INGENIUX

www.ingeniux.com

Content Management System 8.0

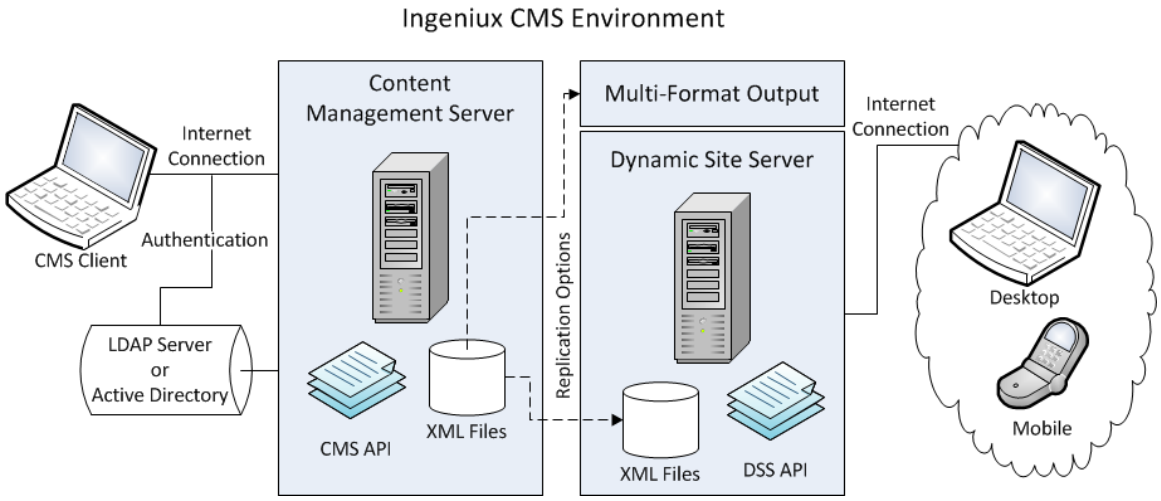
Installation Guide

Revision 2

Table of Contents

1 Ingeniux CMS Basics

A standard implementation of the Ingeniux Content Management System (CMS) comprises two server environments: the Content Management Server (CMS) and the Dynamic Site Server (DSS). Each server has its own API and its own store of XML content. Typically, XML content is created and updated on the CMS and served to site visitors from the DSS.



In many implementations, the CMS server is placed behind a firewall and integrated with an enterprise's authentication processes, data repositories, and legacy applications.

1.1 Content Management Server

The CMS provides an environment in which users can create, update, and publish content. As content is published, it is replicated to the DSS, where site visitors can view it.

One server environment can have several CMS instances installed. Each instance can manage one or more websites.

The CMS API—called the CSAPI, or Content Store Application Programming Interface—is the interface for programmatically managing a CMS instance, its settings, and all of its content. The CSAPI runs behind an IIS web server. Every feature available in the CMS web application user interface is available through the CSAPI.

The CSAPI also provides the interface for several extensibility features that are often used by customers. Extensibility features include the CMS event model (called Custom Hooks) and various customer-specific UI components (called Custom Tabs).

All content in the CMS is stored in native XML documents, and the CMS and DSS applications manage and process this content using internal XML processors. Content can be published from the CMS in several formats:

- As XML to be consumed by a DSS instance

- As Multi-Format-Output (MFO) content for other technologies
- As static HTML

The CMS can be implemented in each of the following configurations:

- With a standard TCP connection over port 80
- Using a TCP port other than port 80 for TCP traffic
- Using a secure SSL connection (HTTPS)
- Behind a reverse proxy

The CMS requires additional configuration to support a reverse proxy. To specify the URL for a reverse proxy server, go to **Administration** > **System Options** > **CMS** > **Reverse Proxy**.

1.2 Dynamic Site Server

Built on the Microsoft .NET framework, the Dynamic Site Server (DSS) delivers content published by the CMS. The DSS supports ASP.NET 4.0, MVC 3, and mobile device detection. It also provides out-of-the-box support for existing XSLT implementations.

The DSS is designed to serve site content in dynamic, multi-format environments. Key features include:

- Default support for ASP.NET MVC 3 and the Razor view engine
- Leveraging of the MVC 3 output cache and authentication/authorization system
- Simple structured traversing of XML documents with the .NET API
- Deferred execution of query statements using LINQ syntax
- Strong-typed support for CMS elements, including Navigations, Links, and Taxonomy Navigations
- Support for runtime-executed element types, including Insert, Component, ComExecute, and Password elements
- Strong-typed support for the CMS taxonomy system at runtime
- Out-of-the-box support for the CMS 7.5 structured URL system
- Integrated support for CMS preview and In-Context Editing
- Support for the User Agents and Sites model employed by the CMS
- Support for all transform options

1.3 Publishing vs. Replication

Updating content on the DSS is a two-step process. First, content is published on the CMS. Then the published content is replicated to the DSS. This two-step process ensures the availability of the DSS and prevents complications that could arise if the DSS accessed files as they are being published.

The two-step model works as follows:

- **Publishing** – Performed by the CMS. Publishing is the process of creating XML files from the pages in the site tree and sending these files to a publishing target folder. This folder is always located on the CMS server in the \xml\pub\ directory.

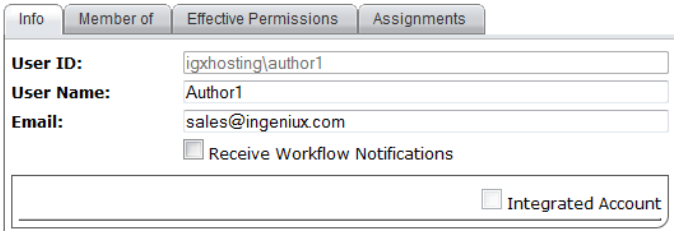
- **Replication** – The process of copying XML from a publishing target on the CMS to a target directory on the DSS. On the DSS, the replicated XML functions as the model (the data store) for the MVC site. For a single-server implementation, the XML can also be replicated to another directory on the CMS.

1.4 Authentication

The CMS supports two types of authentication: Windows domain and LDAP directory.

For both types, the CMS captures user credentials, passes them to the authenticating agent, and authorizes the user based on a successful authentication. For a particular user to work with content in the CMS, the user's Ingeniux ID has to match the user ID in the authenticating directory or database.

In a Windows domain environment, the user ID syntax has to follow the pattern domain\user when a new user is created in the Users/Groups manager.



For LDAP and for custom authentication mechanisms, the user account syntax must match that used by the authenticating method.

The DSS relies on standard IIS security to determine access to published content. Typically, this entails the use of anonymous access in conjunction with an account such as IUSR.

Both the CMS and the DSS can be configured in IIS to use Secure Sockets Layer (SSL) connections. The server application handles this level of security, so configuring SSL won't impact the CMS or DSS sites as long as client requests can get through.

2 System Requirements

This section of the Installation Guide describes basic hardware and software requirements for the CMS system.

2.1 CMS Environment

For optimum performance and stability, Ingeniux recommends installing the CMS and DSS on different physical servers. Under this architecture, the CMS is typically located within an organization's Local Area Network (LAN), behind a firewall. Published content is then replicated to the DSS.

Although Ingeniux recommends using two physical servers, the software can run on a single server. This is sometimes useful for an internal intranet site or for low-load, low-security external sites.

In a typical two-server configuration, the two servers perform under significantly different loads. A CMS generally handles a small number of content contributors performing processor-intensive and disk-intensive activities. A DSS handles a comparatively large number of visitors browsing the site. As a result, the DSS has to manage less processor and disk load.

Content contributors access the CMS through the web-based CMS client. Users can work remotely as long as they have a web connection and permission to access the software. The CMS client is browser and platform independent.

2.2 CMS Requirements

The CMS is a Windows-based platform that runs on Windows Server 2003 or Windows Server 2008/R2. Detailed hardware and software requirements are described below.

2.2.1 Minimum Hardware Requirements

- Quad core Intel Xeon (3 GHz or better)
- 2 GB RAM
- 4 GB free disk space (plus 2x content)
- Serial ATA or Ultra SCSI disks; 10,000 RPM minimum; RAID optional

2.2.2 Hardware Recommendations

Note: Increased hardware requirements are based on server load and user count.

- 4 GB RAM or better
- Gigabit Ethernet

- Dual NICs

2.2.3 Software Requirements

- Operating Systems
    - Microsoft Windows Server 2003 or 2003 R2
        - 32-bit only
    - Microsoft Windows Server 2008 or 2008 R2
        - 32- and 64-bit
        - Web Server Role – all features
    - Windows Vista Business or Windows 7 Professional (for test environments only)
        - 32- and 64-bit
        - Web Server Role – all features
- MSXML 4.0 XML Parser SP3
- Microsoft .NET Framework 4
- Microsoft IIS 6.0 or later
- SMTP or MAPI compliant messaging system

2.3 DSS Requirements

The DSS is an optional part of the Ingeniux CMS solution. The DSS runs on Windows Server 2003 or Windows Server 2008/R2. Detailed hardware and software requirements are described below.

2.3.1 Minimum Hardware Requirements

These are minimum hardware requirements. Load factor may require higher performance servers or multiple servers. Multiple servers can be load balanced.

- Quad core Intel or AMD processor, 2 GHz or better, depending on site traffic and load. Hyper-threaded servers are recommended for heavily trafficked sites.
- 1 GB RAM or better
- 500 MB free disk space (plus 2x content)
- Serial ATA or Ultra SCSI drive; 7,200 RPM minimum; RAID optional

2.3.2 Hardware Recommendations

- 4 GB RAM or better
- 500 MB free disk space (plus 2x content)
- Serial ATA or Ultra SCSI drive; 7,200 RPM minimum; RAID optional

2.3.3 Software Requirements

- Operating systems
    - Microsoft Windows Server 2003 or 2003 R2
        - 32-bit only
- Microsoft Windows Server 2008 or 2008 R2
    - 32- or 64-bit
    - Web Server Role – all features
- Windows Vista Business or Windows 7 Professional (for test environments only)
    - 32- or 64-bit
    - Web Server Role – all features
- MSXML 4.0 Parser SP3
- Microsoft .NET Framework 4
- Microsoft .NET Framework 3.5
- ASP.NET MVC 3.0
- Microsoft IIS 6.0 or later

To Install .NET 4 and MVC 3:

1. Download and install *Microsoft Web Platform Installer*, located at http://www.microsoft.com/web/downloads/platform.aspx
2. Click **Products**.
    - Search for .net 3.5, and click **Add** for *.NET Framework 3.5 SP1*: http://prntscr.com/lhsss
    - Search for .NET 4, and click **Add** for *Microsoft .NET Framework 4*: http://prntscr.com/lhqr6

- Search for MVC 3, and click **Add** for *ASP.NET MVC 3 (Visual Studio 2010)*: http://prntscr.com/lhqus

3. Click **Install**.

2.4 Browsers Supported by the CMS Client

The CMS client requires one of the following Internet browser/operating system combinations:

Windows XP/Vista/7:

- Internet Explorer 6
- Internet Explorer 7
- Internet Explorer 8 (supported as of CMS 6.0 SR5)
- Internet Explorer 9 (supported as of CMS 6.0 SR6 and higher)
- Firefox 3 or higher
- Chrome 9 or higher
- Safari 3.2x or higher

Mac OS X 10.4+:

- Safari 3.2x, 4.x, and 5.x
- Firefox 3 or higher

Support for additional browsers may be added at a future date. Check the Ingeniux Support Site for the most current list of supported browser/operating system combinations.

3 Installation

Most installation projects fall into one of the following categories:

- Installing a new CMS
- Upgrading to a new major release of the CMS
- Upgrading to a new minor release of the CMS

This section describes, in general terms, the process of installing the CMS. Upgrades are described in section 8. Before proceeding with an installation or upgrade, read the pertinent sections of this guide. If you have questions, contact Ingeniux Support.

The steps for installing the CMS are as follows:

1. Install Internet Information Services (IIS) in Native Mode to support Application Pools.
2. Install Indexing Service.
3. Install ASP.Net support.
4. Install Microsoft XML 4.0 SP 3 Parser.
5. Run IGXSetup on the CMS.
6. Install and configure the CMS site by running either the IGX_CMS_Site_Setup or IGX_CMS_Site_Upgrade.
7. Run IGXSetup on the DSS.
8. Install and configure the DSS site by running IGX_Dynamic_Site_Server_Setup.

3.1 Installation Recommendations

Before you begin installing a CMS, it's a good idea to consider site configuration and file storage options. Thoughtful planning will help you maximize site performance and organize the CMS environment effectively.

Ingeniux recommends separating the CMS system and supporting files from the CMS site and log files. One way to separate the CMS system from the CMS site is to use two physical drives: a system drive and a site drive.

This configuration provides several advantages:

- Easy backup and restoration
- Limited impact of drive failures

- Limited system drive capacity requirements
- Clearly delineated resource allocation
- Simplified troubleshooting

The system drive should contain:

- Windows system files
- IIS files excluding IIS log files (installed in the Windows directory)*
- ASP.NET (installed in the Windows directory)
- CMS system files
- .NET Framework (installed in the WINDOWS\Microsoft.NET\Framework\v3.5 directory)
- MSXML 4 SP3 (installed in the Windows\system32 directory)
- Any additional supporting software

  * The associated \inetpub directory and its contents are not necessary for a CMS installation unless otherwise noted by the site developer.

The site drive (the drive on which the CMS site and log files will be installed) should contain:

- Site files
- Indexing catalogs
- Ingeniux log files
- IIS log files

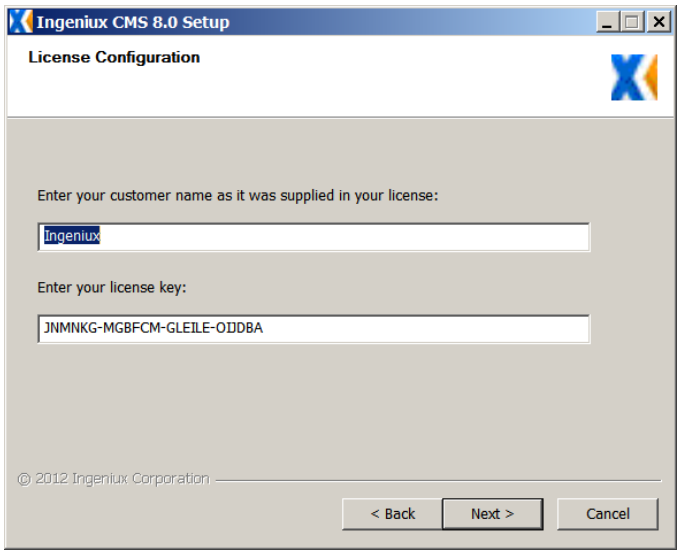
Ingeniux also recommends that the servers running the CMS and DSS software:

- Run only the CMS/DSS applications and the services needed to support them.
- Perform no additional network services (for example, DNS, domain, or LDAP services).
- Host only Ingeniux CMS-managed websites. If additional websites are hosted on the same server, they should run under a separate website and Application Pool in IIS.

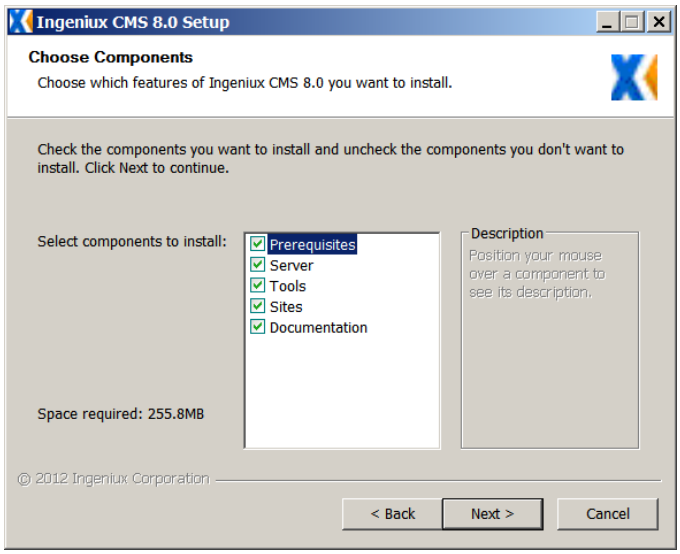
3.2 Installing the CMS System

To install the CMS system, run the file IGXSetup, which can be downloaded from the Ingeniux Support site: http://support.ingeniux.com/downloads.

Read and agree to the license terms, and then enter your customer name and license key in the License Configuration dialog. (Ingeniux Support provides the product license key. Keys are typically valid for one year.) You can paste the key, with dashes, directly into the license key box.



Enter a valid license key and click **Next**. The Choose Components dialog opens with a list of components to install.

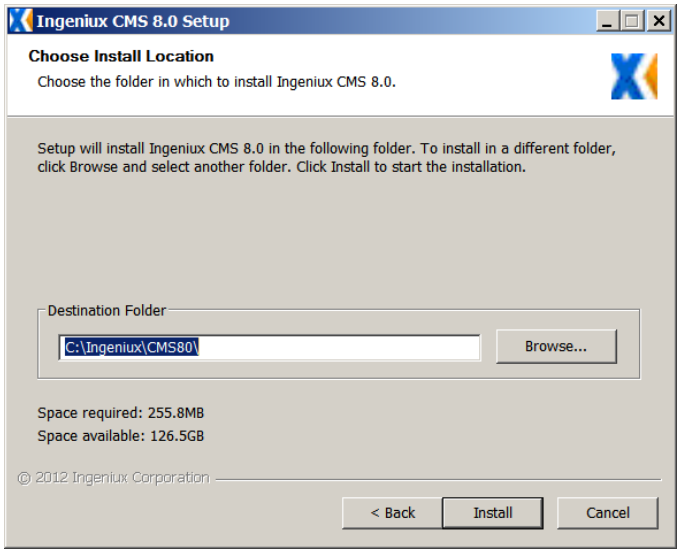


For a standard installation, select all components: **Prerequisites**, **Server**, **Tools**, **Sites**, and **Documentation**. Then click **Next**. If your license key doesn't provide for a DSS installation, only the CMS component will be installed.

Next, the Setup wizard prompts you to choose the destination folder where the software will be installed.

- Do not select the \Program Files directory. The space between the words can cause problems registering the DLLs.

The CMS system should be installed on the same drive as the Windows installation. The default installation folder is C:\Ingeniux\CMS80.



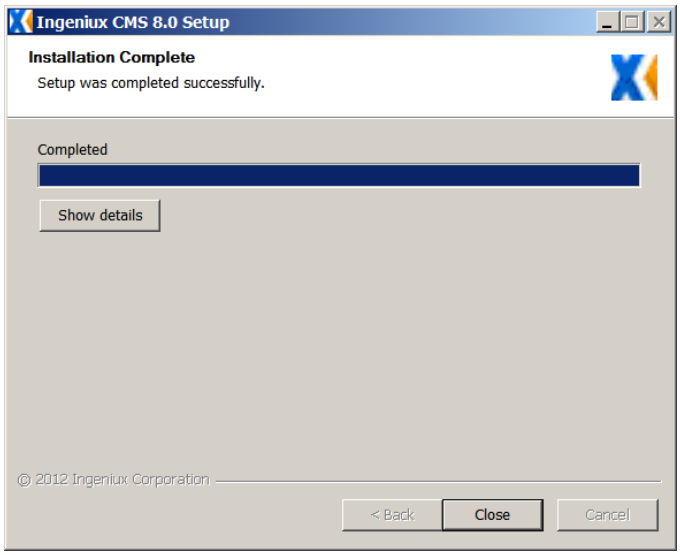
Select a destination folder or leave the default location, and then click **Install**. The space required is an estimate only. Actual installations may vary.

You may be notified that Microsoft Sync Framework will be installed. If you are, click **OK**.

You'll be prompted for permission to stop both the web service (IIS) and the content indexer. Click **Yes** at both prompts.

If the Setup wizard detects a previous installation of the CMS, it will prompt you to keep the previous settings. Click **Yes** unless you are intentionally reconfiguring the existing installation.

An installation progress dialog appears. You can click **Show Details** to see what operations are being processed by the Setup wizard.



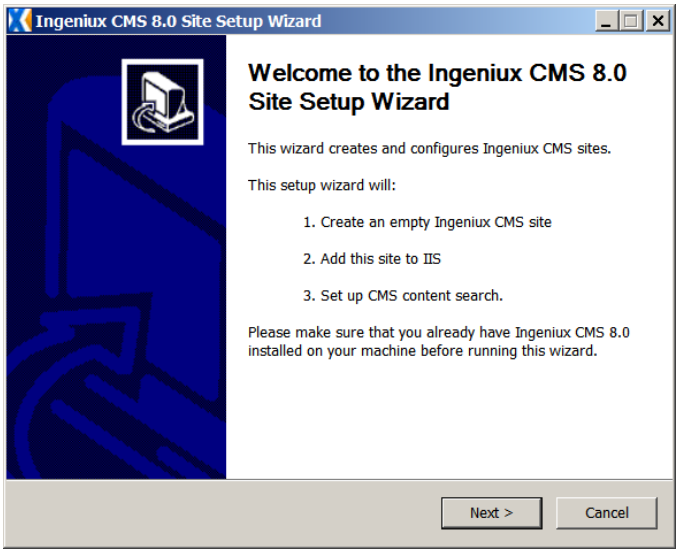
When the installation is complete, click **Close**. This completes installation of the CMS system. Next you'll need to install the CMS and DSS sites.

3.3 Installing a CMS Site

To install a CMS site, you need to run the Site Setup Wizard on the server that will host the site. The Site Setup Wizard creates an empty site, adds the site to IIS, and installs supporting files. To run the Wizard successfully, you have to be a user with administrative access to the server.

