Contents

[Deploying iRING under Windows Server 2008 R2 2](#_Toc333584319)

[Starting Point: 2](#_Toc333584320)

[Server Hardware 2](#_Toc333584321)

[Install/Add Microsoft .NET 3.5 3](#_Toc333584322)

[Installing Required Software 6](#_Toc333584323)

[Install Microsoft .Net Framework 4 6](#_Toc333584324)

[Install Java JDK 1.7 7](#_Toc333584325)

[Setting the JAVA\_HOME Environment Variable. 8](#_Toc333584326)

[Install Tomcat 9](#_Toc333584327)

[Change the Tomcat Service to Autostart 9](#_Toc333584328)

[Deploying iRING Exchange Manager 10](#_Toc333584329)

[Deploy the Web Application Archives 10](#_Toc333584330)

[Test the iRING Exchange Manager Web Site 10](#_Toc333584331)

[Deploying iRING Adapter Manager 11](#_Toc333584332)

[Allow ASP .NET 4 Extensions to Run on the Server 11](#_Toc333584333)

[Create and Application Pool for iRINGTools 11](#_Toc333584334)

[Configure IIS to deploy the Adapter Manager “apps” and “services” as Applications 13](#_Toc333584335)

[Authentication and Access to iRING Configuration 13](#_Toc333584336)

[Create an IRING\_CONFIG Active Directory (AD) Group 13](#_Toc333584337)

[Set Permissions for the Account the Application Pool is Running Under 14](#_Toc333584338)

[Add MIME Types to Process json and SPARQL results 15](#_Toc333584339)

[Registering Assemblies in the Global Assembly Cache (GAC) 16](#_Toc333584340)

[Getting the GACUTIL Utility 16](#_Toc333584341)

[Register an Assembly in the GAC 16](#_Toc333584342)

[Install Database Drivers 17](#_Toc333584343)

[Installing MS-SQL Assemblies 17](#_Toc333584344)

[Oracle Client Setup 19](#_Toc333584345)

# Deploying iRING under Windows Server 2008 R2

## Starting Point:

The Operating System has been installed and the Web Server (IIS) Role has been configured. The server has been added to the corporate domain and the user deploying iRING to the server has been added to the Local Administrators group on the machine.

### Server Hardware

CPU: X5650 @ 2.67GHz

RAM: 2.00 GB

OS: Windows Server 2008 R2 (64-bit OS)

HDD’s: C: 60 GB

D: 20 GB

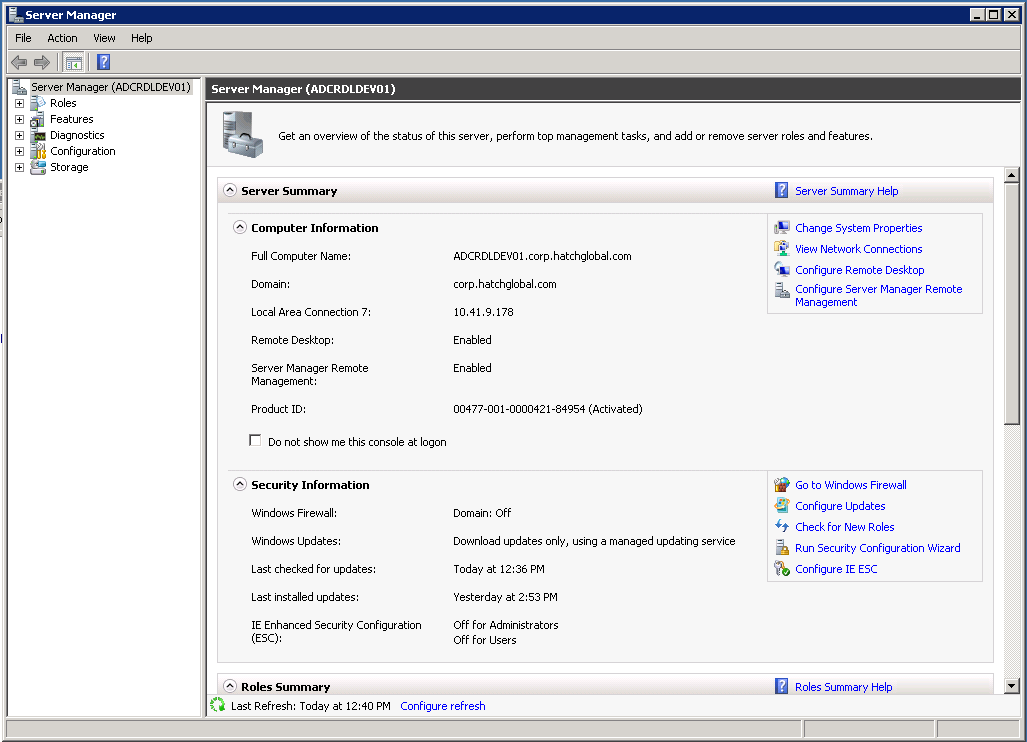
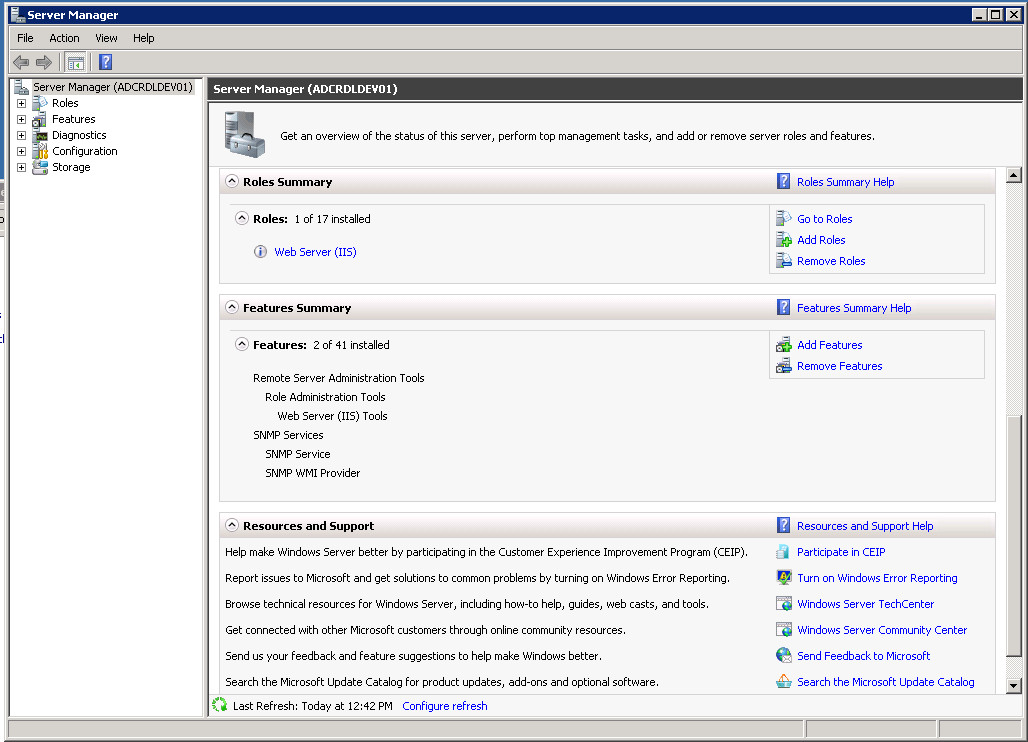
 

Figure : Server Manager Console showing Server Summary

## Install/Add Microsoft .NET 3.5

Using Server Manager scroll down to the “Feature Summary” panel and select “Add Features”.

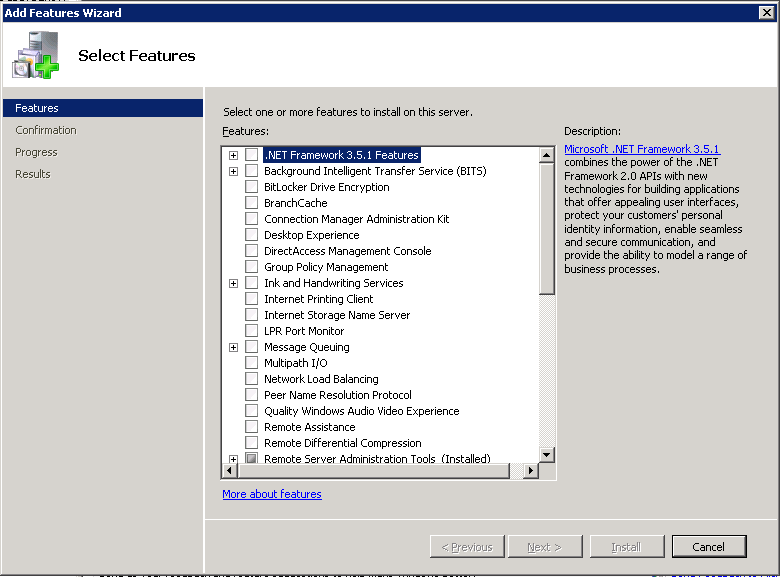


Figure : The Add Feature Wizard

Tick the “.NET Framework 3.5.1 Features” check box. As this feature has a dependency upon another feature you will be prompted to also add “Window Process Activation Services”, choose the “Add Required Features” button.

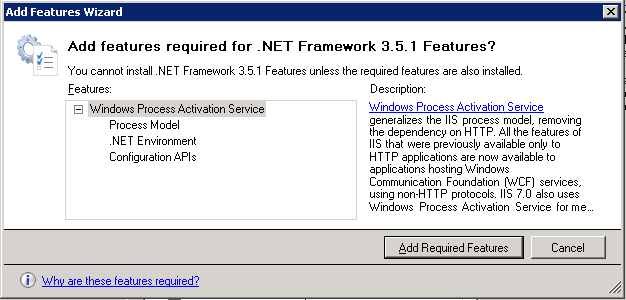
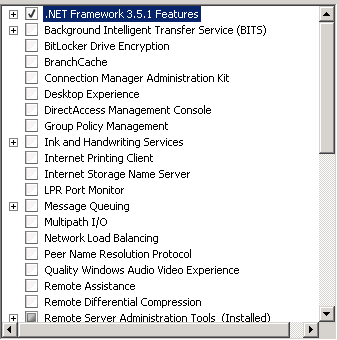


Figure : Selecting Feature to add to the server

If you scroll through the list of features to be added you will see that the “Window Process Activation Services” now also has a tick against it.

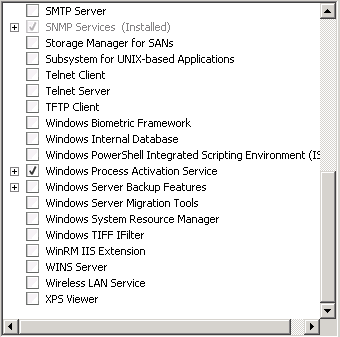


Figure : Windows Process Activation is now selected

Press the “Next” button. You will be asked to the confirm your selection, press “Install” to have the features added.

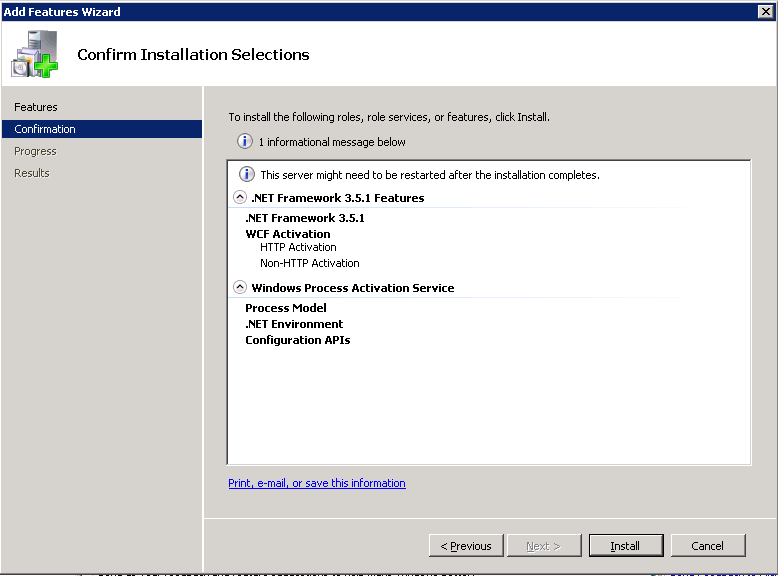
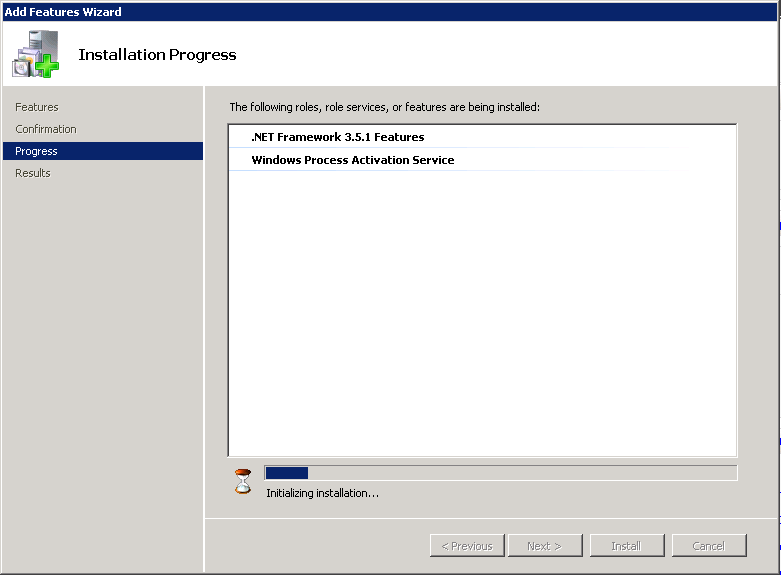
 

Figure : Confirming feature selection and installation progress

Once completed the installation results should show that the features have been added successfully.

