



SQL Server 2012 on Windows Server Core White Paper

Date: 1 March 2012

Version: 1.0





Document Revision History

Version	Revision date	Author	Description
1.0	1 March 2012	Henry Rooney	Initial Release



Contents

Introduction	4
SQL Server 2012 on Server Core.....	5
Installing Windows Server Core	6
Installation Process.....	8
Managing SQL Server.....	10
Connecting to SQL Server	11
Appendix A References.....	12
WARDY IT Solutions Profile.....	13

Introduction

Microsoft Windows Server Core is a minimal server installation option for the Standard, Enterprise or Datacentre editions of Windows Server 2008 and Windows Server 2008 R2. Windows Server Core has been designed to reduce maintenance and administration effort whilst increasing the security of the server. Windows Server Core only installs the services and components that are required for a particular usage scenario. The reduced footprint of Windows Server Core means that it requires less patching and therefore increases the availability of the applications supported by the server.

It must be noted that only specific products can be installed on Server Core and prior to installing refer to the vendor's technical documentation. For more information on the Microsoft Applications and Services supported by Windows Server Core refer to the following article <http://technet.microsoft.com/en-us/library/dd184075.aspx>.

With the release of SQL Server 2012 SQL Server is now supported on Windows Server Core installations.

Windows Server Core has no Graphical User Interface (GUI). Server Core has none of the Windows graphical features such as Windows Explorer, the Start menu and the Taskbar. As a result of the lack of GUI there is a common misconception that it is a time consuming process to build and administer Windows running on the Server Core architecture. The role of Windows Server Core is to create and run an environment that once it is configured requires minimal changes. In this white paper you will see how easy it is to install SQL Server 2012.

SQL Server 2012 on Server Core

Microsoft SQL Server 2012 is the first version of SQL Server to be supported on Windows Server Core. Deploying SQL Server on Windows Server Core provides many advantages over traditional installations of SQL Server, including but are not limited to:

- Reduced patching
- Fewer vulnerabilities
- Less administration

The ability to deploy SQL Server on Windows Server Core provides an opportunity to create an appliance like deployment of SQL Server.

Installing Windows Server Core

The first step in the process of installing Microsoft SQL Server 2012 on Windows Server Core is to install the operating system. SQL Server 2012 is supported on the following Operating Systems running Windows Server Core:

- Windows 2008 SP2
- Windows 2008 R2 SP1

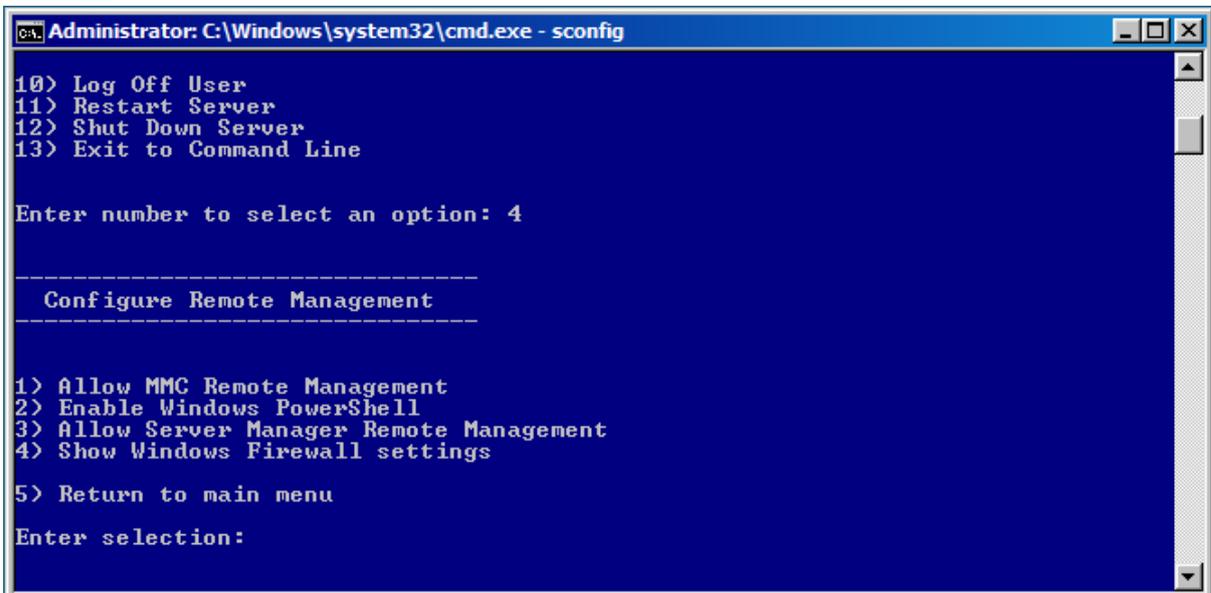
Once the Operating System is installed, it is recommended to enable the operating system features outlined below to aid in the initial setup and configuration of the Server.

