160-6400 Roberts Street | Burnaby BC | V5G 4C9

Install and Configuration Guide

Active Content Manager Version 10

Last revised April 13, 2011

The Active Network, Ltd.
160 - 6400 Roberts Street
Burnaby, BC
Canada V5G 4C9
Office telephone: 604.438.7361 or 1.800.661.1196
Fax: 1.604.432.9708

Contents

Introduction

Overview

This guide provides detailed instructions for installing the Active Content Manager. For details on Site and Export Management, see the *ACM Administration Guide*.

Note The audience for this guide is IT staff and/or system administrators who will be installing, configuring and maintaining the Active CM. This guide assumes that you are skilled at configuring Microsoft Servers (OS, IIS, SQL Server, etc.), and have a thorough understanding of DNS and TCP/IP.

It is recommended that you read through this document completely before installing the Active Content Manager.

Installing and Configuring the Active CM involves the following high-level steps:

- Conduct deployment planning
- Review Hardware and Software Requirements
- Install the Active CM Application Files
- Install and Configure Health Monitor
- Configure Microsoft SQL Server or Oracle DBMS
- Configure IIS
- Configure the Active CM through web.config
- Configure Directory Security for ASP.NET
- Configure Index Server
- Configure your DNS
- Test Correct Operations

Once the Active CM is operating correctly, remember to plan:

- Backup and Restore Procedures

Deployment Planning

The Active Content Manager is a flexible enterprise software system that can service the needs of small/simple installations and very complex and highly scalable installations:

- In a small deployment, the Active CM and all supporting software will be installed on a single computer; this would be appropriate for small or

departmental websites. This deployment scenario offers little in the way of scalability, redundancy or survivability[1].

- In a large deployment, the Active CM can be installed a web-farm scenario with dozens of application servers connected to a SQL Server Cluster. This would be appropriate to support 100's of complex and/or high-traffic websites. This deployment scenario offers very high scalability, redundancy and survivability.

It is important that you understand your requirements and plan the appropriate deployment environment for your needs. Some items to consider:

- How many websites will you be hosting?
- How much traffic will the websites get?
- Are you deploying internet, intranet or extranet sites? How many of each?
- How many authenticated viewers will you have?
- How many content contributors will you have?

It is beyond the scope of this Guide to provide help with Deployment Planning. For detailed analysis of deployment options, please contact your Active Account Executive who will arrange an analyst to work with your organization.

1 Survivability is the ability of the system to survive in the face of catastrophic (and therefore infrequent) failure. For example, can the system continue to service requests in the event of a disk-array failing, a computer ceasing to function, or in the event of a power outage?

Installation

Installing the Application Files

Installation of the Active CM Application files is a simple process. A directory must be created on the server to hold the files and the application files will be copied to that directory. No changes are made to the registry. All configuration of the application is performed by modifying files and/or settings contained within the application directory.

Note Multiple instances of the Active Content Manager can be installed on a single server, with each instance running one or more sites. Each instance is completely isolated from all other instances as long as they are using separate application pools.

It is recommended that you close all other programs before you begin the Active CM installation process.

Create the Active CM Directory

Create a directory that will be used to install the Active CM application files. For example, **C:\Inetpub\cmsroot**.

- You can use any local drive for the Active CM application files directory.
- The Active CM does not rely on the name used for the application files. However, this path will be used when configuring IIS.

Copying the Active CM application files

The ACM application files exist in a folder called **Install files** which can be found in the release zip file. Copy all of the files from the **Install files** folder into your newly created Active CM application directory. (eg. C:\Inetpub\cmsroot)

We'll call this folder **CMS root**.

Renaming the default.aspx files

Once the Active CM files and folders are copied into CMS root, you need to rename the IronPointdefault.aspx file in the CMS root and Admin folder to default.aspx.

Default Administrator Password

Contact ACM support for the default Administrator username and password.

Note It is important to change the default administrator password after you login to the system for the first time.

Health Monitor

Installation and Configuration

The Health Monitor is used to ensure that all scheduled tasks are executed exactly at their scheduled time. HealthMonitor is implemented as a system service to prevent it from being disabled during a server reboot.

Note Even though ACM 10.0 requires .Net 3.5, the installutil.exe referenced below is still found in the .Net v2.0 directory. This is because .Net3.5 is only an extension of the .Net 2.0 runtime.

The Health Monitor package

The Health Monitor files can be found in a folder called **Health Monitor** which can be found in the release zip file.

Installation

1. Create Installation folder on any drive you like. C:\Program Files\HealthMonitor is recommended.
2. Copy the contens of the **Health Monitor** folder into the installation folder
3. Install the service using .NET InstallUtil.exe which is located in the following directory:
   C:\WINDOWS\Microsoft.NET\Framework\v2.0.50727\
4. Open an command prompt by typing CMD into the "Run" option on your start Menu.
5. Type in the following Command string:
   cd C:\WINDOWS\Microsoft.NET\Framework\v2.0.50727
6. Now type in the following to complete the install:
   installutil.exe "C:\Program Files\HealthMonitor\IronPoint.CM.HealthMonitorService.exe"

Configuration

Edit HealthMonitor.Config to specify the URL for hitting the keepalive.aspx file.

Enable Health Monitor Auto Start

Once the HealthMonitor service has been installed, it will need to be configured to start automatically.

1. Go to Services and find IronPoint Health Monitor Service.
   → Start → Administrative Tools → Services
2. Right Click on the service and select Properties.

3. Within the properties, change the start option to be Automatic
4. Save you changes
5. Start the service

Microsoft SQL Server

Create and Populate the Active CM Database

Note SQL Server MUST be installed in 'Mixed Mode'.

Note Some IT groups want to change the default location of the database data and log files and possibly the database name used. Steps 7 and 8 below provide instructions to do this.

1. Open **SQL Enterprise Manager**
2. Click on **Local Server** if you are configuring SQL Server on the same computer as the Active CM application files are installed on. Otherwise, choose the SQL Server instance (located on a different server) you wish to configure.
3. Choose the **Tools** menu and then click **SQL Query Analyzer**
4. Click on **File** and then click on **Open**
5. Browse to the Active CM application directory (eg. C:\inetpub\cmsroot\). Then browse to **\DatabaseScripts** directory.
6. Choose **IronPoint.Install.sql.**
7. Within the Schema script the path to the database **data** and **log** files can be changed. These paths are defined on the line of the file that starts with CREATE DATABASE. Locate these settings and review them to be sure they are what you want. Default is **C:\Program Files\Microsoft SQL Server\MSSQL\Data**
8. By default the database is named **IronPointCM**. While not recommended, if you want to change the database name that the Active CM will use, complete a search and replace for 'IronPointCM' using the new name as a replacement. Please note that you will need to change the name of the Database in subsequent Data scripts.
9. Click the **Execute Query** toolbar button (the green arrow) in **SQL Query Analyzer**
10. Once the query has run successfully the Active CM database has been created.

Note The schema creation script creates two users in SQL Server: CM and mcw. The CM (db_owner) user has password 'CMCM'. The mcw user has password 'iggy'.

Oracle

Create and Update the Active CM Database

Note These instructions have been written with the expectation that a qualified Oracle DBA will be executing these steps.

1. Create a new Oracle Database or identify an existing database into which the ACM schema can be imported.
2. Open SQL Plus and edit and run the `create_tablespaces.sql` and `create_users.sql` scripts. These scripts are located in the \DatabaseScripts\Version7.5Oracle\ folder in the installation package.
3. Import the `IronPointCM_7.500.017_Oracle.dmp` file using the following command in a DOS command prompt.

   ```
   imp <DBAUser>/<DBApassword>@<SID> buffer=32000
   file='C:\<path to .dmp file>' grants=y indexes=y
   rows=y log=OracleImport.log fromuser=CM touser=CM
   commit=y constraints=y compile=y
   ```

   Replace `<DBAUser>`, `<DBApassword>` and `<SID>` with the appropriate values for your database. Replace `<path to .dmp file>` with the directory the dump file is saved.

Note Due to its size, the dmp file is not included with the installation package. Please contact ACMsupport@ActiveNetwork.com or your Active Account Manager to get access to this file.