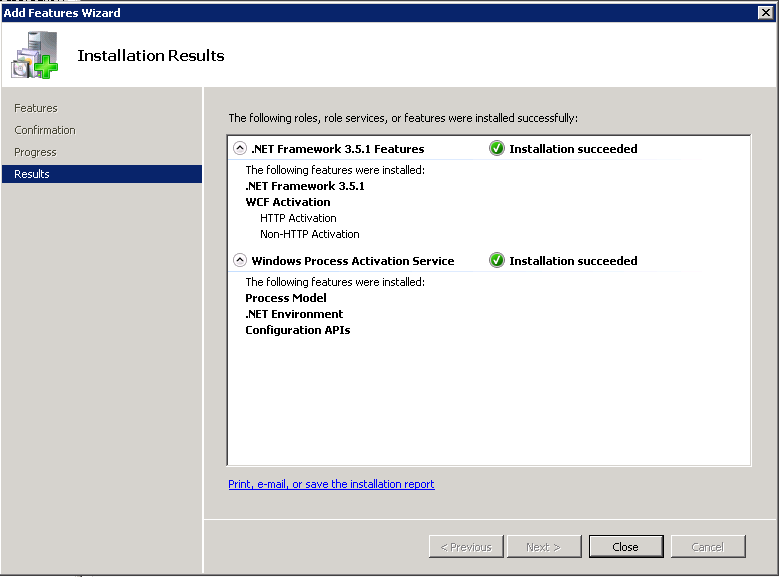
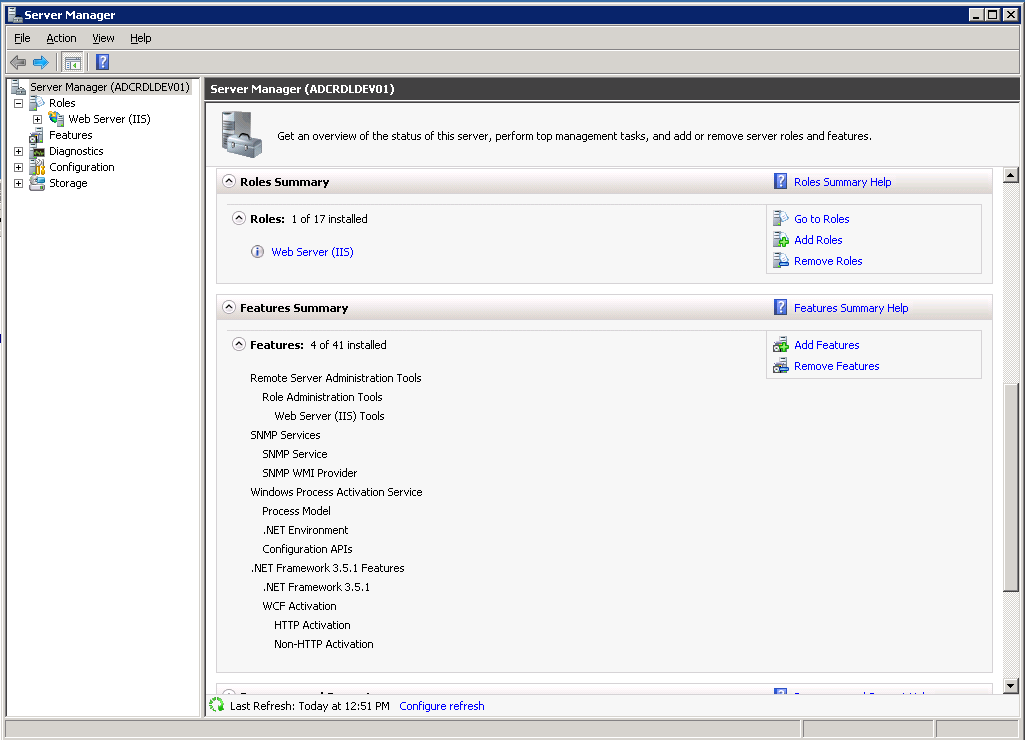


Figure : Successful feature installation

Returning to the Server Manager the additional Feature should now be visible.



## Installing Required Software

To view programs already installed you can open “Program and Features” the Windows start menu select: Start/Control Panel/Programs/Programs and Features.

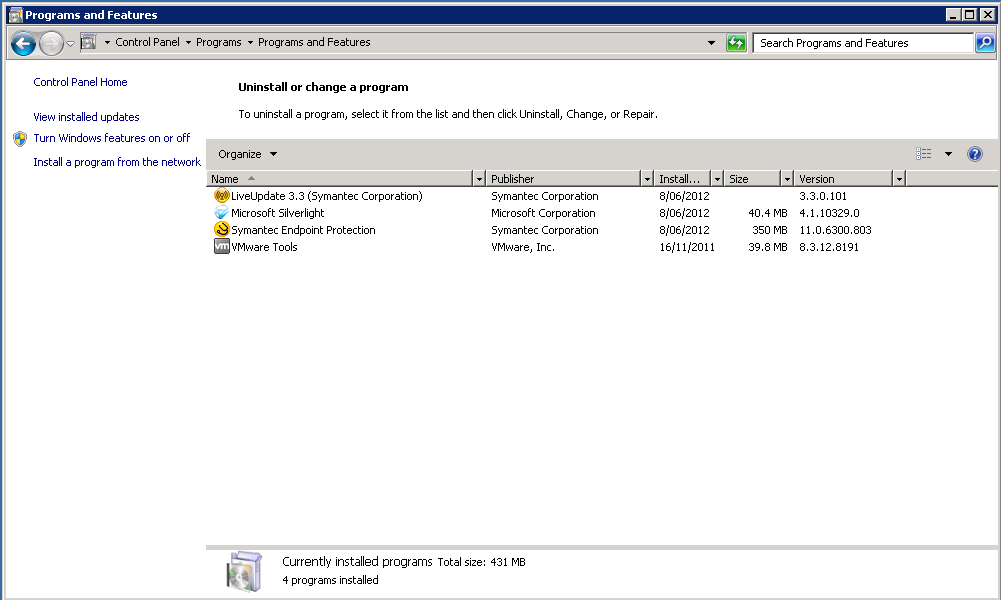


Figure : Example of programs already installed on hte server

### Install Microsoft .Net Framework 4

Download the .NET 4.0 stand alone installer (dotNetFx40\_Full\_x86\_x64.exe) from a trusted site, then execute the installer. Accept the license terms and click the “Install” button.

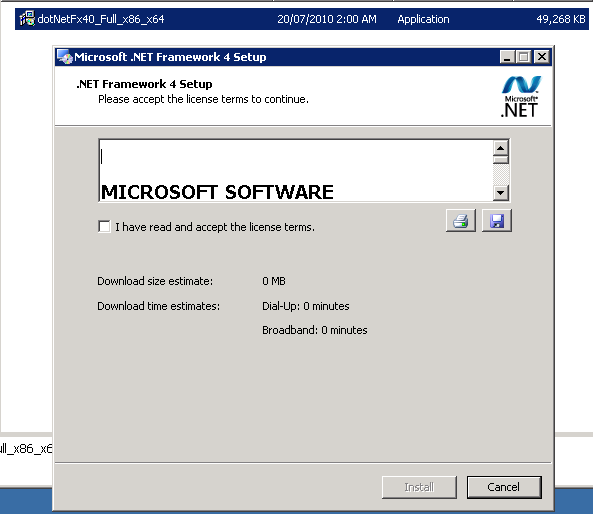
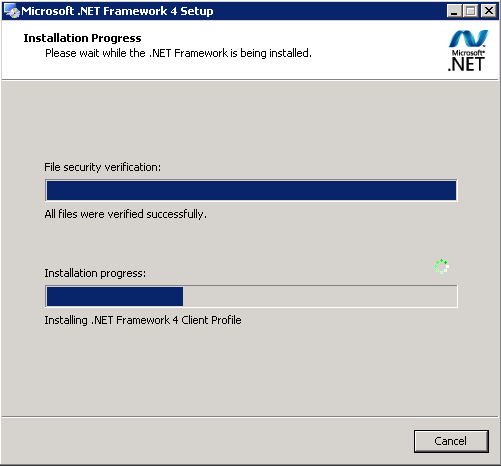
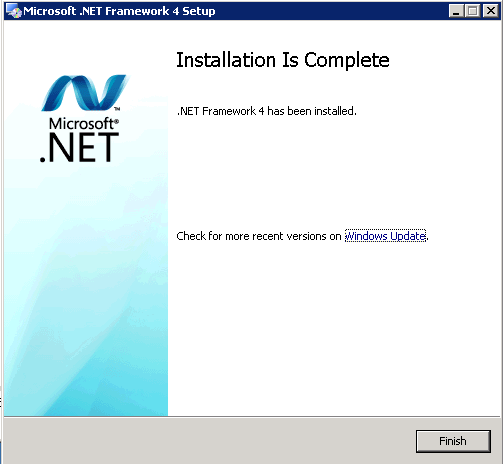
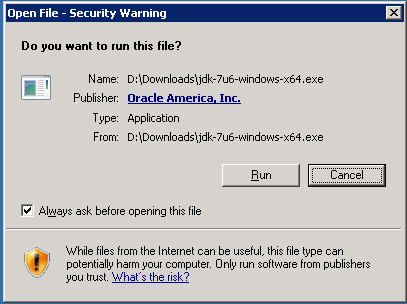
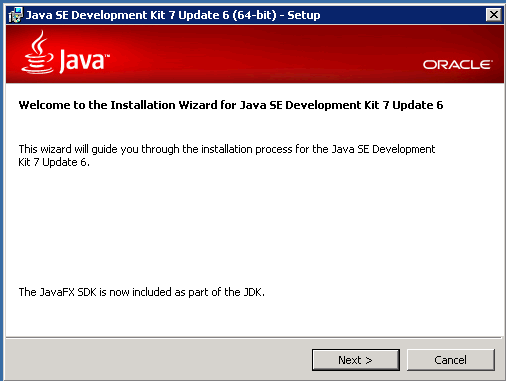
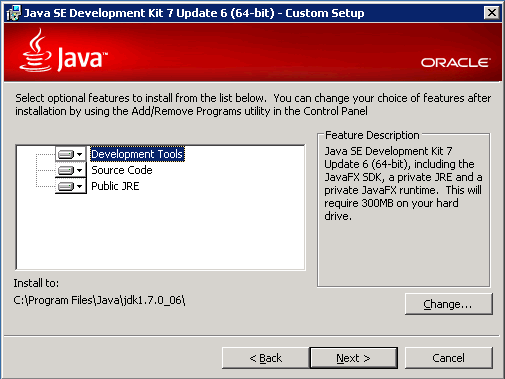
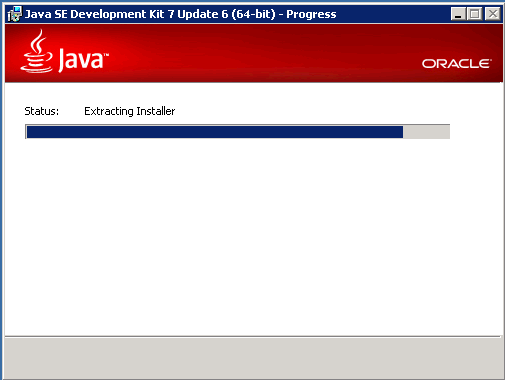
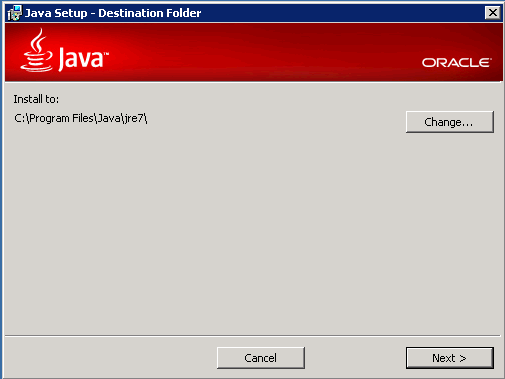
  

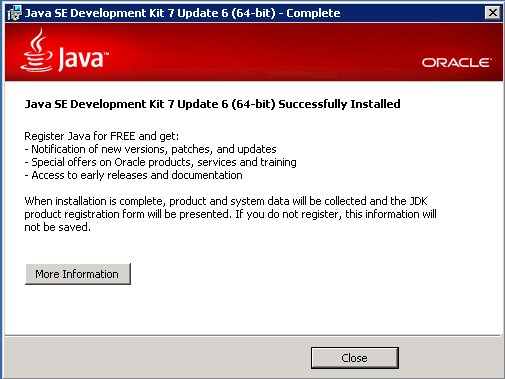
Figure : Accept the terms and conditions and installation progress.