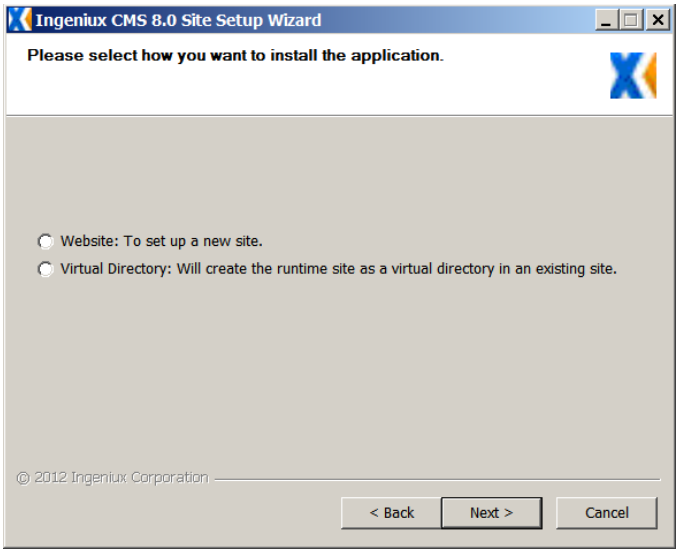
To start the site installation process, open the Tools folder of the CMS system directory (for example, C:\Ingeniux\CMS80\Tools) and double-click **IGX_CMS_Site_Setup**.

The Site Setup Wizard opens.



Click **Next**.

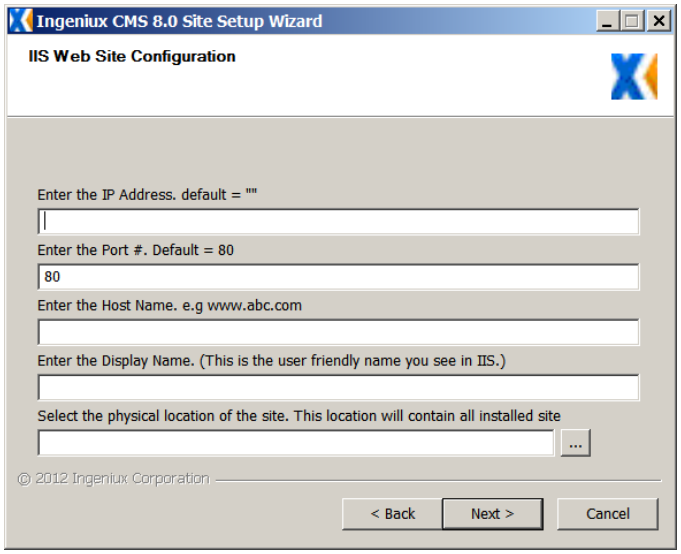
You can install the CMS site as either a website or in a virtual directory under an existing site.



Click either **Website** or **Virtual Directory**, and then click **Next**.

3.3.1 Website

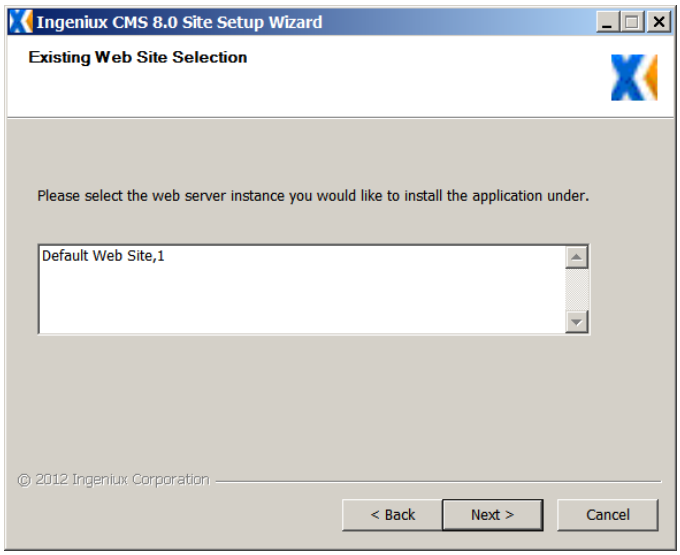
If you choose the Website option, the IIS Web Site Configuration dialog opens.



Enter website configuration values and click **Next**. The User and Group Configuration dialog opens.

3.3.2 Virtual Directory

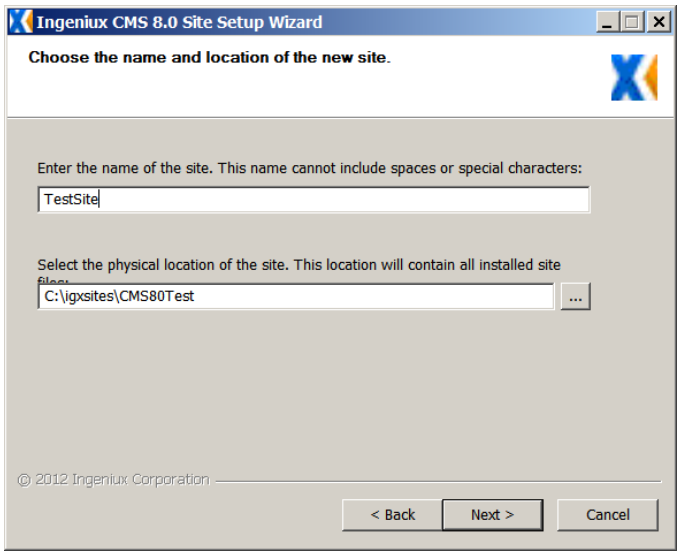
If you choose the Virtual Directory option, the Existing Web Site Selection dialog opens.



To set up a virtual directory, you have to create an IIS website prior to running the Site Setup Wizard. Then you can select the IIS website as the instance under which to run the virtual site.

Click the website under which you want to create a virtual directory, and then click **Next**.

The Site Setup Wizard prompts you for the name and location of the new virtual site.



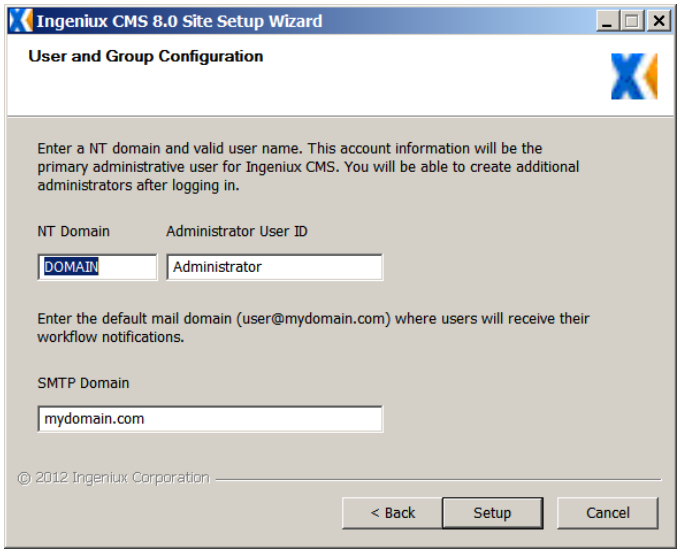
If the location of the virtual site is a remote directory, the directory should be mapped, and the mapped drive should be used for the duration of the installation. Once the Site Setup Wizard has completed, the IIS website or virtual directory can be modified to use a UNC path (for example, \\computername\sharename).

Enter the name and location of the new virtual site, and then click **Next**.

The User and Group Configuration dialog opens.

3.3.3 User and Group Configuration

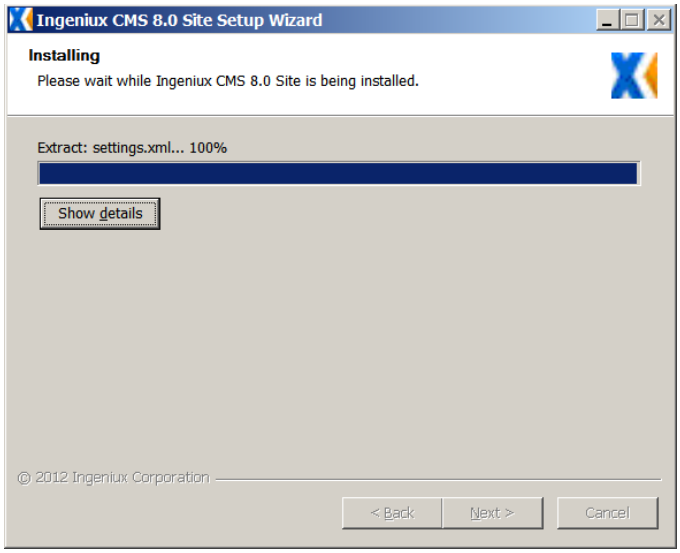
After you configure the website or virtual directory, the User and Group Configuration dialog opens.



Here you can configure the primary administrator and the default mail domain for the CMS site. The administrator ID should extend to both the CMS server and the SMTP domain.

Type the NT Domain, Administrator User ID, and SMTP Domain. If you don't have the account info, or you expect it to change, leave the default values and reconfigure them later.

Click **Setup**. The installation process begins.



IIS will restart during the installation.

3.3.4 Authentication Type

The final step in setting up a CMS site is configuring authentication.

You can use Active Directory to authenticate CMS users, or you can set up a different authentication method (for example, LDAP or a SQL database).

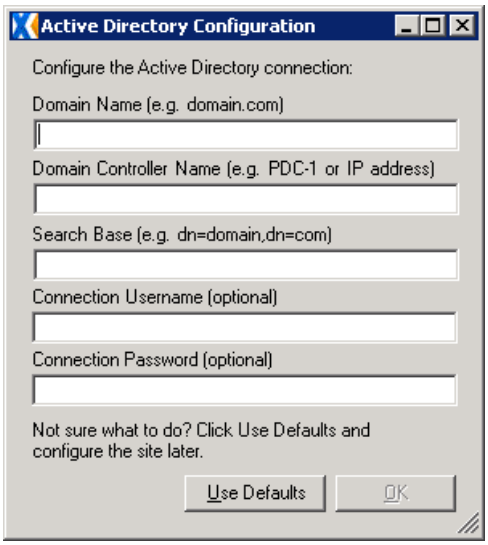
Click **Active Directory Wizard** to set up Active Directory authentication, or choose **Configure Manually** to set up an alternative method.

If you choose Configure Manually, generic versions of the following files will be copied to the site folder:

- Web.config
- local-appsettings.config
- local-connection-strings.config
- local-membership.config

These files need to be configured after the site installation is complete. For more on configuring authentication, see *Configuring Authentication with LDAP Servers*.

If you click **Active Directory Wizard**, you'll be prompted to configure an Active Directory connection.

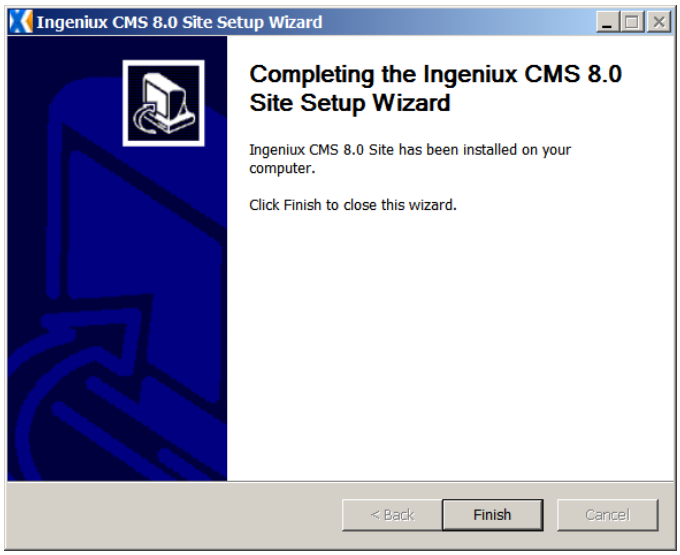


The following values can be configured:

- **Domain Name** – The domain name to be added in front of a user name. The value entered here will be included before each user name at login, following the syntax domain\username.
- **Domain Controller** – The Windows domain controller that the CMS site uses to authenticate users.
- **Search Base** – The path to the domain's Active Directory service (for example, `dc=domain,dc=com`).
- **Connection Username [optional]** – A user name for connecting to the domain controller to authenticate users.
- **Connection Password [optional]** – A password for connecting to the domain controller to authenticate users.

Enter appropriate values and click **OK**, or, if you want to configure these values later, leave the text boxes empty and click **Use Defaults**.

When you're finished configuring authentication for the CMS site, click **Finish**.



3.4 Configuring Authentication with LDAP Servers

To configure an LDAP directory to authenticate users, choose the Configure Manually option during the CMS site installation process (see *3.3.4 Authentication Type*). Then, after the Site Setup Wizard has completed, navigate to the site directory and locate the following files:

- local-connection-strings.config
- local-membership.config

Follow the steps below to configure the site for LDAP authentication:

1. Open local-connection-strings.config in an editor. You'll see several commented-out `<add>` elements containing connection string examples. Delete these elements, including the comment brackets `<!-- -->` enclosing them.
2. Between the `<connectionStrings></connectionStrings>` tags, in place of the deleted elements, add the following script, with appropriate values in brackets:

```
<add name="IGXLDAPConnectionString"
connectionString="LDAP://[ServerPathToLDAPServer]/[LDAPSearchBase]" />
```

In the example above, `ServerPathToLDAPServer` represents the path to the LDAP server, (for example, `oldap.university.edu`), and `LDAPSearchBase` represents a directory path to the object containing the users, (for example, `ou=users,dc=university,dc=edu`).

The entire element should look something like this:

```
<add name="IGXLDAPConnectionString" connectionString="LDAP://
oldap.university.edu/ou=users,dc=university,dc=edu" />
```

3. Save the file.
4. Next, open local-membership.config in an editor.
5. Locate the commented-out `<add>` element that begins as follows:

```
<add name="IGXLDAPMembershipProvider" type="IGX.LDAPMembershipProvider"
```

This is a sample LDAP membership provider. Delete this sample and the comment enclosing it.

6. If no credentials are required to bind to the LDAP directory, add the following in place of the deleted sample element:

```
<add connectionStringName="IGXLDAPConnectionString"
connectionSecurity="anonymous"
ldapFilter="(objectClass=person)"
name="MyAnonLDAPMembershipProvider"
type="IGX.LDAPMembershipProvider" />
```

If an account is required to bind to the LDAP directory, add the following in place of the deleted sample element:

```
<add connectionStringName="IGXLDAPConnectionString"
bindUsername="[bindUserAccount]"
bindPassword="[BindPassword]" ldapFilter="(objectClass=person)"
ldapUserAttribute="uid"
name="MyLDAPMembershipProvider" type="IGX.LDAPMembershipProvider" />
```

Here `[bindUserAccount]` represents the actual LDAP user account and `[BindPassword]` represents the password for the LDAP user account used to bind to the LDAP directory.

7. Save the file and recycle IIS.
8. Connect to the site and confirm that user authentication is functioning.
9. Save copies of the following files in a different directory:

- local-appsettings.config
- local-connection-strings.config
- local-membership.config
- Web.config

After a site upgrade, you can copy these files into the site directory so that you don't have to reconfigure them.

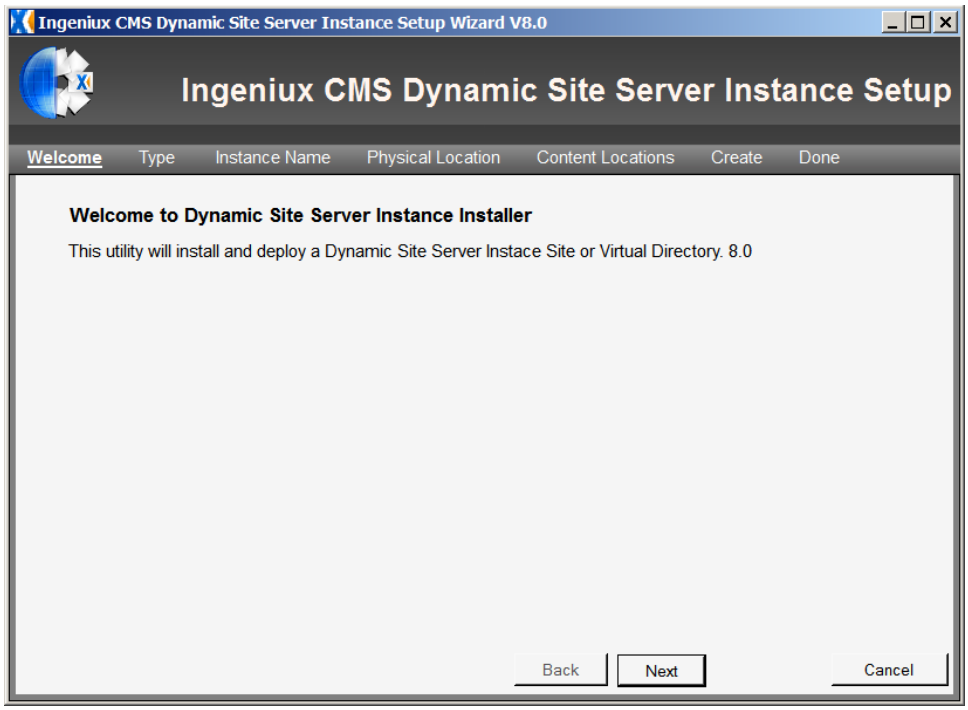
3.5 Installing a DSS Site

To install a DSS site, you will need to run the DSS Setup Wizard on the server that will host the DSS. The DSS Setup Wizard installs and deploys an instance of the DSS.

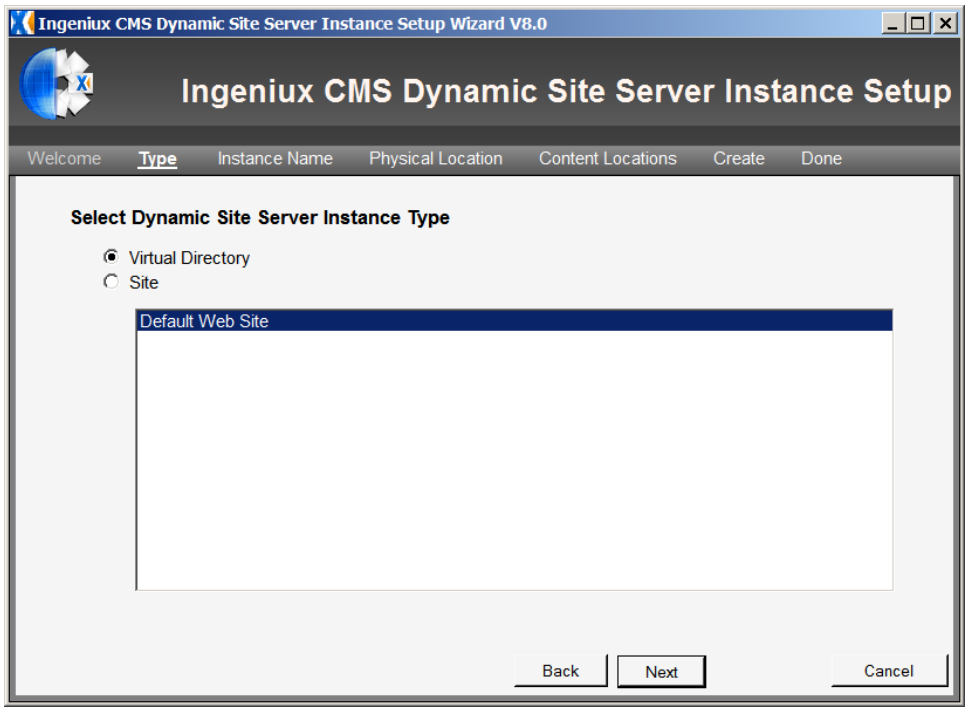
- If the CMS system is installed with a license key that doesn't license a DSS, the DSS can't be installed.

To start the DSS installation process, open the Tools folder of the CMS system directory (for example, C:\Ingeniux\CMS80\Tools) and double-click **IGX_Dynamic_Site_Server_Setup**.

The DSS Setup Wizard opens.



Click **Next**. The Setup Wizard prompts you to install either a virtual directory or a site. By default, the virtual directory option is selected.



To set up a virtual directory, leave **Virtual Directory** selected, and select the IIS website under which you want to create the virtual directory. Websites have to be created in IIS in order to appear in the list.

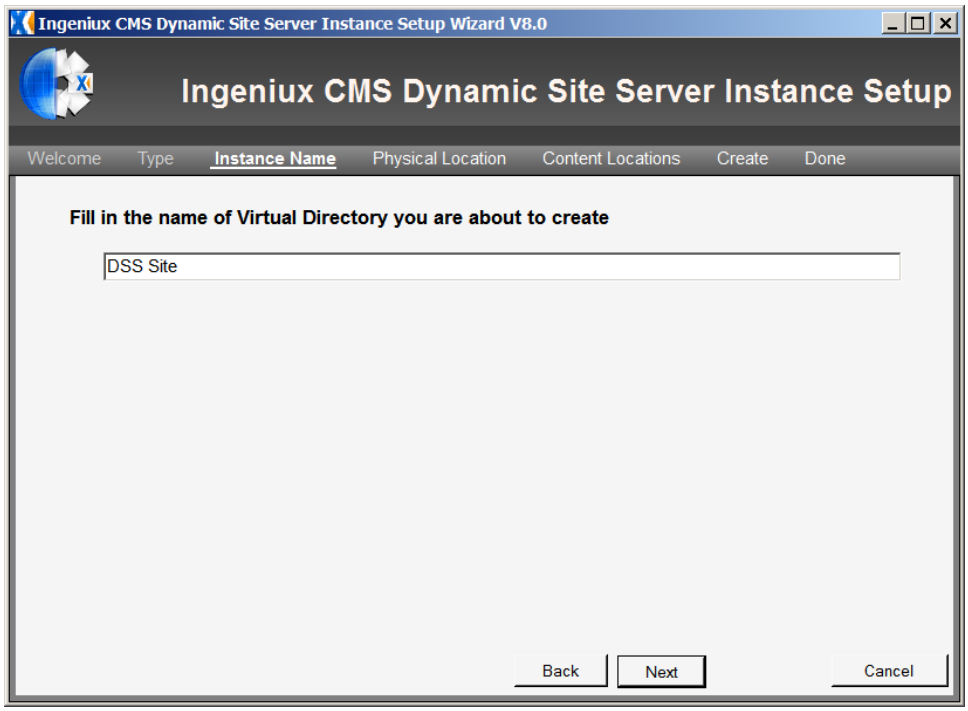
Or

To install the DSS as its own site, select **Site**.

Click **Next**.