4. In SQL Plus, execute all sql scripts in the version folders (Version8.0Oracle, Version8.1Oracle, Version8.2Oracle, Version8.3Oracle, Version8.4Oracle, Version8.5Oracle, Version8.6Oracle, Version8.7Oracle, Version8.8Oracle and Version10.0Oracle) **beginning** with the script named `7.500.018AlterUsers(add HTMLEditor).sql` located in the \DatabaseScripts\Version7.5Oracle\ folder. **Be sure to run these scripts in the order they are listed.**

Configure the Database Connection

Default

By default, Active Content Manager is configured to connect to a SQL Server database running on the default instance of SQL Server on the same computer as the application. If your SQL Server is on a different computer, if you are using a named instance of SQL Server or if you are using an Oracle Database, then you must follow the appropriate steps below to modify the data source identifier used by the Active CM application to connect to the database.

SQL Server

To connect to a SQL Server other than the default local instance or to configure advanced settings:

1. Navigate to the Active CM Application directory, and then navigate to the **BIN** directory.
2. Right click on the file **IronPoint.DataAccess.dll.config**, select Properties, and clear the Read-only attribute.
3. Right-click the same file again and open it in Notepad or WordPad.
4. In the section of the config file named **dataSources**, locate the **dataSource** element with the name attribute **"SQL"** (this may be the only dataSource entry in your config file).
5. In the **connectionString** attribute you can configure how the Active CM connects to SQL Server. There are four settings you can adjust in the connectionString:
    a. Data Source - This is the name of your SQL Server instance. If you have SQL Server installed on the same computer use "localhost".
    b. Initial Catalog - This is the database name within the specified SQL Server instance.
    c. User ID - This is the user that the Active CM will use to connect to SQL Server.
    d. Password - This is the password used to connect to SQL Server.

Note As an advanced configuration option, you can configure the Active CM to connect to SQL server using a different User ID and Password.

Note You now have the ability to configure your database connection through a separate application. This application encrypts the **IronPoint.DataAccess.dll.config** file and writes the database connection information to the registry. You can then manage this information from within ACM. See the *Save Database Connection*

Information to the Registry section for details.

6. To change which SQL Server to use, modify the value for **Data Source**. Replace **(local)** with the name of your SQL Server instance. For example, if your SQL Server is named MyServer, this section of the config file will look like this:

```
<dataSource name="SQL" default="true"
   provider="SqlClient"
   connectionString="Data Source=MyServer;...
```

Oracle

To connect to an Oracle Database or to configure advanced settings:

1. Navigate to the Active CM Application directory, and then navigate to the **BIN** directory.
2. Right click on the file **IronPoint.DataAccess.dll.config**, select Properties, and clear the Read-only attribute.
3. Right-click the same file again and open it in Notepad or WordPad.
4. In the section of the config file named **dataSources**, locate the **dataSource** element with the name attribute **"Oracle"** (this may be the only dataSource entry in your config file).
5. In the **connectionString** attribute you can configure how the Active CM connects to Oracle. There are three settings you can adjust in the connectionString:
    a. Data Source – This is your database SID.
    b. User ID - This is the user that the Active CM will use to connect to Oracle.
    c. Password - This is the password used to connect to Oracle.

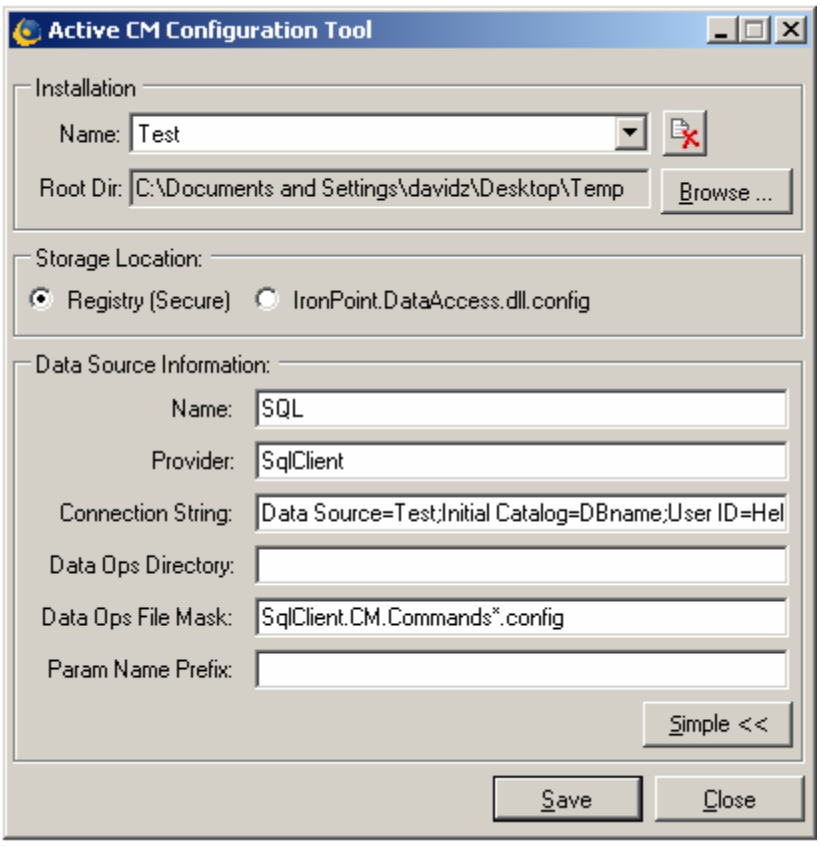
Note You now have the ability to configure your database connection through a separate application. This application encrypts the **IronPoint.DataAccess.dll.config** file and writes the database connection information to the registry. You can then manage this information from within ACM. See the *Save Database Connection Information to the Registry* section for details.

Save Database Connection Information to the Registry

You now have the option to save the data source connection information included in the **IronPoint.DataAccess.dll.config** file to the registry rather than leaving it in the config file. This provides a more secure location for your connection information as it no longer resides within the ACM installation directory structure.

Note Please note that this is only an optional process and only needs to be performed if you require the additional security.

1. Browse to the **\bin** folder and double-click on the **DataAccessSetupApp.exe** file to launch the Active CM Configuration Tool.



2. Enter a name value for this ACM installation.
3. Click the **Browse** button to browse to and select the root directory of your Active CM installation.
4. Select the **Registry (Secure)** option to write the connection information to the registry.

> **Note** This utility can also be used to write the connection information to the **IronPoint.DataAccess.dll.config** file. To do this, select the IronPoint.DataAccess.dll.config radio option.

5. In the **Name** field, type "SQL" if you are connecting a SQL database or "Oracle" if you are connecting to an Oracle database.
6. In the **Provider** field, type "SqlClient" for SQL or "Oracle" for Oracle.
7. In the **Connection String** field, type the following connection string format substituting the connection information where necessary.

```
Data Source=ServerInstance;Initial
Catalog=DatabaseName;User
ID=Username;Password=Password
```

   a. Data Source. This is the name of your SQL Server instance. If you have SQL Server installed on the same computer use "localhost". If you are connecting to an Oracle database, enter the database SID.