### Install Java JDK 1.7

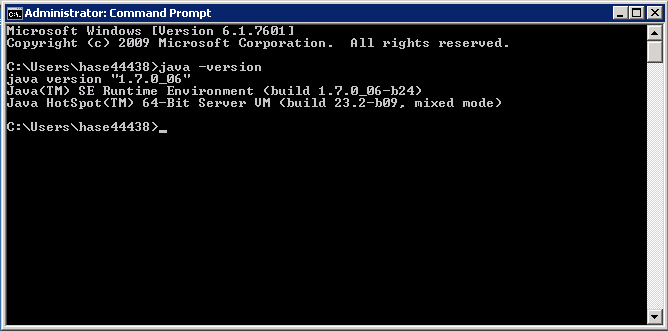
Download the latest 64 bit Java Development Kit stand alone installer (for example jdk-7u6-windows-x64.exe) from a trusted site, then execute the installer. You will need to select “Run” to give permission to run the installer, then follow the installation wizard prompts to install the JDK and JRE.

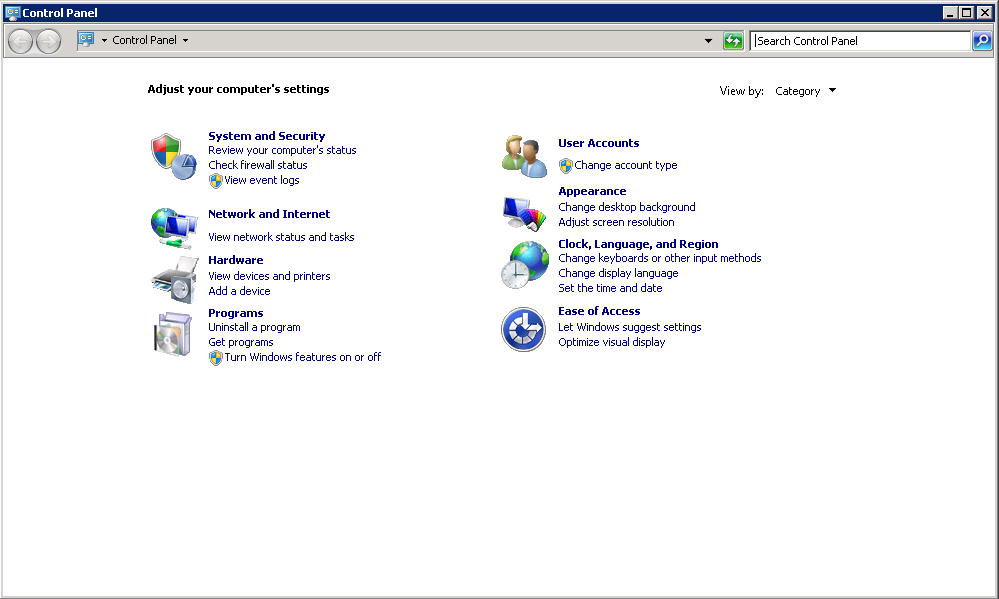
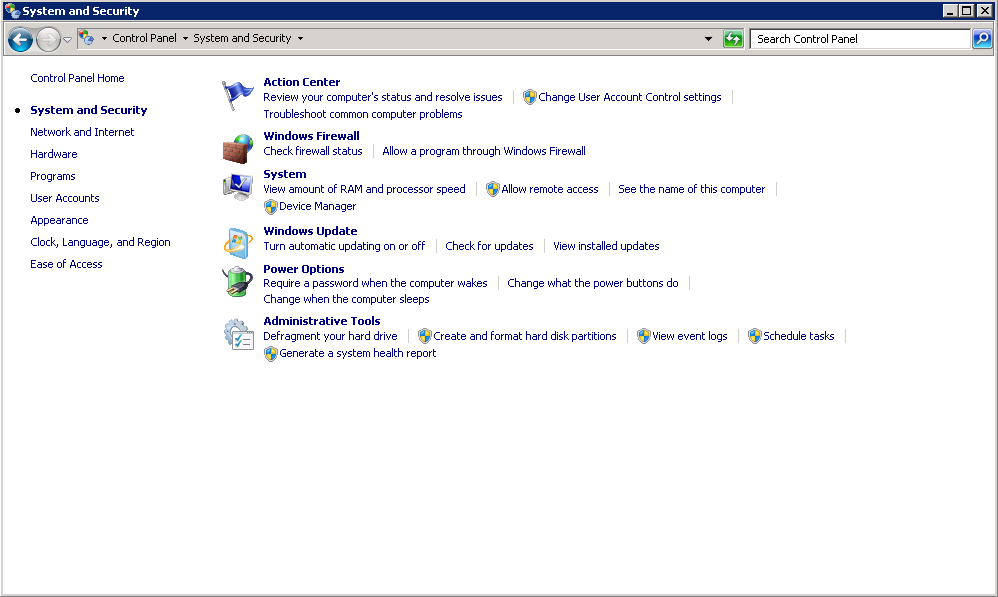
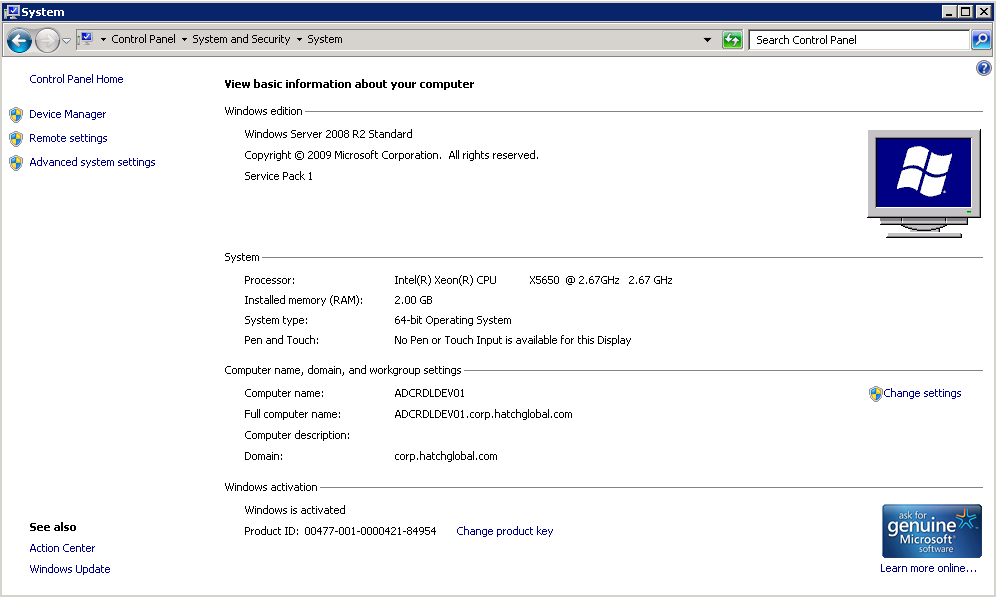


You can confirm that Java has been installed properly by opening a command prompt windows and issuing the command: java –version

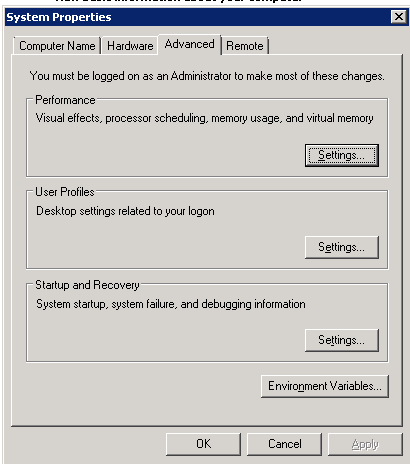
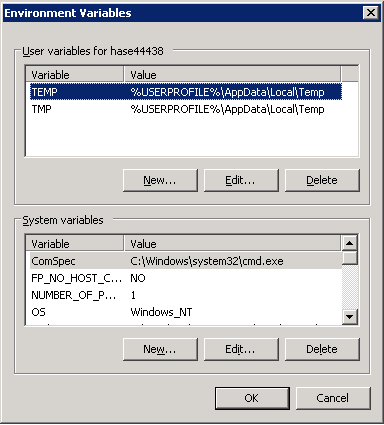
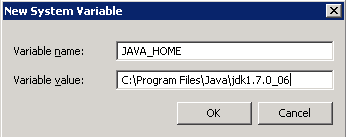


### Setting the JAVA\_HOME Environment Variable.

Adding an environment variable named JAVA\_HOME that points to the Java folder will simplify running Java based applications on the server. Use the “Advanced System Settings” link from “Start\Control Panel\System and Security\System” to add environment variables.

Select the “Environment Variables” from the “Advanced” Tab, then the lower “New” button to add a new System variable. The name of the variable must be JAVA\_HOME and the value of the variable needs to be the full folder path into which the JDK was installed.

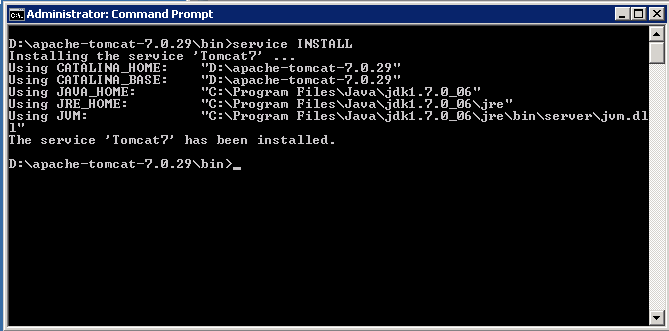
  

### Install Tomcat

Note that we will not be installing Tomcat with a Windows installer and so it will not appear in the “Program and Features” list for the server.

Download Tomcat, apache-tomcat-7.0.29-windows-x64.zip, from a trusted site. You may also like to download the associated documentation, apache-tomcat-7.0.29-fulldocs.tar.gz. Extract apache-tomcat-7.0.29-windows-x64 to D:\apache-tomcat-7.0.29.