3.5.1 Installing the DSS as a Virtual Directory

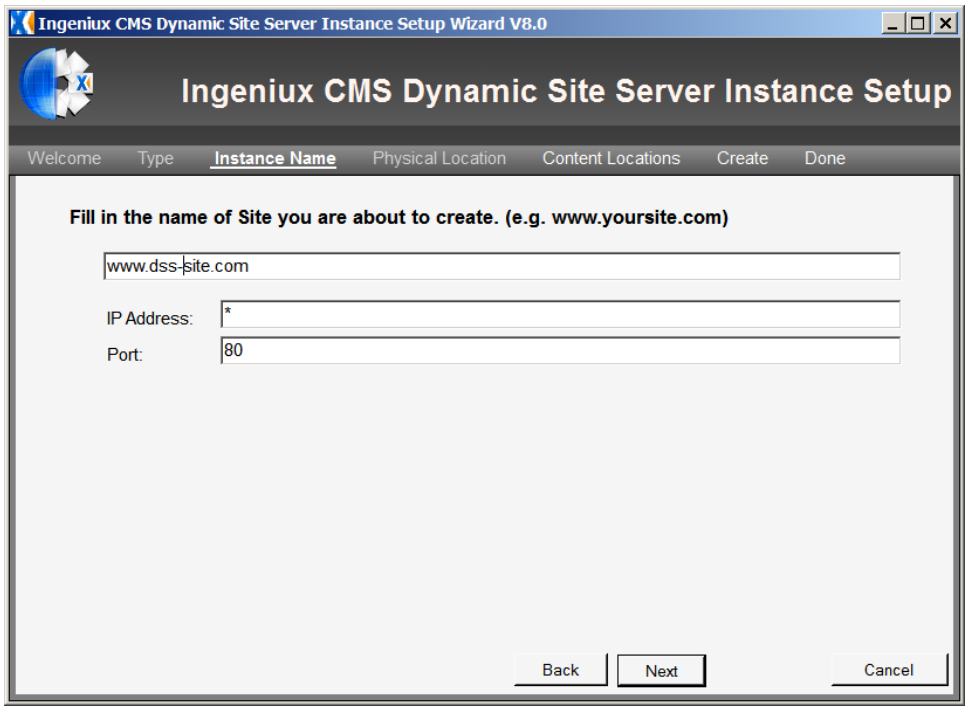
If you choose to set up the DSS site in a virtual directory, you'll be prompted to enter a name for the directory.



Type a name for the virtual directory you want to create. Then click **Next**.

3.5.2 Installing the DSS as a Site

If you choose to set up the DSS as a site, you'll be prompted to enter a host name for the site.



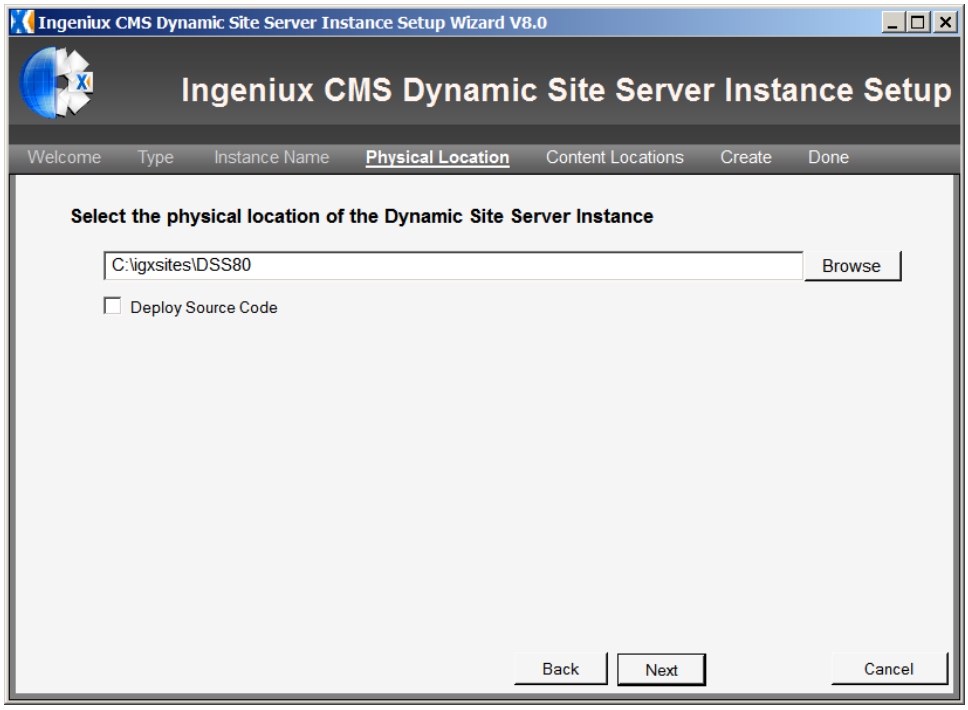
The host name represents the DNS value associated with the website. This value must be registered with the appropriate DNS server so that the host name is associated with the designated IP address. The host name should follow the syntax *www.hostname.com*.

By default, the IP address is set to a value of *, which indicates the IP address of the local server. The port is set to 80.

Type a host name for the site, make any necessary changes to the IP address and port (in most cases you won't need to change these values), and click **Next**.

3.5.3 Finishing the DSS Installation

For both a virtual directory and a site, you'll need to enter a physical location for the DSS instance.



The physical location is the path to the site on the disk (for example, C:\igxsites\DSS80). Click **Browse** and select a physical location for the site.

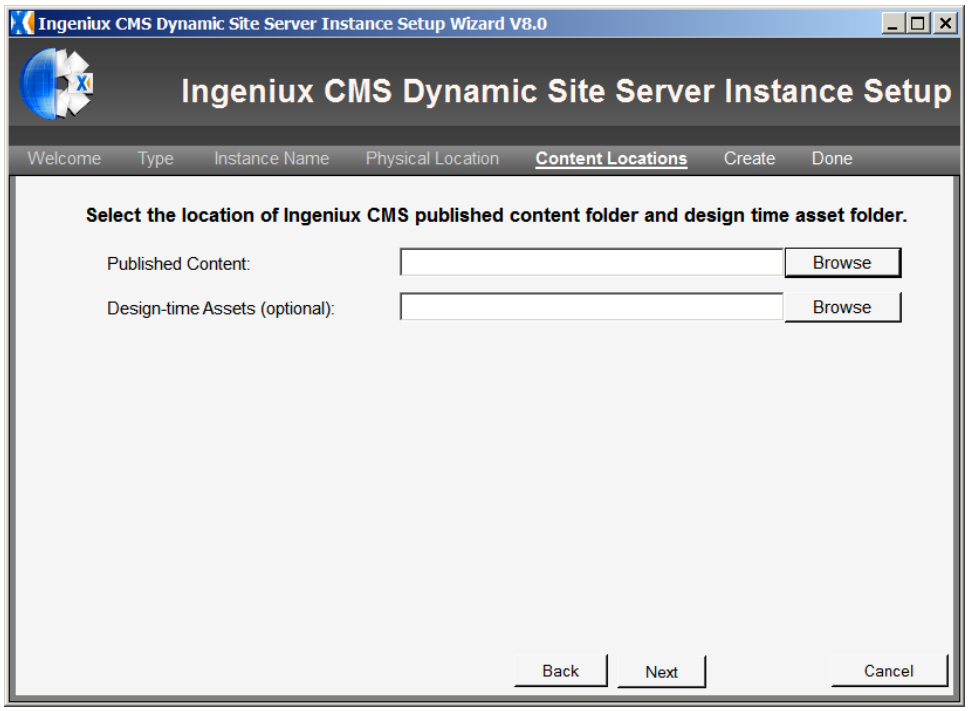
If you select **Deploy Source Code**, a sample ASP.NET MVC solution is deployed with the installation. For legacy installations using an XSLT runtime, and for installations deploying an existing MVC solution, this option should be cleared. Most installations fall into one of these two groups, so in most cases the Deploy Source Code option should not be selected.

However, if you plan to build a new MVC solution, you can deploy the source code and use that as the framework for your project.

- The DSS supports XSLT transformations out of the box. To deploy a DSS that uses XSLT stylesheets instead of the MVC framework, leave **Deploy Source Code** unselected.

When you're finished configuring the physical location and the source code option, click **Next**.

The Setup Wizard prompts you to select a location for the CMS published content folder and a location for the Design-Time assets folder.



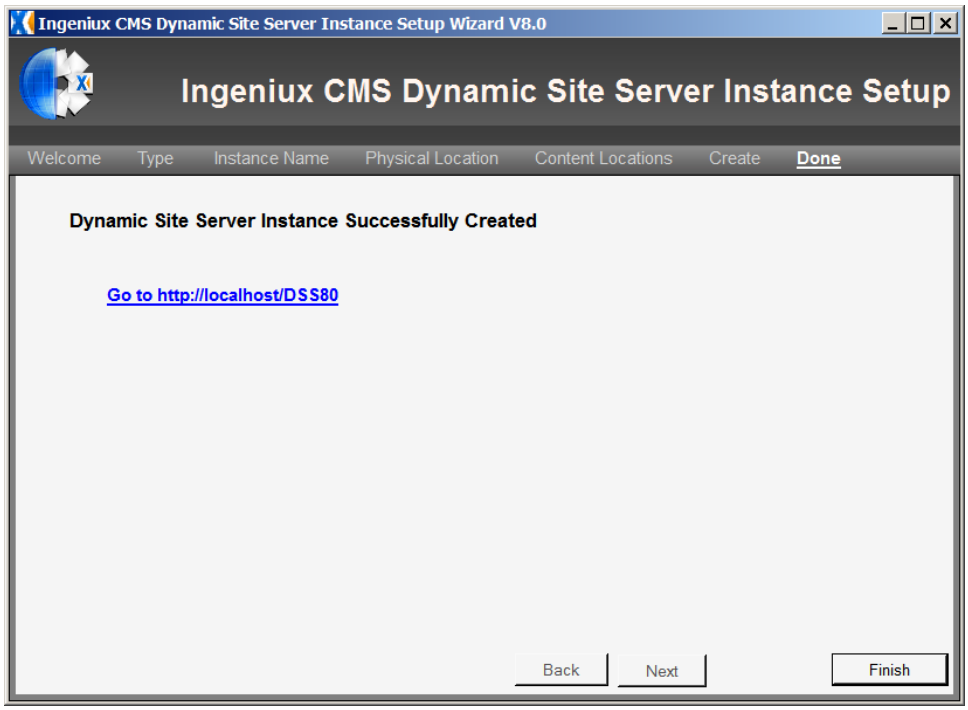
The Published Content folder is the folder to which the CMS replicates content. The DSS then displays content from this folder on the live website.

- To configure replication, log in to the CMS, open a publishing target, and configure values on the Replication tab. (For more on replication, see the *Administrator Guide*.)

The Design-Time Assets folder points to an assets folder on the DSS.

Use the **Browse** buttons to select a Published Content folder and (optionally) a Design-Time Assets folder. Then click **Next**.

The Setup Wizard creates a DSS instance.



The Setup Wizard displays a link to the new DSS site, but the site won't be populated with content until the first publish and replication from the CMS. To complete the DSS installation process, click **Finish**.

3.6 Installing the ComExecute Component

The ComExecute component extends the standard functionality of the CMS. ComExecute can be used to perform database functions, send email, or digest content from web services via SOAP.

3.6.1 Installing ComExecute

The ComExecute component is available from the Ingeniux Support site as a zip file. Typically, the component files will be installed at C:\Program Files\Ingeniux Corp\IGX ComExecute Component 8.0.The component files need to be installed on both the CMS and DSS servers to work properly.

3.6.2 Upgrading ComExecute

If you use the Site Upgrade Wizard to upgrade to CMS 8.0, you will still need to upgrade the ComExecute component manually. To do so, go to the Ingeniux Support site and download and install the ComExecute component for CMS 8.0. If the ComExecute component is not upgraded, associated functions (e.g. sending email) won't work.

3.7 Configuring HTTPS/SSL

The secure sockets layer (SSL) protocol provides encryption and authentication services for HTTP transactions. HTTP using SSL for data security is called HTTP Secure (HTTPS).

The CMS supports three HTTPS/SSL configurations:

- No SSL; HTTP for login and the site
- SSL for login only; HTTP for the site
- SSL for the entire site

These three configurations are governed by three settings:

- requireSSL (in Web.config)
- loginUrl (in Web.config)
- redirectToHttpAboutLogin (in local-appsettings.config)

To enable one of the SSL configurations, use the appropriate combination of settings:

No SSL; HTTP for login and the site:

- requireSSL = false
- loginUrl = secured/login.aspx
- redirectToHttpAboutLogin = true

SSL for login only; HTTP for the rest of the site:

- requireSSL = true
- loginUrl = secured/seclogin.aspx
- redirectToHttpAboutLogin = true

SSL for the entire site:

- requireSSL = true
- loginUrl = secured/seclogin.aspx
- redirectToHttpAboutLogin = false

To enable HTTPS/SSL for a CMS site, complete the following steps:

1. Back up the Web.config and local-appsettings.config files before editing.
2. Enable SSL on the virtual directory or website in IIS that is associated with the CMS.

Note: This portion of the configuration is not related to the Ingeniux CMS and is standard IIS/server administration. Additional information on enabling SSL in IIS can be found on the Microsoft TechNet site at http://goo.gl/PNLN.

3. Configure the Web.config <forms> node for SSL:

   a) Browse to the root directory of your site.

   b) Open the Web.config file in a text editor.

   c) Navigate to the <forms> node.

```
<authentication mode="Forms">
  <forms name="IGXAuth" path="/" loginUrl="secured/login.aspx"
      protection="All" timeout="30" slidingExpiration="true" >
```

4. Add the `requireSSL="true"` attribute to the <forms> node.

```
<authentication mode="Forms">
  <forms name="IGXAuth" path="/" loginUrl="secured/login.aspx"
      protection="All" timeout="30" slidingExpiration="true" requireSSL="true" >
```

5. Adjust the `loginUrl="secured/login.aspx"` attribute as needed:

```
<authentication mode="Forms">
  <forms name="IGXAuth" path="/" loginUrl="secured/seclogin.aspx"
      protection="All" timeout="30" slidingExpiration="true" requireSSL="true" >
```

6. Configure local-appsettings.config:

```
<appSettings >
  <add key="userdomain" value=""/>
    <!-- time out of temp images created by image manipulations -->
    <add key="tempImageTimeOut" value="30"/>
    <!-- settings for https only. If true, redirect to http, otherwise, stays https -->
    <add key="redirectToHttpAboutLogin" value="false"/>
</appSettings>
```

If the `redirectToHttpAboutLogin` setting isn't present, you may need to add it. The default value for the node is `<add key="redirectToHttpAboutLogin" value="true"/>`.

4 CMS Site Verification

A CMS site can be set up as either a website or virtual directory in IIS. If the CMS is implemented as a website, the URL is the hostname of the website. If it's implemented as a virtual directory, the URL is the hostname/virtual directory name. The simplest installation method places the CMS site as a virtual directory underneath the Default website.

IIS Configuration	URL	DNS Configuration
Default website	Hostname of server	None: The server should already have a hostname registered on the internal network.
Virtual directory under default website	[Hostname]/virtual directory name	None: The server should already have a hostname registered on the internal network.
Website	Hostname assigned to website	A DNS entry needs to be created to map the IP address of the NIC to the hostname for the website.
Virtual directory under website	[Hostname]/virtual directory name	A DNS entry needs to be created to map the IP address of the NIC to the hostname for the website.

For most CMS implementations, you can use the Setup Wizard to install and configure a CMS site. If you need to perform a manual installation, or if you need to verify the configuration of a CMS site, use the following sections for reference.

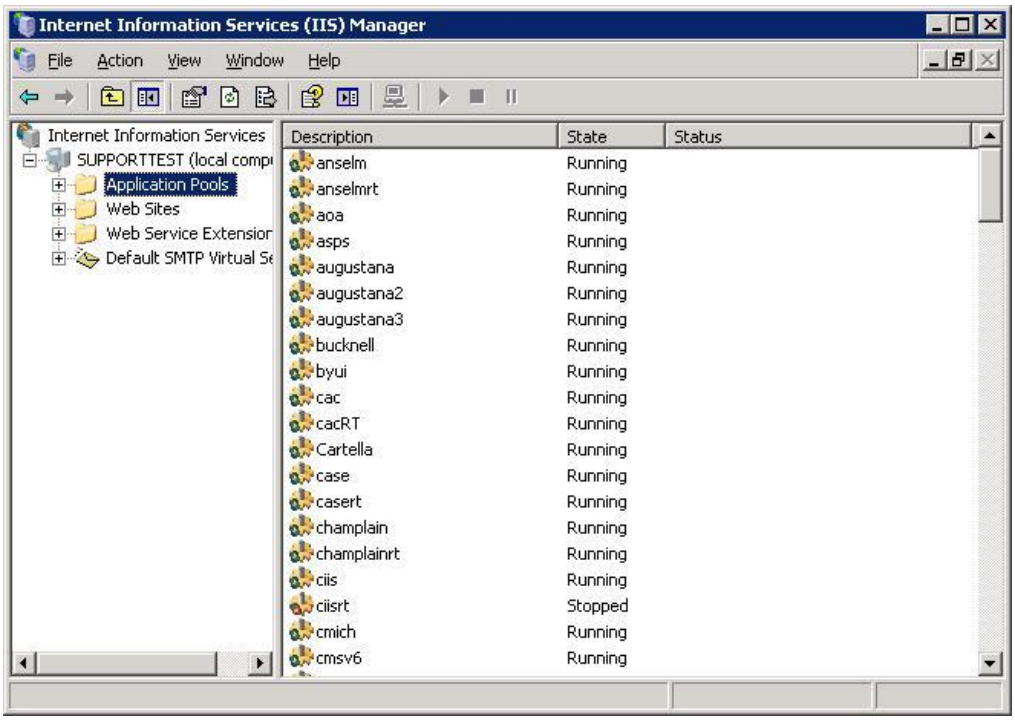
4.1 CMS Site Configuration (IIS 6.0)

This section describes CMS site configuration in a Windows Server 2003/IIS 6 environment.

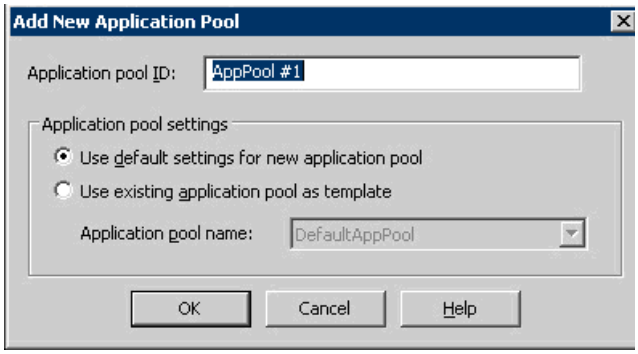
4.1.1 Creating an Application Pool

Ingeniux recommends using separate Application Pools for each site (whether an IIS website or a virtual directory) running on a CMS server. Using separate Application Pools limits the impact sites have on each other and allows easier identification of problem websites, as each Application Pool runs as a separate process.

Before you create a new Application Pool, make sure that an Application Pool hasn't already been created for your site. The Application Pool would be listed in the IIS Manager in the Application Pools folder.



If it hasn't been created, right-click the Application Pool in the IIS Manager and select **New** > **Application Pool**).



Enter an application Pool ID that corresponds to the CMS site, and select **Use default settings for new application pool**.

4.1.2 Configuring the Application Pool

To maximize CMS site availability while maintaining site performance, Ingeniux recommends recycling an Application Pool daily.

- **Note**: The Application Pool should not be configured to shut down idle processes, because some CMS processes (for example, large check-ins and publishes) may appear idle to the Application Pool. Enabling the shutdown of idle processes could prematurely stop an important CMS task.

Also, with regard to recycling the Application Pool, keep the following best practices in mind:

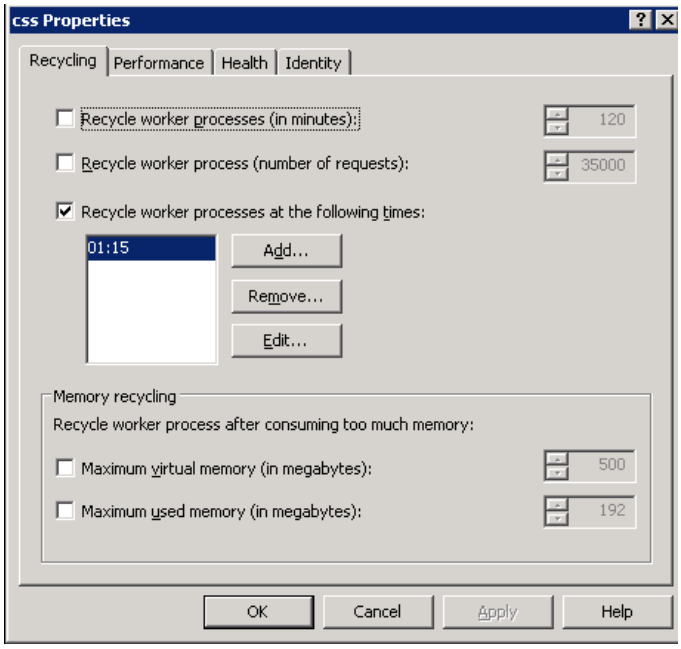
- The daily Application Pool recycle should occur during a time of little or no activity, because recycling an Application Pool during certain CMS processes could cause site corruption.
- Recycling the Application Pool of an IIS website containing a virtual directory hosting a CMS site could cause corruption. Ingeniux recommends preventing an Application Pool recycle in this context.

Ingeniux recommends the following standard Application Pool settings for a CMS site. These settings may need to be adjusted to meet the demands of a specific environment, especially with regard to server hardware, server performance, and site usage volume. Please use these settings as a starting point and modify them as needed.

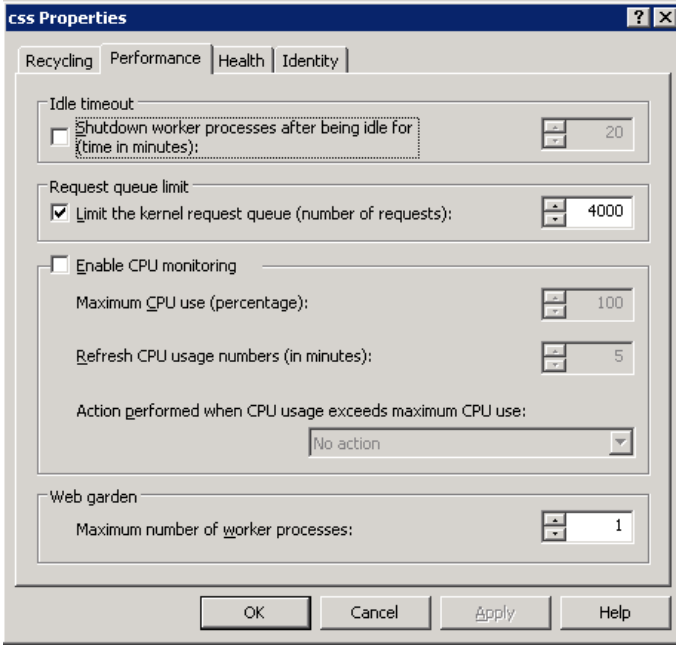
To configure an Application Pool, right-click it under the IIS Application Pools folder and select **Properties**.