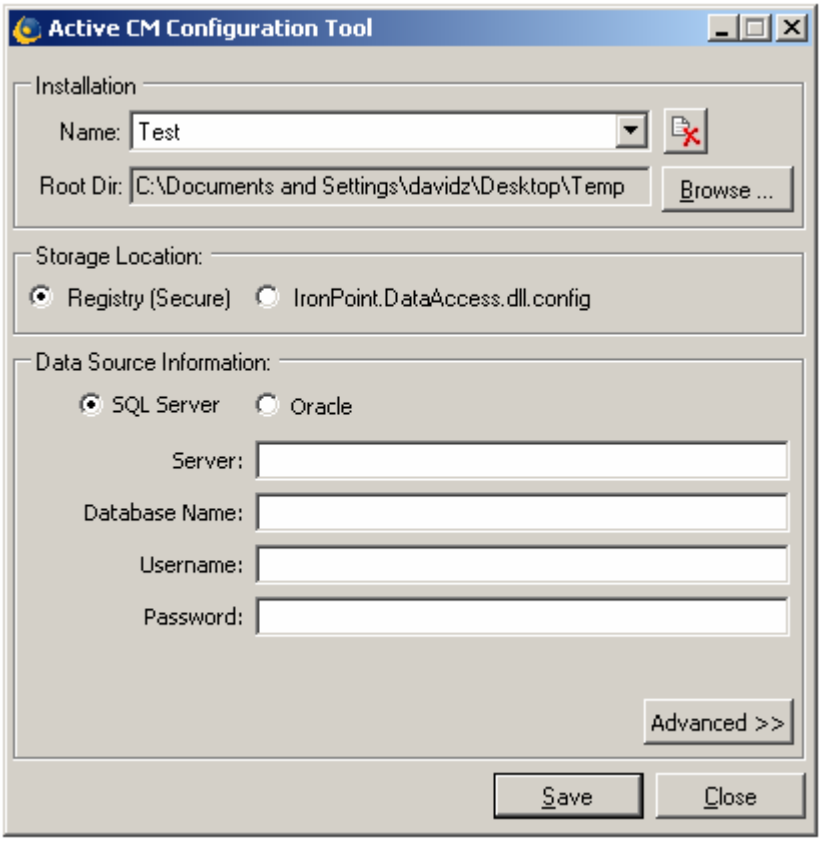
b. Initial Catalog. This is the database name within the specified SQL Server instance. **This can be excluded if you are using an Oracle data source.**

c. User ID. This is the user that the Active CM will use to connect to SQL Server or the Oracle data source.

d. Password. This is the password used to connect to SQL Server or the Oracle data source.

8. You can leave the **Data Ops Directory** field blank.
9. In the **Data Ops File Mask** field:
   a. Type "SqlClient.CM.Commands*.config" if you are using a SQL database.
   b. Type "Oracle.CM.Commands*.config" if you are using an Oracle database.
10. If there are any parameter name prefixes associated with the SQL files, type the prefix in the **Param Name Prefix** field, otherwise you can leave it blank.
11. Click **Save** to write the changes to the registry and remove the connection information from the **IronPoint.DataAccess.dll.config** file.

Note The "**Simple <<**" button allows you to revert to a simpler screen where you can simply enter the name of your database server, the name of your database and your username and password.



License Files

License Files Installation

You will receive a separate .zip file of standard Active CM licenses (e.g. Licenses_standard.zip)

To add these files to your Active CM installation you need to do the following:

- Unzip the license file.
- Copy all files (.lic format) into the **bin** folder of your Active CM.

File System Data Storage ("/Sites" folder)

In addition to storing data in a Database (SQL Server or Oracle), the Active CM also stores data on a NTFS Volume in the **Shared Folder** (by default 'Sites'). The Shared Folder contains all data that is not stored in the database.

Understanding Shared Folders

The contents of the Shared Folder include:

- One folder for each site, denoted by Site ID (ex. C:\inetpub\cmsroot\Sites\3).
- The current templates design package and archived packages used for each site (ex. C:\inetpub\cmsroot\Sites\3\Templates).
- A Digital Assets folder - that contains all documents, images, pdfs, etc that are managed by the Active CM (ex. C:\inetpub\cmsroot\Sites\DigitalAssets).
- HTML files generated for Search functionality (ex. C:\inetpub\cmsroot\Sites\3\SearchSite)
- HTML files generated for Static File Exports (ex. C:\inetpub\cmsroot\Sites\3\HTMLSite)

If the Shared Folder is located within the IIS WebSite or Virtual Directory it is HTTP accessible, and therefore all content can be accessed with a browser. If desired, the Active CM can be configured to secure content contained in the Shared Folder. This is accomplished by separating the Shared Folder into a Private Shared Folder and a Public Shared Folder. The Private Shared Folder is stored on an NTFS Volume that is not http accessible, while the Public Shared Folder is stored within the IIS WebSite or Virtual Directory.

Storing the Shared Storage (/Sites folder) on a separate server

You can set up ACM to store the Sites folder on a separate server. This is useful if you want to keep all the files related to your ACM content on a File Server or NAS which is regular backed up in case of a failure. This setup is also required as a central storage location for a multi-server setup. But for typical ACM setups, you can skip this section.

1. Create a Virtual Directory pointing to the physical path of your Shared Sites folder
    a. Open IIS
    b. Right-click on your ACM site
    c. New
    d. New Virtual Directory
    e. Name this "SharedSites" (You can use any name as long as you update your web.config file appropriately in step 3)
    f. Point the Virtual Directory to the root of your ACM site

g. In the "Allow the following permissions" window, select "Read" only
h. Finish creating your virtual directory
i. Right-click on the Virtual Directory
j. Properties
k. Virtual Directory tab
l. Select "A share located on another computer"
m. Specify your UNC path in the "Network directory:" field
n. Click on Connect As
o. Type in the username and password of a user which has read and write access to this share
p. OK
q. Select the Virtual Directory you just created in the left pane of IIS to ensure that you can see all the contents of the Virtual Directory in the right-pane

2. Update the web.config file to tell ACM which physical path to look for your Shared Sites folder ex. <add key="PhysicalPathToSharedFolder" value="\\FileServer\ShareName" />

3. Update the web.config file to tell ACM how to access this shared storage from the web (through the virtual directory you created in step 1). <add key="VirtualPathToSharedFolder" value="/SharedSites" />

4. Create a domain user which has read/write access to the file share, and has access to impersonate the ACM application.
   a. Create a domain user NOTE: do not give this user Administrator privileges to the ACM Web server
   b. Enable impersonation for your site. In the web.config file, add the following to the <system.web> section:

```
<identity impersonate="true"
   userName="DOMAIN\UserAccount"
   password="Password" />
```

   c. Enable the domain user to run as an IIS Application:
   Add the user to the "IIS_WPG" group in the domain (more information here: http://technet.microsoft.com/en-us/library/cc739233%28WS.10%29.aspx

Securing Access to Shared Folder Storage from the Web

If you need to secure the digital assets you're uploading to your site, as well as the files in your templates, follow the steps below. This type of setup is usually used in Intranets which may contain sensitive information which you do not want someone browsing to see. For most publically accessible sites, you may not need to set this up if all of your content is publically accessible anyways.

Content Access Control

Generally, websites are in the public domain and controlling access to files is not much of a concern. However, with the Active CM it is easy to create Intranets or Extranets based on personalized content. Content that is personalized is restricted to

a specific group of logged in users. Within this personalized content you may have confidential content, images or links to sensitive documents. Personalized content can not be accessed by a user or malicious person unless they have the proper username and password. Further, passwords are encrypted and stored in the database.

If you require the highest level of security, you need to use the Active CM features that allow you to configure Private and Public shared folders. It is important to understand that some of the files that are used to generate personalized pages are stored within an http accessible folder by default.
The content of each page is contained within the Search Site.

The physical documents that are uploaded into the Digital Asset Manger have their file names rewritten as a GUID but are stored within the Digital Asset directory.

Previous versions of your site designs are in the Archive Directory.

To configure the Shared Folder paths, you adjust settings in the web.config document that is located in the home directory. It is important that directory names and paths are accurately defined in the Web.config file.

Shared Folder (Http accessible)

There are two web.config keys that control the location of the Shared Folder. Remember that the Shared Folder is http accessible and that all items stored in this folder can be accessed by a web browser.

```
<add key="PhysicalPathToSharedFolder" value="Sites" />
```

By default in the Active CM installation the Http accessible folder is named ‘Sites’; however, it can be whatever you like as long as it is defined in the key noted above.

```
<add key="VirtualPathToSharedFolder" value="/" />
```

If a virtual directory is used to define the path to the Public Shared Folder, make sure include it in this key. If a WebSite is used, the value should be a single forward slash.

Private Shared Folder (non-Http accessible)

The Private Shared Folder contains all resources and data that you want the Active CM to manage but do not want users to have direct browser access to. For example, all Digital Assets can be secured such that access is controlled by the Active CM.

```
<add key="PhysicalPathToPrivateSharedFolder"
value="Sites" / >
```

This directory, if different than the Public Shared Folder, should not be within an Http accessible folder.

