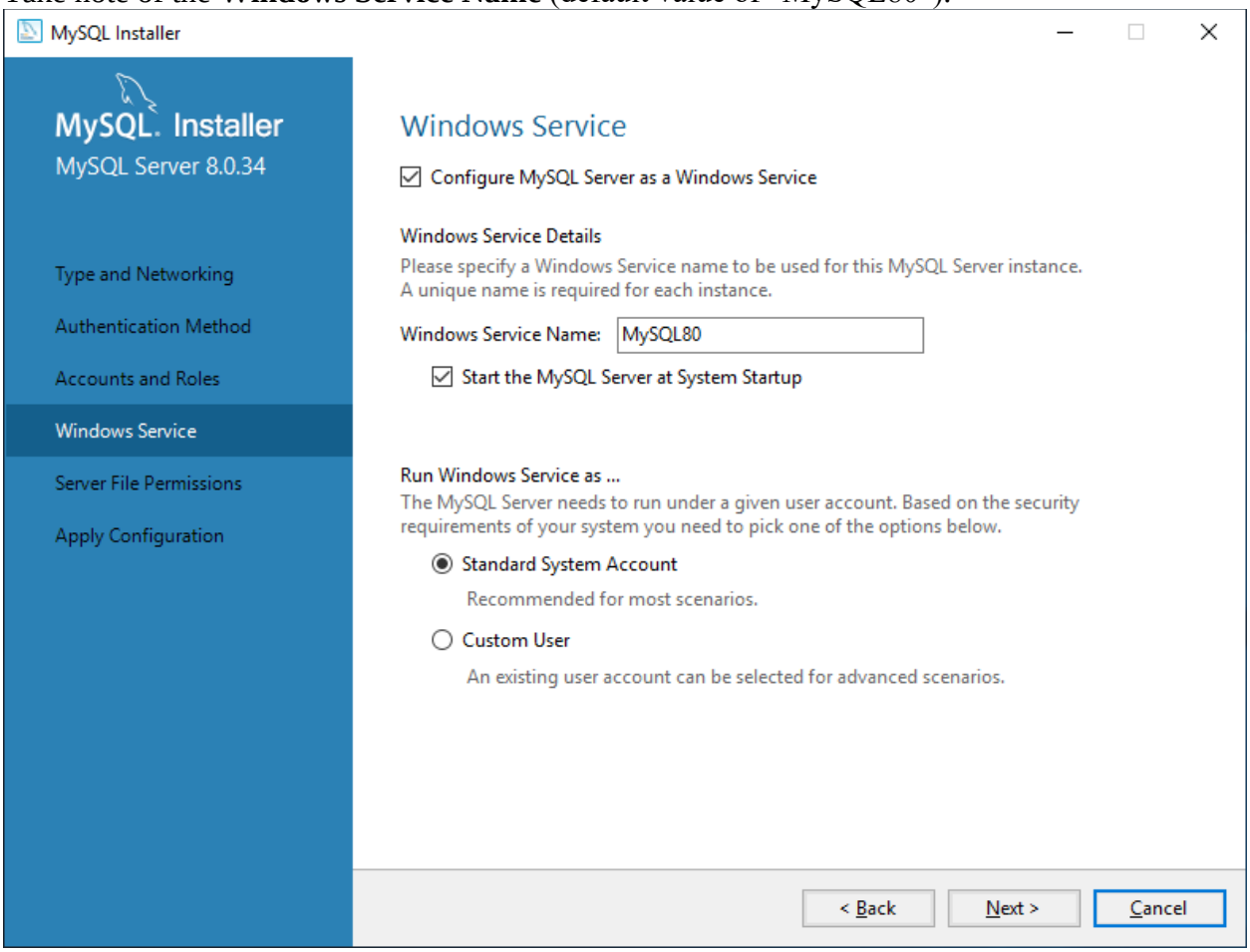


On the Accounts and Roles page, configure the root password for the server and click [Next>]