Use the following tabs/settings to configure the Application Pool:

Recycle – Set up a nightly recycle of the Application Pool.

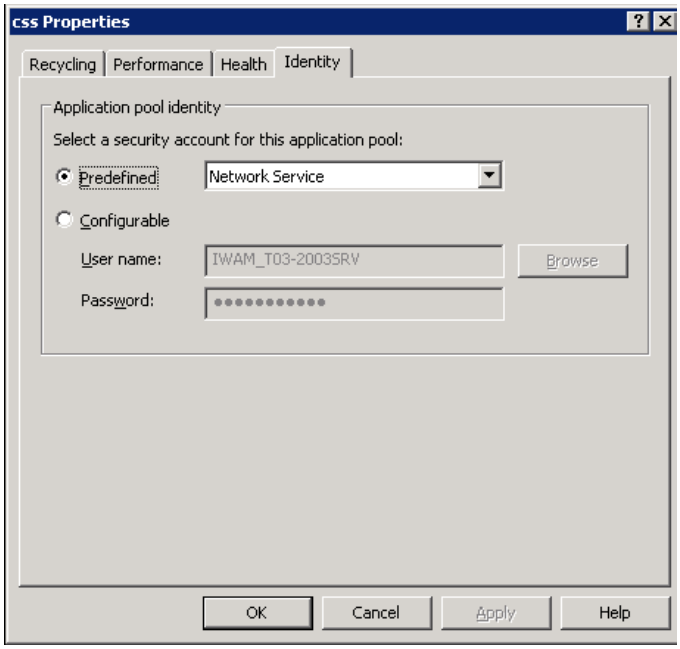


Performance – Clear **Shutdown idle worker processes**… .



Health – Leave the settings on the Health tab at their defaults.

Identity – Confirm that the Identity tab is configured to use the Network Service account.

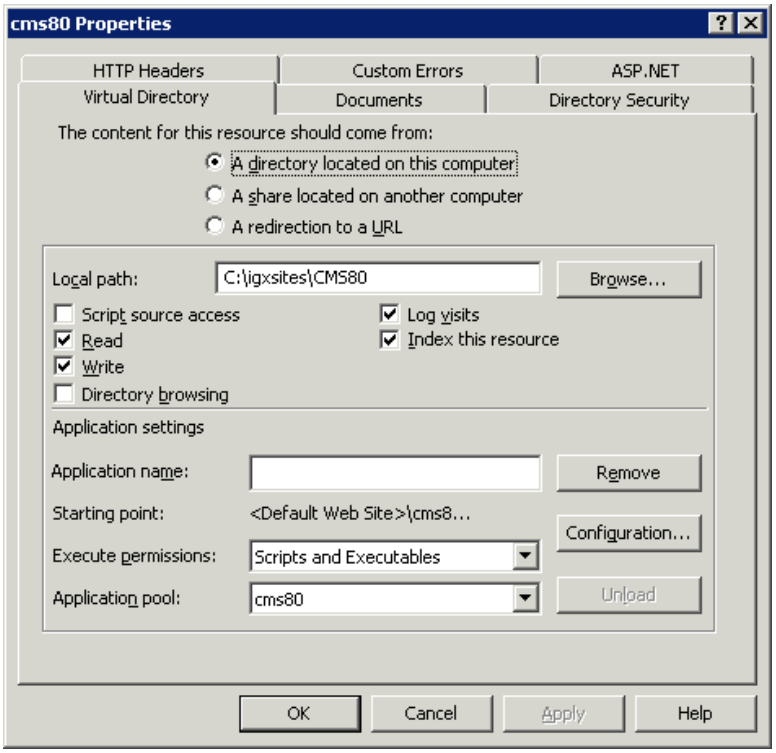


4.1.3 Configuring an IIS Website or Virtual Directory

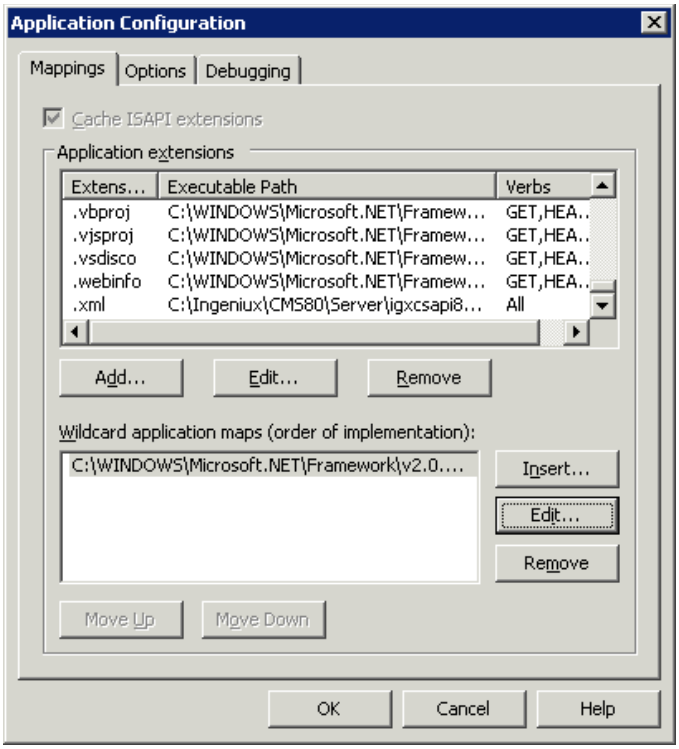
After configuring the Application Pool for the CMS site, you'll need to configure the site itself in IIS. Locate the CMS site in the Web Sites section of the IIS Manager directory and right-click the site. Select **Properties** and configure the following tabs and settings.

Home/Virtual Directory Tab – This tab contains basic site information and will be labeled either "Virtual Directory" or "Home Directory" depending on your site configuration. For a CMS site, configure the following values and settings:

- **Local Path** – Path to the CMS site files
- **Permissions** – Check boxes selected for **Read**, **Write**, **Log Visits**, and **Index this resource**
- **Application pool** – Application Pool for the site

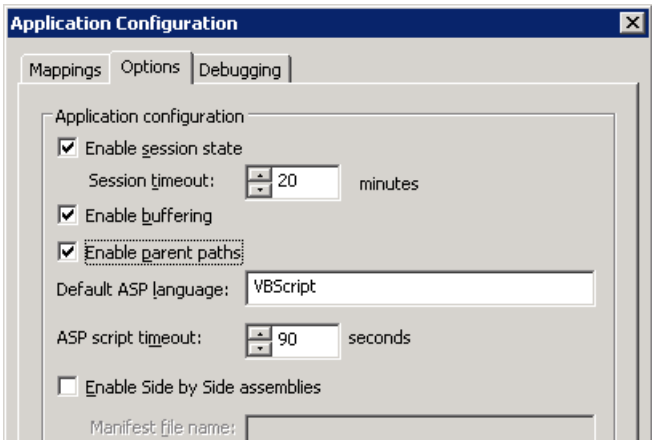


- **Configuration** – Options for extension and wildcard mappings. To configure mappings, click **Configuration**. The **Application Configuration** dialog opens.

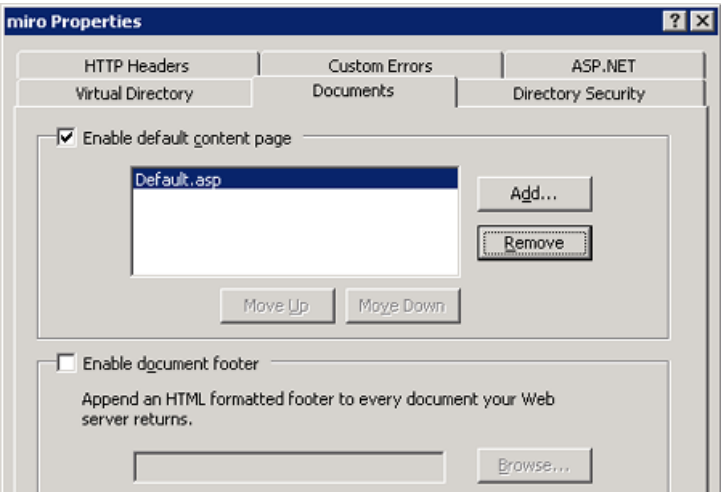


The .xml extensions for a CMS site should be mapped to igxcsapi80.dll. (Note that the DSS mapping differs here.) Also, aspnet_isapi.dll has to be implemented as a wildcard application map.

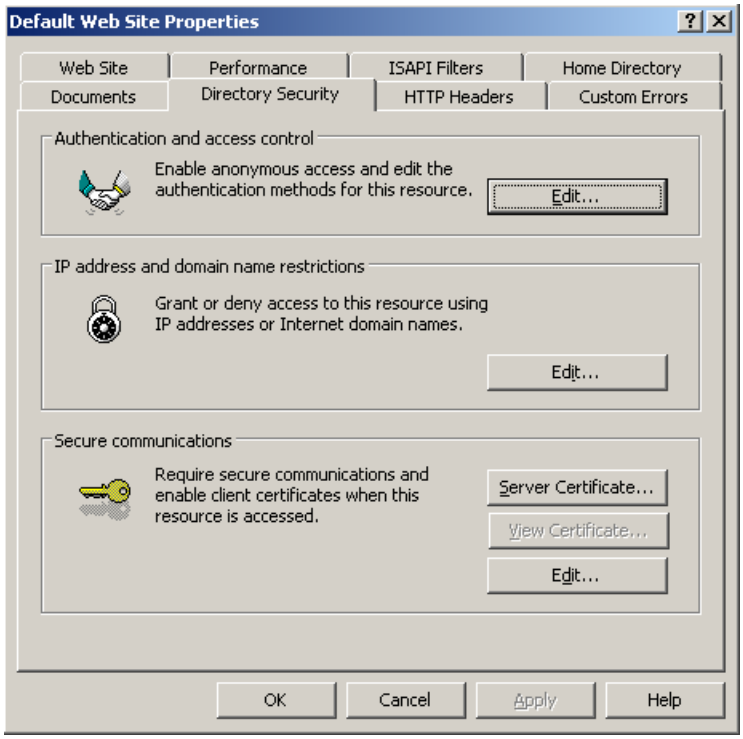
On the **Options** tab, **Enable Parent Paths** should be selected.



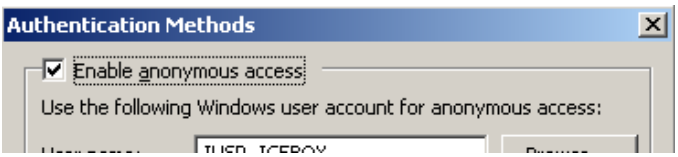
Documents – This tab defines the default document IIS loads when the site is requested. This should be set to Default.asp for CMS servers.



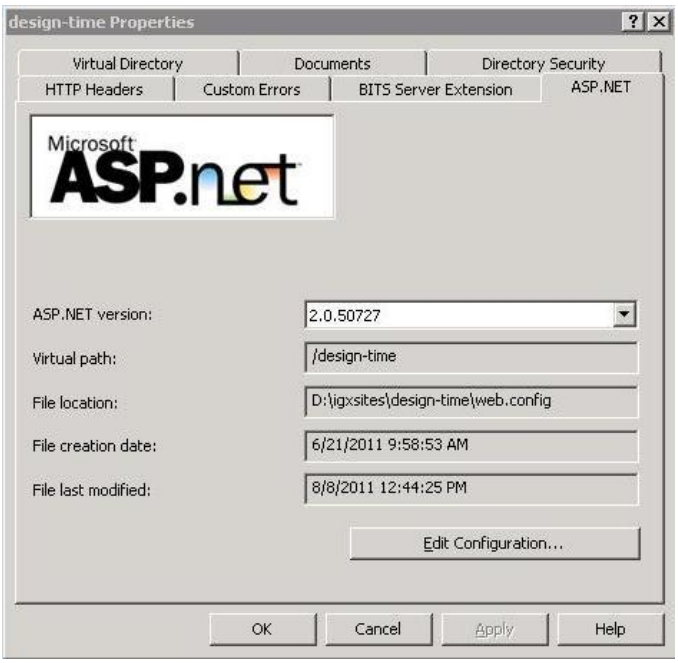
Directory Security – This tab defines access to the site. Authentication and access control should be set to **Enable anonymous access**. This setting enables the CMS, in conjunction with ASP.NET, to provide authentication in place of local Windows security.



To enable anonymous access, on the Directory Security tab in the authentication and access control section, click **Edit** and select **Enable anonymous access** in the dialog that appears:



ASP.NET – Specifies the version of ASP.NET to use for the site. If only one version of ASP.NET is present, the ASP.NET tab is not available.



Note: If two versions of ASP.NET are installed but the ASP.NET tab is not present, this may indicate an ASP.NET registration problem. The registration problem could prevent the CMS from functioning properly. In such a case, consult the Microsoft Support site for possible resolutions.

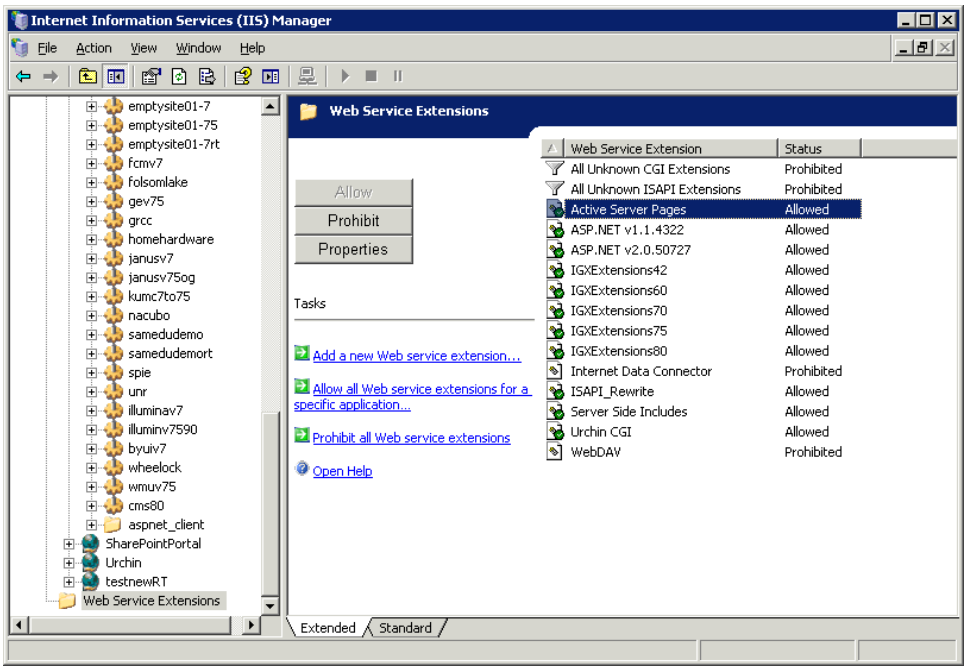
4.1.4 Configuring Web Service Extensions

For a CMS site to run, the following Web Service Extensions have to be allowed:

- Active Server Pages
- ASP.NET
- IGXExtensions80
- Indexing Service
- Server Side Includes

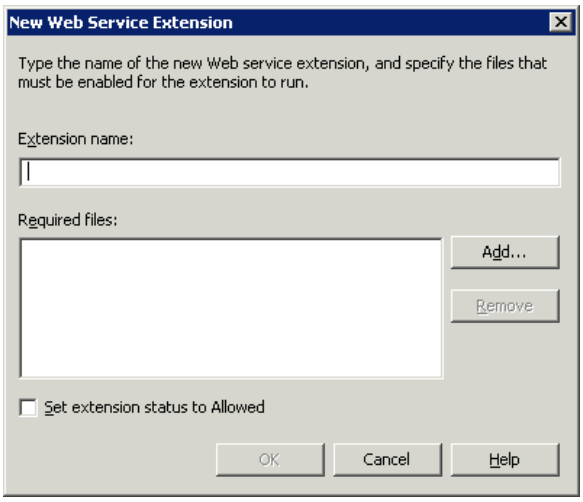
These services should be present and enabled by default. If not, follow these steps to enable them:

1. In **IIS Manager**, find and open the **Web Service Extensions** folder.
2. In the right pane, select **Active Server Pages** and click **Allow**.
3. Repeat for **Indexing Service**, **Server Side Includes**, and **ASP.NET** if installed.



The Ingeniux DLLs for a CMS site also need to be added and allowed. To add the Ingeniux DLLs, follow these steps:

1. Click **Add a New Web Service Extension**. The New Web Service Extension dialog opens.



2. Enter the following under Extension Name*:*

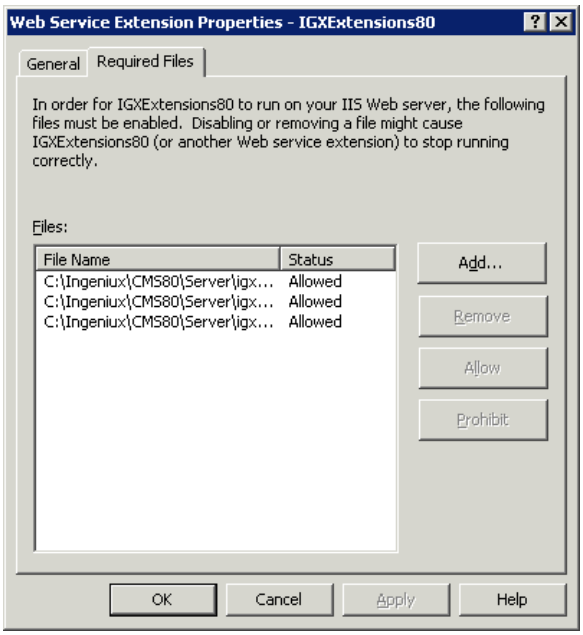
   `IGXExtensions80`

3. Select **Set extension status to Allowed**.

4. Click **Add** > **Browse** and select **igxcsapi80.dll**.
5. In the Web Service Extensions lists in the right pane of IIS Manager, double-click **IGXExtensions80** and click **Required Files**. The list of required files should ultimately include igxcsapi80.dll and igxxmlsvr80.dll.

To add a DLL:

1. In Web Service Extension Properties on the Required Files tab, click **Add**.



2. Click **Browse** and select the DLL you want to add. Typically, these files are located in the Ingeniux\CMS75\Server directory.
3. Repeat the process for all the DLLs, and make sure that all of the appropriate DLLs and web service extensions are set to **Allow.**

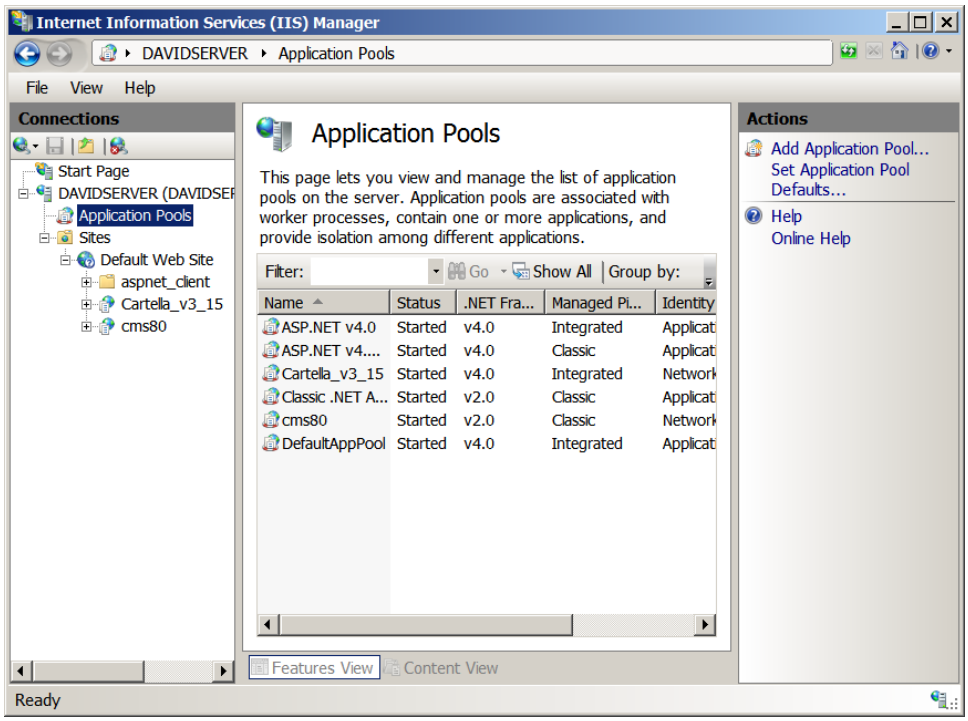
4.2 CMS Site Configuration (IIS 7.0)

This section describes CMS site configuration in a Windows Server 2008/IIS 7 environment.

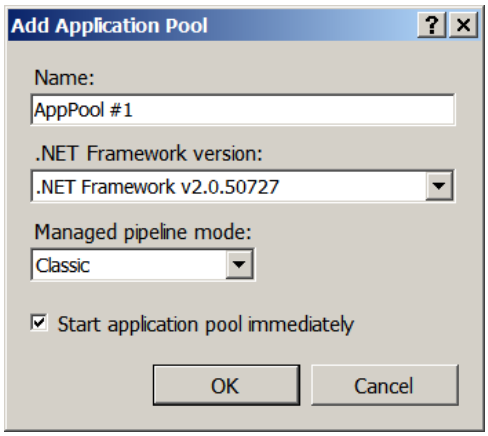
4.2.1 Creating an Application Pool

Ingeniux recommends using separate Application Pools for each site (whether an IIS website or a virtual directory) running on a CMS server. Using separate Application Pools limits the impact sites have on each other and allows easier identification of problem websites, as each Application Pool runs as a separate process.