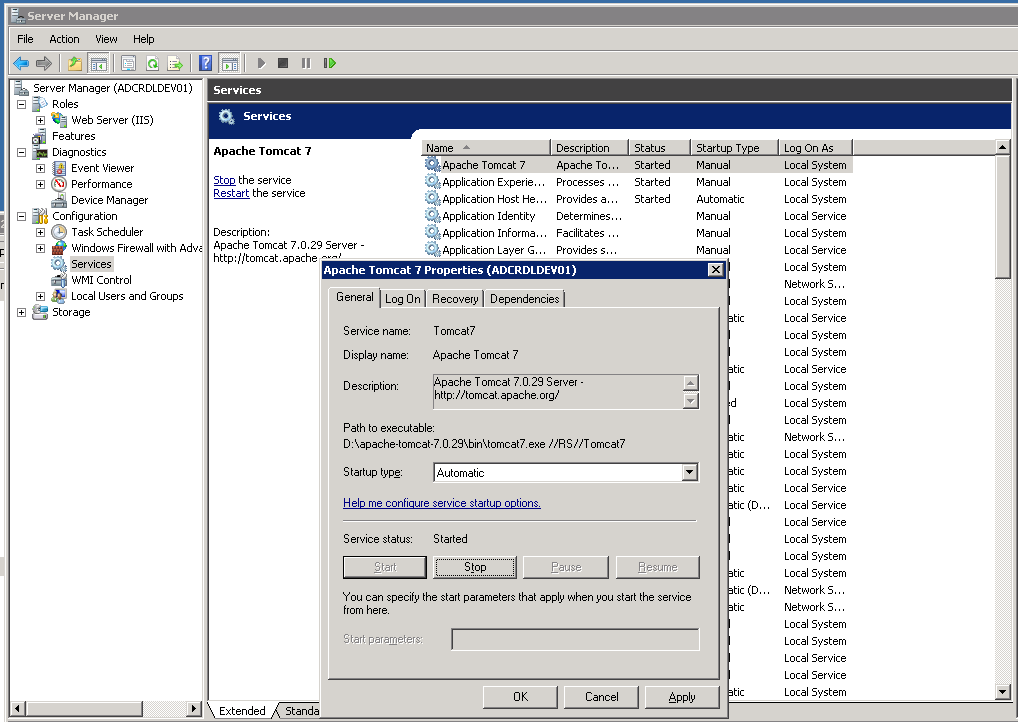
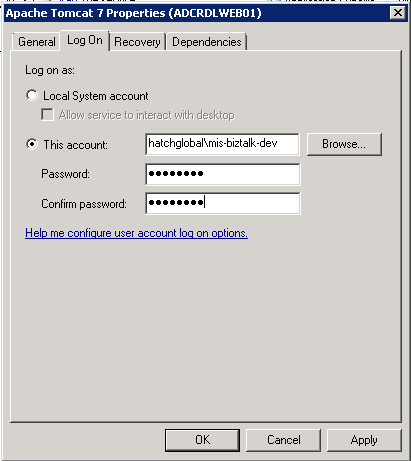
We want Tomcat to run as a Windows service, automatically starting whenever the Server is running. To achieve this open a command prompt and change to the Tomcat “bin” folder the run the service.bat command file passing INSTALL as the parameter.



### Change the Tomcat Service to Autostart

Use Server Manager to display the services installed on the server, Apache Tomcat 7 should appear at the top of the list. Double click on this entry and from the ensuing dialog use the “Startup type” combo box to “Automatic” and press the “Start” button to actually start Tomcat. Finally press OK to save your changes.

You may also need to run Tomcat under a domain account because your iRing implementation may need to access resources protected by domain security, for example a Web Service that uses NTML or Kerberos authentication.

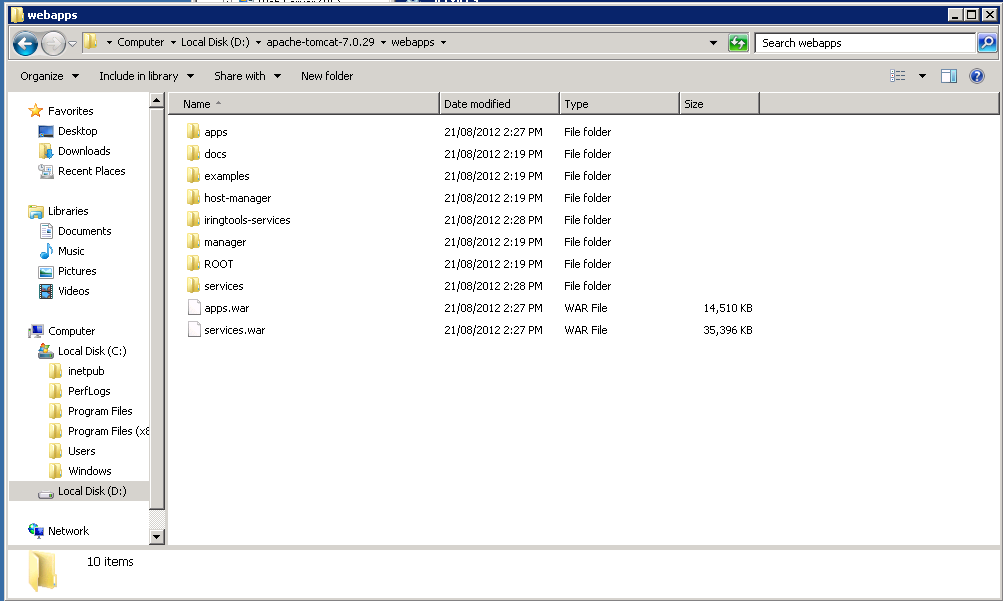
 

## Deploying iRING Exchange Manager

Download the iRINGCore-2.4.0.zip file from a trusted site. Extract from this file the iringtools-apps.war and iringtools-services.war Web Application Archive files to a temporary folder. Note, do not extract them directly to the Tomcat WebApps folder otherwise they will be deployed automatically by Tomcat before you get a change to rename them.

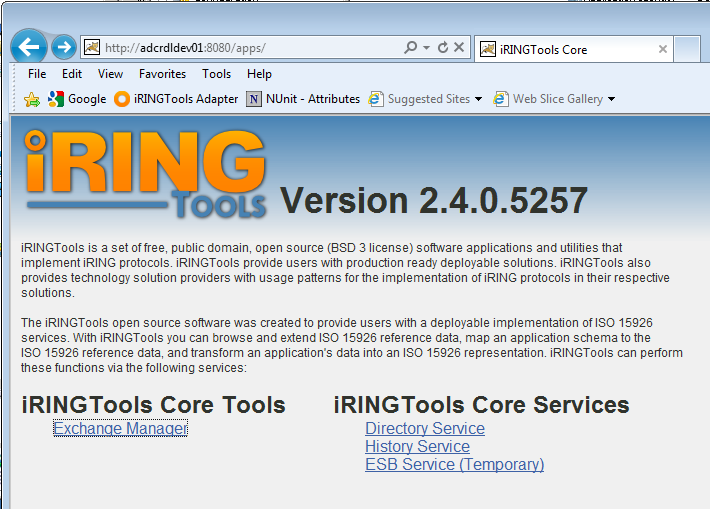
### Deploy the Web Application Archives

Rename the two files to just apps.war and services.war then copy these two files to the Tomcat Web Applications folder, D:\apache-tomcat-7.0.29\webapps. Provided Tomcat is running (as per previous instructions) then the archives will be automatically deployed to the apps and services folders respectively.



### Test the iRING Exchange Manager Web Site

At this point the Exchange Manager applications and services should be available. Open <http://yourservername:8080/apps> to confirm this.

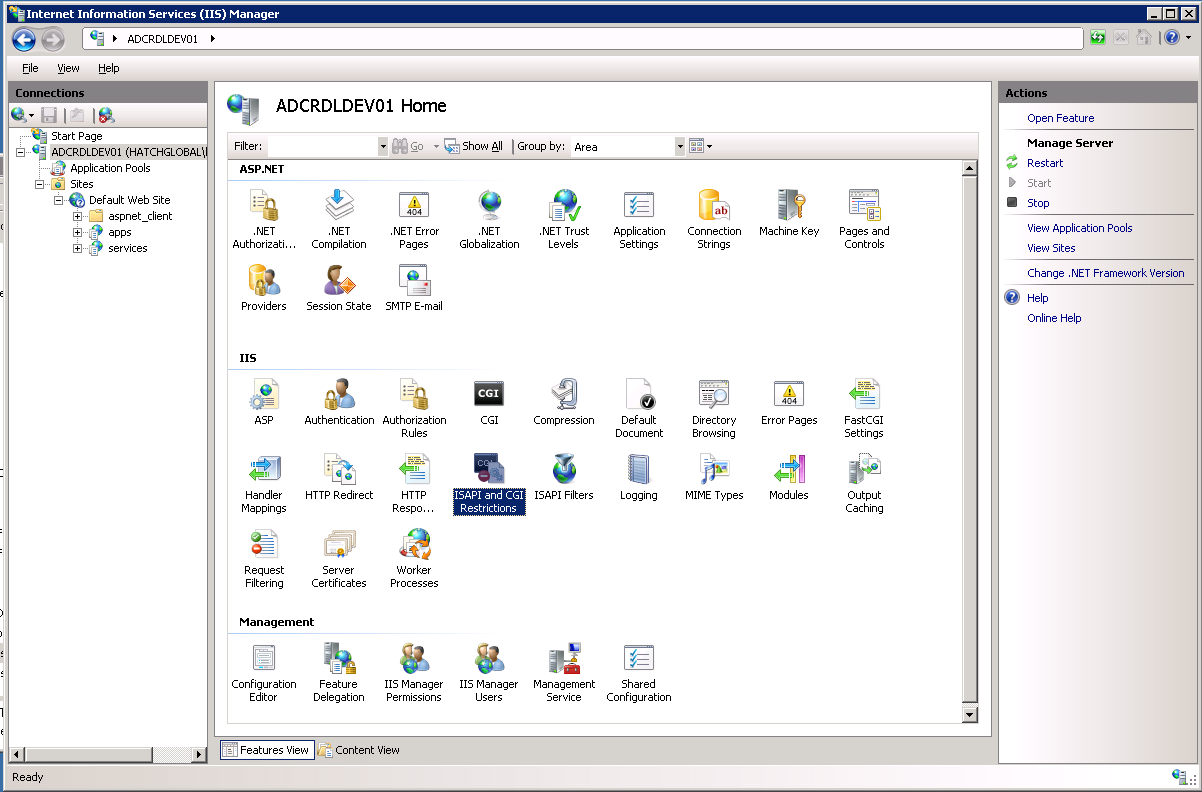
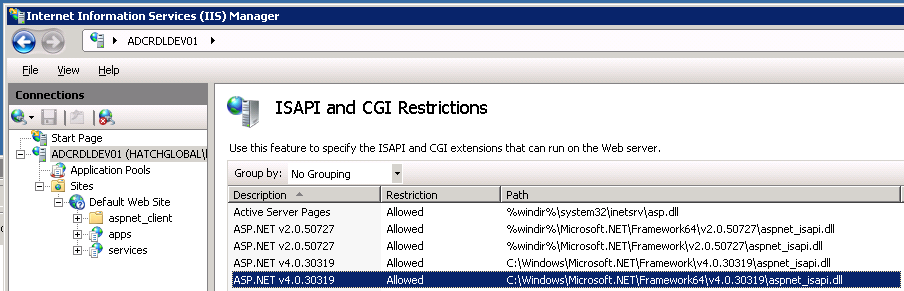


## Deploying iRING Adapter Manager

Download the iRINGTools-2.4.0.zip file from a trusted site. Extract the contents to D:\iRINGTools.

### Allow ASP .NET 4 Extensions to Run on the Server

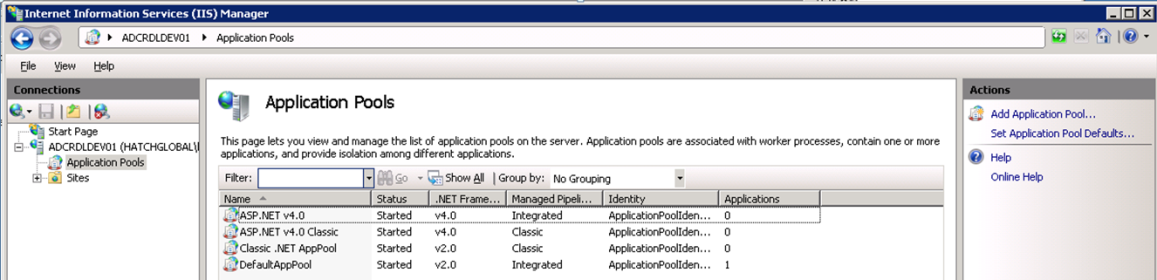
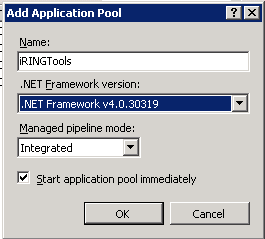
Highlight the Server name, the root of the connections tree in the left hand panel, then double click on ISAPI and CGI Restrictions. Change the ASP.NET v4.x selections to “Allowed”.

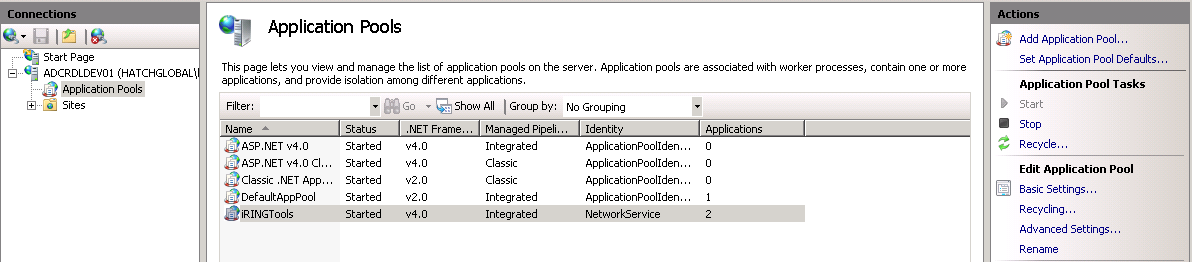
### Create and Application Pool for iRINGTools

Open the IIS Manager, “Start\All Programs\Administrative Tools\Internet Information Services (IIS) Manager” and highlight the Application Pools node.