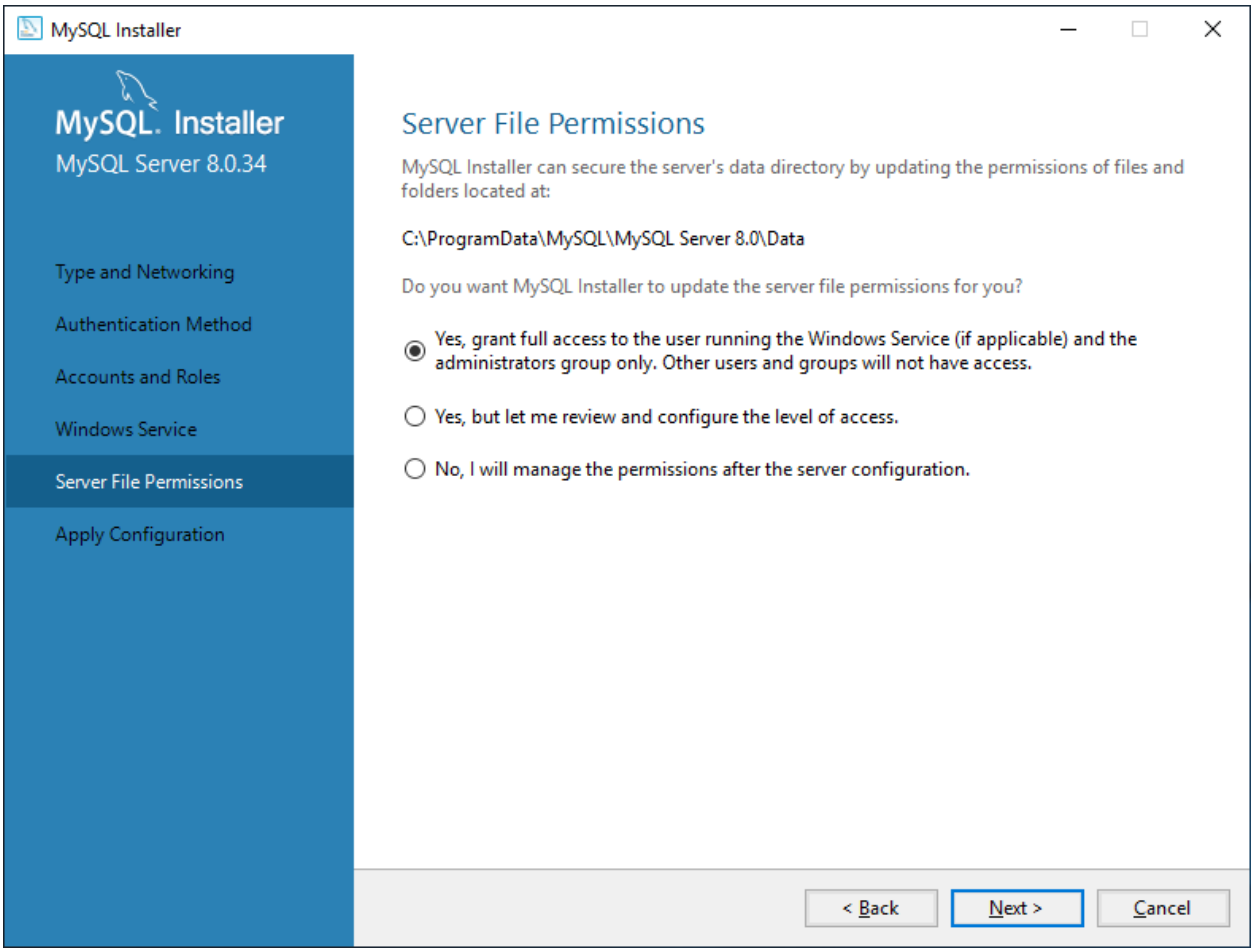


On the Windows Service page, enable the option to **Configure MySQL Server as a Windows Service** and select the **Start the MySQL Server at System** Startup option. Also, select the option to Run Windows Service as a **Standard System Account**.

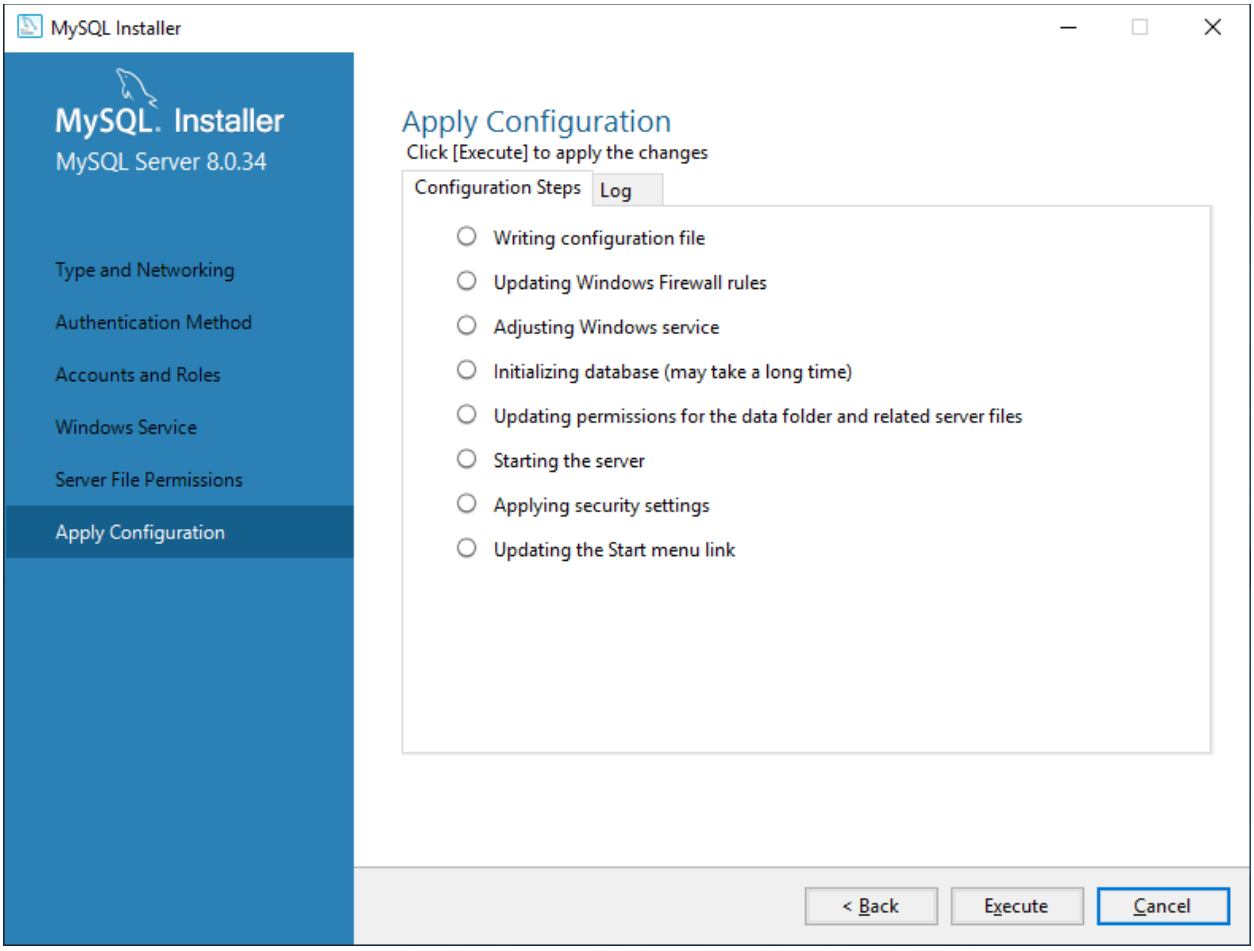
Take note of the **Windows Service Name** (default value of "MySQL80").