Once Windows Server Core is installed at the command prompt, type:

```
SCONFIG
```

This will open the Server Configuration Dialog box. In this interface there are several tasks that can be performed, such as adding the server to a domain, changing computer names, etc. This will not be discussed in detail in this document, however for more information on SCONFIG refer to the following article: [http://technet.microsoft.com/en-us/library/ee441254\(v=ws.10\).aspx](http://technet.microsoft.com/en-us/library/ee441254(v=ws.10).aspx).

For the purpose of this installation, select option 4) Configure Remote Management - the following screen will be displayed:



```
Administrator: C:\Windows\system32\cmd.exe - sconfig
10> Log Off User
11> Restart Server
12> Shut Down Server
13> Exit to Command Line

Enter number to select an option: 4

-----
Configure Remote Management
-----

1) Allow MMC Remote Management
2) Enable Windows PowerShell
3) Allow Server Manager Remote Management
4) Show Windows Firewall settings
5) Return to main menu

Enter selection:
```

We need to enable the following options:

- 1) Allow MMC Remote Management
- 2) Enable Windows PowerShell
- 3) Allow Server Manager Remote Management



You may want to simplify your configuration tasks by creating PowerShell scripts to automate the process and configure the server locally.

Once the server is configured, it is recommended to disable options 1) and 3) to reduce the surface area of the server.

Installation Process

Before installing SQL Server 2012, it must be noted that not all features are supported on Windows Server Core. If you are deploying SQL Server Reporting Services then you have to install this component on the Full Installation of Windows Server. The following is a list of SQL Server 2012 features that can be installed on Windows Server Core:

- Database Engine Services
- SQL Server Replication
- Full Text Search
- Analysis Services
- Client Tools Connectivity
- Integration Services Server
- Microsoft Sync Framework

It may have occurred to you that because we are installing SQL Server on an Operating System without GUI support that the installation process will be more difficult. This is not the case and in fact provides many advantages within an organisation. By removing the GUI from the installation process this reduces the opportunity for user error, as all installations are managed through a command line script using a configuration file.

By generating a standard SQL Server build for the organisation it ensures that there is consistency in the installation and configuration process. This also ensure that each server has the same layout so administration overhead is reduced and the deployment of a new SQL Server instance is exponentially faster than traditional deployments.

One of the biggest hurdles is generating the standard SQL Server installation is generating the configuration file. The most efficient method is to perform an installation of the SQL Server features supported on Windows Server Core through the SQL Server Setup and once the installation is certified then make a copy of the "ConfigurationFile.ini" that resides in the setup bootstrap folder "C:\Program Files\Microsoft SQL Server\110\Setup Bootstrap\Log". This configuration file contains all the necessary parameters required to install SQL Server on Server Core.

Once you have a configuration file or command line parameters the next process is to run the SQL Server installation process. SQL Server 2012 will configure and install the required prerequisites for SQL Server which removes the necessity to install and configure .Net components.

For demonstration purposes, this install is being performed at the command line. The server name is SERVERCORE and a named instance of SQL Server, INS1.

```
Setup.exe /q /ACTION=INSTALL /FEATURES=SQLENGINE /INSTANCENAME=INS1  
/SQLSVCCACCOUNT="<DomainName\UserName>" /SQLSVCPASSWORD="<Password>"  
/SQLSYSADMINACCOUNTS="<DomainName\UserName>"  
/AGTSVCCACCOUNT="<DomainName\UserName>" /IAcceptSQLServerLicenceTerms
```

You will not see any progress indicators during the install, so don't be concerned as the install is happening.

To perform the equivalent installation using a configuration file, you will need to execute the following command:

```
Setup.exe /QS /ConfigurationFile=MyConfigurationFile.INI
```

Once the installation is completed, open up a PowerShell window from the command prompt by running the following command:

```
PowerShell
```

This will Display "PS" in front of the standard path. To check that SQL server is installed, execute the following PowerShell CmdLet:

```
Get-Service [m-s]*
```

This will list all installed services starting with the letters between m and s. You will notice that the SQL Agent service is set to manual, this can be altered post install or can be managed during install and set to an automatic start up.

In the PowerShell window type:

```
Exit
```

The interface will return to the standard Command Prompt.