Select the “Add Application Pool” link to create an application pool under which to run the iRING Adapter Manager facilities. Call the pool “iRINGTools” and change the “.NET Framework version” to “.NET Framework v4.0.xxxx”

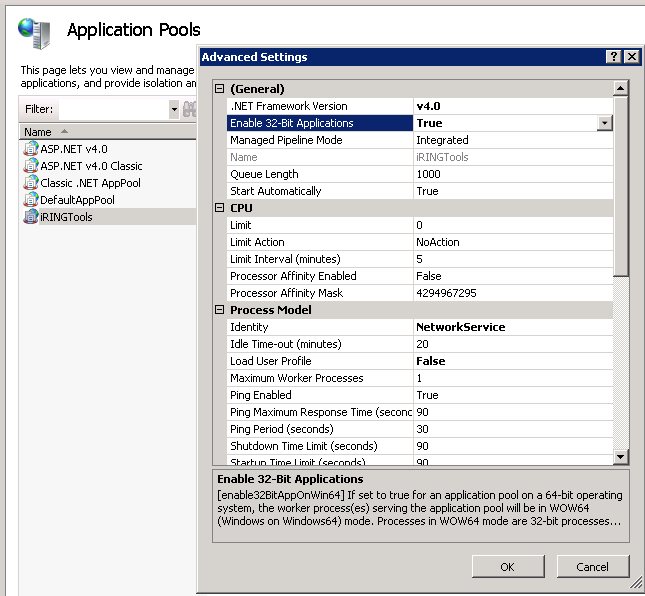
Now highlight the “iRINGTools” pool and select “Advanced Settings…”



From the Process Model pane select the ellipse button of the Identity value and change the Application Pool Identity to “Network Service”. Note: Depending on your organization’s security paradigm you may want to run the iRING Sites under a specific “service” account, if so then instead of using the “Network Service” select “Custom Account” and enter an appropriate domain account and credentials.

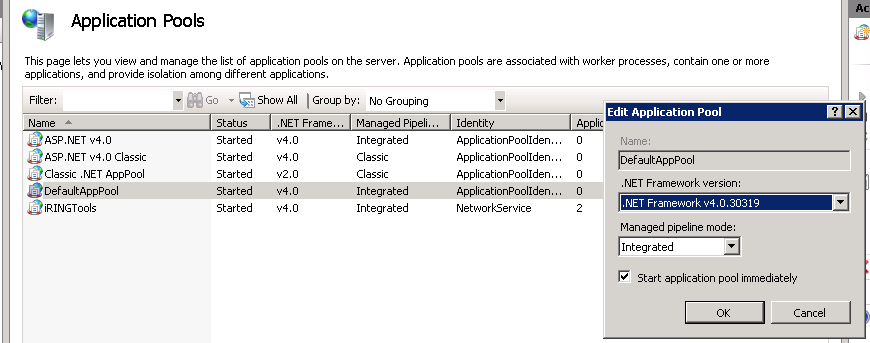
### Allow 32-Bit Applications

To allow 32 Bit Applications, for example the Oracle .Net 4 Client, highlight the “iRINGTools” pool and select “Advanced Settings…” and change the Enable 32-Bit Applications to “True”



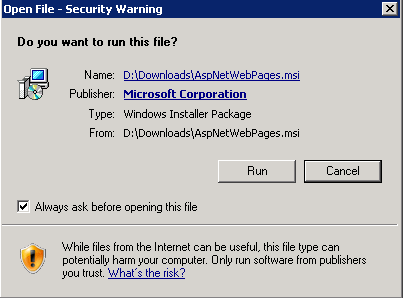
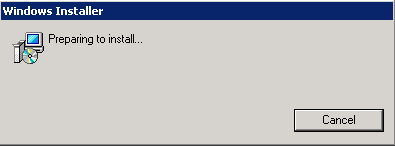
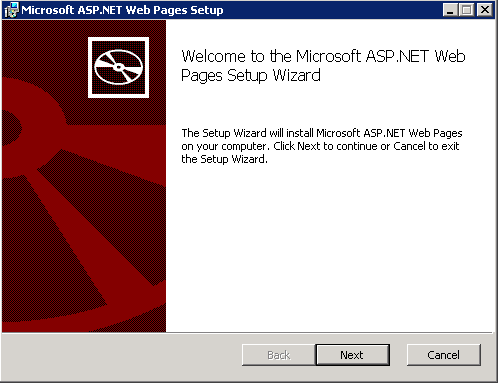
### Change the DefaultAppPool to .Net 4.0

Double click on the DefaultAppPool then change the .NET Framework version: to “.NET Framework v4.xxxx



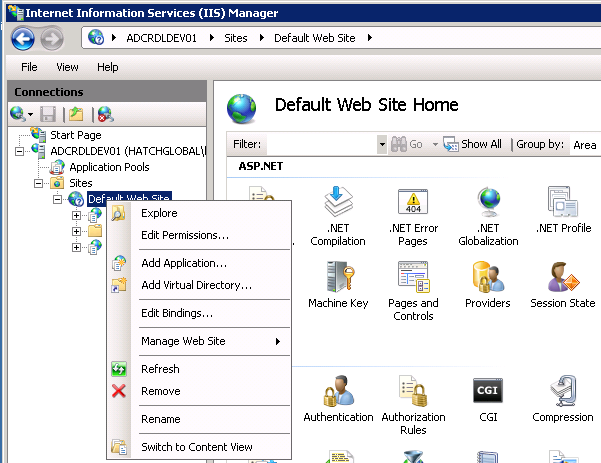
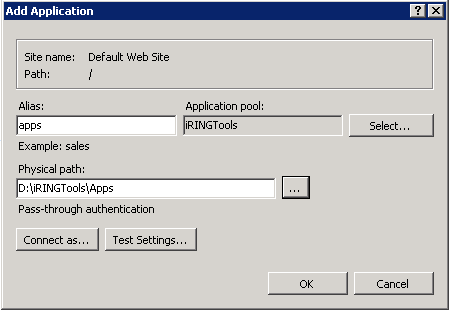
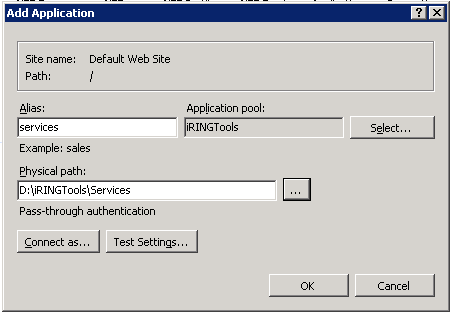
### Install Microsoft AspNetWebPages.

Download AspNetWebPages.msi from a trusted site and double click the msi to launch the installer. Selet “Run” to allow the installer to run. Following the installer prompts, accepting the license agreement, and press the Next and Install buttons as required.

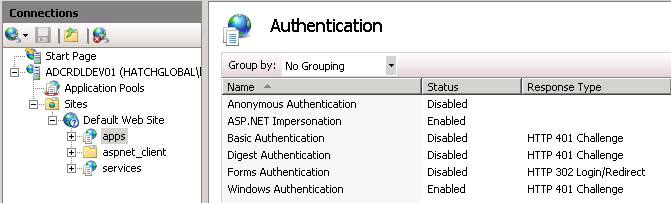
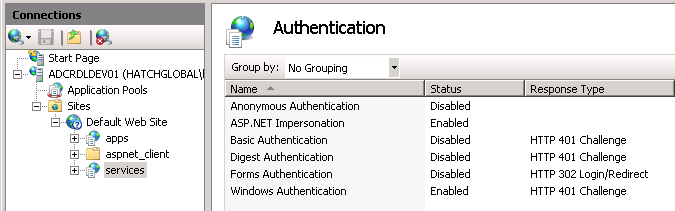
  

### Configure IIS to deploy the Adapter Manager “apps” and “services” as Applications

In IIS Manager right click over “Default Web Site” and then choose “Add Application”. Enter the alias of apps, use the “Select…” button to choose “iRINGTools” as the application pool and use the ellipse button to nominate “D:\iRINGTools\Apps” as the Physical path. Then follow the same process for the “Services” app.

For both “apps” and “services” applications choose the “Authentication” icon and ensure that “ASP .NET Impersonation” and “Windows Authentication” are enabled and that all other selections are disabled.

### Authentication and Access to iRING Configuration

By choosing Windows Authentication and .NET Impersonation users authenticated via your Active Directory will be able to access your iRINGTools Web Site and will access those resources as themselves (IIS will impersonate the user passing on their credential token to access resources on the server.

Note that this means that general users, who will only have the default read access to the Web Site, will be restricted from using iRING services that directly access the server’s resources (files, registry etcetera), they will still be able to see the data exposed by the end points (provided they satisfy any data layer specific security rules), but they will not be able to alter the scope or end point configurations.

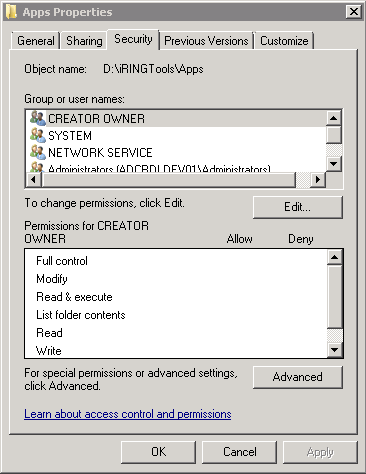
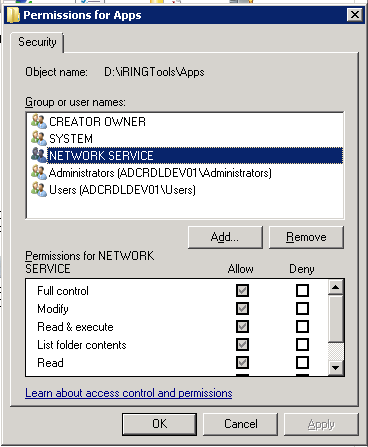
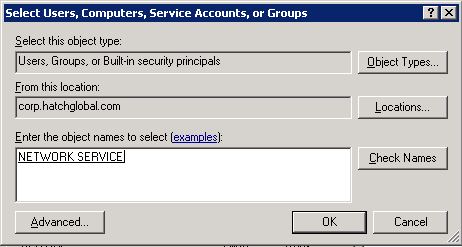
### Create an IRING\_CONFIG Active Directory (AD) Group

You will almost certainly want to create and AD group (for example IRING\_CONFIG) and add to this group domain users that are permitted to configure iRING. For example you would grant IRING\_CONFIG “Full control” in the D:\iRINGTools\Services\App\_Data folder so they would be allowed to create and update the iRING .xml configuration files.