Before you create a new Application Pool, make sure that an Application Pool hasn't already been created for your site. The Application Pool would be listed in the IIS Manager in the Application Pools page.



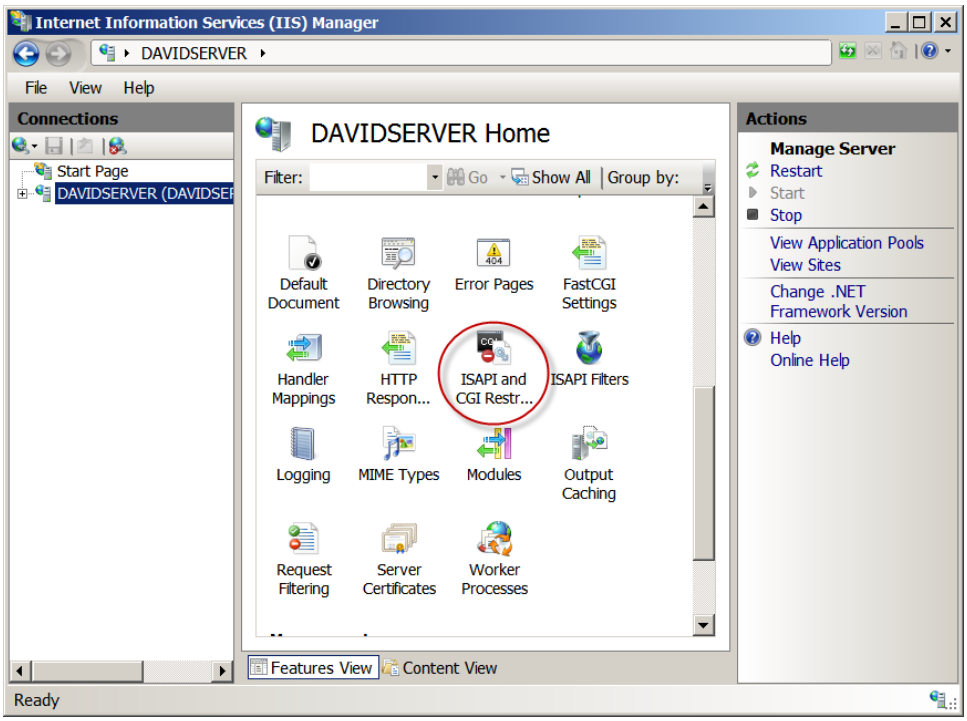
If it hasn't been created, right-click the Application Pools node in IIS Manager and select **Add Application Pool**.



Enter an application Pool ID that corresponds to the CMS site, and, if necessary, select **Classic** as the managed pipeline mode. Leave the .NET Framework version set to 2.0.

4.2.2 Configuring Web Service Extensions

To configure Web Service Extensions in IIS 7, select the server node in the IIS Manager directory tree. Then, in the main pane, select **ISAPI and CGI Restrictions**.



A list of DLLs is displayed. The list should include igxcsapi80.dll and igxxmlsvr80.dll. Verify that the IGX extensions are installed and set to **Allowed**.

To add a DLL (if necessary):

1. Click **Add** in the **Actions** pane.

2. Click the **Browse** (…) button.
3. Navigate to, and select, the desired file. Typically, these files are located in the C:\Ingeniux\CMS80\Server directory.
4. Select **Allow extension path to execute**, and then click **Okay.**
5. Repeat the process as needed for DLLs.
6. Once the igxcsapi80.dll and igxxmlsvr80.dll have been added, confirm that their restriction level is set to **Allowed**. To allow a file, select it, click **Edit**, and select **Allow extension path to execute**.

4.2.3 Confirming Server Role Service

Under Windows Server 2008 and IIS 7, you may have to verify that the proper Role Services are installed. This verification step is new with IIS 7.

In Server Manager, navigate to the Role Service list and verify that the list matches the following screen shot:

Role Service	Status
Web Server	Installed
Common HTTP Features	Installed
Static Content	Installed
Default Document	Installed
Directory Browsing	Installed
HTTP Errors	Installed
HTTP Redirection	Installed
Application Development	Installed
ASP.NET	Installed
.NET Extensibility	Installed
ASP	Installed
CGI	Installed
ISAPI Extensions	Installed
ISAPI Filters	Installed
Server Side Includes	Installed
Health and Diagnostics	Installed
HTTP Logging	Installed
Logging Tools	Installed
Request Monitor	Installed
Tracing	Installed
Custom Logging	Installed
ODBC Logging	Installed
Security	Installed
Basic Authentication	Installed
Windows Authentication	Installed
Digest Authentication	Installed
Client Certificate Mapping Authentication	Installed
IIS Client Certificate Mapping Authentication	Installed
URL Authorization	Installed
Request Filtering	Installed
IP and Domain Restrictions	Installed
Performance	Installed
Static Content Compression	Installed
Dynamic Content Compression	Installed
Management Tools	Installed
IIS Management Console	Installed
IIS Management Scripts and Tools	Installed
Management Service	Installed
IIS 6 Management Compatibility	Installed
IIS 6 Metabase Compatibility	Installed
IIS 6 WMI Compatibility	Installed
IIS 6 Scripting Tools	Installed
IIS 6 Management Console	Installed
FTP Publishing Service	Not installec
FTP Server	Not installec
FTP Management Console	Not installec

4.3 File Level Permissions

For the CMS to access the physical resources on the server, certain file level permissions have to be enabled for the \ingeniux and \igxsites directories. These permissions don't grant network access to these resources, but they enable the CMS to access these directories on behalf of users. The permissions are as follows:

\ingeniux			
Account/Group	Access Level	CMS	DSS
IIS Application Pool Account[1]	Full Access	Required	Required
IUSR_[Computername][2]	Full Access	Required	Required

\temp			
Account/Group	Access Level	CMS	DSS
IIS Application Pool Account[1]	Full Access	Required	Required
IUSR_[Computername][2]	Full Access	Required	Required

\igxsites[3]			
Account/Group	Access Level	CMS	DSS
IIS Application Pool Account[1]	Full Access	Required	Required
IUSR_[Computername][2]	Full Access	Required	Depends on Site[4]

\%windir%\Microsoft.NET\Framework\v2.0.50727			
Account/Group	Access Level	CMS	DSS

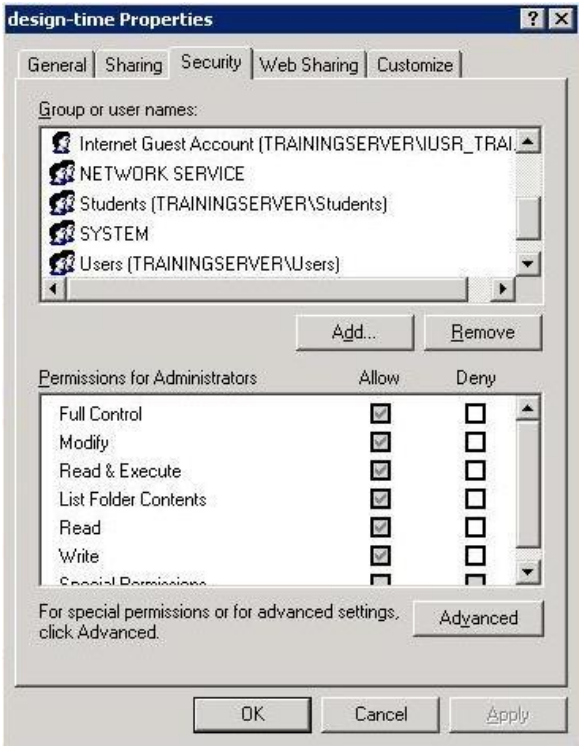
IIS Application Pool Account[1]	List Folder Contents, Read & Execute, Read, and Write	Required	Not Required
IUSR_[Computername][2]	List Folder Contents, Read & Execute, Read, and Write	Required	Not Required

Notes:

1. This account is configured in the **Properties** > **Identity** tab of the Application Pool used by the site. It defaults to [localmachine]\network service.
2. This account is used by the IIS Website to support anonymous access to site resources after a user has been validated via ASP.NET authentication.
3. If the log files, XML directory, and Index catalogs are in different directories, each account/group, with the appropriate access, will need to be added to each directory.
4. Permissions on the DSS server may vary depending on whether resources are password protected.

To add directory permissions for a user or user group, follow these steps:

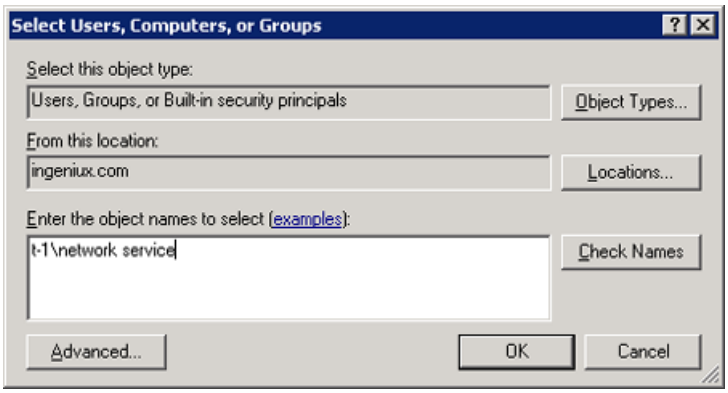
1. Navigate to the parent directory or drive.
2. Right-click the directory and select **Properties**.
3. On the **Security** tab, click **Add**.



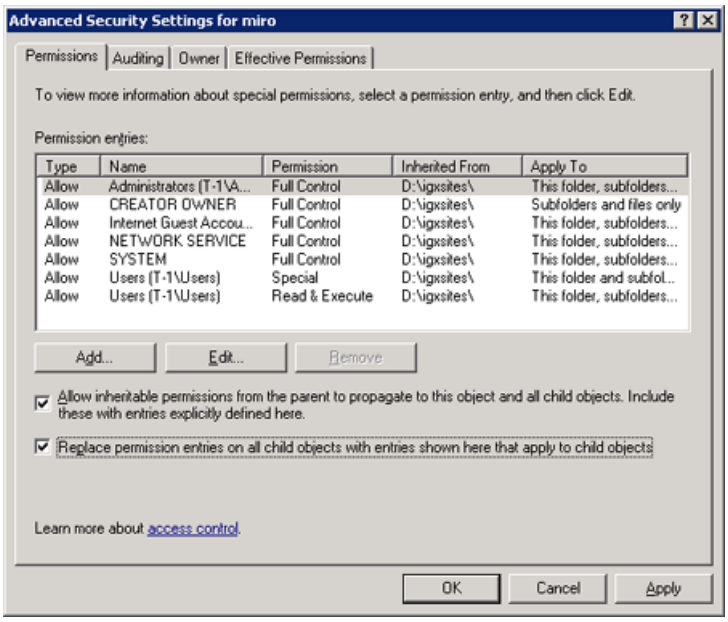
4. Enter the group or account name in the **Enter the object names to select** field. Use the following syntax:

```
domain\[user or group name]
```

Click **OK**.



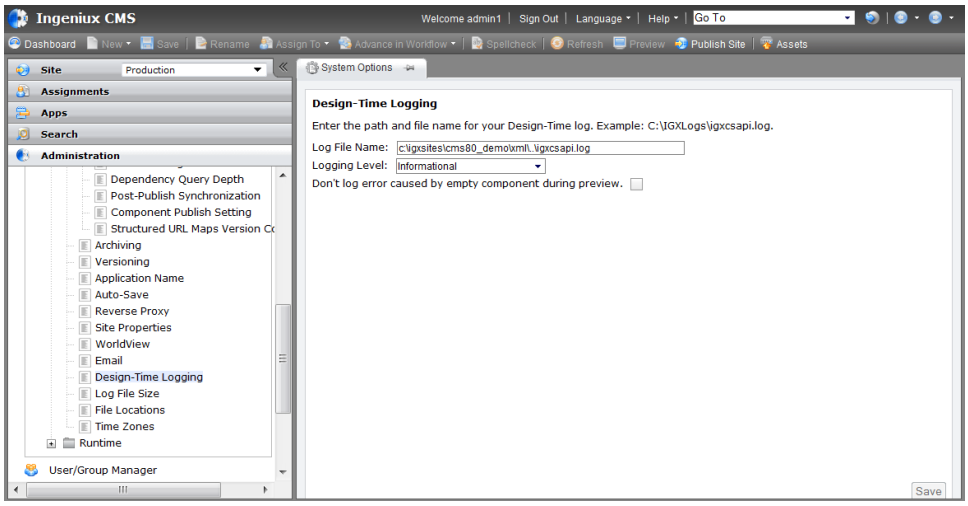
5. On the **Permissions** tab, select the access level (e.g. Full Access).
6. Repeat steps 1 through 6 to add additional accounts and/or user groups.
7. Next, click **Advanced** and select the **Replace permissions** check box.
8. Click **OK** and then **Yes** to the resulting dialog.



4.4 Log File Configuration

Ingeniux Log Files – To configure the location of the Ingeniux log files, follow these steps:

1. In the CMS Client, go to **Administration** > **System Options** > **CMS** > **Design-Time Logging**.

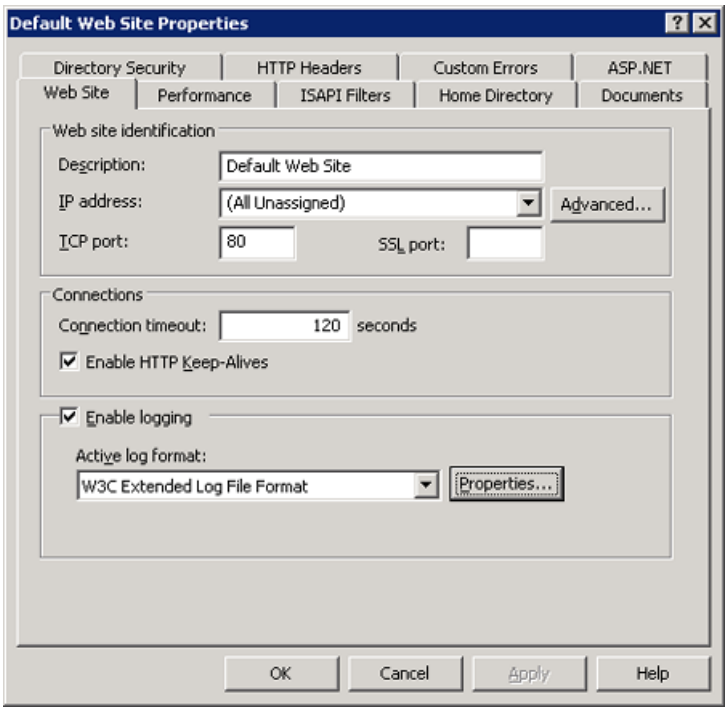


2. Enter the directory path and file name for the log. Then configure the logging level (for more on logging, see the *Administrator Guide*).
3. Click **Save** to confirm the settings.
4. Reset the Application Pool for the site.
5. To configure DSS Logging, go to **Administration** > **System Options** > **Dynamic Site Server** > **Runtime Logging** and repeat steps two through four with appropriate DSS settings.

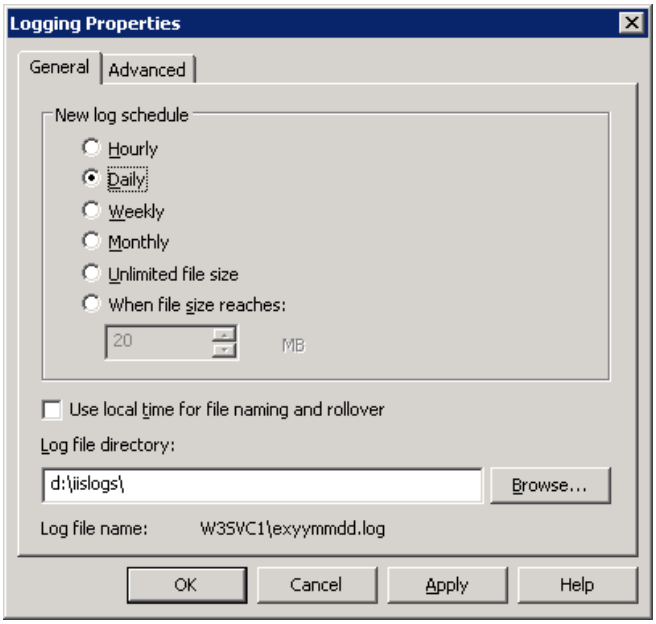
IIS Log Files – To set the log file location, follow these steps:

1. In IIS Manager under the **Web Sites** folder, right-click the CMS site (or the site above the CMS virtual directory) and click **Properties**.

2. On the **Web Site** tab, click **Properties**.



3. On the General tab, click **Browse** and select a log file directory.



4. Reset IIS for the changes to take effect.

4.5 Site Registry Entry Verification

To verify or configure the Ingeniux CMS site registry settings, follow these steps:

1. Go to **Start** > **Run**, and run the following command:

   ```
   regedit
   ```

2. In the Registry Editor, navigate to the registry key for your site. The path should look something like this:

   ```
   HKEY_LOCAL_MACHINE\SOFTWARE\Wow6432Node\Ingeniux\CMS\8.0\Sites\
   ```

   In this key, there should be a subkey corresponding to the CMS site name. Select this site subkey. If the site subkey is not present, skip to step 4.

3. The following value pairs should be listed:

   ```
   catalog=[catalogName]
   contentType=1
   disableSearching=0
   hostname=[HostNametoWebsite]
   ipaddress=[IPofServer]
   remotePassword=
   remoteTimeout=30
   remoteUserName=
   sitename=[SiteNameValue]
   sitepath=[SiteDirectoryPath]
   ```

Verify the following values:

Hostname – The domain name of the CMS site (for example, if the site URL is *http://www.designsite.gov/publisher*, the hostname is *www.designsite.gov*).

Sitename – The name of the virtual directory created for the CMS site. In the example above, the sitename is *publisher*. If the CMS is set up as a site (not as a virtual directory), leave `[SiteNameValue]` empty. Once you've verified registry values, skip to step 5.

4. To add registry values, copy and paste the text below to Notepad:

   ```
   Windows Registry Editor Version 5.00
   [HKEY_LOCAL_MACHINE\SOFTWARE\Wow6432Node\Ingeniux\CMS\8.0\Sites\[SiteNa
   me]] @=""
   "catalog"="[CatalogName]"
   "contentType"="1"
   "disableSearching"="0"
   "hostname"="[HostName]"
   "ipaddress"="[IPAddressOfServer]"
   "remotePassword"=""
   "remoteTimeout"="30"
   "remoteUserName"=""
   "sitename"="[SiteName]"
   ```

```
"sitepath"="[PathToXMLFolder]"
```

Replace the values between brackets [] with the appropriate information. You may also need to adjust the registry key path. For `sitepath`, use the following syntax:

```
[driveletter]\\[SiteDirectory]\\XML
```

For example - `D:\\igxsites\\cms80\\xml`

5. Save the file as Site.reg. Then right-click the Site.reg file and select **Import** (or **Merge**). Click **Ok**.
6. If any values were changed or added, verify that all users are logged off and no publishes or check-ins are in progress. Then restart IIS by going to **Start** > **Run** and running the following command:

   ```
   IISreset
   ```

4.6 Cleaning Up Publish Logs

To avoid clutter, Ingeniux recommends deleting old publish logs on a monthly basis.

To delete publish logs:

1. In the site directory, open the *pub* folder. The file path to the pub folder looks something like this: [Drive]\[SiteDirectory]\xml\pub.

2. Delete the publishLog files. The file names will look something like this: publishLog1-2011-06-14-16-04.xml.

5 DSS Site Verification

The DSS can be set up as either a website or a virtual directory in IIS. If the DSS is a website, the URL is the hostname of the site. If the DSS is a virtual directory, the URL is the hostname/virtual directory name.

IIS Configuration	URL	DNS Configuration
Default website	Hostname of server	None: The server should already have a hostname registered on the internal network.
Virtual directory under default website	[Hostname]/virtual directory name	None: The server should already have a hostname registered on the internal network.
Website	Hostname assigned to website	A DNS entry needs to be created to map the IP address of the NIC to the hostname for the website.
Virtual directory under website	[Hostname]/virtual directory name	A DNS entry needs to be created to map the IP address of the NIC to the hostname for the website.

For most implementations, you can use the Setup Wizard to install and configure a DSS site. If you need to perform a manual installation, or if you need to verify the configuration of a CMS site, use the following sections for reference.

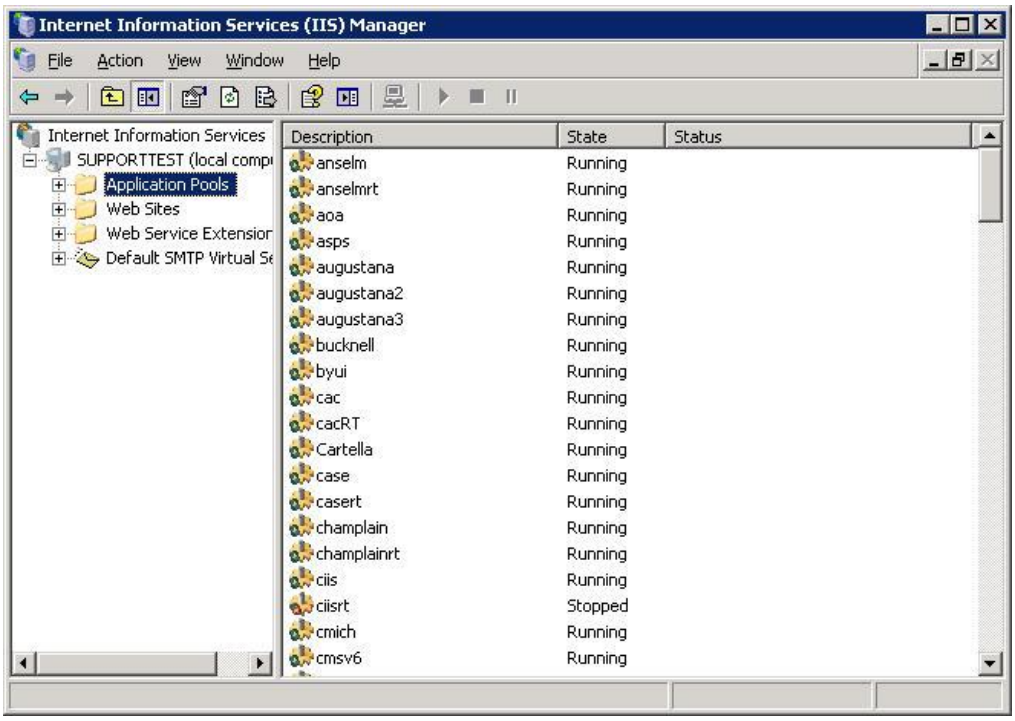
5.1 DSS Site Configuration (IIS 6.0)

This section describes DSS site configuration in a Windows Server 2003/IIS 6 environment.

