Minion Install Guide

This guide will show you everything you need to know to install the Minion Maintenance Modules. Do not proceed with the install until you have read this entire document as it contains important information on the effect of the install on your servers.

# Important--Unblock Files--Important

Before continuing with the install you must unblock the files. Windows blocks files when you download them from the internet, so the first time you run this you’ll need to unblock them. The easiest way to do this is to open a Powershell window and navigate to the MinionWare folder you just unzipped. Next type this command:

**dir -recurse | unblock-file**

Nothing will be returned and it’ll happen almost instantly. Now you should be ready to run the installer. If you want to check to make sure that they were all unblocked, you can right-click on each one and go to Properties. If you don’t see an ‘Unblock’ checkbox, you’re good to go. You only need to unblock the .ps1 files. Check all the folders to make sure you got them all.

# Quick Start

This installer is designed to run from a single location. There’s no longer any need to specifically connect to a specific server, and then to another one. You can install Minion across your entire environment from a single script execution.

If you want to skip all the intricacies of explanation and just run the installer (it’s ok, this other stuff will still be here for you), go into Powershell and simply type the following:

./MinionSetupMaster.ps1 <servername> <DBName> <Product>

**Example 1:**  
Installing a single product on a single box.

./MinionSetupMaster.ps1 Server1 DBAAdmin CheckDB

**Example 2:**  
Installing a single product on multiple boxes.

./MinionSetupMaster.ps1 “Server1|Server2|Server3” DBAAdmin CheckDB

**Example 3:**  
Installing a 2 products on multiple boxes.

./MinionSetupMaster.ps1 “Server1|Server2|Server3” DBAAdmin “Backup|CheckDB”

**\*Note: When installing Backup and CheckDB, it’s best to install Backup first because CheckDB relies on elements of Backup and you’ll get a warning. It’ll still install ok, but you probably want a clean install. If you install CheckDB by itself you’ll get the warning as well, and you’ll be able to run every aspect of CheckDB except the remote features.**

# Benefits of a Unified Installer

There are many benefits to a unified installer.

1. Much easier to run upgrades.
2. Allows for excellent customization for your shop by adding logic to the process where needed.
3. Much easier to add customized code that sets up your environment the way you want it. This would be used to add backup locations, snapshot locations, dynamic tuning settings, etc. You can even use it to add views or define security for your installation.
4. You can install from a centralized location much easier.
5. You can write your own installer if the one we provide doesn’t suit your needs.
6. The installer automatically knows what the latest version of the modules is, and will install that version by default.
7. You now have a more unified error log so you only have to go to a single place to see install errors.

# Installer Parts

The Minion installer consists of 2 scripts: **MinionSetup.ps1** and **MinionSetupMaster.ps1**.

## Folders

Includes: This folder contains the data you’re able to customize for each install. Every module has a set of parameters that you can configure so you don’t have to do it after the install.

**InstallLog:** Holds all the dated versions of the install logs. Search the log after each install for errors.

**Shared:** This folder holds all the install elements that are common among all the modules. The elements here get installed for every module whether they use it or not.

## Scripts

**MinionSetup.ps1** – Runs the setup for a specific module on an individual server.  
**MinionSetupMaster.ps1** – Controls the installation of multiple modules on any number of servers.  
**BackupInclude.ps1** – Holds the configurable elements for the backup install. This is the file you’ll change when you want to customize your backup install.  
**CheckDBInclude.ps1** – Holds the configurable elements for the checkdb install. This is the file you’ll change when you want to customize your checkdb install.  
**ReindexInclude.ps1** – Holds the configurable elements for the reindex install. This is the file you’ll change when you want to customize your reindex install.

Even when you’re only installing a single module on a single server, you should use the **MinionSetupMaster.ps1** script.

# MinionSetupMaster Params

$ServerName – Name of the server you’re installing Minion on. This can be a single server, or a pipe-delimited list. If you want to pass in a list of servers through a text file, or a SQL query, then look at the section on multi-server deployment for more information.

$DBName - All modules should be installed in the same DB and in the case of CheckDB and Backup, they must be installed in the same DB because there’s some inter-dependency between them.

$MaintType – You can install all Minion Modules from the same unified installer. You can name the module you want to install, or you can use the keyword ‘All’ to install all the Minion Modules at the same time. If you want to only install 2 modules for instance, you could pass it in like this: ‘Backup|CheckDB’.  
$User – SQL username for connecting to the server you’re installing on.  
$PWord – SQL password for connecting to the server you’re installing on.

Both of the SQL login parameters are NULL by default, so the installer will run with Windows authentication unless you specify otherwise. Do NOT specify a Windows account and password in the above parameters as this will not work.

# Customizing Parameters

You will want to customize your install to your environment. Take for example, setting your backup location. To customize install parameters, each module has a file in the Includes folder. Simply open the file for each module and make the changes you’re interested in.

Of course, you don’t have to customize the parameters. You’re free to leave everything in the installer as-is and alter the data with T-SQL statements after the install. The easiest way to do this is to write the statements and save them as a file in one of the module folders. Then add your new script to InstallOrder.txt in that module folder. Directions for using InstallOrder.txt are found below under ‘Customizing your Environment Data’.

# Multi-Server Deployment

If you pass in $ServerName = $null then you will need to open the Master script and fill the $ServerList variable using a query or a text file, or whatever you prefer. If you use a query to fill this variable then you’ll need to change the variable $currServer = $\_ to use the current column coming back from the DB. For example: if you fill the $ServerName like this:

$ServerName = invoke-sqlcmd -database master -query “select ServerName from dbo.MyServers”

Then you would need to change the other variable to this:

$currServer = $\_.ServerName;

# Minion Security

The Minion Modules are very secure routines to run on your systems. They not only allow you to setup the jobs under any user account you wish, but they also use Powershell in RemoteSigned mode. As well, they turn on, and rely on xp\_cmdshell. Go here to read about how xp\_cmdshell is as secure, if not more so, than any other SQL Server feature.

# Customizing your Environment Data

It’s incredibly easy to setup your environment install with your own data. All of the Minion Modules come with their own seed data so you can easily install and run them, but you may want to put in dynamic tuning thresholds, or custom backup paths, or specialized schedules, etc. The unified installer makes this very easy. In the folder for each module, you’ll see a file called **InstallOrder.txt**. This file is at the heart of customizing your data. This file defines the order in which each of the scripts is run by the installer, and it also defines which scripts will be run to begin with. So, if you have any updates or inserts you need to customize your environment, then simply put your customized statements into a file and add that file name to the end of the list in the InstallOrder file. Your file will be added to the installer. And of course you can enter as many custom files as you like. You could add your custom data to the main installer scripts if you like, but we advise that you perform customized data statements in your own scripts and add them to the routine, and leave the default install files alone. The reason is that when you add your own data to the install scripts, then you have to sift through and take it back out when you upgrade and add it to the new scripts. However, if you simply add the script to the InstallOrder file, then that’s all you have to do with the new version when you download it. Well, that and copy the custom scripts to the new module folder.