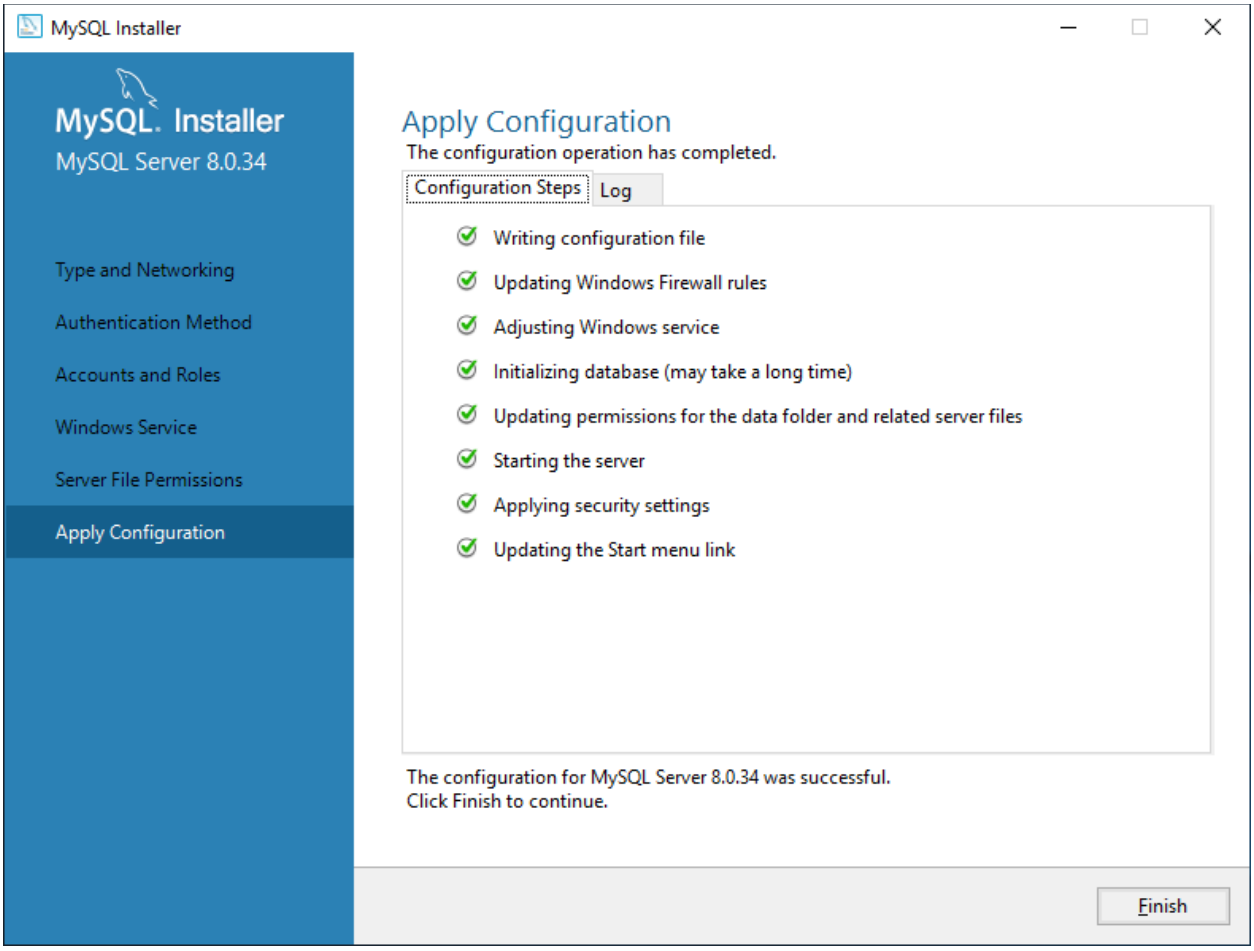
On the Server File Permissions page choose the option: **Yes, grant full access to the user running the Window Service (if applicable) and the administrators group only. Other users and groups will not have access.**



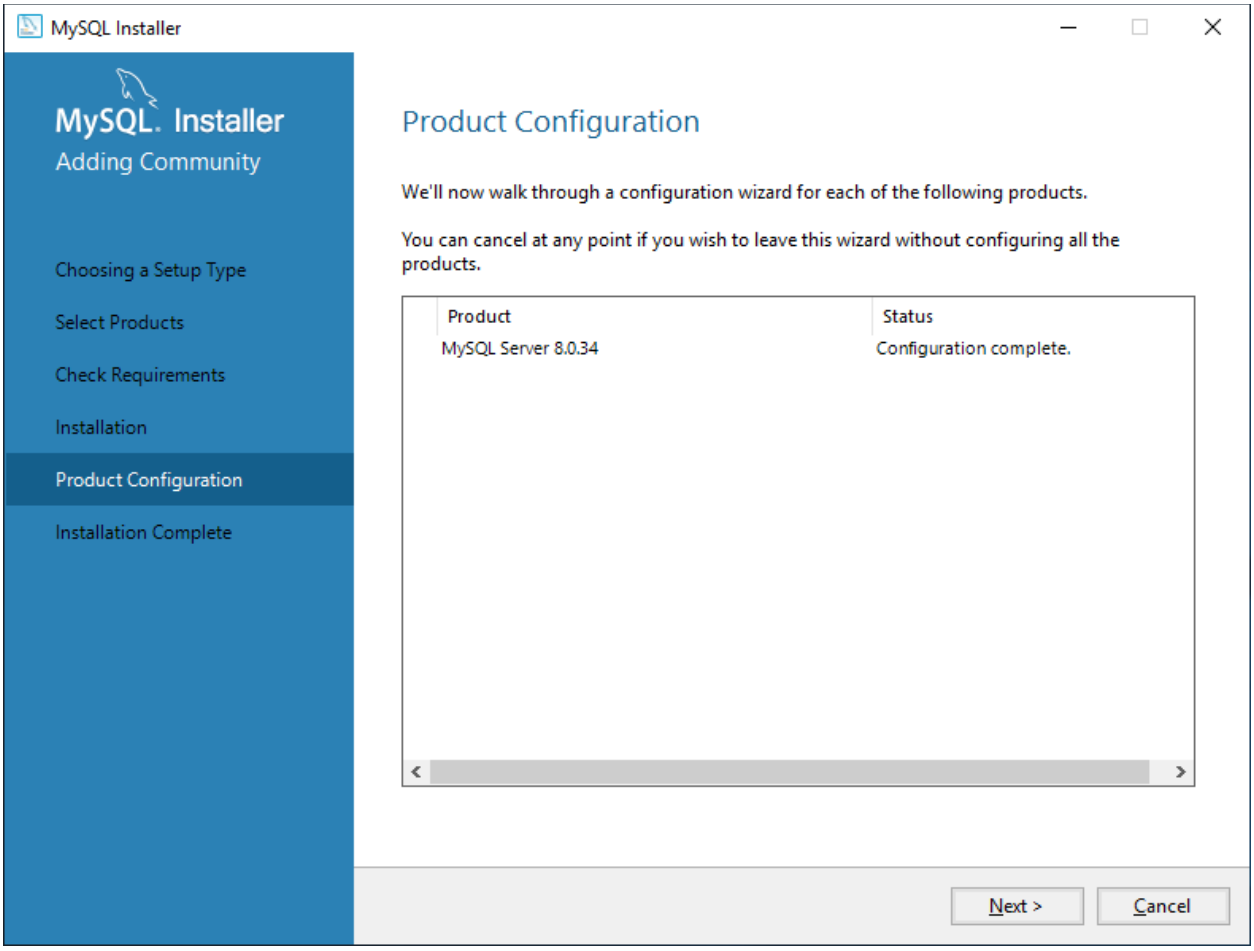
Click[Execute] at the Apply Server Configuration screen of the installer.