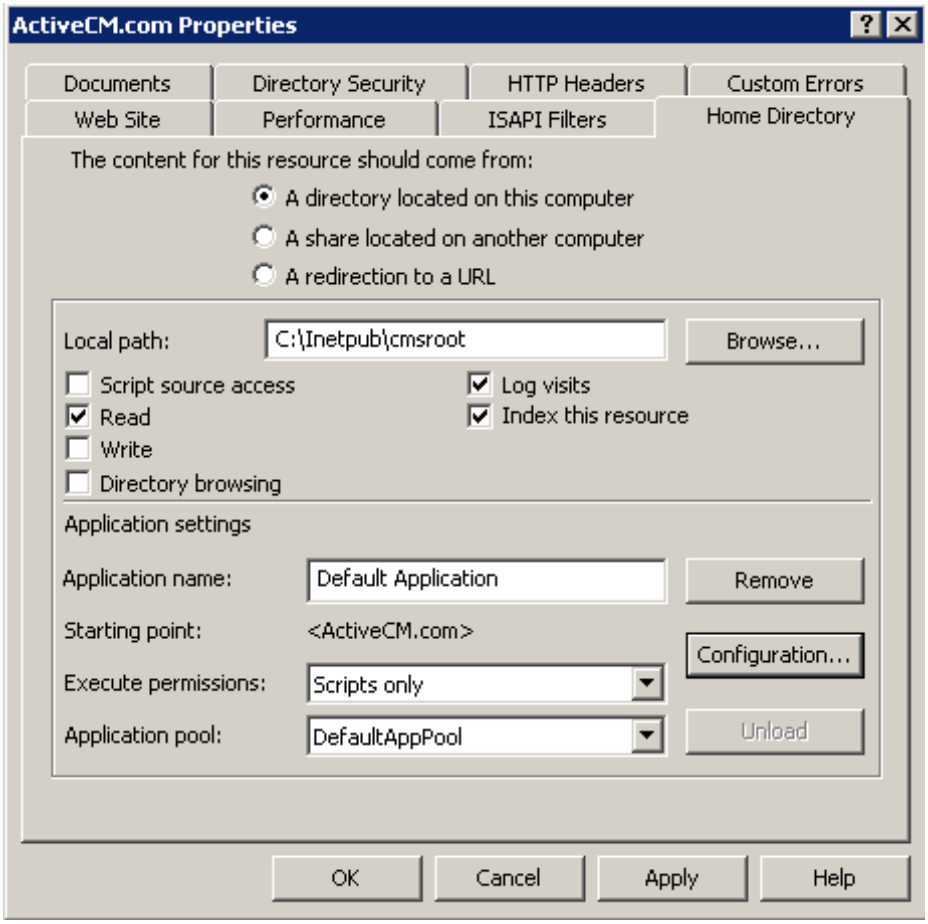
Note By default, the Active CM performs access checks to ensure that only authorized Digital Assets are returned to users. If you are configuring this instance of the Active CM for public sites, you can set the web.config key “DoAssetAuthorizationCheck” to False and save processing time.

Configuring IIS

Single Website Configuration

For a basic installation of the Active CM, IIS requires very little setup. Use the default configuration to create a website and set the home directory to point to the Active CM application files directory.

Note The Active CM application files directory can be placed on any local drive and path, however it is usually C:\Inetpub\cmsroot\.



Multiple Website Configuration

If you wish to use different URLs or Domains to access the individual Sites created in the Active CM application, you must use a customized default.aspx file and either multiple IIS websites or Host Headers. If you are going to use different websites, make sure that each one has its virtual directory pointing to the root of the Active CM installation. See below for details on Host Headers and the default.aspx file.

Host Headers

Microsoft Internet Information Services (IIS) permits you to map multiple Web sites with the same port number to a single IP address by using a feature called Host Header Names. By assigning a unique host header name to each Web site, this feature permits you to map more than one Web site to an IP address.

Note For more information, see Microsoft Knowledgebase article: 324287 - How to use host header names to configure multiple Web sites in Internet Information Services.

Configure Web Sites Using Host Header Names

To configure Web sites by using the Host Header Names feature, follow these steps:

1. Click **Start**, point to **Administrative Tools**, and then click **Internet Information Services.**
2. Expand *** server name** (where server name is the name of the server), and then expand **Web Sites**.
3. Right-click the Web site that you want, and then click **Properties**. The **Web site Properties** dialog box appears.
4. Click the **Web Site** tab, and then in the **IP Address** list, select the IP address that you want assigned to this Web site.
5. Click **Advanced**.
6. Under **Multiple identities for this Web Site**, click the IP address, and then click **Add**. The **Advanced Web Site Identification** dialog box appears.
7. In the **Host Header Name** box, type the host header that you want. For example, type www.example1.com. Add the port number, select the IP address in the list, and then click **OK**.
8. In the **Advanced Multiple Web Site Configuration** dialog box, click **OK**.
9. In the **Web site name Properties** dialog box, click **OK**.
   You return to the IIS window.
10. Right-click the next Web site that you want, and then click **Properties**.
11. In the **IP Address** list, select the same IP address that you selected in step 4, and then click Advanced.
12. Under **Multiple identities for this Web Site**, click the IP address, and then click **Edit**. The **Advanced Web Site Identification** dialog box appears.
13. In the **Host Header Name** box, type a unique host header for this Web site. For example, type www.example2.com, add the port number, select the IP address in the list, and then click **OK**.
14. In the **Advanced Multiple Web Site Configuration** dialog box, click **OK**.
15. In the **Web site name Properties** dialog box, click OK. You return to the IIS window.

16. Repeat steps 10 through 15 for each Web site that you want hosted on this IP address.
17. Register the host header names with the appropriate name resolution system -- for example, a Domain Name System (DNS) server or, in the case of a small network, a Hosts file.
18. The Web sites are now configured to accept incoming Web requests, based on their host header names.

Default.aspx purpose and locations

As noted in the beginning of this section, one instance or installation of the Active CM software is capable of hosting multiple Sites. Each site has its own unique Site ID. In the root directory of the Active CM installation there is a default.aspx file. This file is used to redirect IIS to the correct Site ID. For example, the following is what is included in this file by default.

```
<% Response.Redirect("~/site3.aspx") %>
```

If www.mysite.com was typed into web browser, the following would be returned back into the address bar: http://www.mysite.com/site3.aspx

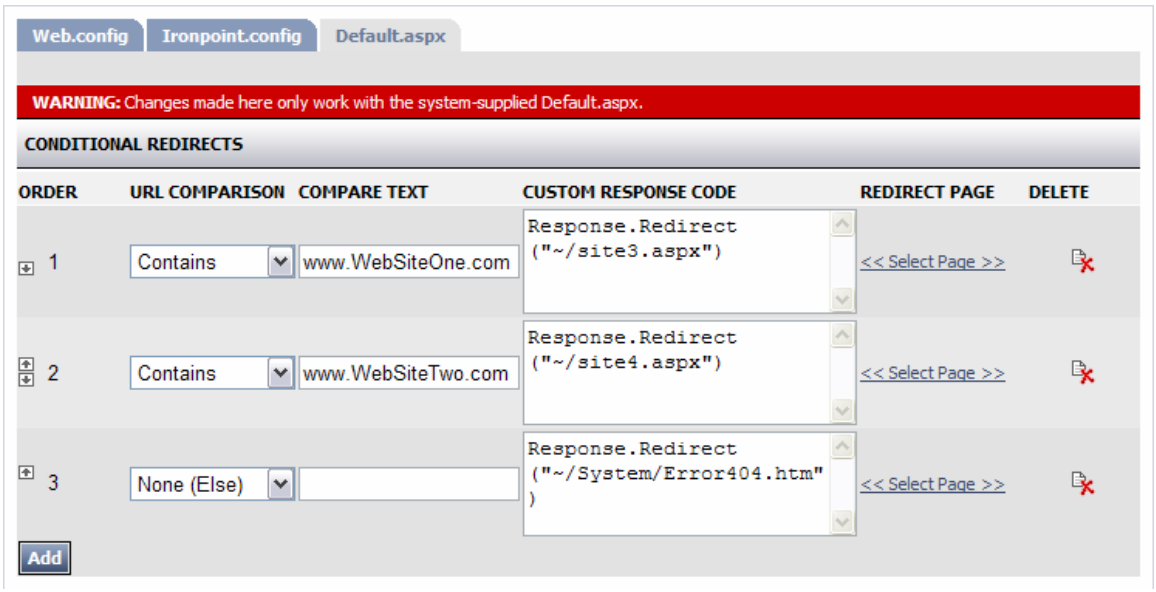
If there are two URLs or domains that are going to be used to access two Active CM websites, the default.aspx file will need something similar to the following VB.NET code in it to recognize which address is to be directed to which Site ID.

```
<%
dim strURL as string = UCase(Request.URL.tostring)

If InStr(strURL, UCase("www.WebSiteOne.com")) > 0 then
Response.Redirect("~/site3.aspx")

elseIf InStr(strURL, UCase("www.WebSiteTwo.com")) > 0 then
Response.Redirect("~/site4.aspx")

else
Response.Redirect("~/System/Error404.htm")

end if
%>
```