### Set Permissions for the Account the Application Pool is Running Under

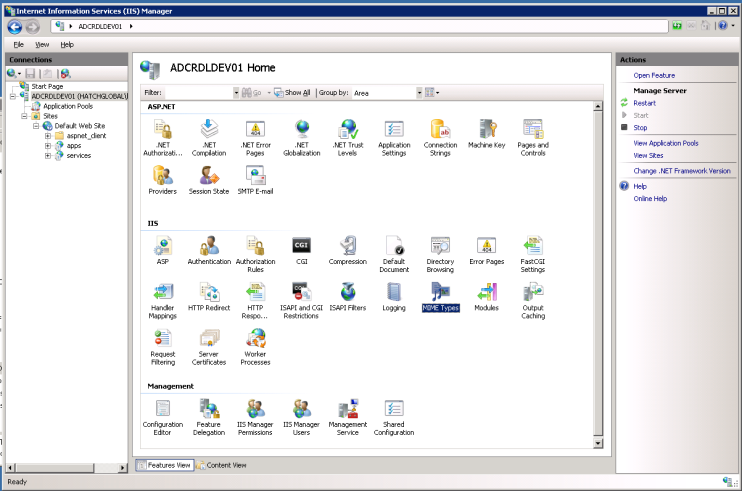
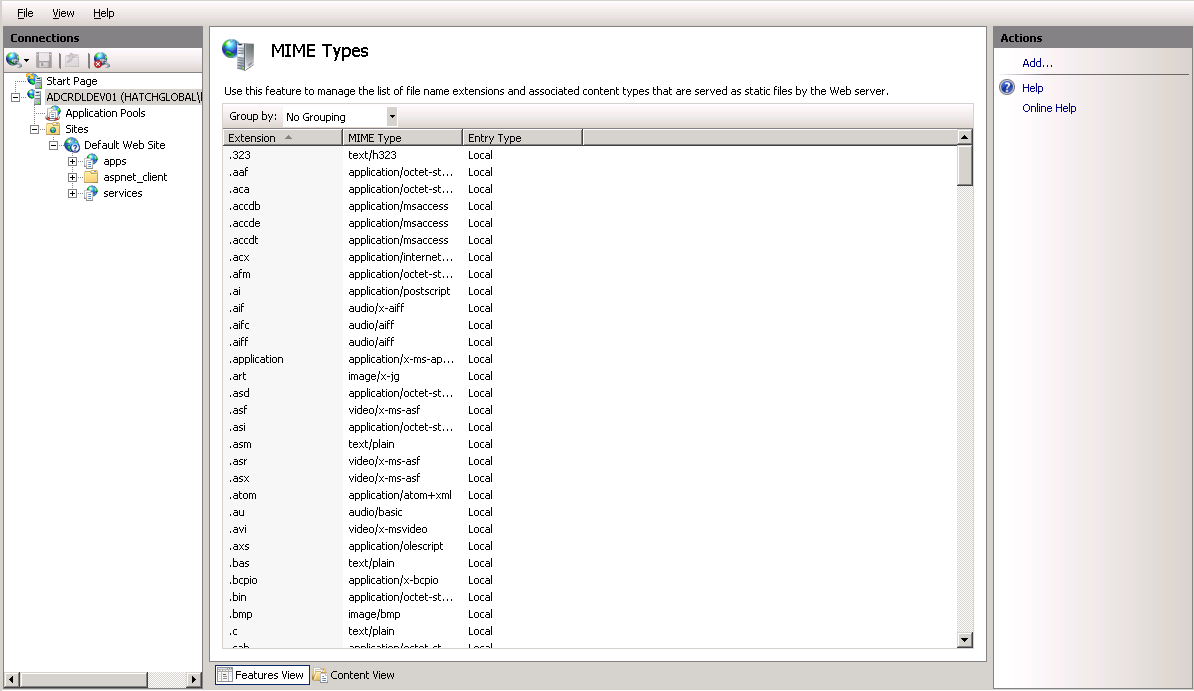
The account that was nominated to run the iRINGTools Application Pool needs full permissions to the files and folders for apps and services. Repeat the following for both apps and services applications.

Highlight the application then from the far right pane select “Edit Permissions”. Select the “Security” tab and then the “Edit” button, and from the ensuing dialog select the “Add…” button. Enter the name of the account used to host the Application Pool, “Network Service” [or your alternate account if running it under a different service account], after clicking OK to accept the name tick the “Full control” check box, the ok again (twice) to complete setting permissions. Remember to repeat for both apps and services.

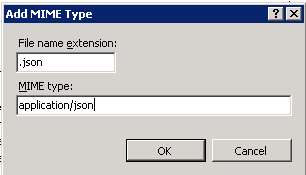
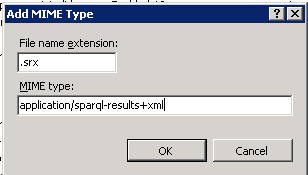
  

### Add MIME Types to Process json and SPARQL results

Highlight the root (server) then double click the MIME Types icon.

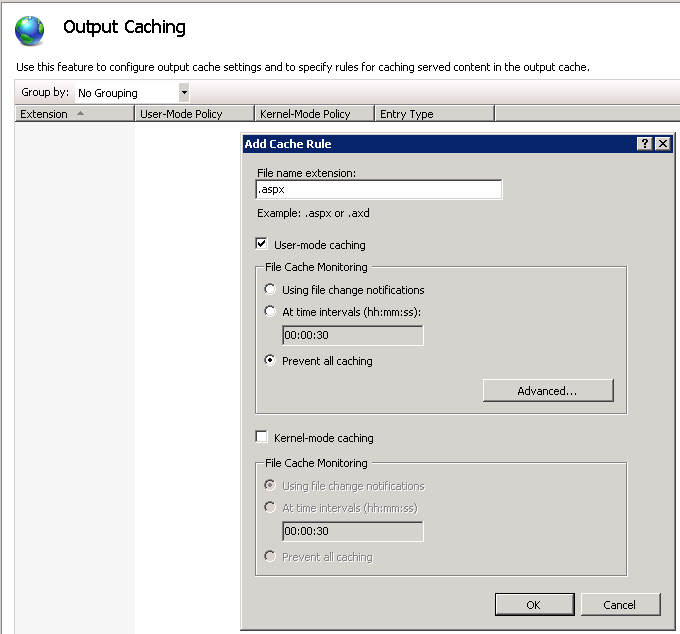
 

Select the “Add…” link from the right hand panel and add “.json” – “application/json” then repeat the process for “.srx” – “application/sparql-results+xml”.

### Prevent Caching of .aspx

Select the root (server name) in the IIS tree then select Output Caching, then ”Add” a configuration to prevent caching of .aspx.



## Registering Assemblies in the Global Assembly Cache (GAC)

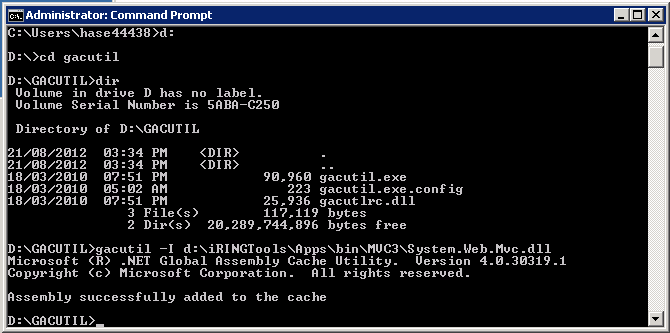
### Getting the GACUTIL Utility

“GACUTIL” is a utility is provided as part of the .NET SDK and can be used to register assemblies in the GAC. You would not generally want to install the entire .NET SDK on your server, instead install it on a workstation and copy the following files from the “C:\Program Files\Microsoft SDKs\Windows\v7.0A\bin\NETFX 4.0 Tools” folder to a folder (for example D:\GACUTIL) on the server: gacutil.exe, gacutil.exe.config, gacutlrc.dll.

### Register an Assembly in the GAC

There may be assemblies that you require on the server to support your organizations custom data layers. For example you might require MVC2 and MVC3 assemblies. Given that both versions of the assemblies are named System.Web.Mvc.dll you obviously cannot copy both assemblies into the iRINGTools applications or services bin folders. Instead you would create two sub-folders, bin\MVC2 and bin\MVC3, each with the applicable version of the assembly, and then register the assemblies in the GAC.

To register an assembly in the GAC invoke GACUTIL using the –I parameter and the full path to the assembly, for example: gacutil /i d:\iRINGTools\Apps\bin\MVC3\System.Web.Mvc.dll



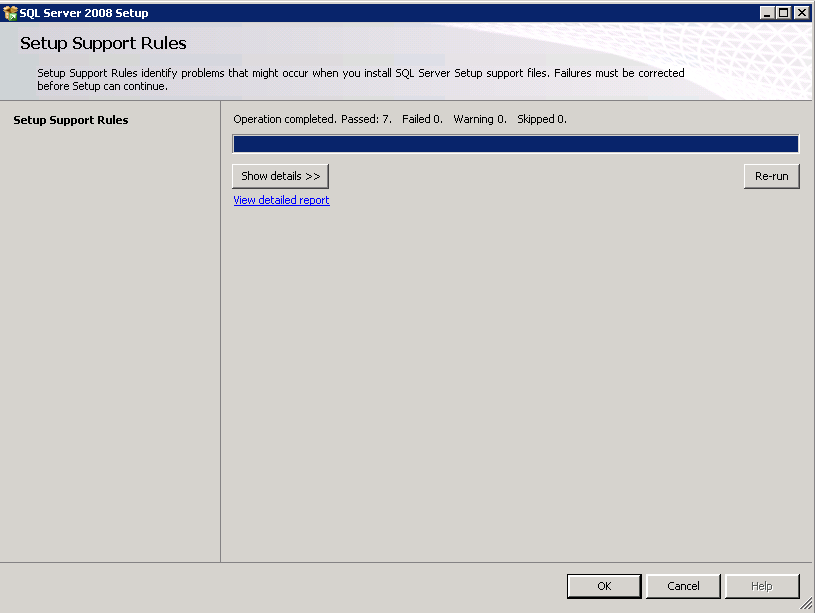
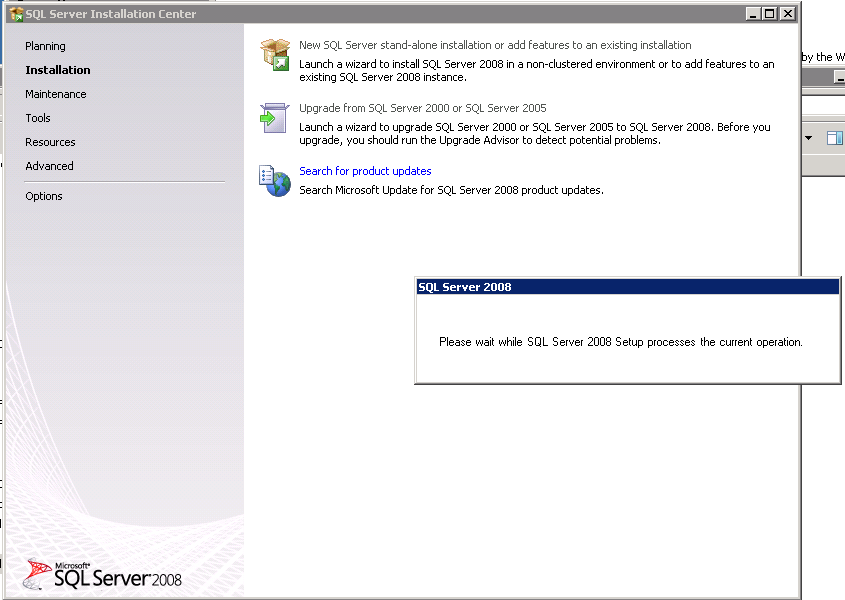
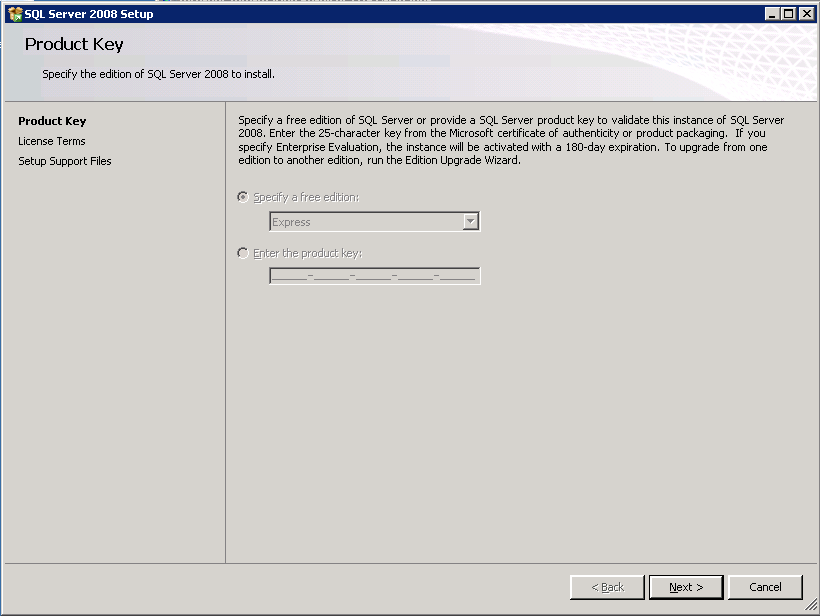
## Install Database Drivers