Next we want to open the SQL Server PowerShell Component, to do this type the following into the command prompt:

```
Sqlps
```

Enter the following command to navigate the SQL instance that was just installed:

```
Set-Location SQLSERVER:\SQL\SERVERCORE\INST1
```

We are navigating to this location to execute a query against the SQL Instance that has been installed. Entering the following command in the PowerShell window will return the version of SQL Server that has been installed.

```
Invoke-SQLCmd "SELECT @@VERSION"
```



```
Column1
Microsoft SQL Server 2012 RC0 - 11.0.1750.32 (X64) ...
```

As you can see we have confirmed that we have installed SQL 2012.

Managing SQL Server

With a Windows Server Core install we cannot manage the SQL Server instance by running Management Studio on the server so remote access needs to be enabled. This is probably the most complex step in configuring SQL Server. Firstly TCP/IP must be enabled, the script to perform this action has not been included. However you can find an excellent article that provides samples and explanations of how to perform this here <http://msdn.microsoft.com/en-us/library/dd206997.aspx>.

Following on from this, Windows Firewall needs to allow communications on the port that SQL Servers is listening on (by default port 1433). To enable this eExecute the following command at the command prompt:

```
netsh advfirewall firewall add rule name="SQLServer" dir=in
action=allow protocol=TCP localport=1433 profile=DOMAIN
```

The command included here will enable communication on Port 1433 however if preferred, Windows firewall settings can be configured through PowerShell. For a listing of the port numbers used by SQL Server refer to the following article:

[http://msdn.microsoft.com/en-us/library/cc646023\(v=sql.110\).aspx](http://msdn.microsoft.com/en-us/library/cc646023(v=sql.110).aspx)

Finally the SQL Server Instance needs to be enabled for remote access. This can be performed through the SQL PowerShell control that was opened earlier by invoking the `Invoke-SQLCmd CmdLet`.

```
EXEC sys.sp_configure N'remote access', N'1'
GO
RECONFIGURE WITH OVERRIDE
GO
```



Connecting to SQL Server

Now that the SQL Instance is installed and enabled for remote access, Management Studio can be launched from your desktop and the instance managed remotely through the traditional interface.



Appendix A References

[http://msdn.microsoft.com/en-us/library/hh231669\(v=sql.110\).aspx](http://msdn.microsoft.com/en-us/library/hh231669(v=sql.110).aspx)

WARDY IT Solutions Profile

WARDY IT Solutions is a specialist Microsoft SQL Server consulting, training and Business Intelligence provider. Since its establishment in 2004, WARDY IT Solutions has assisted customers to successfully deliver SQL Server solutions ranging from database administration to Business Intelligence reporting and training. We have a comprehensive range of consulting services designed to fulfil every Microsoft SQL Server requirement for our Clients and is in the unique position as a business to have direct access to the Microsoft SQL Server product team in Redmond – USA by being a member of the Technology Advisor Group for future releases of SQL Server.

WARDY IT Solutions has established itself amongst the top solution providers in SQL Server technologies and recently we have received the following industry accolades:

- 2011 MAPA Data Platform Partner of the Year
- 2009 Microsoft APAC Data Management Partner of the Year
- 2009 & 2010 Finalist for the QLD ACS Award for Service Delivery
- 2010 Finalist for the QLD AIIA Award

In the new 2010 Microsoft Partner Network, WARDY IT Solutions has achieved Gold Competency for Data Platform and Business Intelligence, and Silver for Learning, and prides itself on having the best SQL Server consultants and trainers. Within our team we have recognised industry experts in Microsoft SQL Server including; several Microsoft Certified Trainers (MCT's), the authors and editors of several SQL Server books and each of our consultants are Microsoft Certified IT Professionals (MCITP's).

WARDY IT Solutions has vast experience in private and government sectors, including: Manufacturing and Industrial, Retail and Distribution, Banking and Finance, Health, Transport, Utilities, Education, and Telecommunications. Our clients range from the largest banks and stockbrokers in Australia to Australia's largest retailers; our solutions deliver results for the largest Microsoft SQL Server implementations in Australia and we built a quality and lasting relationship based on trust and mutual respect with our Clients.

"The SQL Server Healthcheck provided by WARDY IT Solutions was invaluable. The level of detail that the findings went into was truly remarkable. By implementing the changes recommended by the Healthcheck we were able improve the performance of our SQL Server instance by over 160%."

- Greg Norris, Trinity

"The vast wealth of knowledge in the WARDY IT Solutions team means they are able to assist with all of our various SQL Server needs, and their customer service commitment is second to none. We feel secure in knowing they are there for everything from the simplest question through to an operational emergency."

- Cherie Whelan, RPS Group



Data Management Solutions
Partner of the Year



Data Platform
Partner of the Year