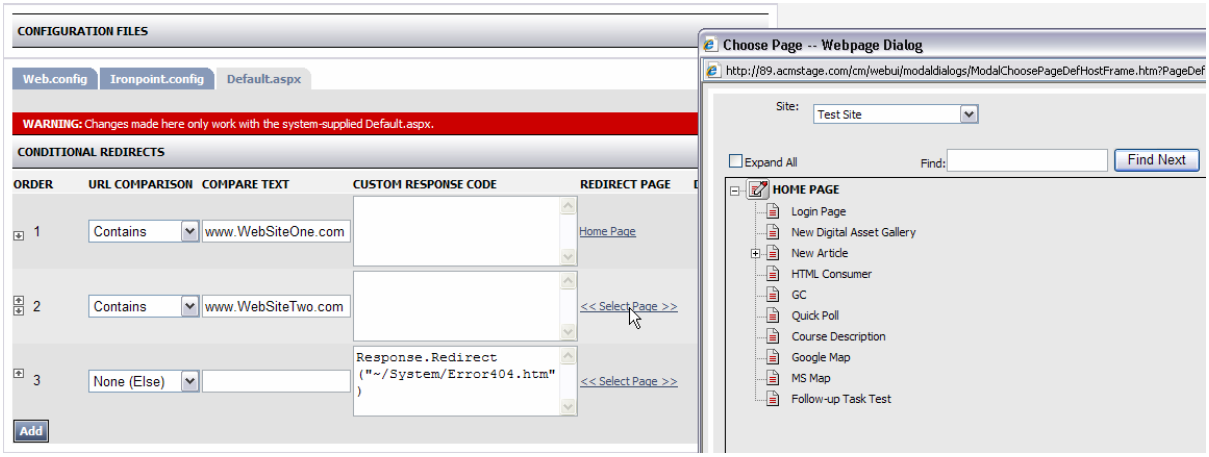
In the above code snippet the first website www.WebSiteOne.com is redirected to Site ID equal to 3. The website URL and the Site ID can be changed as appropriate. There are many way to achieve this using VB.NET code, the above is simply one example.

Note Do not add directories onto the address path. They are not needed.

The Default.aspx can also be managed directly from the Admin Center of ACM as well. Browse to the Configuration Files screen and select the Default.aspx tab. Note the warning that these setting will only work if you are using the system default.aspx, not the ironpointdefault.aspx. The example below will have the same result as editing the default.aspx file directly.



Additionally, if you know what page you would like the **www.WebSiteOne.com** to load, you can exclude the custom response code and simply select the page in the redirect page column.



Application Pool Setup

In IIS 6.0, application pools are a method of isolation between Web sites and applications. An application pool can contain one or more applications. A worker process services requests for the Web sites and applications in an application pool. The more application pools you have, the more you isolate each site from others. To help create a very secure and reliable environment, you can place each site in its own application pool and each worker process could run as a unique identity. Errors in one application pool cannot cause another application pool, or the server itself, to fail.

It is important to create separate Application Pool for each Active CM instance. To do it you will need to perform the following steps:

1. Start IIS
2. Expand **COMPUTER (local computer)**
3. Right-click on **Application Pools**

4. Select **New, Application Pool**
5. Label it (for example you could use your site name e.g. MySite App. Pool)
6. Leave all the default selections
7. Click **OK**
8. Right-click on newly created application pool (e.g. MySite App. Pool)
9. Select **Properties**
10. Under the **Recycling** tab clear the check mark for "Recycle worker process (in minutes):
11. Check "Recycle worker process at the following times:
12. Click on Add…
13. Type 01:00 (that is 1 AM)
14. Click OK
15. Click on **Health** tab
16. Remove check mark before Enable Rapid-fail Protection
17. Click OK
18. If your site is already setup in IIS, right-click on it under Web Sites
19. Select Properties
20. Click on **Home Directory** tab
21. Under **Application Pool** select application pool you created
22. Click **OK**

web.config Configuration

The Active CM has certain specific application settings that are controlled through the web.config file located in the root of the application files directory. The web.config variables can be edited using notepad or through the **Web.config** tab found in the Administration Centre > Configuration Files.

The following values can be configured:

KEY	DESCRIPTION
PhysicalPathToSharedFolder	The physical path to the shared folder. This is the folder that contains all data that is not contained in the database. For example: C:\Inetpub\cmsroot\Sites
VirtualPathToSharedFolder	The virtual path to the shared folder. For example: /cmsroot/Sites
PhysicalPathToPrivateSharedFolder	The physical path to the private shared folder. This is the folder that contains all site resources that must be secured. If the Active CM is configured to secure all access to digital assets, this folder must be located in a non-http accessible location. For example: C:\Inetpub\cmsPrivate\.
CustomErrors	Determines if the Active CM should show detailed error messages to users. Default value: RemoteOnly
Compilation	Determines if the application is set to ASP.NET "debug" mode. Default value: false:
Trace	Determines if the application should use ASP.NET tracing.
SessionState	Determines configuration settings for session management. Can be in-process for single application servers, or out-of-process for webfarms.
Httpruntime	Set max request length and request timeout.
AsyncKeepAliveEnabled	False
AsyncJobServerEnabled	True
AsyncJobInterval	1
AsyncPollingInterval	1
AsyncJobRestartInterval	60
AsyncJobRemovalInterval	1440
SMTP_ServerHost	The SMTP server host to use for sending email from the system. For example: smtp.fusemail.com. Default value: none.
SMTP_ServerPort	The SMTP server port to use for sending email. Default value: 2500
SMTP_UserName	The SMTP user name to use to authenticate against the SMTP server. Default value: none.

SMTP_Password	The SMTP password to use to authenticate against the SMTP server. Default value: none.
SMTP_SAS	2
SMTP_SSLProtocol	0 - None, 240 - Default, 12 - SSL2, 48 - SSL3, 192 - TLS
DoDatabaseValidation	Determines if the application should perform a database validation check at application start. Default value: False
DoAssetAuthorizationCheck	Determines if the application should perform Digital Asset authorization checks. This setting is only required if digital assets access must be controlled. In a public Internet deployment, this value should be False. Default value: True
UseRadEditor	Determines if the application with use the Telerik R.a.d. Editor for WYSYWIG content. If False, a normal text editor is used. Default value: True
RequestHistory	Sets the number of requests to record in logging: Default value: 5
EnableCaching	Determines if the application should enable caching. Default value: True
EnableMultiServerCache	Determines if the application instance will support a multi-server cache configuration. This setting is only required if multiple application servers are used. Default value: False
ShowQuickEditIcon	Determines if the application will show the QuickEdit icon when editing content. Default value: True
UseWindowsEventLog	Provides for the option to record Active CM system events in the Windows Event Log. This can make supporting multiple instances of the Active CM application on a single server more convenient. Default value: False
SessionTimeoutAnonymous	Provides the option to set the anonymous session timeout value in minutes. Default value: 60
SessionTimeoutAuthenticated	Provides the option to set the authenticated session timeout value in minutes. Default value: 60
SessionTimeoutAnonymousKeepAliveInterval	Provides the option to send an Ajax call to keep the anonymous session alive. If this value is less than the SessionTimeoutAnonymous value, the session will be preserved until the user closes their browser or browses off the site. Default value: 30
SessionTimeoutAuthenticatedKeepAliveInterval	Provides the option to send an Ajax call to keep the authenticated session alive. If this value is less than the SessionTimeoutAuthenticated value, the session will be preserved until the user closes their browser or browses off the site. Default value: 30

Note You can encrypt the information within the web.config file as it contains password information for your SMTP server. To do so, go to System Information in the Admin Center and expand the WEB.CONFIG SETTINGS section. You will see the current web.config settings and two buttons to encrypt or decrypt the web.config. If you encrypt the web.config file, you can only manage its content through the

Web.config tab found in the Configuration Files section of the Admin Center.