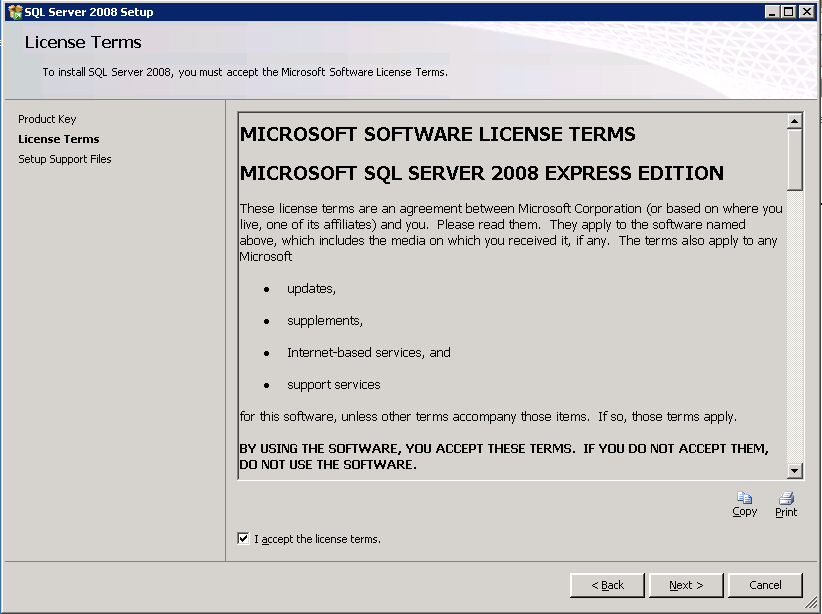
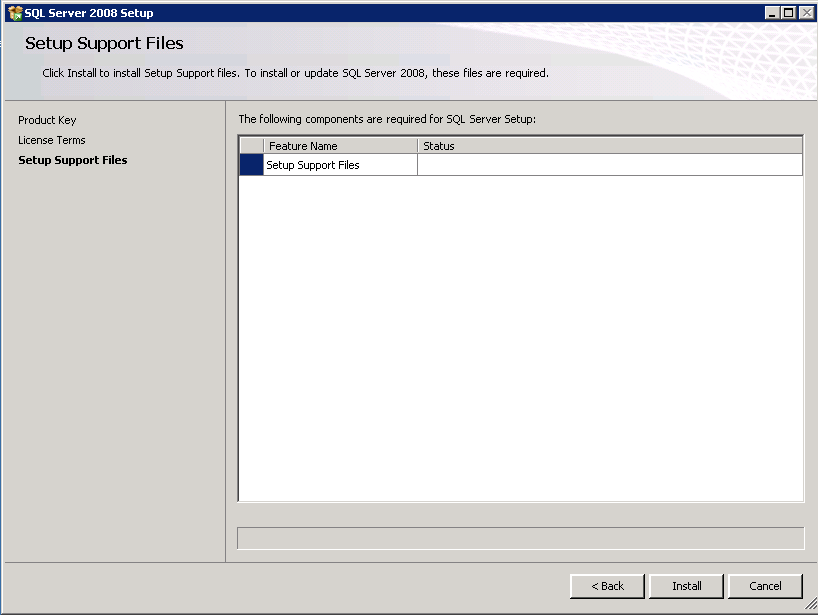
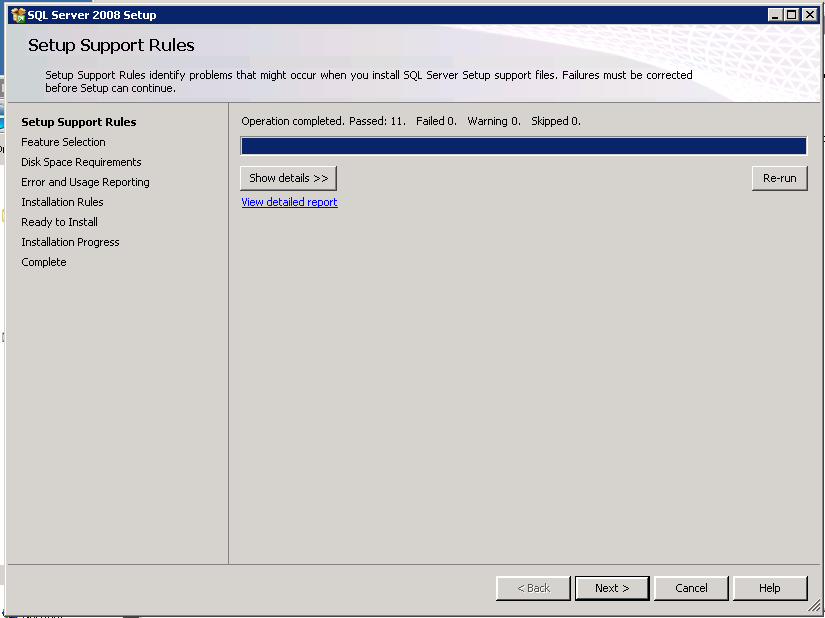
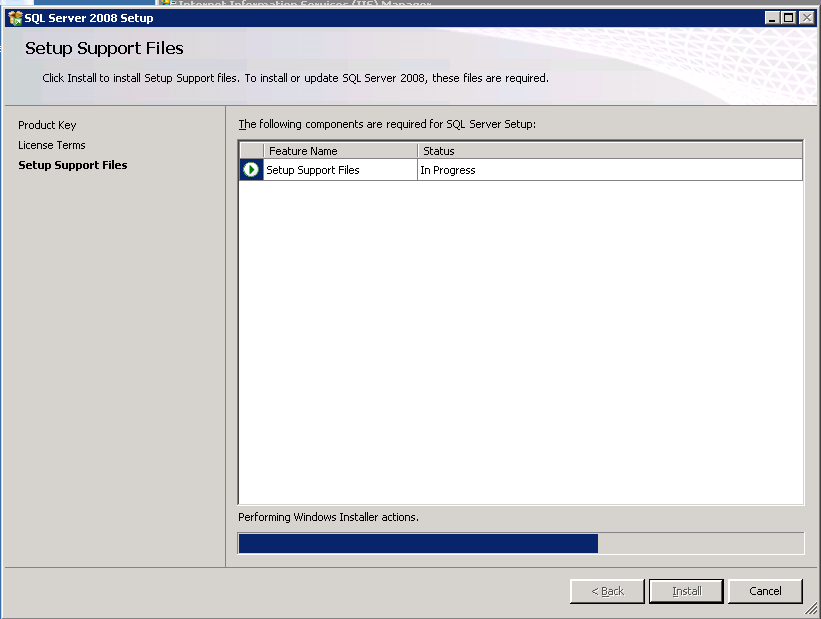
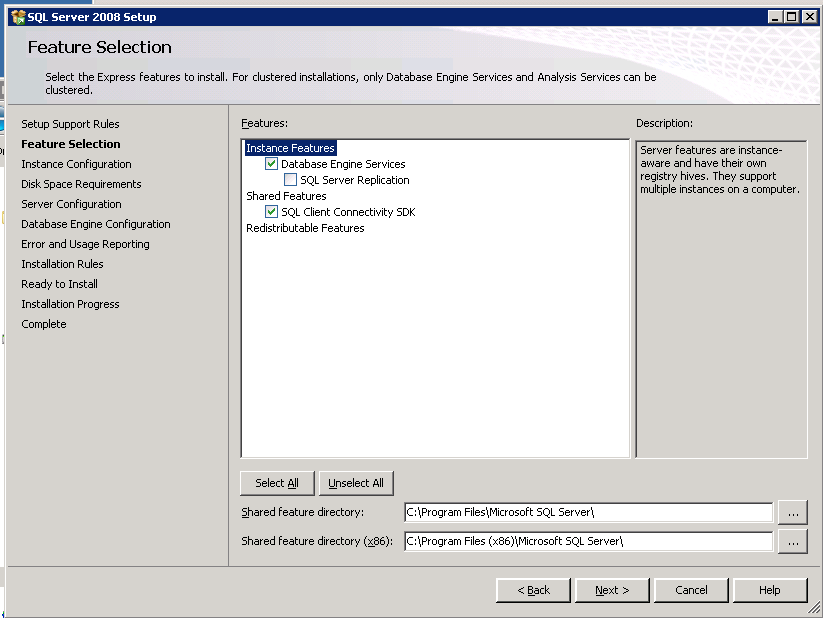
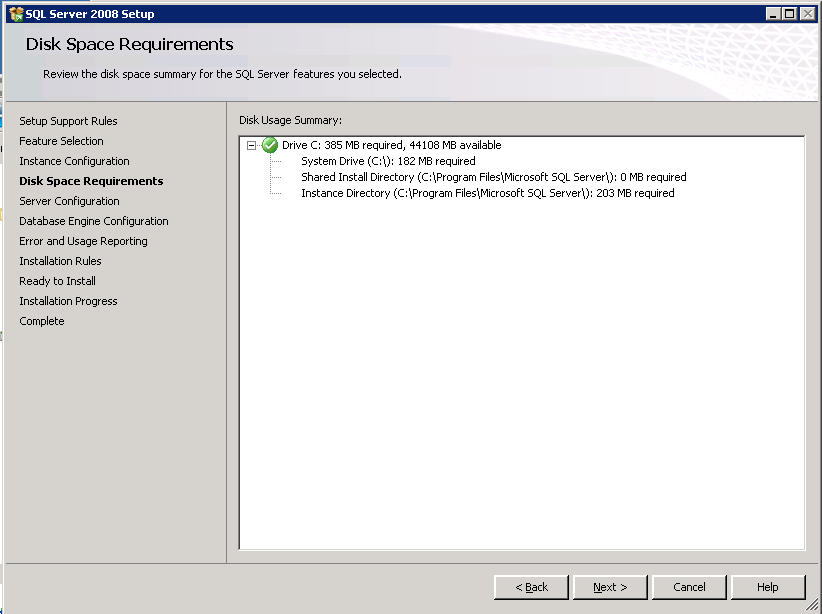
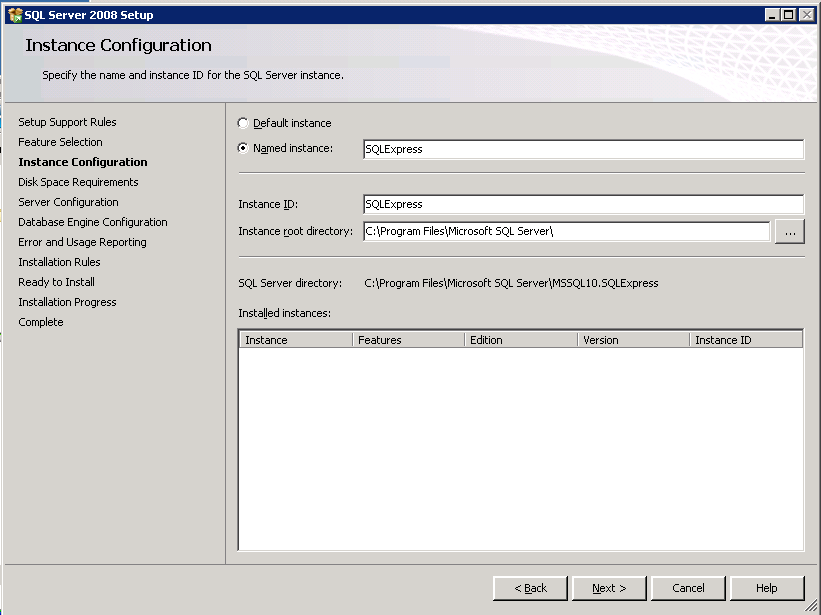
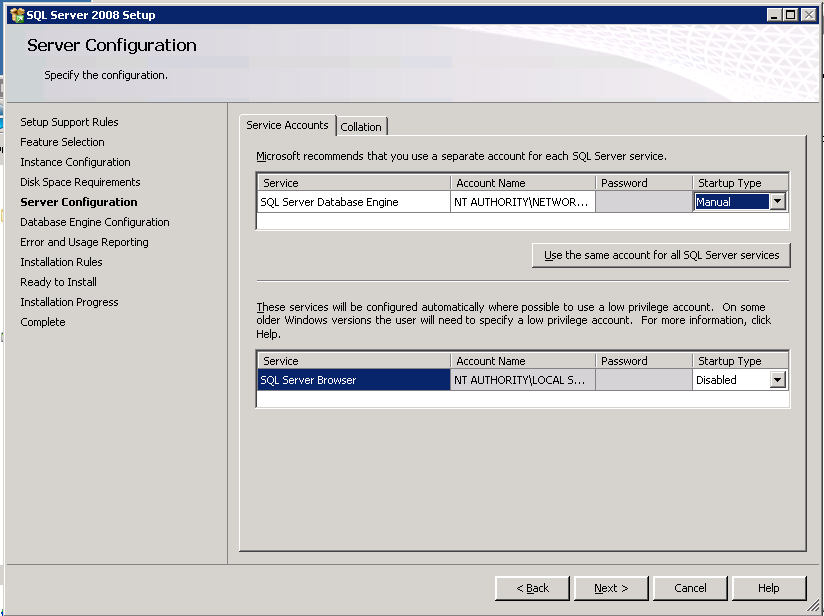
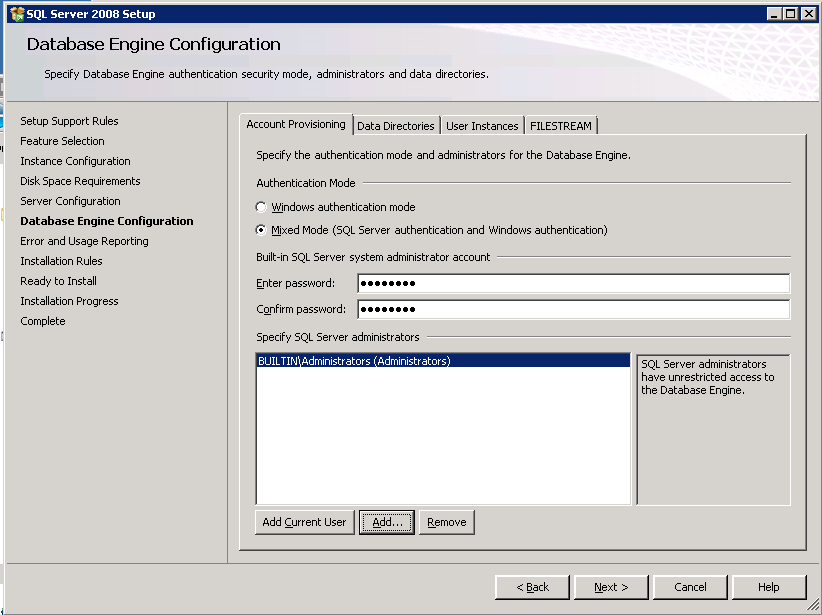
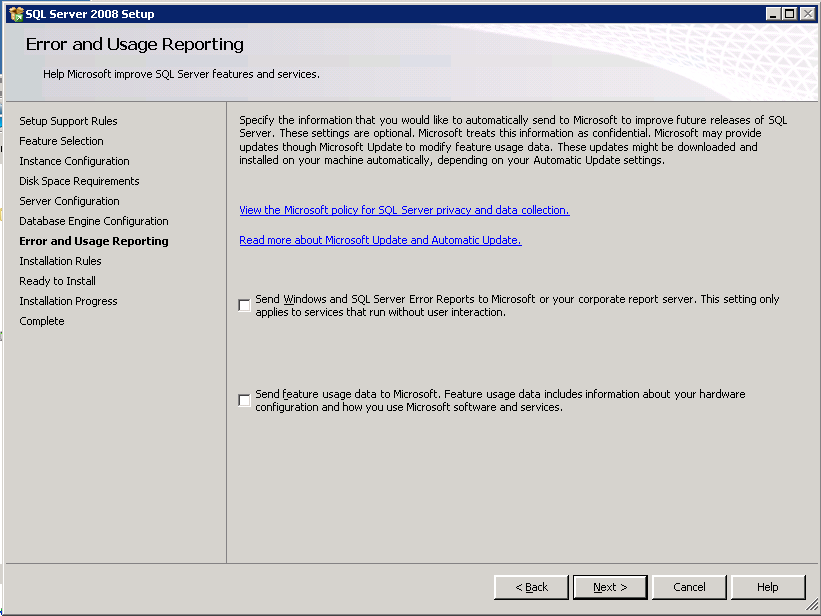
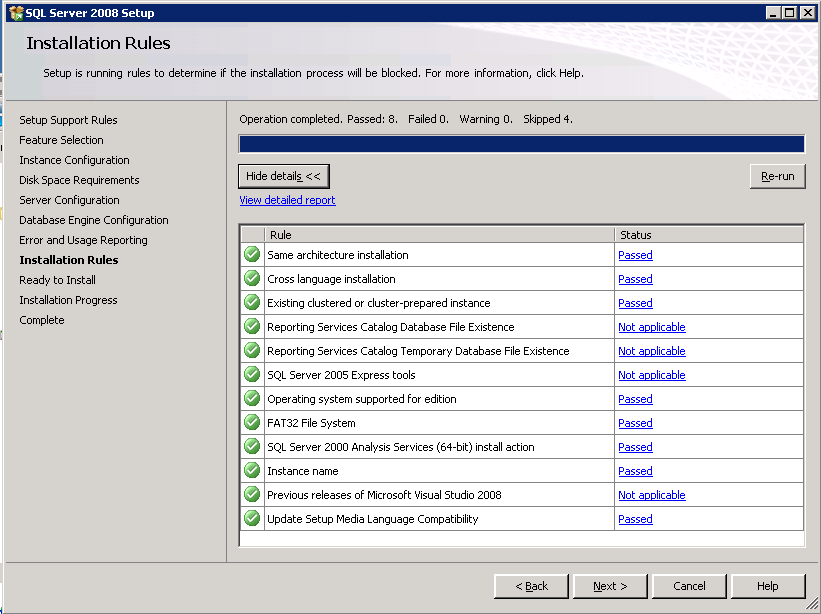
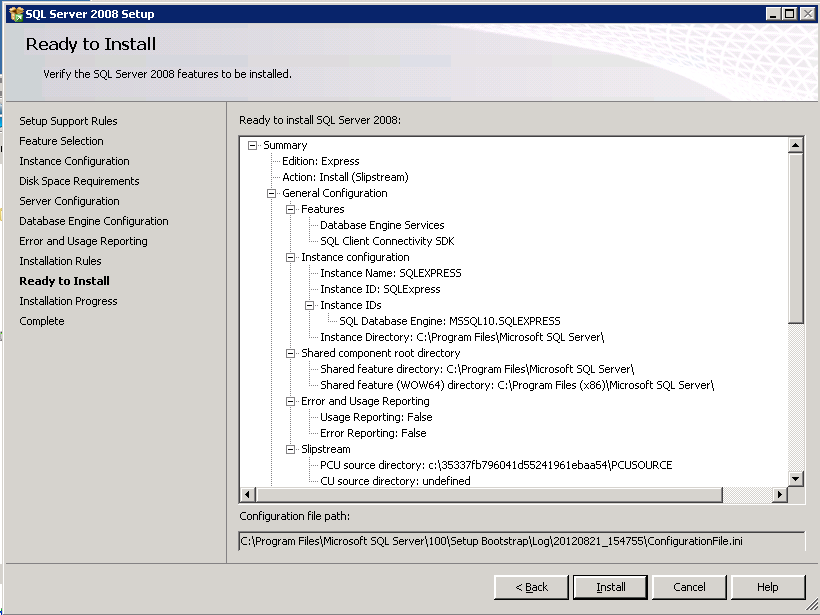
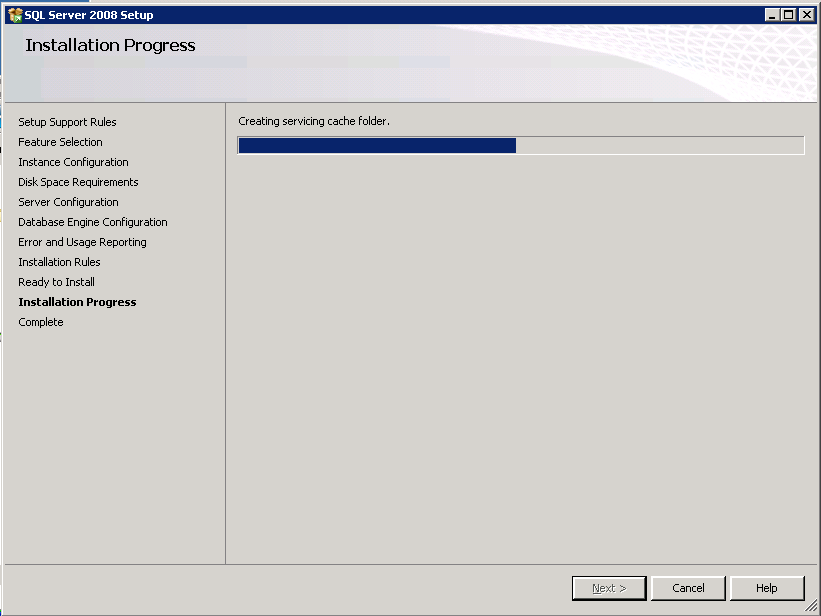
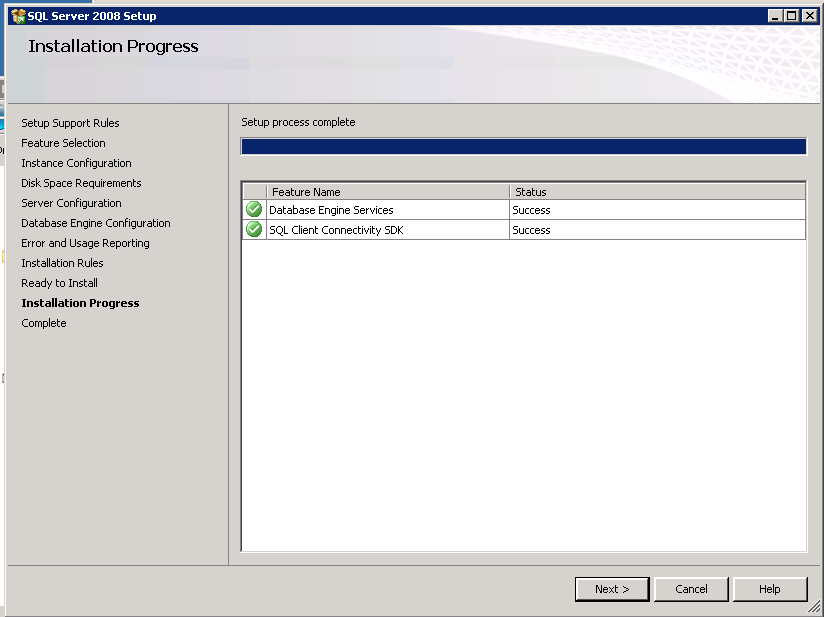
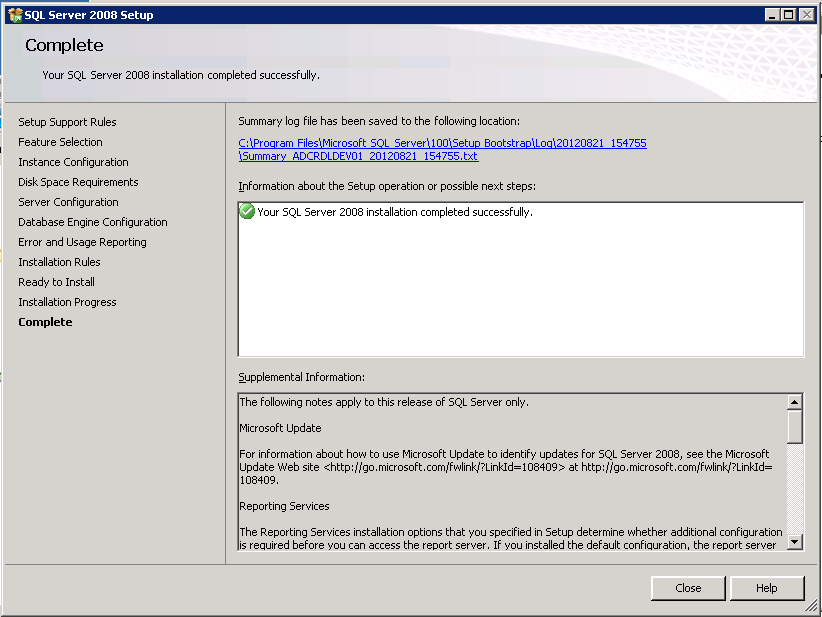
iRINGTools data layers commonly access data stored in database repositories, for example MS-SQL and Oracle databases. For these data layers to work the database client assemblies need to be installed on the server.

### Installing MS-SQL Assemblies

Note at the time of writing is document not all required MS-SQL assemblies were automatically included in the iRINGTools bin folders and so the easiest way to ensure all the necessary MS-SQL assemblies were installed and registered was to install SQLExpress on the server. Note that during the installation you can choose NOT to run the database server automatically so there is no performance hit on the server from running an unneeded database [aside from the cost of some disk space for the installed database server files].

Download the SQLExpress 64 bit stand alone installer, SQLEXPR\_x64\_ENU.exe, from a trusted site and execute it. Follow the installation wizard, taking care to change the startup type to Manual on the Server Configuration dialog.

### Oracle Client Setup

At the time of writing the latest Oracle Client release was 11.2.0.3, the selected deployment vehicle was the Oracle Developer Tools with Oracle Data Access (ODTwithODAC11203.zip). Unzip the aforementioned .zip and run the setup.exe.