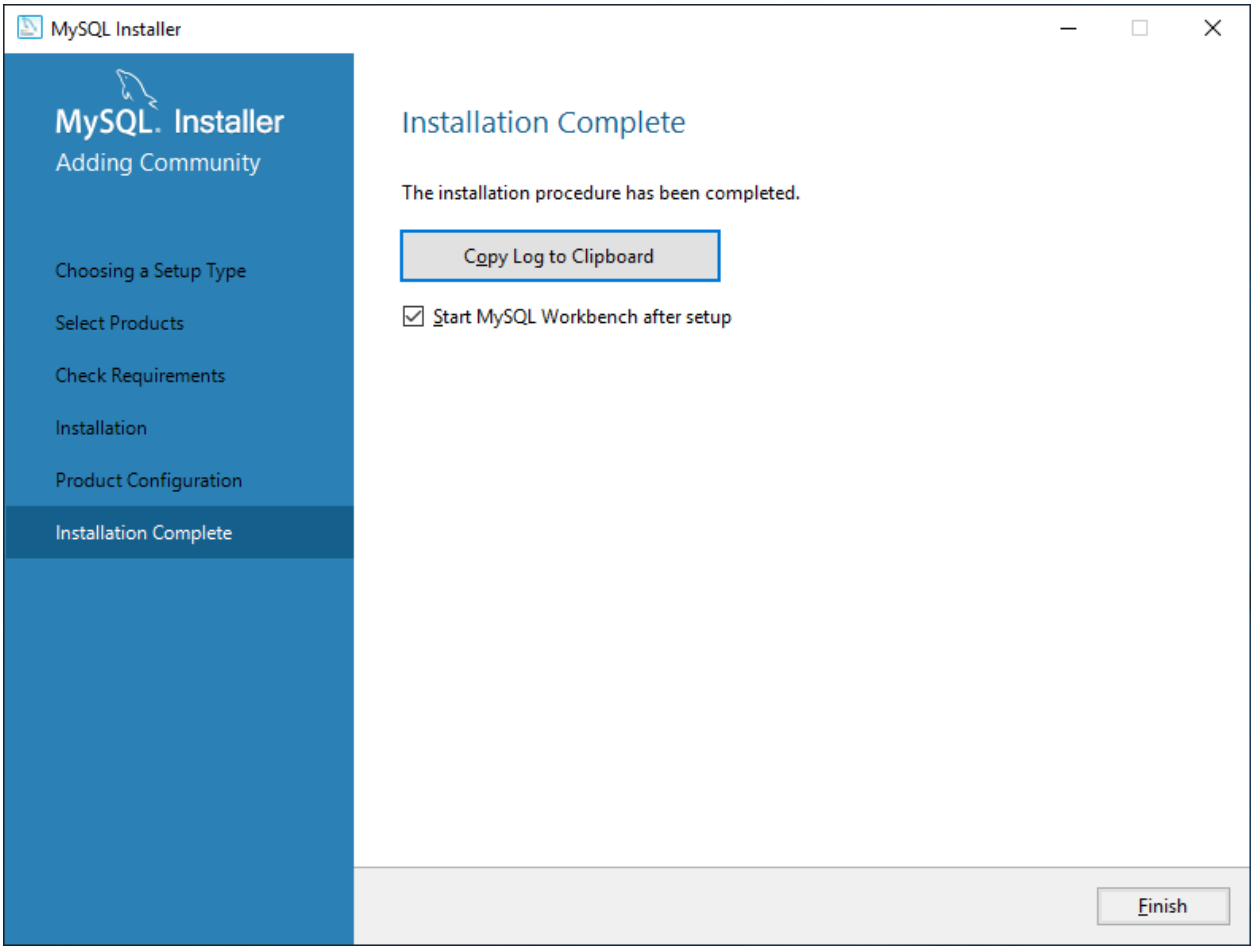
Click [Finish] once all configuration steps are completed.



On the Product Configuration screen click [Next>].



On the Installation Complete screen ensure the **Start MySQL Workbench after setup** option is selected and click [Finish].



D.2 Linux Installations

Most Linux distributions already include MySQL in their package management tools. As long as the included version is version 8.0, let your package manager handle the initial installation.