Domain Resolution

As of ACM version 9.0, the Ironpoint.config file is now stored in the database. If you have upgraded to 10.0 from a version prior to 9.0, please refer to the 10.0 Upgrade Guide for instructions on updating the database with the Ironpoint.config settings.

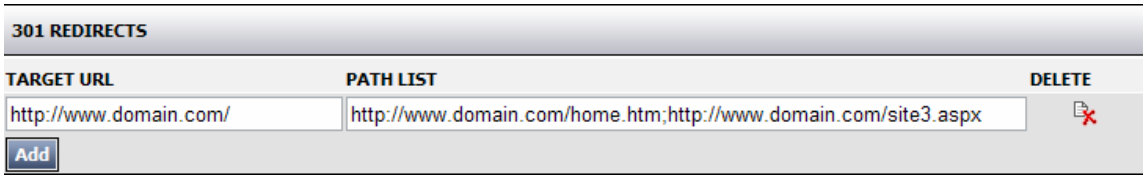
ACM allows you to create Authority Redirects, 301 Redirects and FQDN Mappings. These settings can be configured in the Ironpoint.config tab located in the Configuration Files section of the Admin Center. These 3 settings take priority in the order mentioned. Examples of the new settings are below:

Authority Redirects

Multiple authority redirects can be configured and each one takes an 'authorityIn' and a 'targetAuthority'. Upon receiving a request, ACM will check the authority of the request URL to see if a redirect has been configured. If a match is found, it will rebuild the URL and do a 301 redirect. Using the example below, 'http://domain.com' would be 301 redirected to 'http://www.domain.com'.

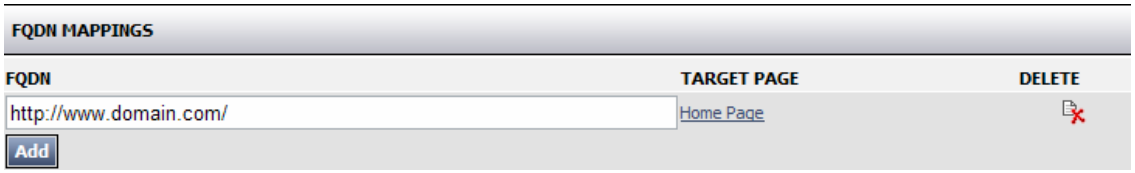
301 Redirects

Multiple 301 redirects can be configured, each taking a 'targetURL' and defining multiple semicolon delimited URLs that will redirect to the target. Upon receiving a request, ACM will see if the requested URL matches any of the 'Redirect301' path list URLs and perform a 301 redirect to the target URL. Using the example below, both 'http://www.domain.com/home.htm' and 'http://www.domain.com/site3.aspx' will be 301 redirected to 'http://www.domain.com/'.



FQDN Mapping

Multiple FQDN mappings can be configured, each requiring an 'FQDN' and a target page. This will allow a user to click or type in the 'FQDN' and see that remain in the address bar when the actual page is loaded. Upon receiving a request that contains only the authority or the authority and the virtual root, ACM will check if the request URL matches with one of the configured FQDN settings in the list of mappings. If one is found, much like a friendly URL mapping, ACM will rewrite the path server side to the specified target page. Using the example below, requesting (or getting 301 redirected to) 'http://www.domain.com/' will load the Home page while remaining 'http://www.domain.com/' on the address bar.



Directory Security for ASP.NET

Note As of Version 10.0 you must use ASP.NET 3.5 when installing the Active Content Manager.

The Active CM application, via the ASP.Net worker process, needs file permissions to create and change files in the Shared Folder (both Private and Public).

If you are using Windows 2003, the ASP.Net worker process runs under the Windows account called NETWORK SERVICE.

Note It is also possible to configure a new Windows account and instruct ASP.Net to use this account. This is called impersonation and may result in enhanced security.

To adjust the file permissions to the Shared Folder, ensure that the NETWORK SERVICE or ASPNET user account has modify permissions on the Shared Folder. Note that you may need to check "Replace permission entries on all child objects with entries show here that apply to child objects".

Here are the folders that need proper access rights set for NETWORK SERVICE and/or ASPNET:

- **\CM\WebUI**
- **Sites**
- **System**
- **Web.config** (only if you want to allow modification of these settings through the application via Admin Center > Configuration Files > Web.config tab)

1. Right-click each folder
2. Select Properties
3. Click on Security tab
4. Click on Add
5. Type NETWORK SERVICE
6. Click on Location button
7. Make sure to select local server. Do not select domain name
8. Click on Check Name button
9. Click on OK
10. Select Modify under Permissions for NETWORK SERVICE
11. Click on OK
12. Repeat these steps for all three folders and Web.config file.

Note On a Windows 2000 Server, the local ASP.NET also needs modify permissions on the same folders

Shared Folder Contents

This section describes the location and security of files that are generated or uploaded by the users of the Active CM.

These files include:

1. Design Packages
    - CSS
    - Templates
    - JavaScript Files
    - Images that are displayed in the templates
    - Configuration files
2. Documents, images, PDFs or the like that are uploaded into the Digital Asset Manager.
3. HTML documents that are generated by the application.
    - SearchSite folder, containing page search versions
    - HTMLSites folder, the HTML File Export staging folder
4. Previous versions of Design Packages

Indexing Service Setup

The following steps are necessary to configure the Active CM and server to support the Search page type. Microsoft's Index Server is the backbone on which the Search page type relies. The indexing service is uses the Search Site files generated as part of the Active CM export process. Regardless of whether or not your site is dynamic or static, export to the Search Site folder is necessary for search indexing to work.

How the Search Page works

The Active CM supports searching of content by creating a Search Site. Using Microsoft's Index Server, search results are built against this Search Site.

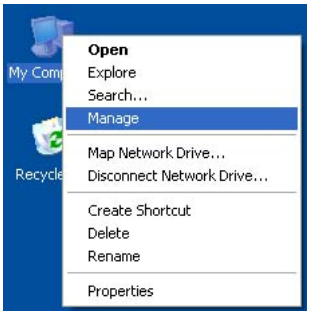
Note Before the Search Page will return results you must: 1) Properly configure Index Server, 2) Configure the Active CM for Export (so that the Search Site is created and maintained).

Indexing Service Configuration

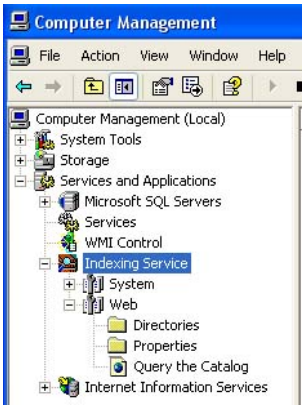
1. Open the Computer Management Microsoft Management Console (MMC).

   The Computer Management MMC is available by default in your Administrative Tools folder in your Start menu:

   Alternately, right clicking on My Computer and selecting Manage from the context pop-up will also open the Computer Management MMC.