5.1.1 Creating an Application Pool

Ingeniux recommends using separate Application Pools for each site (whether an IIS website or a virtual directory) running on a DSS. Using separate Application Pools limits the impact each site has on the other(s) and provides for easier identification of problem websites, because each Application Pool runs as a separate process.

To create a new Application Pool, right-click the **Application Pools** folder in IIS Manager and select **New** > **Application Pool**.



The Add New Application Pool dialog opens.

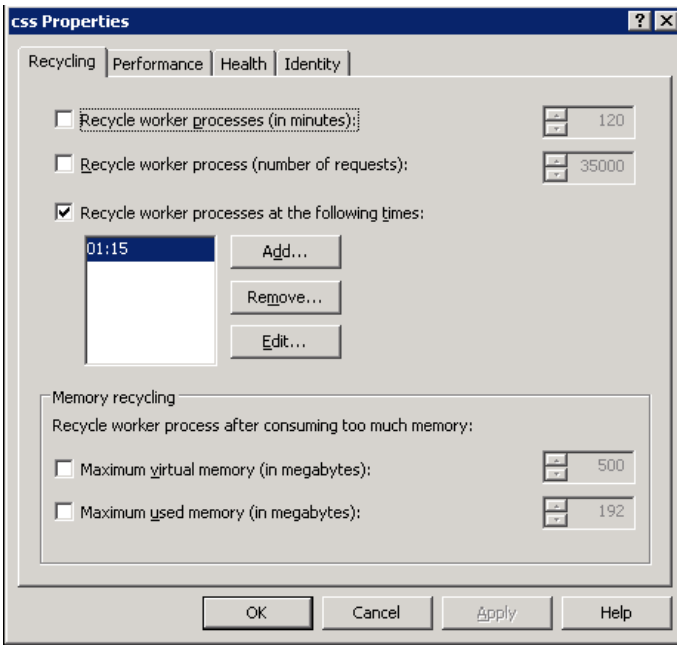
Enter an Application Pool ID for the DSS site.

5.1.2 Configuring the Application Pool

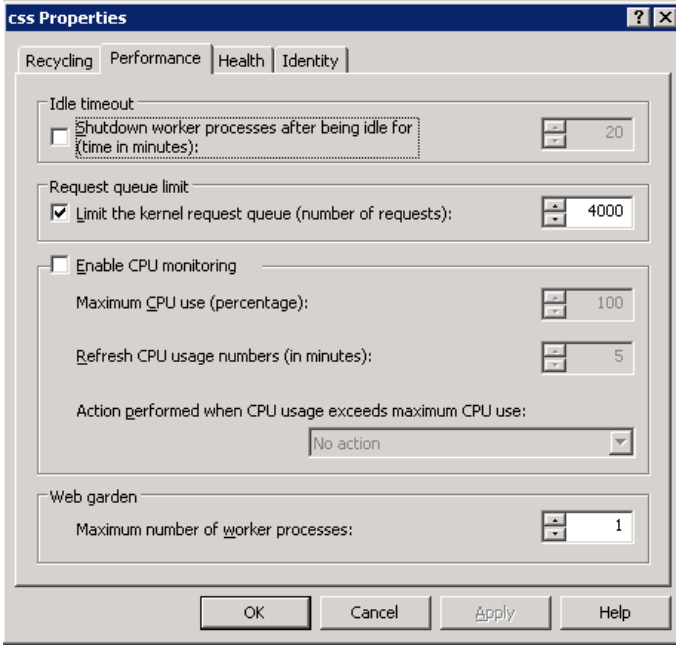
Ingeniux recommends certain Application Pool settings for DSS sites. These settings may need to be modified later if server demands or site usage change. Please use these settings as a starting point only.

To configure the Application Pool, right-click it under the IIS Application Pools folder and select **Properties**. Then configure as follows:

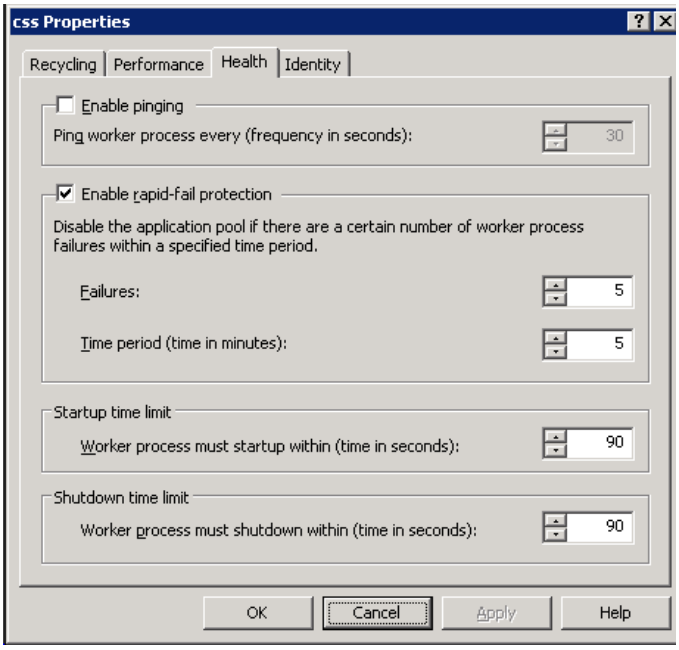
Recycling – Set up a nightly recycle of the Application Pool. To maximize DSS site availability while maintaining site performance, Ingeniux recommends recycling the Application Pool daily.



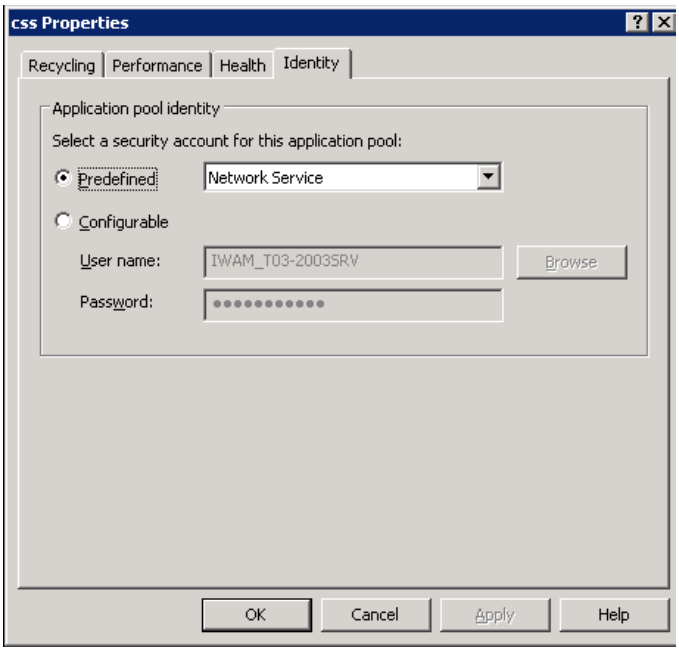
Performance – Increase the **Request queue limit** to 4000. Additional web gardens should not be added if the DSS site relies on the maintenance of session state for any application hosted on the DSS site, since extra web gardens allow multiple sessions.



Health – Clear **Enable rapid fail protection** and increase the startup and shutdown timeout values to 300 seconds.



Identity – Configure the Network Service account.

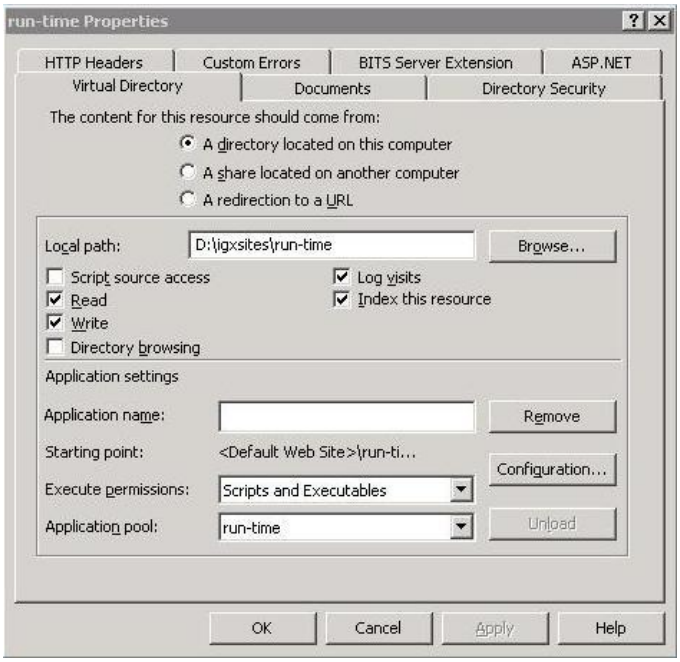


5.1.3 Configuring an IIS Website or Virtual Directory

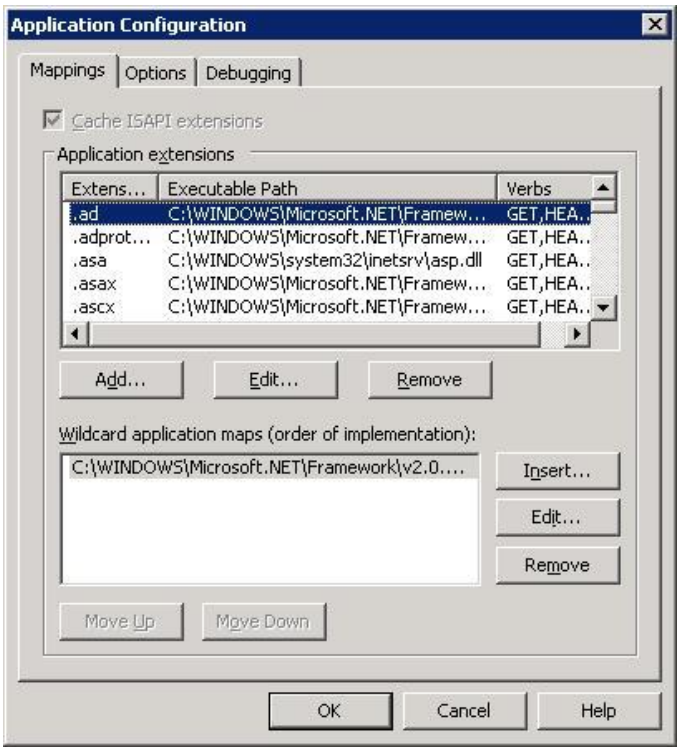
To view configuration information for a DSS site, right-click the site in IIS Manager and select **Properties**. The settings for the site are distributed among seven tabs. To configure a DSS site, you'll need to ensure that the appropriate settings are selected at the **Home/Virtual Directory** tab, the **Documents** tab, and the **Directory Security** tab.

Home/Virtual Directory – This tab contains basic site settings. The tab is labeled either "Home Directory" or "Virtual Directory" depending upon your configuration. The following settings should be configured:

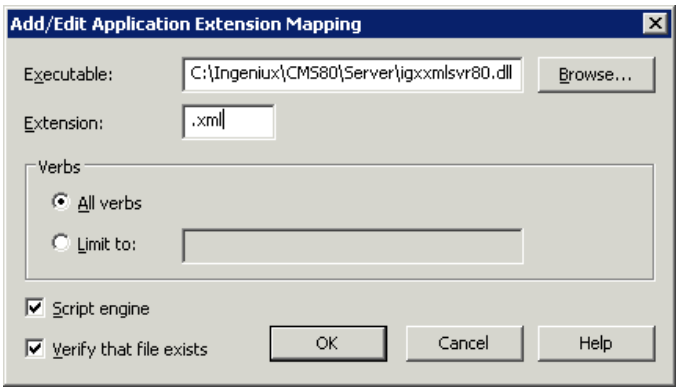
- **Local Path** – Specifies the path to the DSS files.
- **Permissions** – Should be set to Read, Write, Log Visits, and Index this resource.
- **Application Pool** – Specifies the application pool for the site.



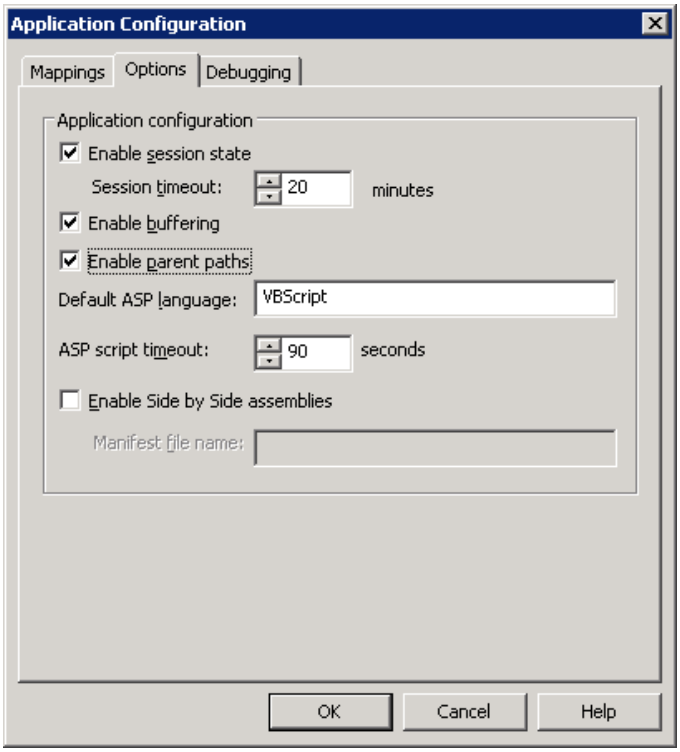
- **Configuration** – Contains extension mappings as well as additional application settings. The .xml extensions should be mapped to igxxmlsvr80.dll. If they are not present, click **Add** in the **Mappings** tab.



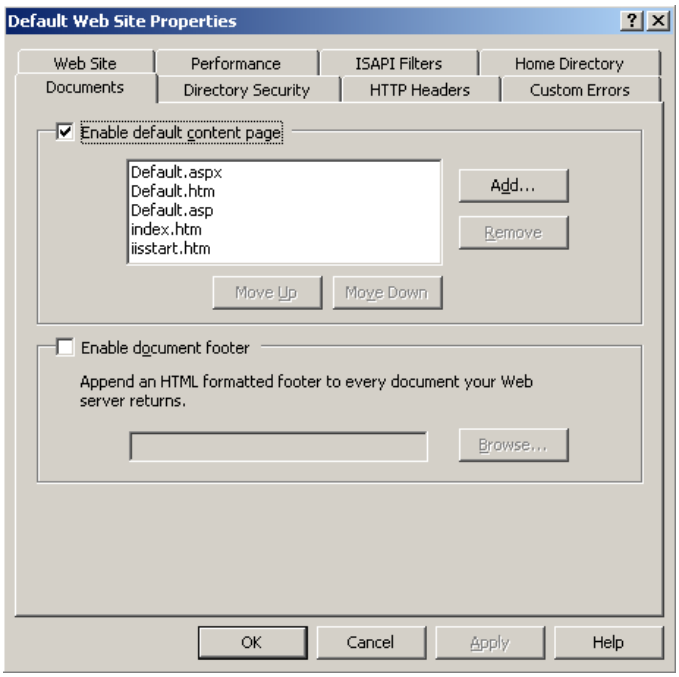
Enter the path to the desired executable (DLL) or use the **Browse** button to locate the DLL on the disk. Add the related extension in the appropriate field and click **OK**.



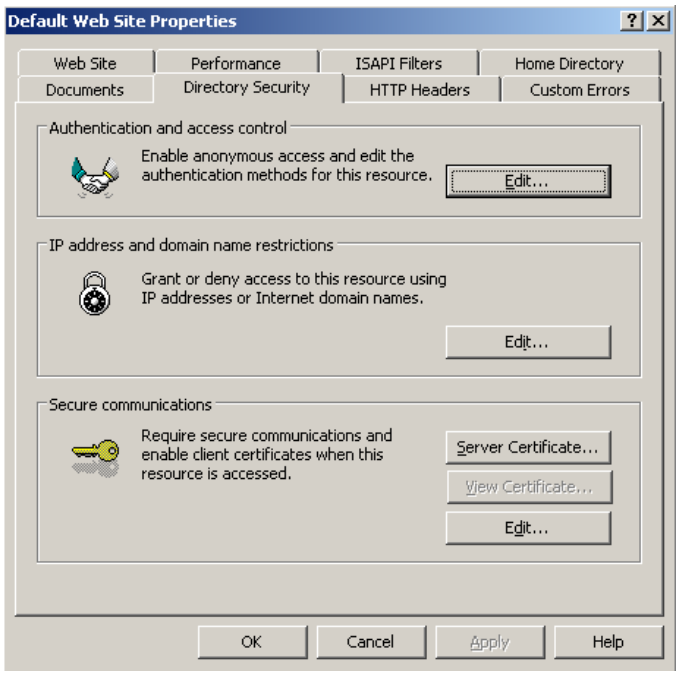
Enable Parent Paths should be selected in **Configurations** > **Options**.



Documents – This tab defines the default document loaded by IIS when the site is requested. This should be set to the home page for the site, e.g. x127.xml.



Directory Security – Defines access to the site. Authentication and Access Control should be set to use Integrated Windows Authentication.



Confirm that Anonymous Access is granted to the site by clicking **Edit** in the "Authentication and access control" section of the **Directory Security** tab and selecting **Enable anonymous access**.

5.1.4 Configuring Web Service Extensions

The DSS web service extension settings for Windows 2003 mirror the CMS settings. For details, see 4.1.4.

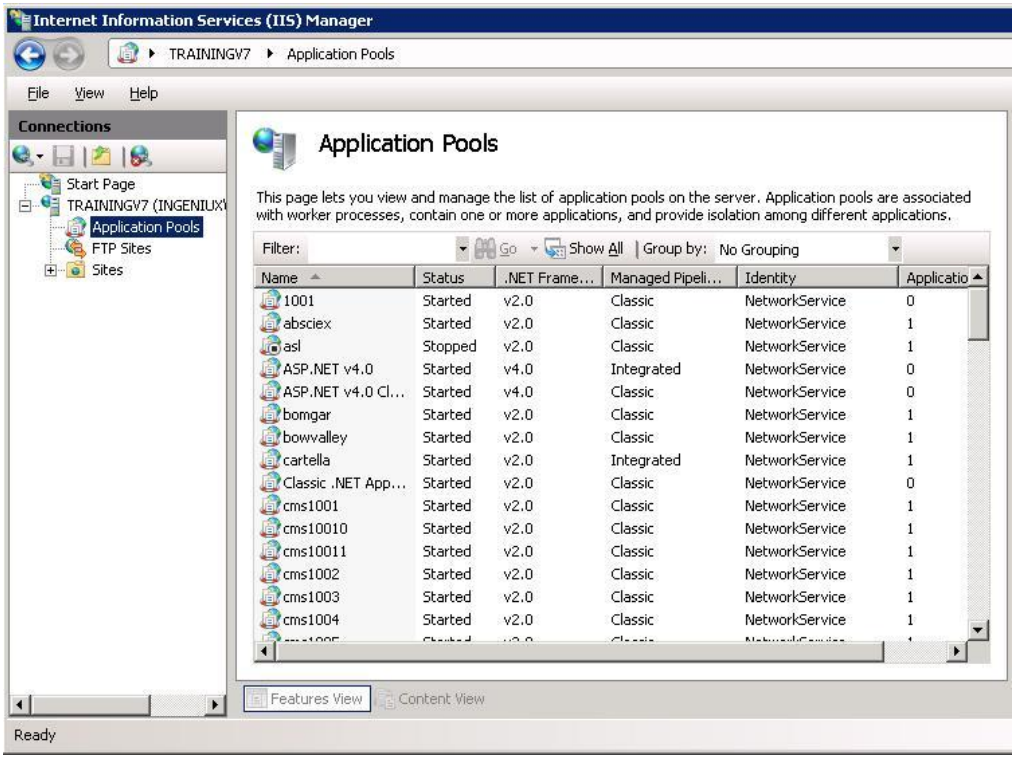
5.2 DSS Site Configuration (IIS 7.0)

This section describes DSS site configuration in a Windows Server 2008/IIS 7 environment.

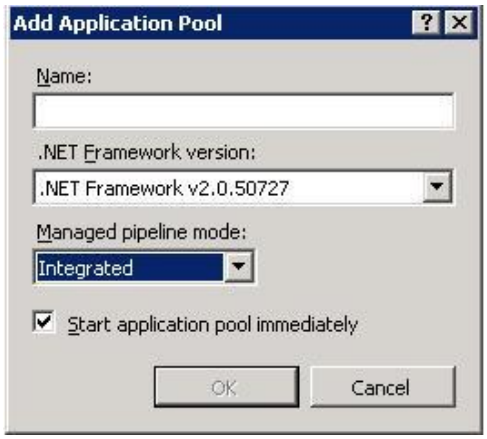
5.2.1 Creating an Application Pool

Ingeniux recommends using separate Application Pools for each site (whether an IIS website or virtual directory) running on a Dynamic Site Server (DSS). Using separate Application Pools limits the impact each site has on the other(s) and makes it easier to identify problem websites, as each Application Pool runs as a separate process.

To create a new Application Pool, right-click **Application Pools** in IIS Manager and select **Add Application Pool**.



The Add Application Pool dialog opens.