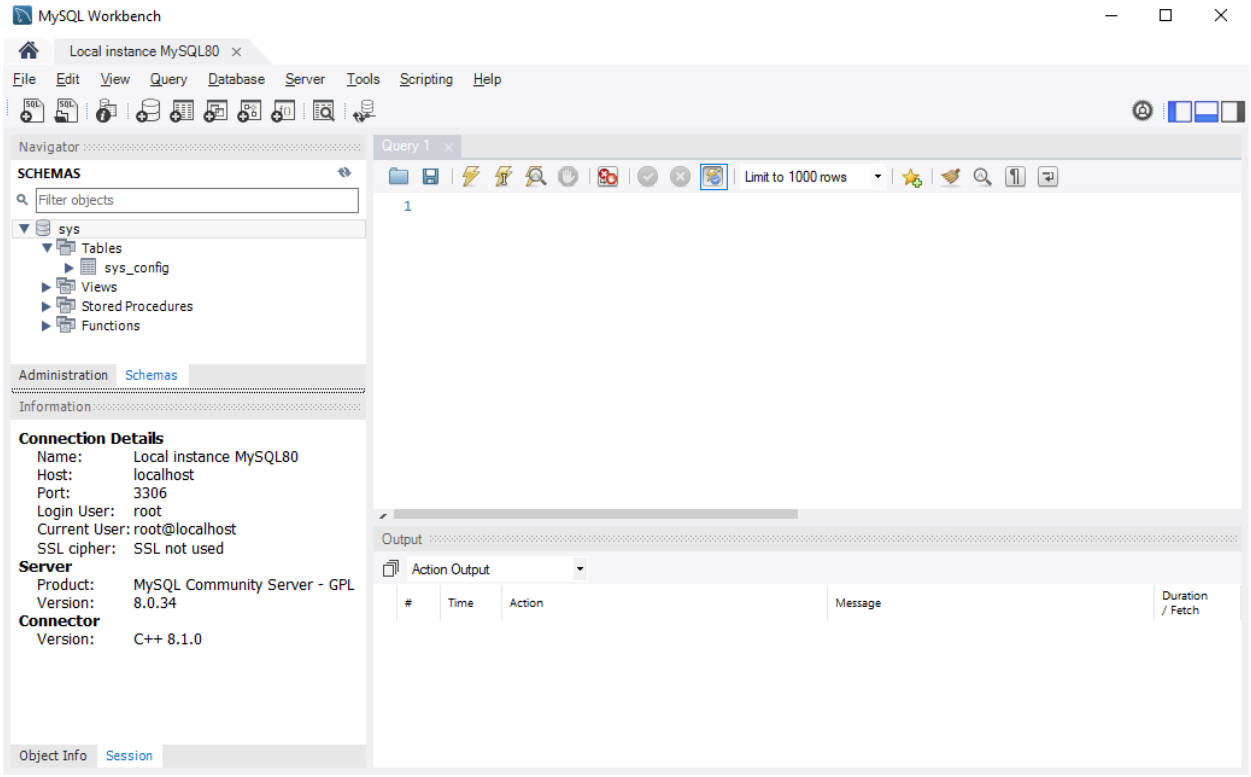
If your Linux package does not include version 8.0 of MySQL, refer to https://dev.mysql.com/downloads/mysql/ for instructions on configuring your package manager to get the latest 8.0 version of MySQL.

It is also recommended that you install the MySQL Workbench for Linux.

E. MySQL Configuration

E.1 Verify the Installation

After installation, you should be able to launch the MySQL workbench and connect it to your MySQL Server.



E.2 Locate the Configuration File

There are a few parameters which must be set on your MySQL server to ensure compatibility with DTS.

These parameters are typically placed in a file named my.cnf (or my.ini)

The location of the configuration file on Windows is typically within the folder where the database data is stored (which was selected during installation). On Linux, the typical location is in the `/etc/` path.

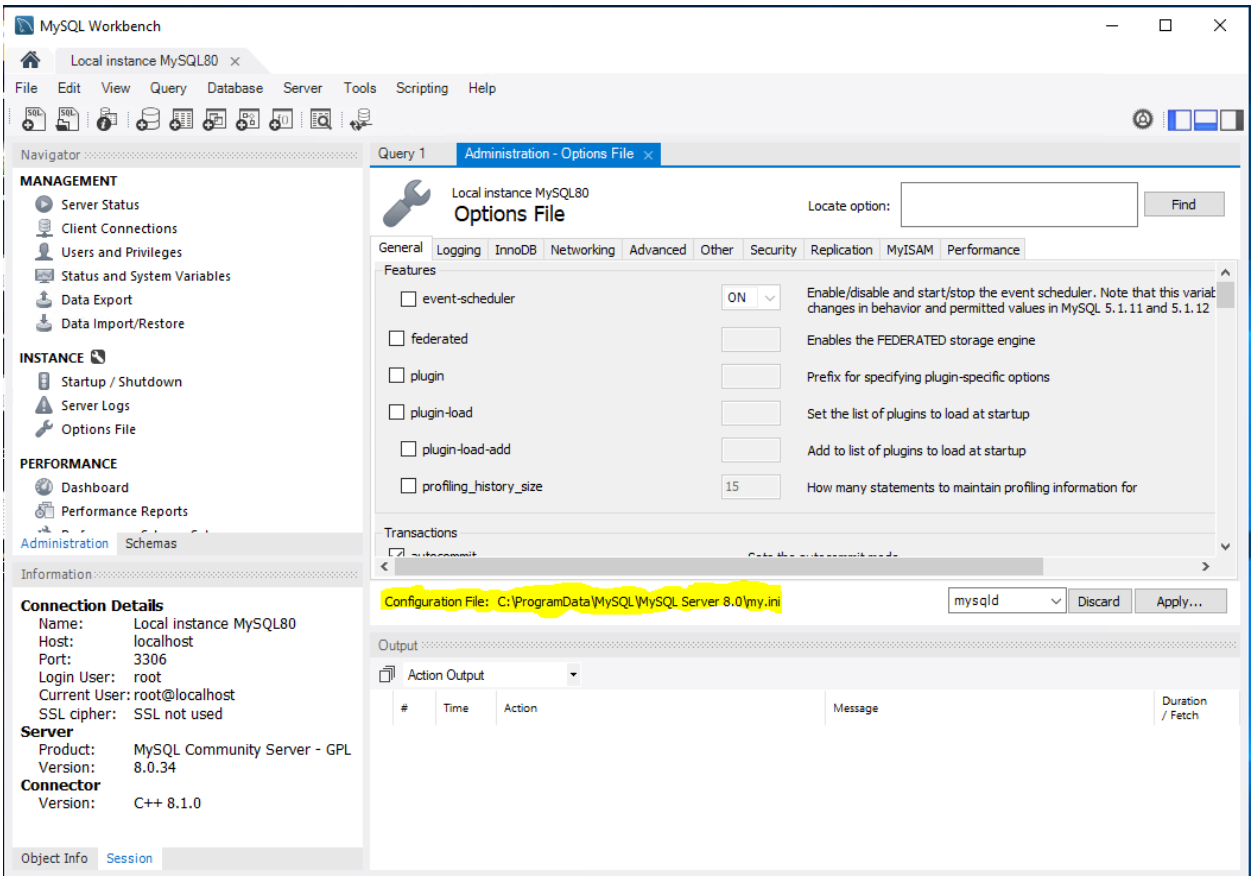
However, there are many other locations which are read for configuration data, with most (but not all) options being merged. See https://dev.mysql.com/doc/refman/8.0/en/option-files.html for the official documentation.

> **CAUTION:** On Windows when MySQL is installed as a system service, it uses the parameter '`—defaults-file`' to specify the path to the MySQL configuration file. When the '`defaults-file`' parameter is given, no other paths except for the specified path are referenced for configuration information, and none of the merging behavior from the hierarchy of configuration files takes place.