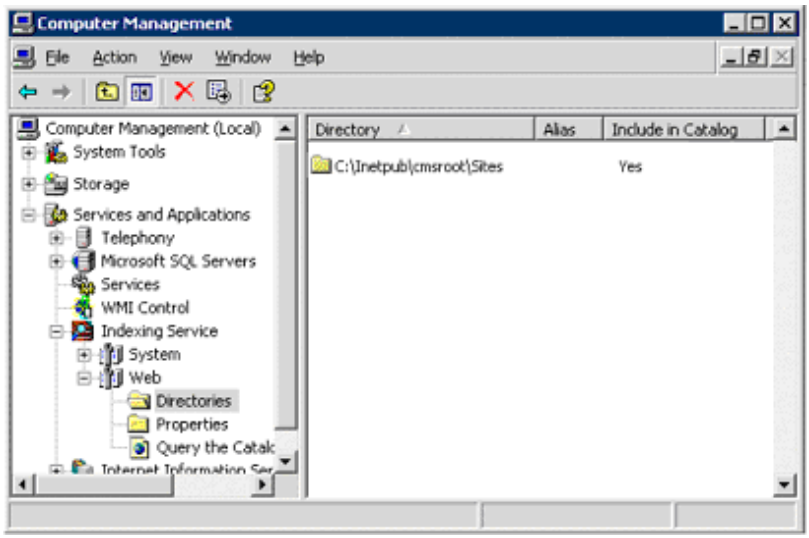


2. Inside the Computer Management MMC, expand Services and Applications, expand Indexing Service, and finally, expand the "Web" catalog.

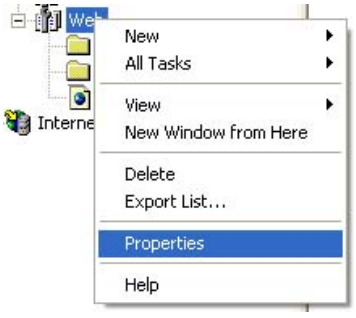


Note If the web catalog doesn't exist in your installation of Indexing Services, IIS may not be properly set up. IIS automatically generates the web catalog when instantiated. If you still do not have a Web catalog after IIS has been set up, please refer to Index Server documentation in order to create a new "Web" catalog.

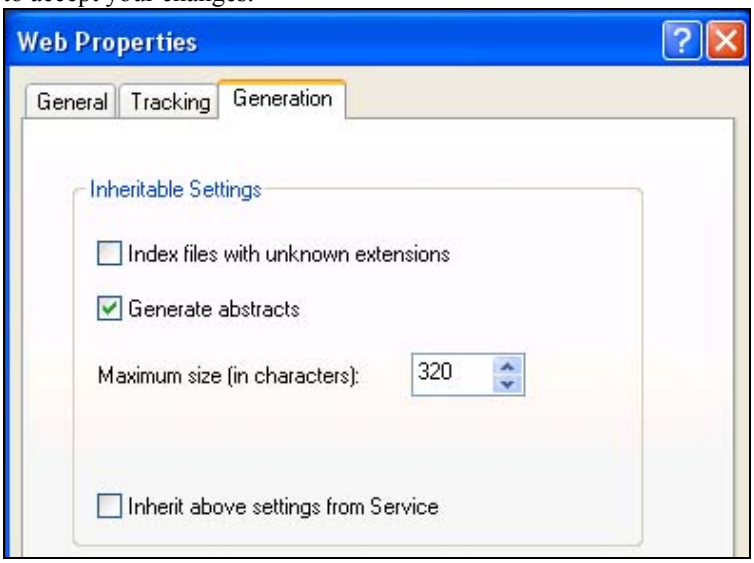
3. Select the Directories folder to make sure there is an existing CMS Root\sites folder in the right hand frame. There should be a folder by default as long as enabled the "index this resource" option in IIS for this website.



4. Right click on the Web catalog in the left frame of the Computer Management MMC and select Properties (this is not the Properties sub-folder).



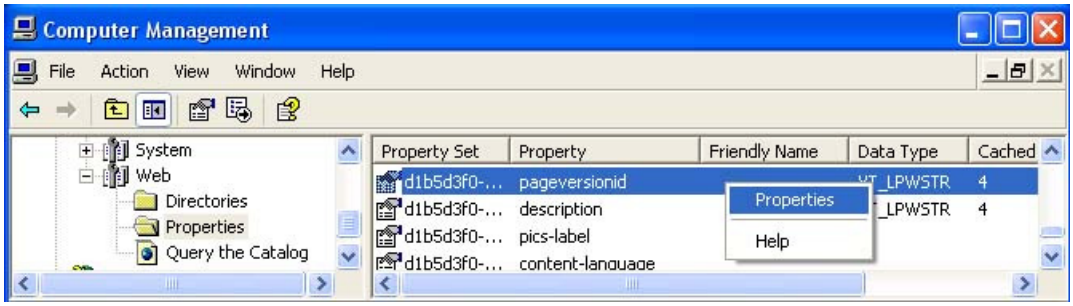
5. On the Generation tab, ensure that the “Inherit above settings from Service” checkbox is not selected. Ensure that the “Generate abstracts” checkbox is selected. If you wish for all content of all types to be searched, regardless of whether or not they are recognized by the system, ensure that the “Index files with unknown extensions” checkbox is selected; otherwise, ensure that it is not selected. After all checkboxes are confirmed, click the OK button to accept your changes.



6. Open the Properties subfolder of the Web catalog.

7. Open the Properties of the “pageversionid” property.

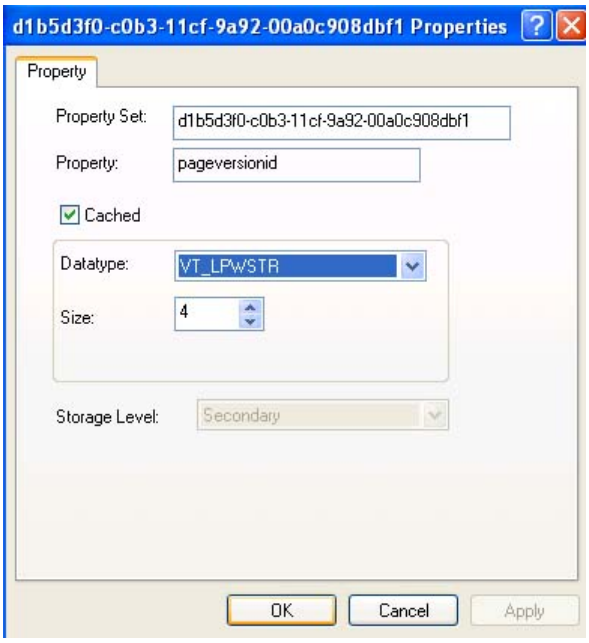
   Scroll through the list of Properties until you locate a property with the “Property” name of “pageversionid”. Right click on the property and select “Properties” from the context menu that appears.



Note If you are unable to find the “pageversionid” property (or any of the properties identified in step 8 below), it means that the Active CM sub-tags haven’t been registered in Indexing Services. You should initiate a full rescan of the CMS root folder in your Web catalog in order to ensure that the sub-tags are properly

registered. Refer to a later section in this Chapter for information on how to Rescan Index Server. If the "pageversionid" property is still not displayed after the re-scan, right click on the Properties folder and select All Tasks > Refresh List.

8. With the "pageversionid" property Properties dialog displayed, click on the "Cached" checkbox. By default, the "Datatype" drop down box should be set to "VT_LPWSTR"; if it isn't, select "VT_LPWSTR" from the list. (The default size should be set to "4". This default size is fine and can be left as is.)



Note After changing a property, you will be prompted with a "Pending Change" message box. This box warns you that the Property cache modification takes effect when Indexing Service is restarted. Only newly filtered documents will have the modified set of properties. A full rescan should be initiated to extract newly cached properties from already filtered documents.

Ignore this message for now. It informs you that when you're finished updating the properties, the system will need a restart and full rescan, as described in step 9.

9. Repeat steps 7 and 8 for the properties with Property names:
   a) "pageversionid"
   b) "locationline"
   c) "digitalassetid"
   d) "contenttype"
   e) "lastmodifieddatetime"
   f) "pageancestors"
   g) Any property sets with Friendly Name "Characterization".

Note Index Services has a property called "content-type" in addition to the Active CM "contenttype" property. Ensure that the non-hyphenated property is being cached.

10. After you have updated all properties for caching, restart Indexing Services and then perform a full rescan on your CMS root folder.

    The process for performing a full rescan and/or restarting Index Services is detailed later in this chapter.