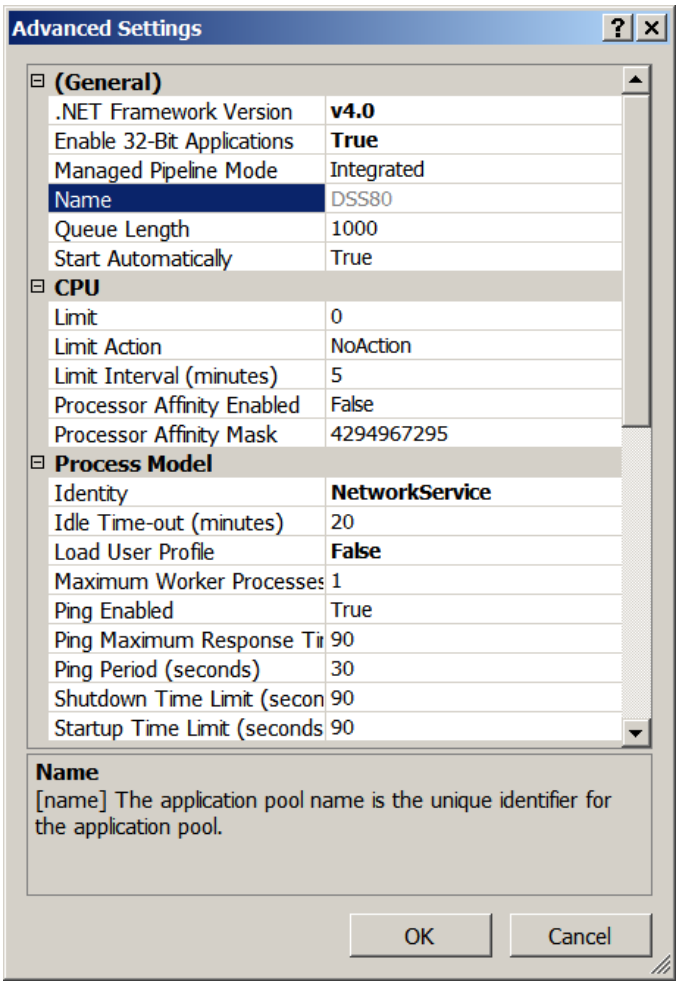


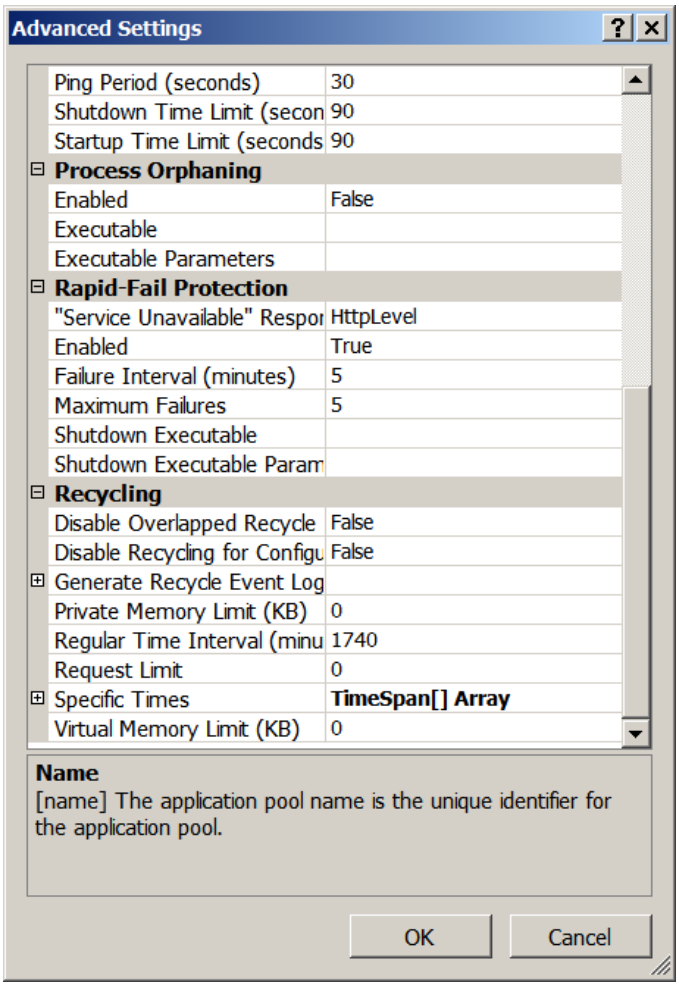
Enter a name for the DSS Application Pool, set the .NET Framework to version 4.0, leave the managed pipeline mode set to **Integrated**, and click **OK**.

5.2.2 Configuring the Application Pool

To configure the Application Pool, click **Application Pools** in IIS Manager, select the Application Pool for the DSS site, and click **Advanced Settings**.

The advanced settings for both 32-bit and 64-bit are the same, with the addition in 64-bit of **Enable 32-bit Applications** being set to True. The two windows below show the advanced settings for a DSS site:





5.3 File Level Permissions

The DSS file level permissions mirror those of the CMS. For details on file level permissions, see 4.3.

5.4 Log File Configuration

The DSS log file configuration mirrors the CMS configuration. For details on log file configuration, see 4.4.

5.5 Site Registry Entry Verification

The DSS registry values mirror CMS registry values. For details on site registry values, see 4.5.

6 Maintenance Guidelines

Ingeniux recommends conducting CMS server maintenance at regular intervals. The frequencies suggested below are for a standard implementation. Depending upon the environment, these tasks may need to be performed more or less frequently.

Action	Frequency	Access Requirements	Automation	Impact to Users	Benefits
Back up \xml directory	At least once a week	File level access to server for initial setup	Depends on backup solution	Potential impact to site performance. Should be performed during non-peak hours.	Provides a means to recover from server/ application crashes and/or other types of catastrophic failures.
Empty recycle bin	Weekly	Administrator access to the site	Manual process	Potential impact to site performance. Should be performed during non-peak hours.	Removes page dependencies no longer needed for the site.
Dependency graph rebuild	Weekly	Administrator access to the site	ASP page triggered via scheduled task	Potential impact to site performance. Publishes will queue up until a full publish is completed. Should be performed during non-peak hours.	Maintains accurate page dependencies; purges dependencies based on deprecated pages, etc.
Archive publishing logs to a directory other than the publishing target directory	Weekly	File level access to server for initial setup	Script triggered via scheduled task	None	Provides a record of previous publishes.
Archive igxcsapi.log files	Weekly	File level access to server for initial setup	Script triggered via scheduled task	None	Decreases disk space usage.
Reset Application Pool	Daily	File level access to server for initial setup	IIS and application configuration	Last publish request may be lost. Should be performed during	Frees memory used by the IIS application.

				non-peak hours.	
Archive/purge IIS logs	Weekly	File level access to server for initial setup	Script triggered via scheduled task	None	Decreases disk space usage.
Rebuild Index catalog(s)	Weekly	File level access to server for initial setup	Script triggered via scheduled task	CMS search unavailable during process.	Decreases disk space usage; refreshes search catalogue.
System/swap file defragment	Weekly	File level access to server for initial setup	Script triggered via scheduled task	Potential impact to site performance. Requires two server reboots to complete. Should be performed during non-peak hours.	Maintains system performance.

6.1 CMS Site Maintenance Tasks

Before undertaking the system maintenance described in 6.2, it's a good idea to perform a few simple site maintenance tasks. Ingeniux recommends completing the following tasks in the order listed.

6.1.1 Back Up the \xml Folder

Description – Prior to any maintenance work, the [site]\xml directory, excluding the \xml\backup and \xml\pub directories, should be backed up and archived in a safe location.

Automation – There are various avenues to schedule backups. Ingeniux does not recommend any particular method. However, the option selected should meet the specific requirements of the environment.

Frequency – The frequency of backups should be in keeping with the level of site activity. Large sites with frequent publishes and heavy maintenance demands benefit from more frequent backups.

System Impact – The system impact will vary depending upon the capabilities of the I/O device and the size of the site being backed up. This process should occur during a period of limited usage.

User Access – During a backup, users can still access the system. But it's better if users are logged out, for two reasons:

- Changes to content may not be reflected in the backup, depending on when a page is modified. And new pages may not be listed in the reference.xml file, depending on when the backup is started.

- The backup process may interfere with a user's ability to access a file.

Publishing – Publishes shouldn't be conducted during a backup, because file access issues may arise if the processes are run simultaneously.

6.1.2 Empty the Recycle Bin

Description – Because the CMS can restore pages, pages that have been deleted to the recycle bin continue to impact dependency calculations. Having too many items in the recycle bin can negatively impact check-in and publishing times.

Frequency – In determining how frequently to empty the recycle bin, you should take into account two factors: 1) performance concerns, and 2) user expectations. If the recycle bin is emptied too frequently, users won't have time to restore deleted pages. If the recycle bin is not emptied on a regular basis, check-in and publishing times begin to increase.

Also, be aware that it takes longer to empty the recycle bin when there are a lot of files in it, and that emptying the recycle bin removes all files, no matter how long they've been there.

For a typical site, Ingeniux recommends emptying the recycle bin once a week. However, the specific needs of the users and the size of the site should dictate the frequency. It's a good idea to communicate the schedule to users, so that they know how long deleted pages will be available for restoration.

System Impact – The number of pages in the recycle bin determines the impact of emptying it. As the number of pages increases, the time needed to empty the recycle bin increases.

User Access – Depending on the number of pages in the recycle bin, publishes and check-in times can increase during this process. Removing a large number of pages may cause the application to appear unresponsive.

Publishing – Ingeniux recommends not publishing pages while the recycle bin is being emptied.

6.1.3 Dependency Graph Rebuild

Description – A dependency graph is a database that tracks relationships between pages. Ingeniux recommends periodically rebuilding the dependency graph to ensure the accuracy and efficiency of page dependencies.

To rebuild a dependency graph:

1. Stop IIS.
2. Delete the depgraph*.db file(s) in [site directory]\xml.
3. Restart IIS.
4. Conduct a full publish for each publishing target.

This process also removes XML page files for deleted pages.

Frequency – Ingeniux recommends rebuilding page dependencies once a week for a typical site. Large sites with a lot of user activity require more frequent dependency graph rebuilds.

System Impact – The CMS may not be available or may appear unresponsive during a dependency graph rebuild.

User Access – Ingeniux recommends that users not access the system during this process.

Publishing – Ingeniux recommends not conducting any incremental publishes after dependency graphs have been deleted or cleared, until a full publish has completed. Any incremental publishes submitted will be queued and won't start until after the full publish has completed.

6.2 System Maintenance

In addition to application-specific tasks, a number of regular tasks need to be conducted to maintain an optimal CMS environment. The tasks below address parts of the server operating system that can impact performance if not regularly maintained.

6.2.1 Archive Publishing Logs

Description – With each publish, the application generates an XML file that lists the files published and provides associated publishing performance data. These log files are located in the \xml\pub directory, and the Publish Monitor can display logs for a given publishing target.

Over time, log files can accumulate and impact the performance of the Publish Monitor. Ingeniux recommends periodically deleting these files or archiving them to another location.

Frequency – Ingeniux recommends archiving the publishing logs weekly under normal usage. With heavy site use, archive the logs more often.

System Impact – The system impact will vary depending on the capabilities of the I/O device and the number of publishing logs to be archived. Typically, archiving log files will not impact performance. Ingeniux recommends retaining publishing logs for two months, in case they are needed for troubleshooting or performance monitoring.

User Access – Archiving the publishing logs will only impact user access to the log files in the Publish Monitor.

Publishing – Archiving publishing logs will not impact publishing.

6.2.2 Reset Application Pool

Description – IIS uses Application Pools to manage memory. Ingeniux recommends resetting the Application Pool associated with the Ingeniux site once a day. During the reset, the Ingeniux site will not be available. In addition, any Ingeniux process (for example, a publish or check-in) will abruptly end. In some cases, this may cause the program to lose track of a file or to corrupt the dependency graphs. As a result, the Application Pool should be reset during periods of light use.

Automation – Application Pool recycling can be automated in IIS.

Frequency – The Application Pool for the Ingeniux site should be reset daily.

System Impact – All IIS sites associated with the Application Pool are unavailable during the reset.

User Access – Because IIS sites associated with the Application Pool are unavailable during the reset, users should not attempt to access the CMS just before or during this process.

Publishing – Because IIS sites associated with the Application Pool are unavailable during the reset, users shouldn't try to publish content just before or during this process.

6.2.3 Archive/Purge IIS Logs

When enabled to do so, IIS writes log files for requests processed. These log files take up space on the hard drive. Ingeniux recommends archiving these files, especially if they are written to the system drive.

Frequency – The frequency of IIS log archiving should be proportional to the size of the log files and the number of requests processed by IIS. Ingeniux recommends archiving the IIS logs weekly, in a typical installation. Archived IIS logs should be kept for two months for troubleshooting and performance monitoring.

System Impact – This process shouldn't significantly impact system performance, provided that the number and size of log files are typical.

User Access – User access will not be impacted by this process.

Publishing – Publishing will not be impacted by this process.

6.2.4 System File Defragmentation

Over time, as files are moved and copied, they become fragmented (stored in non-contiguous sectors on the hard disk). Files take longer to access as they become more fragmented. As this fragmentation increases, system performance begins to degrade. A regular cycle of defragmentation (re-writing files on the hard disk so they're contiguous) is required to prevent the degradation of system performance.

The Windows Disk Defragmenter tool does not defragment the system paging file. This paging file is used by the operating system to augment a computer's physical memory. It's important to defragment this file as well. There are several utilities available that support defragmentation of the Windows Paging file. To download one of these utilities, PageDefrag, go to

http://technet.microsoft.com/en-us/sysinternals/bb897426

Frequency – Ingeniux recommends defragmenting the system once a week in a typical CMS environment, but the frequency should be based on system usage. Larger sites with heavy usage may need defragmentation more frequently; smaller sites with light usage may not need it as

often. The time needed to defragment a system is proportional to the amount of data and the degree of fragmentation. Frequent defragmenting will ensure short defragmentation times.

System Impact – The defragmentation process consumes significant system resources because the process affects every file. Defragmenting the Windows Paging file requires a system reboot. Ingeniux recommends preventing users from accessing the system during both processes.

User Access – Users should not access the CMS site while this process is being performed.

Publishing – No publish actions should be performed during this process.

6.3 Maintenance Schedule Example

For this example, assume a medium-sized liberal arts college runs a large CMS site. This site contains over ten thousand pages. Over a hundred users log in, edit pages, and create content. They conduct numerous publishes between 7:00 A.M. and 6:00 P.M., Monday through Friday.

Also assume two people administer the site, a site administrator and a server administrator.

Both administrators work between 7:00 A.M. and 5:00 P.M., Monday through Friday. Both are available after hours as needed and during maintenance periods in the event of difficulties.

The table below shows a recommended schedule for maintenance tasks on this site. The administrators agree that most maintenance should occur on Thursday afternoons and Friday mornings, because these are periods of relatively limited user activity and maximum support availability. The IIS Application Pool will be recycled nightly.

Operation	Frequency	Start Time	Estimated Time to Complete	Manual/Automated	Required User/Automation
Back-Up \XML	Weekly/ Thursdays	3:00 PM	~30 minutes	Manual	Server Admin
Empty Recycle Bin	Weekly/ Thursdays	3:30 PM	~20 minutes	Manual	Site Admin
Dependency Graph Rebuild	Weekly/ Fridays	1:00 AM	~40 minutes	Automated	Scheduled Task
Rebuild Indexing Catalog	Weekly/ Fridays	2:00 AM	~15 minutes	Automated	Scheduled Task
Defragment System Drive	Weekly/ Fridays	2:30 AM	~45 minutes	Automated	Scheduled Task

Defragment Page File/ Server Reboot	Weekly/ Fridays	3:30 AM	~20 minutes	Automated	Scheduled Task
Application Pool Reset	Daily	4:15 AM	~3 minutes	Automated	IIS Process

6.4 Site Optimization

The following are best practices for implementing a site in the Ingeniux CMS:

- Set start pages on all ancestor navigations. The start page marks the point in the site tree from which pages are pulled into a navigation. Pages up to, but not including, the start page are pulled into ancestor navigations. At a minimum, the start page should be set to the home page.
- As a general rule, avoid the use of subtree navigations.
- Use a Site Control to hold elements used by each page.
- To lessen the XML content load, use `mode` to display a single element in multiple ways.

For a system that employs a legacy XSLT runtime, the following best practices are also relevant:

- Optimize style sheets and coding practices to improve the processing of XSLT.
- Do not use `//` to indicate an XPath.
- Use `<xsl:variable>` whenever possible to represent commonly used XPaths.
- Use `<xsl:apply-templates>`.
- Limit the use of style sheets. For example, use one main style sheet (such as default.xsl) and include style sheets from there.

6.5 Restoring a Site from a Backup

When replacing server hardware or recovering from a system failure, you may need to restore a CMS site from a backup. The \xml directory located under the site directory contains all critical files specific to a CMS site. To restore a CMS site from a back-up, follow these steps:

1. Review and implement the HTML upgrade strategy if appropriate.
2. Backup the localstyles.css file.
3. Run IGXSetup and install the CMS system. (If it is already installed and configured, skip this step.)
4. Run IGX_CMS_Site_Setup and create a site.
5. Ensure that the permissions for the new \xml directory are configured.

6. Go to **Start** > **Run**, and run the following command: `IISreset /stop`
7. After the Command Window disappears, navigate to [sitedirectory]\xml for the new site.
8. Select **Edit** and choose **Select All**.
9. Select **File** and choose **Delete** to delete the contents of the \xml directory.
10. Copy the contents of the backed-up \xml directory into the \xml directory of the new site.
11. Once the file copy has completed, right-click the \xml directory, click **Properties**, and select the **Security** tab.
12. Select **Advanced** and verify that **Allow inheritable permissions**… is checked. Then check **Replace permission entries**… and click **OK**. This will ensure that the user has access to the directory.
13. Go to **Start** > **Run**, and type in the following command: `IISreset /start`.
14. Use section 7 of this manual to verify that the CMS is installed correctly.
15. Restore the backed up localstyles.css file.
16. Launch the CMS Client to verify access and to verify that the original site has been restored.

7 Installation Checklist

Once the full suite of CMS system components is installed and configured, it's a good idea to verify the success of the installation. This section provides a checklist of settings and components for the CMS and DSS. For more in depth configuration instructions, see sections 4 and 5.

- The following checklists refer to a Windows Server 2003 environment. The verification process will be slightly different for Windows Server 2008.

7.1 CMS Installation Checklist

1. Verify that MSXML 4 SP3 is installed.
2. Verify that .NET Framework 3.5 is installed.
3. Verify that ASP.NET is installed.

Using the Computer Management tool, open **Services and Applications** > **Internet Information Services (IIS) Manager** > **Web Sites**:

1. Right-click the IIS Website or Default Web Site and go to **Properties** > **Documents**. Verify that a valid default document exists.
2. Go to **Directory Security** and verify that anonymous access is enabled.
3. Verify the following for the **Virtual Directory** or **Home Directory** tab:
    a. **Read** and **Write** permissions are selected.
    b. Execute permissions is set to Scripts only.
    c. The Application Pool is valid for Windows Server 2003 (for Windows Server 2008, the Application Pool pipeline mode must be set to Classic).
    d. The .xml extensions are mapped to igxcsapi80.dll in **Configuration**.
    e. The option to **Verify that file exists** is unchecked for the .xml extension (you can check this by double-clicking the mapping in **Configuration** > **Mappings**).
    f. The file path \windows\Microsoft.NET\Framework\v2.0.50727\aspnet_isapi.dll is mapped under Wildcard application maps.
    g. The option to **Verify that file exists** is unchecked for aspnet_isapi.dll.
    h. In **Configuration** > **Options**, the option to Enable parent paths is selected.
4. In the **ASP.NET** tab (if present), verify that the ASP.NET version is set to 2.0.50727. This tab will be present only if there are multiple versions of .NET on the system.
5. Verify that the appropriate file-level permissions are set on the \xml directory and all files and subdirectories contained in the \xml directory.
6. Verify that the appropriate file-level permissions are set on the \server directory and all files and subdirectories contained in the \server directory.
7. Verify that the appropriate web extensions including ASP.NET have been allowed.
8. Verify that the Ingeniux DLLs have been added as web service extensions and set to **Allowed**.

7.2 DSS Installation Checklist

1. Verify that MSXML 4 SP2 is installed.
2. Verify that .NET Framework 4.0 is installed.

3. Using the Computer Management tool, open **Services and Applications** > **Internet Information Services (IIS) Manager** > **Web Sites**.
4. Right-click the IIS Website or Default Web Site and go to **Properties** > **Documents**. Verify that a valid default document exists.
5. Go to **Directory Security** > **Edit** and verify that Enable anonymous access is selected.
6. Verify the following for the **Virtual Directory** or **Home Directory** tab:
    a. **Read** permission only is selected.
    b. The **Execute permissions** setting is set to **Scripts only**.
    c. The Application Pool is valid for Windows Server 2003 (for Windows Server 2008, the Application Pool pipeline mode must be set to Integrated).
    d. The .xml extensions are mapped to igxcsapi80.dll in **Configuration**.
    e. The option to **Verify that file exists** is unchecked for the .xml extension (you can check this by double-clicking the mapping in **Configuration** > **Mappings**).
    f. If structured URLs are enabled and configured to use the .htm and/or .html extensions, verify that these extensions are mapped to igxxmlsvr80.dll and that the option to **Verify that file exists** is unchecked for these extension mappings.
    g. The option to Enable Parent Paths is selected in **Configuration** > **Options**.
7. Verify that the appropriate file-level permissions are set on the \[site] directory and all of its subdirectories.
8. Verify that the appropriate file-level permissions are set on the \server directory and all of its subdirectories.
9. Verify that the appropriate web extensions have been allowed.
10. Verify that the Ingeniux DLLs have been added as web service extensions and set to **Allowed**.

8 Upgrades

Ingeniux regularly releases new versions of the CMS system. To deploy an updated version of the CMS, you will need to install the software and then upgrade sites.

To upgrade a CMS site, run IGX_CMS_Site_Upgrade.

Before upgrading a functional site, you may want to make a copy of the site and work through the upgrade process on that copy. Once issues of authentication, authorization, and familiarization with the new features have been resolved, the new CMS site can be deployed.

The following steps outline the general process for upgrading to CMS 8.0 and may not apply to the specific requirements of all CMS configurations. Ingeniux recommends contacting Support for assistance with outlining a specific upgrade strategy.

Ingeniux also recommends backing up the following files before performing an upgrade on a working site:

- /xml directory
- local-appsettings.config
- local-connection-strings.config
- local-membership.config
- Web.config

8.1 Upgrading from CMS 4.2

To upgrade a CMS 4.2 implementation, follow these steps:

1. Review and implement the HTML upgrade strategy.
2. Copy the contents of the \xml directory to a safe location (i.e. to a directory that is backed up).
3. Run IGXSetup to install the Ingeniux 8.0 DLLs on the CMS server.
4. Run IGX_CMS_Site_Upgrade, which is located in \cms80\Tools, to upgrade the site on the CMS.
5. Go to **Start** > **Run**, and type the following:

   ```
   IISRESET /Stop
   ```

6. Navigate to the \XML directory and delete the following file(s):

   ```
   DepgraphX.db
   ```

   where `X` represents some number.