If you wish to use a configuration file from a different folder for your MySQL instance on Windows, you must uninstall the service and re-install it with the path to the new configuration file. See https://dev.mysql.com/doc/refman/8.0/en/windows-start-service.html for details. This step is optional – the easiest method of customizing the installation is to simply edit the file that is currently in use.

We suggest you manually edit the configuration file.
The MySQL Workbench can be used to locate the current my.ini file of the running instance by clicking on the '**Options file'** page. To locate the current my.ini file for the running instance from the MySQL Workbench, click the **"Options File"** item under the **"Server"** menu.

Note: On Windows, this path may be marked as a 'System' or 'Hidden' folder, and you may have to enable viewing 'System' and/or 'Hidden' items before you can navigate to this path.



E.3 Customize the Installation

The following options are the configuration settings that should be set for a successful DTS installation. These parameters are all part of the '**mysqld'** section. NOTE: Please make a backup of your configuration file before making any changes.

```
[mysqld]
character-set-server=utf8mb4
collation-server=utf8mb4_general_ci
default-storage-engine=INNODB
lower_case_table_names=1
max_allowed_packet=100M
innodb_buffer_pool_size=2000M
innodb_flush_log_at_trx_commit=2
innodb_redo_log_capacity=512M
log_bin_trust_function_creators=1
local_infile=1
```

- lower_case_table_names – optional for windows, required for Linux
- default-storage-engine – DTS requires the transactional storage engine InnoDB
- max_allowed_packet – required to allow large terminologies to be loaded in bulk
- character-set-server – required to ensure correct handling of data
- collation-server – required to ensure correct handling of data
- innodb_buffer_pool_size – this parameter controls the cache size of the main cache within MySQL. Setting a large value here generates large performance improvements for DTS. The recommendation above is for 2 GB. You may need to customize this value for your installation, depending on the hardware you have available. A value greater than 2 GB may help if you have several large terminologies loaded. A smaller value may be required if your server is memory constrained, and is also hosting the EE server hosting DTS, for example. At least 500 MB is recommended, at a minimum.
- innodb_flush_log_at_trx_commit - an (optional) performance improvement that helps during data loads and large commits, at the cost of a slight risk of data loss (up to the last second worth of commits) in the event of a server failure. For more details see: https://dev.mysql.com/doc/refman/8.0/en/innodb-parameters.html#sysvar_innodb_flush_log_at_trx_commit.
- innodb_redo_log_capacity - Defines the amount of disk space occupied by redo log files. This variable supersedes the innodb_log_files_in_group and innodb_log_file_size variables. For more details see:.https://dev.mysql.com/doc/refman/8.0/en/innodb-parameters.html#sysvar_innodb_redo_log_capacity
- log_bin_trust_function_creators - If log_bin_trust_function_creators is set to "1", the requirement that functions be deterministic or not modify data is dropped. Stored procedure calls are logged at the statement level rather than at the CALL level. For more details see: https://dev.mysql.com/doc/refman/8.0/en/stored-programs-logging.html

- local_infile – this system variable controls server-side LOCAL capability. Depending on the local_infile setting, the server refuses or permits local data loading by clients that request local data loading. For more details see: https://dev.mysql.com/doc/refman/8.0/en/load-data-local-security.html

These options should be set manually within the configuration file (edit, add, and/or replace the existing values).

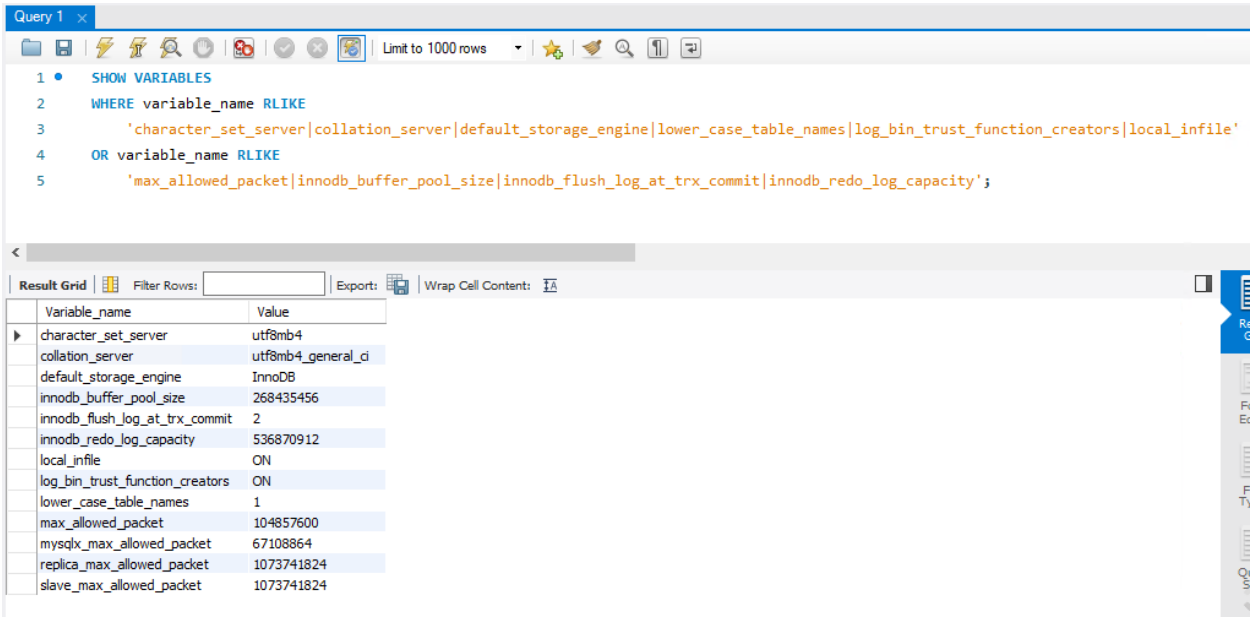
> **Note:** On Both Windows and in many Linux distributions, the installation process for MySQL sets a large number of variables in the configuration file by default – based off of the 'Server Configuration Type' that was selected during installation.

After editing the configuration restart the MySQL server. If the MySQL server fails to start, there is likely an error in the syntax within your configuration file. Compare your file with the default file that you backed up before you started editing.

E.4 Validate the Configuration Settings

After customizing the installation, please run the following query within the MySQL Workbench to validate that the required configuration options are set.

