# C:\Users\stefano.maggi.CONBIPELSPA\Desktop\prom38.png

**Date**: 16/05/2012

**Procedure:** SQL Server 2012 - Installation for Windows Server Core

**Source:** [**LINK**](http://www.sqlmag.com/article/sql-server/creating-sql-server-2012-installation-windows-server-core-142853)

**Permalink:** [**LINK**](http://heelpbook.altervista.org/2012/sql-server-2012-installation-for-windows-server-core/)

**Created by:** HeelpBook Staff

**Document Version:** 1.0

# **SQL Server 2012 – Installation for Windows Server Core**

One of the most important new features in the **SQL Server 2012** release is the ability to run it on **Windows Server Core**. Server Core is perfect for back-end infrastructure applications such as **SQL Server**.

It provides all the **Windows Server** core services but lacks the graphical management shell, which you don’t really need on a server box.

Running on the leaner Server Core requires less overhead, but more importantly, it's also more secure because it has a smaller attack vector and requires significantly less patching. Microsoft says that Server Core reduces the need for patching by almost 60 percent over a full Windows Server installation.

It's important to note that the Server Core implementation in **Windows Server 2012** (originally code-named Windows Server 8) will be significantly easier to use than the current Windows Server 2008 R2 implementation. Unlike that version, where you have to choose either a full installation or a Server Core installation, the upcoming **Windows Server 2012 Server Core** has an installation option that you can switch on and off.

Here's a guide to the steps for creating a **SQL Server 2012** installation for **Windows Server Core**.

**1. Perform the basic Server Core configuration with sConfig.**

If this is the first time you’ve used the Server Core system, you’ll need to use these options on the **Server Configuration** tool interface to perform the basic server setup: 8) Network Settings, 1) Domain/Workgroup, 2)Computer Name.

Then select 4) Configure Remote, followed by 2) Enable Windows PowerShell.

**2. Enable Windows PowerShell and the .NET Framework on Server Core.**

To enable **PowerShell 2.0** and **the .NET Framework 4.0**, run the following commands:

DISM /Online /Enable-Feature /FeatureName: NetFx2-ServerCore

DISM /Online /Enable-Feature /FeatureName: NetFx3-ServerCore

DISM /Online /Enable-Feature /FeatureName: MicrosoftWindowsPowerShell

dotnefFx40\_Full\_x86\_x64\_SC.exe /passive /promptrestart

**3. Open the Server Core firewall ports for SQL Server using netsh.**

Run the following commands in the Server Core command prompt:

netsh firewall set portopening TCP 1433 "SQLServer"

netsh firewall set portopening TCP 1434 "SQL Admin Connection"

Depending on which **SQL Server** features you’re using, you might need to open more ports. For more information you can refer to the **MSDN** article " [Configure the Windows Firewall to Allow SQL Server Access](http://msdn.microsoft.com/en-us/library/cc646023.aspx)."

**4. Run the SQL Server command-line installation.**

Run the following command for the **Server Core** command prompt:

<path to setup>Setup.exe /qs /ACTION=Install /FEATURES=

SQLEngine,Replication /INSTANCENAME=MSSQLSERVER /SQLSVCACCOUNT="<DomainName\UserName>"

/SQLSVCPASSWORD="<StrongPassword>" /SQLSYSADMINACCOUNTS="<DomainName\UserName>"

/AGTSVCACCOUNT="NT AUTHORITY\Network Service"

/TCPENABLED=1 /IACCEPTSQLSERVERLICENSETERMS

Note that the parameters can change depending on the features you want to install. You can refer to the **Microsoft** article " [Install SQL Server 2012 on Server Core](http://msdn.microsoft.com/en-us/library/hh231669.aspx)."

**5. Enable SQL Server remote access.**

Run **SQLCMD** from the Server Core command prompt. Then run the following commands in the **SQLCMD** window:

EXEC sys.sp\_configure N'remote access', N'1'

GO

RECONFIGURE WITH OVERRIDE

GO