7. Go to **Start** > **Run**, and enter the following:

```
IISRESET /Restart
```

8. Turn off any replication of content to the Run-Time site.
9. Launch the CMS client and conduct a full publish on all publishing targets.
10. Run IGXSetup to install the CMS system software on the DSS.
11. Run IGX_Dynamic_Site_Server_Setup and configure the DSS site (the new replication process needs to be configured in the CMS Client).

8.2 Upgrading from CMS 5.x

The upgrade process attempts to transfer toolbar settings from toolbar_sets.xml to tinymceconfig.xml. Uploaded file extension types and Pretty Print options configured in config_igx.asp will not be migrated, as the new editor does not support these options.

To upgrade from CMS 5.x, follow these steps:

1. Review and implement the HTML upgrade strategy.
2. Backup the localstyles.css file.
3. Copy the contents of the \xml directory to a safe location (i.e. to a directory that is backed up).
4. Run IGXSetup to install the CMS system on the CMS.
5. Run IGX_CMS_Site_Upgrade, located in \cms80\Tools, to upgrade the CMS site.
6. Restore the localstyles.css file.
7. Go to **Start** > **Run**, and type in the following:

   ```
   IISRESET /Stop
   ```

8. Navigate to the \xml directory and delete the following file(s):

   ```
   DepgraphX.db
   ```

   where `X` represents some number.

9. Go to **Start** > **Run**, and enter the following:

   ```
   IISRESET /Restart
   ```

10. Turn off any replication of content to the Run-Time site.
11. Launch the CMS client and conduct a full publish to all publishing targets.
12. Run IGXSetup to install the CMS system software on the DSS server.
13. Run IGX_Dynamic_Site_Server_Setup and configure the DSS site (the new replication process needs to be configured in the CMS client).

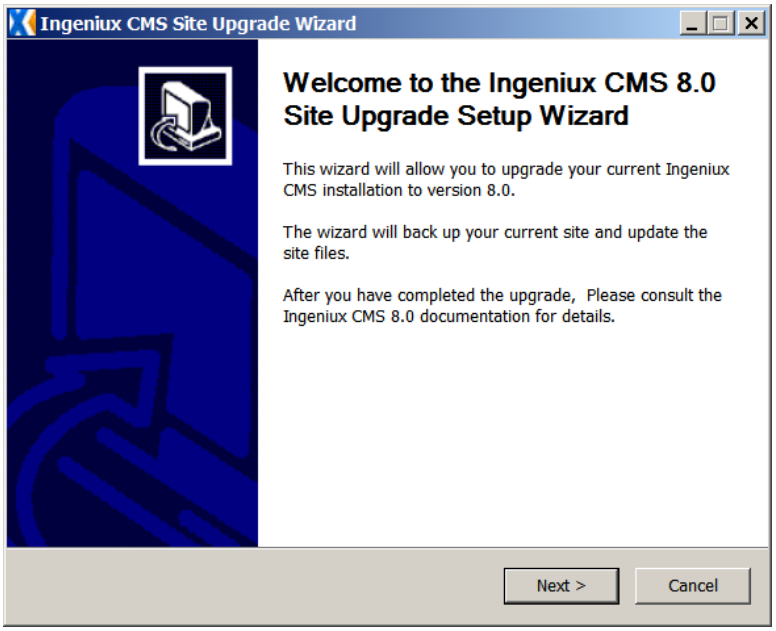
8.3 Upgrading to CMS 8.0

The IGX_CMS_Site_Upgrade utility upgrades a CMS site to the current version of the application. In addition, IGX_CMS_Site_Upgrade configures the site to use the ASP.NET authentication mechanism.

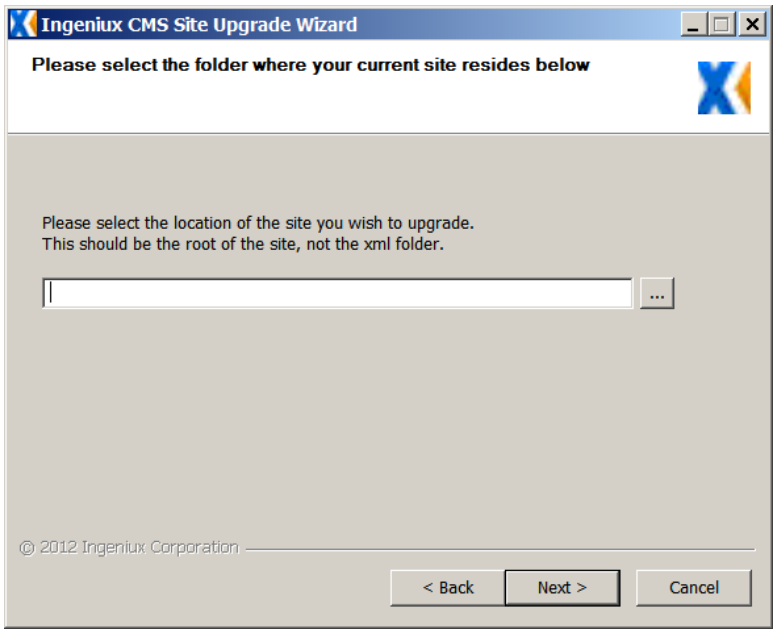
This utility must be run on the host server by a user with administrative access to the Windows server. This user must also have administrative privileges for the remote directory if a remote directory is the target of the upgrade.

- Note: If the existing site is located on a remote directory, a mapped drive should be used for the upgrade. Once the utility has completed, the IIS website or virtual directory can be modified to use a UNC path (for example, \\[computername]\sharename).

To begin the upgrade process, go to [drive:]\Ingeniux\CMS80\Tools and double-click **IGX_CMS_Site_Upgrade**. The Site Upgrade Wizard opens.

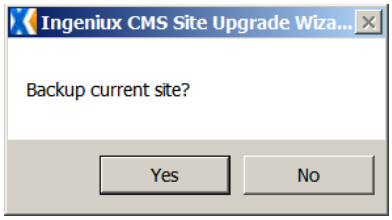


Click **Next**. The Wizard prompts you to enter the location of the site you want to upgrade.



Click the ellipsis button (…) and select the site to be upgraded. Then click **OK** > **Next**.

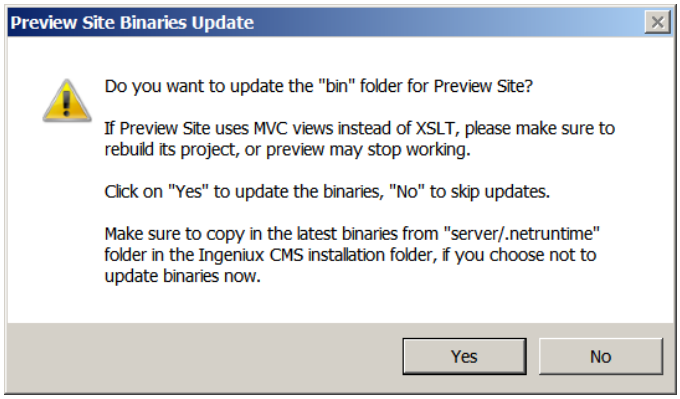
The Wizard asks you whether to back up the site.



To back up the site, click **Yes** and select a backup directory. Then click **Upgrade**. To upgrade without a backup, click **No**.

The upgrade process begins. If the Wizard asks to stop and restart IIS, click **Yes**.

During the upgrade, a dialog asks if you want to update the bin folder for the preview site. In most cases, Ingeniux recommends performing the update.

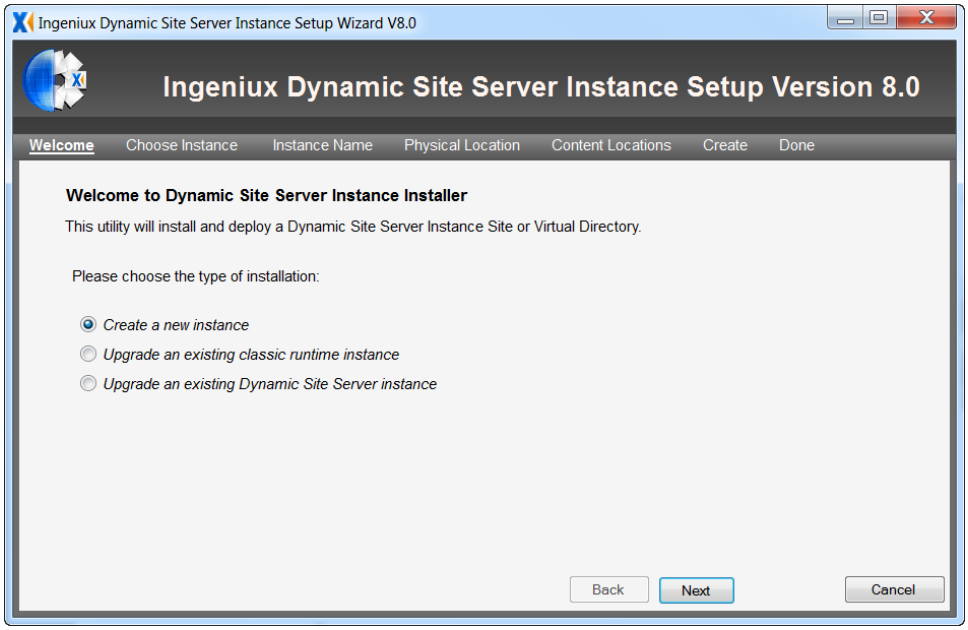


To update the bin folder, click **Yes**. For an XSLT site, no further action is necessary. For an MVC site, you will also need to recompile the preview site and redeploy the DSS site.

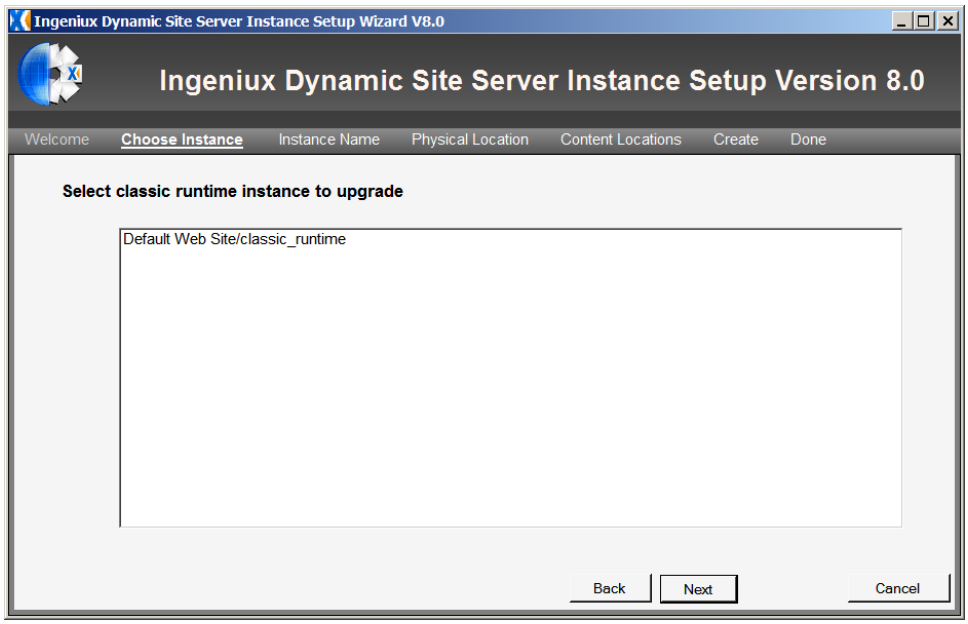
When the upgrade process is complete, click **Finish**.

8.4 Upgrading to DSS 8.0

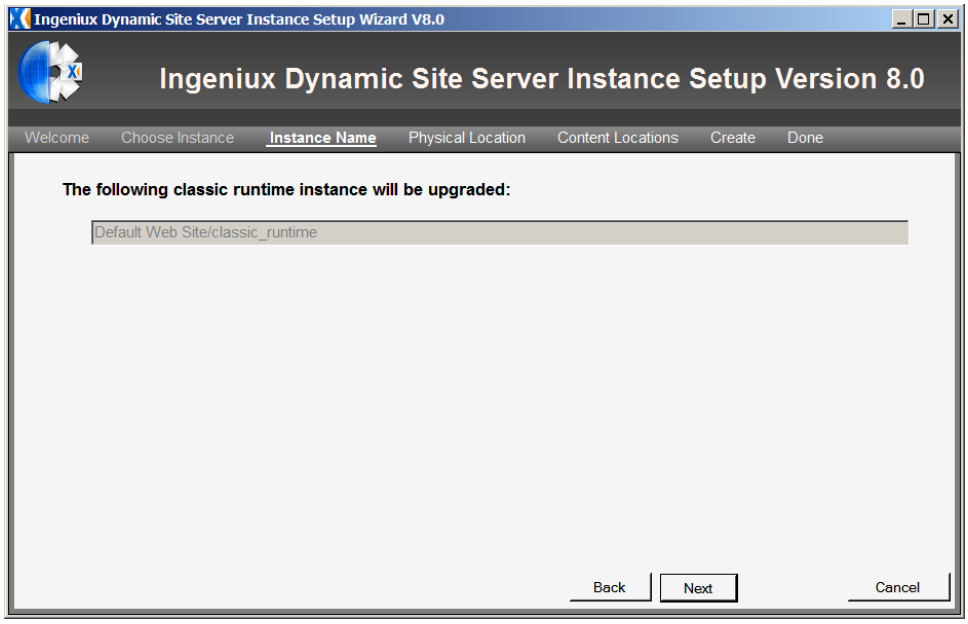
To upgrade a classic Runtime site, go to [drive:]\Ingeniux\CMS80\Tools and run IGX_Dynamic_Site_Server_Setup. The setup wizard opens.



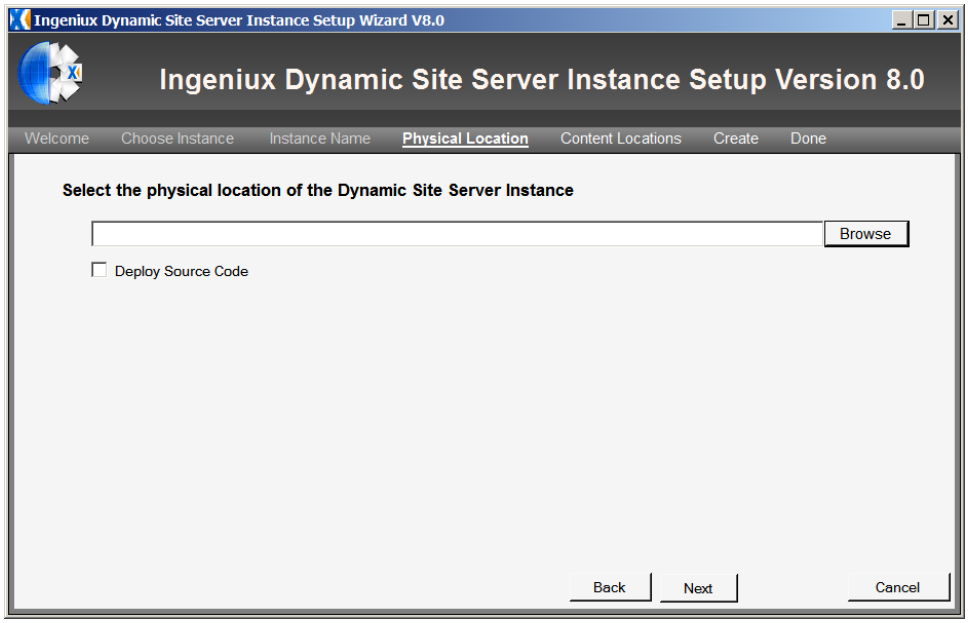
Select **Upgrade an existing classic runtime instance** and click **Next**.



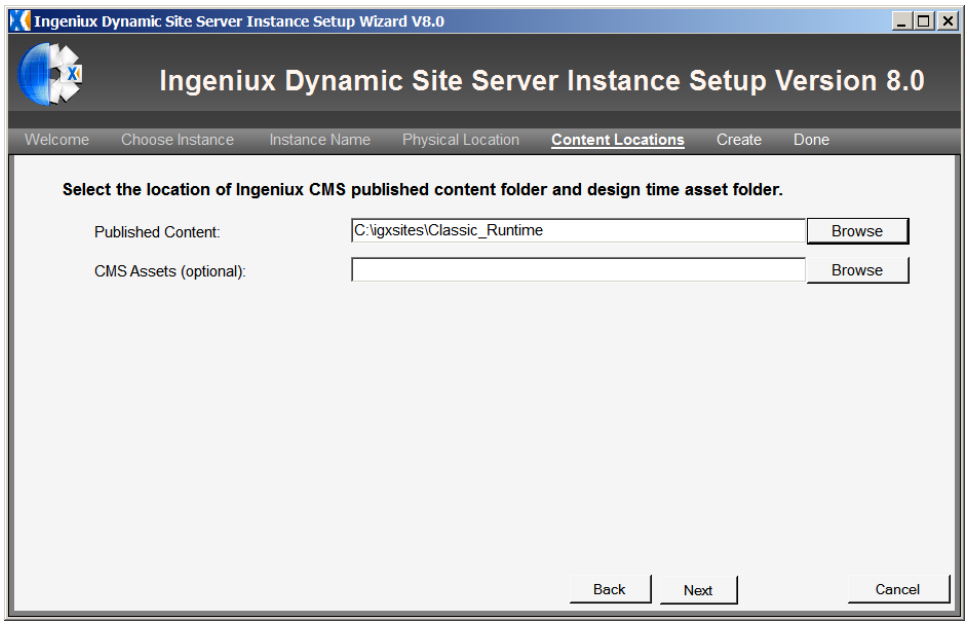
Select a site to upgrade and click **Next**.



To confirm the runtime instance to be upgraded, click **Next**.



Click **Browse** and select a location for the DSS instance that will replace the classic Runtime instance. If you want to deploy the sample MVC solution included with the DSS, enable **Deploy Source Code**. Click **Next**.



Select a published content folder (typically, this will be the location to which the CMS replicates published content) and, optionally, a separate store of CMS assets to be used by the DSS. Click **Next**.

The setup wizard creates a new DSS instance.

9 Glossary of Terms

Application Pool	An area in system memory used by IIS to run one or more IIS applications.
Application Pool Account	A Windows account assigned to an application whose credentials are used by IIS to access system resources.
Archive	A store of CMS pages in a Microsoft SQL database.
ASP.NET MVC	A Microsoft development technology for building dynamic web applications. MVC stands for "Model View Controller" – an application architecture on which the DSS is built.
Authentication	The process whereby a user's credentials are confirmed – specifically that the user exists and that the user's password is valid. In the Ingeniux environment, this validation is provided by another application such as a Windows Domain Controller or an OpenLDAP server.
Authorization	The process whereby the CMS determines what privileges a given authenticated user possesses.
Categorization	An association between a taxonomical term and a particular node.
Check-in	The process of submitting changes to the CMS prior to publishing a page.
Check-out	The process of requesting permission to make changes to a page.
Child	A page existing one level below the current page in the site tree.
Child Navigation	A mechanism for pulling content from nodes below the current page.
CMS	Hosts the CMS site, where content creators build, manage, and publish content (formerly called the Design-Time server).
CMS Client	Used to build, manage, and publish content on the Design-Time server.
Components	Content designed to be used in multiple pages of the site. Content contained in a component cannot be transformed until the component is pulled into a page.
Content Store	XML pages that make up the site.
CSS	Cascading Style Sheets. A simple style sheet language used to manage the presentation of content for a browser. This language is used in conjunction with HTML to render XML for a requesting browser.
Dependencies	The connections between pages.

Dependency graph	A database specific to the publishing target that contains the list of dependencies for each page.
DSS	Dynamic Site Server. A public-facing website that serves content published by the CMS. (Beginning with CMS 8.0, the DSS replaces the Run-Time server.)
Full Publish	Publishes all checked-in pages marked for publish for a given publishing target; deletes the contents of the publishing target before publishing.
Incremental Publish	Publishes selected pages to a publishing target.
Navigation	A mechanism for pulling in content from site nodes based on the site hierarchy.
Page Creation Rules	Automate the creation of new pages and components and specify where in the site tree a new page or component is created. Page creation rules are used in conjunction with workflow to simplify the creation and management of content.
Page Template/Schema	An XML page used as a template for creating new pages. A template specifies the structure of a page, including element types and their attributes.
Pages	XML files that contain site content or components. Pages are identified by unique xIDs.
Parent	A page existing one level above the current page in the site tree.
Permissions	The functions a given user group is able to perform (e.g. see the site tree, delete pages, create pages, etc.).
Publish	The processing of a page by the CMS, readying it for replication to the DSS.
Publishing Target	A sub-directory of the [site]\xml\pub directory to which pages are published. The pub target is specified during an incremental or full publish.
Read Only Access	Allows a specified group the ability to view the node (and the nodes inheriting this permission).
Recycle Bin	Holds deleted pages. Pages can be restored if the recycle bin has not been emptied.
Replication	The process of copying published pages from the CMS to the DSS.
Schemas	Templates from which CMS users can create and manage pages or components.
Site Map	The logical tree structure (hierarchy) for the Ingeniux site or site(s); maintained by the reference.xml file, which exists over the flat file structure in the \xml directory.
Site Tree	A logical representation of pages organized using ancestors, siblings, and children.

Start Page	An attribute of an ancestor navigation element which indicates the highest level node of the tree; navigation stops one page below the page specified.
Structured URLs	Provide a mapping from xIDs to friendly, text-based URLs.
Stylesheet	A separate XSLT file used to format an XML document.
Taxonomy	A hierarchical naming convention used to create navigation based on category/node associations.
Workflow	An automated mechanism for moving content through the CMS. A workflow process is defined by a sequence of workstates that a page must move through as work is completed.
Workstate	The location of a given page within a workflow.
XML	Extensible Markup Language. A mark-up language using tags to structure content. An XML document does not contain any formatting information.
XML Processor	An application used to transform XML and style sheets into complete documents, usually to be consumed by an Internet browser.
XSLT	Extensible stylesheet language. A stylesheet language used to format an XML document.