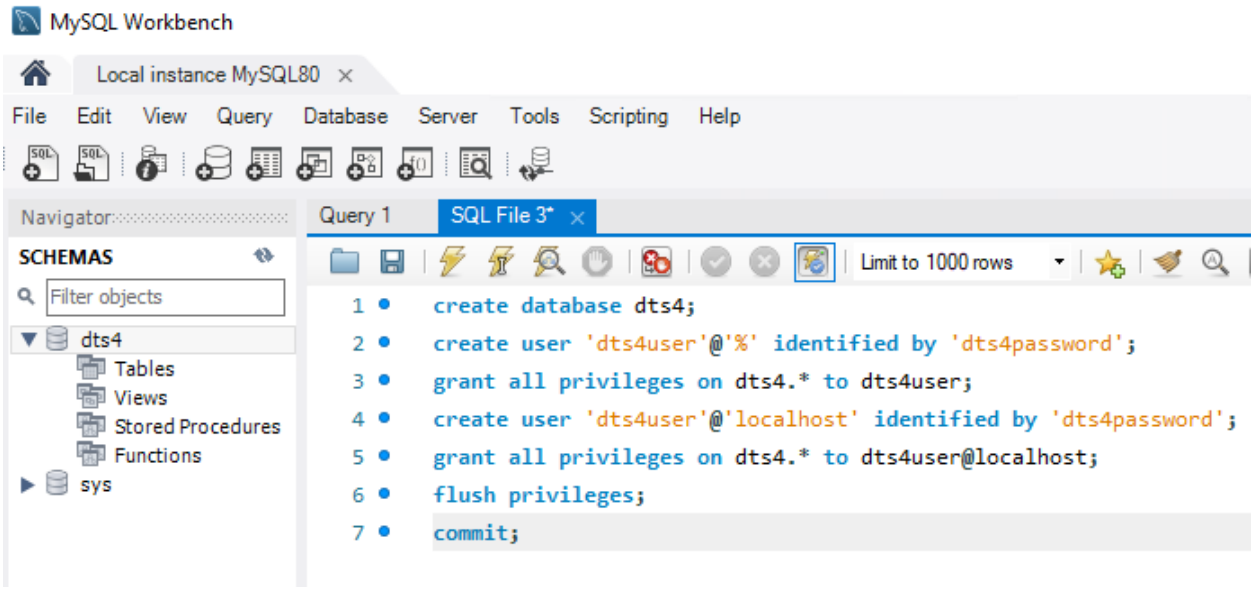
```
SHOW VARIABLES
WHERE variable_name RLIKE
     'character_set_server|collation_server|default_storage_engine|lower_case_table_names|log_bin_trust_function_creators|local_infile'
OR variable_name RLIKE
     'max_allowed_packet|innodb_buffer_pool_size|innodb_flush_log_at_trx_commit|innodb_redo_log_capacity';
```



F. Create the DTS User and Database

Run the following SQL commands to create a DTS user and an empty DTS database.
**NOTE: The below example uses the username "dts4user" and password "dts4password". However, you should specify your own database credentials that are most suitable for your environment. You can use the following example, replacing the brackets [] and their values with ones you prefer for your setup.*

```
create database dts4;
create user '[dts4user]'@'%' identified by '[dts4password]';
grant all privileges on dts4.* to [dts4user];
create user '[dts4user]'@'localhost' identified by
'[dts4password]';
grant all privileges on dts4.* to [dts4user]@localhost;
flush privileges;
commit;
```



- `create database dts4;`
  - Create a new, empty database for DTS
- `create user 'dts4user'@'%' identified by 'dts4password';`
  - The first parameter is the user name – this is the value you will use when configuring your DTS installation to connect to MySQL. The second parameter are the hosts allowed to connect using this username. The third parameter is the password that will be used when configuring your DTS installation to connect to MySQL.
- `grant all privileges on dts4.* to dts4user;`
  - Grant the DTS user created above full permissions to the DTS database
- `create user 'dts4user'@'localhost' identified by 'dts4password';`

    - The first parameter is the user name – this is the value you will use when configuring your DTS installation to connect to MySQL. The second parameter indicates 'localhost' rather than '%' as the host allowed to connect using this username. The third parameter is the password that will be used when configuring your DTS installation to connect to MySQL.
- `flush privileges;`
    - Activate the permissions change
- `commit;`
    - In case this is inside a transaction, ensure it is committed.

G. Creating and Populating the DTS Schema and Tablespace

The next steps assume that you have DTS installed on your system – as they rely on scripts provided by the DTS installation. Please refer to the **DTS Installation Guide** for instructions on installing DTS. Return to this document when you get to the step "**Preparing Your Knowledgebase**".

G.1 Create DTS Knowledgebase Schema

After you create the MySQL 8 username and Database, you must create the DTS tables/schema. Before you can run the Knowledgebase Create utility to create the schema, you must follow this procedure to configure the parameters by which the schema will be created, and by which a connection will be made to the DTS database.

1. Go to `<DTS_HOME>\bin\kb`. If `target-connection.xml` is not there, copy target-connection-<database>.xml (where <database> is mysql) as `target-connection.xml`.

2. Open this new `target-connection.xml` file, where you will set the values for the database connection. For the **user** and **pass** property values (highlighted) enter the username and password for the user you created earlier.

   **NOTE: DTS uses the default username and password of "dts4". However, you must specify your own database credentials that were established in the previous steps of this document.*

```
<!--
  MySQL connection.
-->
<connection>
  <property name="direction" value="target" />
  <property name="type" value="mysql"/>
  <property name="user" value="dts4user"/>
  <property name="pass" value="dts4password"/>
  <property name="host" value="localhost"/>
  <property name="databaseName" value="dts4"/>
  <property name="databasePort" value="3306"/>
  <property name="jdbcDriver" value="com.mysql.jdbc.Driver"/>
  <property name="url_template" value="jdbc:mysql://[HOST]:[PORT]/[DATABASE]"/>
</connection>
```

Be sure to verify/update the **host**, **databaseName**, and **databasePort**. When you run the Knowledgebase Create utility, a database connection will be established based on the values in this file.

3. Save the `target-connection.xml` file.

4. Run the file `kbcreate.bat` (`<DTS_HOME>\bin\kb\create`) (or `kbcreate.sh` for Linux) to start the Knowledgebase Create utility. If this is the first time `kbcreate.bat` is being run, the utility creates all the tables required to load data into the knowledgebase. If `kbcreate.bat` was run previously, the Knowledgebase Create utility checks for existing tables.

   Important Note: If one or more tables are missing from an existing set, the utility automatically drops all existing tables, then recreates them all (without displaying any user prompts).

   If all the tables exist, the utility prompts you to drop, or not drop, existing tables.
   All the tables exist. Do you want to drop all tables (y/n)?

   If you type **y**, the utility drops all existing tables, then recreates them all. Type **n** to bypass the drop of existing tables and creation of new ones, and to exit the utility.

The results of running the Knowledgebase Create utility are written to the log file (the default log file is `kbcreate.log` in `<DTS_HOME>\bin\logs`). If a different log file name and location

are specified in the log configuration file `kbcreatelog.xml` (`<DTS_HOME>\bin\kb\create`) the results are written to that file.
If an error occurs in the knowledgebase creation process, the utility stops and updates the log file to reflect the error.

G.2 Populate the DTS Knowledgebase

Please return to the **DTS Installation Guide** and continue with the step "**Populating the DTS Knowledgebase**".

H. Upgrade MySQL Server from version 5.6 to 8.0

If you are currently running DTS 4 against a MySQL 5.6 database, the following are MySQL upgrade instructions/suggestions for upgrading your MySQL 5.6 database to MySQL 8. *MySQL requires you first upgrade to MySQL 5.7 before you can upgrade to MySQL 8*.

NOTE: Make backups of everything to ensure you can roll back your entire MySQL 5.6 environment and data in case anything goes wrong with the upgrade to MySQL 8.0

H.1 Upgrade from MySQL 5.6 to 5.7

For Windows OS - Please refer to the MySQL 5.7 documentation on "Upgrading MySQL on Windows" that is located on the MySQL Developer Zone site: https://dev.mysql.com/
The MySQL 5.7 documentation on "Upgrading MySQL on Windows" is located here:
https://dev.mysql.com/doc/refman/5.7/en/windows-upgrading.html
We suggest you follow the "Using the Windows ZIP archive distribution" approach.
To get the newest 5.7 server ZIP distribution go to:
https://downloads.mysql.com/archives/community/

We suggest in the Product Version dropdown to select "5.7.40", in the Operating System dropdown select "Microsoft Windows", and choose "Windows (x86, 64-bit)" in the OS Version dropdown. Finally, click the "Download" button for the ZIP Archive "(mysql-5.7.40-winx64.zip)".

A summary of the steps you will be instructed to perform:

1. Shutdown your current MySQL 5.6 server.
2. Extract the upgrade files into your MySQL 5.6 directory (most likely located in ..\ProgramData\MySQL\MySQL Server 5.6). The easiest option is to overwrite the existing MySQL 5.6 files.
3. Start the upgraded MySQL server.
4. Run **mysql_upgrade**. This will take some time to complete. Make sure to wait until it's fully completed before moving on to the next steps.
5. Shutdown and restart the MySQL server and then review the startup logs to check for any issues.

For Linux OS – Please refer to one of the following documents:

MySQL 5.6 installed as a Docker container:
https://dev.mysql.com/doc/refman/5.7/en/docker-mysql-getting-started.html#docker-upgrading
MySQL 5.6 running on Debian or Ubuntu Linux (ensure the target version is 5.7.40, see the "Selecting a Major Release Version" section of the page):
https://dev.mysql.com/doc/mysql-apt-repo-quick-guide/en/index.html#repo-qg-apt-upgrading
MySQL 5.6 running on SLES (SUSE Linux Enterprise Server) - ensure the target version is 5.7.40, see the "Selecting a Release Series" section of the page:
https://dev.mysql.com/doc/mysql-sles-repo-quick-guide/en/index.html#repo-qg-sles-upgrading
MySQL 5.6 using a platform that supports Yum:
https://dev.mysql.com/doc/refman/5.7/en/updating-yum-repo.html
MySQL 5.6 where none of the above apply:
https://dev.mysql.com/doc/refman/5.7/en/upgrade-binary-package.html

H.2 Upgrade from MySQL 5.7 to 8.0

For Windows OS - Please refer to the MySQL 8.0 documentation on "Upgrading MySQL on Windows" that is located on the MySQL Developer Zone site: https://dev.mysql.com/
The MySQL 8.0 documentation on "Upgrading MySQL on Windows" is located here: https://dev.mysql.com/doc/refman/8.0/en/windows-upgrading.html
We suggest you follow the "Using the Windows ZIP archive distribution" approach.
To get the newest 8.0 server ZIP distribution go to: https://dev.mysql.com/downloads/mysql/
We suggest in the Select Version dropdown to select "8.0.xx" and in the Select Operating System dropdown select "Microsoft Windows". Scroll past the MSI installer and click the "Download" button for the "Windows (x86, 64-bit), ZIP Archive (mysql-8.0.xx-winx64.zip)".
A summary of the steps you will be instructed to perform:

1. Stop your MySQL 5.7 server.
2. Extract the upgrade files into your MySQL 5.7 directory (most likely located in ..\ProgramData\MySQL\MySQL Server 5.6). The easiest option is to overwrite the existing MySQL 5.7 files.
3. Start the upgraded MySQL server. This startup may take some time, it will detect the version difference and run the upgrade automatically.
4. When the previous step is complete, shutdown and restart the MySQL server and then review the startup logs to check for any issues.

For Linux OS – Please refer to one of the following documents:

MySQL 5.7 installed as a Docker container:
https://dev.mysql.com/doc/refman/8.0/en/docker-mysql-getting-started.html#docker-upgrading
MySQL 5.7 running on Debian or Ubuntu Linux (ensure the target version is 8.0.xx, see the "Selecting a Major Release Version" section of the page):
https://dev.mysql.com/doc/mysql-apt-repo-quick-guide/en/index.html#repo-qg-apt-upgrading
MySQL 5.7 running on SLES (SUSE Linux Enterprise Server) - ensure the target version is 8.0.xx, see the "Selecting a Release Series" section of the page:
https://dev.mysql.com/doc/mysql-sles-repo-quick-guide/en/index.html#repo-qg-sles-upgrading
MySQL 5.7 using a platform that supports Yum:
https://dev.mysql.com/doc/refman/8.0/en/updating-yum-repo.html
MySQL 5.7 where none of the above apply:
https://dev.mysql.com/doc/refman/8.0/en/upgrade-binary-package.html