Rescanning Index Server

Whenever there are major changes to the search criteria, properties, or the site in general, it's often a good idea to force a re-scan of the CMS root folder in the Web catalog of Index Server. Note that Index Server does automatically re-scan changes to content regularly, but the following procedure allows a user to force this re-scan to take place outside of the regular scanning process or for newly cached properties.

1. Open the Computer Management Microsoft Management Console (MMC). (Detailed instructions on how to do this can be found earlier in this chapter.)
2. Inside the Computer Management MMC, open the Indexing Service and navigate to the Web catalog.

   Click the plus sign next to Services and Applications to open the folder, click on the plus sign next to Indexing Service, and then click on the plus sign next to Web.

   If the Web catalog doesn't exist in your installation of Indexing Services, IIS may not be properly set up. IIS automatically generates the Web catalog when instantiated. If you still do not have a Web catalog after IIS has been set up, please refer to Index Server documentation in order to create a new "Web" catalog.
3. Open the Directories subfolder of the Web catalog and locate the CMS root directory, as defined inside IIS. This directory will be available if the "Index this Resource" checkbox is selected on the Home Directory tab in the IIS website properties.

   When the Active CM application is installed, IIS is configured to direct traffic appropriately through Active CM ASP.NET pages. When that folder is configured, either by default or explicit naming convention, the folder is added to Indexing Services. By clicking on the "Directories" folder on the left-hand side of the Computer Management MMC, the list of all valid folders appears on the right-hand side.

   If the CMS root folder doesn't exist in your installation of Indexing Services, IIS may be properly set up. IIS automatically adds the CMS root folder to Indexing Services when instantiated. If you still do not have a CMS root folder after IIS has been set up, please contact Active CM support.
4. Right click on the CMS root directory, select the "All Tasks" fly-out from the context menu, and select the "Rescan (Full)" option.

   After you select to re-scan a given directory, the Indexing Service starts to perform the scan invisibly. Selecting the main "Indexing Service" item in the Computer Management MMC displays all catalogs for the current service. After selecting to re-scan a directory, the Web catalog should appear with a status of

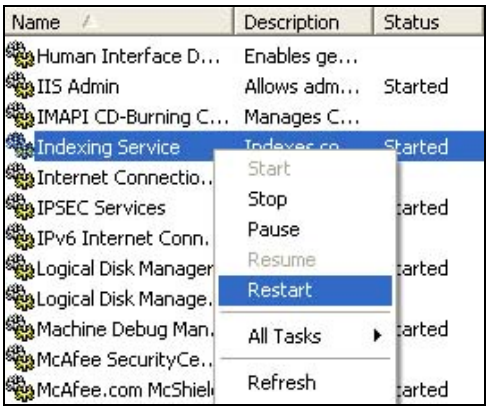
"Scanning". When the "Scanning" status is removed, Indexing Service has finished your new scan.

Note The scan may take a while depending on how much data is being scanned. The scanning speed can be increased by tuning the performance (Right-click on Index Service, select All Tasks > Tune Performance) however increasing indexing performance will allocate more CPU resources to the indexing process and may impact CPU Utilization.

Restarting Index Server

The Index Server should rarely need to be restarted. During configuration, however, Index Server will need to be restarted after configuring property caching and after configuring Adobe iFilter (if desired).

From Computer Management, right click on Indexing Service and select Restart from the context menu.



Installing Adobe's iFilter

To be able to search PDF files, Adobe has provided an enhancement for Microsoft's Indexing Services. This enhancement, called iFilter, is available for download from Adobe's website.

To install Adobe iFilter:

1. Download iFilter from www.adobe.com. The iFilter now comes with Adobe Reader 8 and you can't download it by itself, you need to install Reader to get the iFilter.
2. Stop all appropriate clients and stop the index server service.
3. Uninstall any previous version of Adobe iFilter
4. Double click the downloaded file and follow the on-screen instructions
5. After the installation completes, start the Index Server service and all appropriate clients.
6. Restart and rescan Index Services.

Making pages available for indexing

After you setup Indexing service and properties, you need to configure site properties, export the site and let the Indexing server index those pages.

1. Browse to your new website in a Web Browser.
2. Select the Login page
3. Login to ACM using the administrator password
4. Click on the System button on the Admin Toolbar
5. Select Sites to open the Site Manager.
6. Click on site for which you wish to you want to provide search functionality
7. Enter correct FQDN (Fully Qualified Domain Name) under Dynamic Properties/External and Internal FQDN (e.g. http://www.mysite.com)
8. Save changes
9. Logout
10. Recycle application pool for site you're working on
11. Login
12. Click on Admin Center
13. Select Export
14. Under Unscheduled Export click on the link next to "Page to Export". This will open the "Choose page" window
15. Click on the Top (parent) page if it is not already highlighted
16. Click OK
17. Select the "Include children" check box

18. Click on Export Now
19. Click on System, System Information
20. Monitor status under Job Queue Status
21. You can refresh this page to follow get job status
22. Wait until all pages are exported
23. Start another web browser
24. Go to your web site
25. Test search

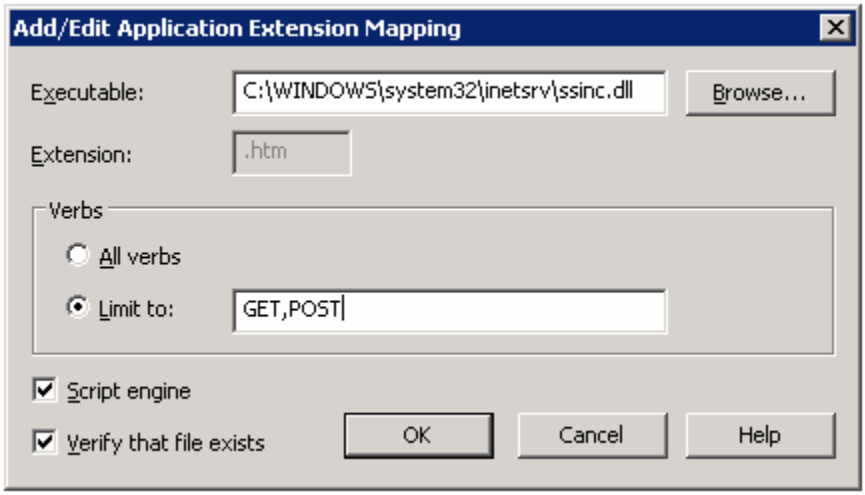
Enabling SSI

Statically Published Sites

The Active CM software is capable of generating Static HTML pages. These static pages are built by IIS or Apache out of several parts. For example, the menu, the template and the content are all separate pages. The web server glues these pieces together by using server side include.

To **add** server side includes for an IIS run HTML website, follow these steps.

1. Click **Start**, point to **Administrative Tools**, and then click **Internet Information Services.**
2. Expand *** server name** (where server name is the name of the server), and then expand **Web Sites**.
3. Right-click the Web site that you want, and then click **Properties**.
   The **Web site name Properties** dialog box appears (where Web site name is the name of the Web site that you selected).
4. Click the **Home Directory** tab, and then click **Configuration**….
5. Click the **mapping** tab, and then scroll the list and see if '.htm' and '.html' are included in the list. If they are not included, add them following the next steps.
6. Click **add**… **or edit**…
7. Configure the application extension mapping as per the following figure.



8. Click **OK**
9. Redo steps 5 to 8 for both '.htm' and '.html'
10. Click **OK**

To **enable** server side includes for an HTML website, follow these steps.

1. Click **Start**, point to **Administrative Tools**, and then click **Internet Information Services.**
2. Expand **server name** (where server name is the name of the server), and then click on **Web Service Extensions**.
3. Click **Server Side Include**
4. Click **Enable**, then close IIS

Multiple Application Server Configuration Files

This discusses how to set up multi-server ASP.NET Web Applications and Web services. For most uses of ASP.NET, a single server can handle all requests in a timely manner. However, many environments must deploy multiple servers to handle consistently high volumes of traffic, to support processor-intensive applications, to respond to sudden bursts in traffic, or to meet redundancy requirements.

In the simplest form, you can deploy Web pages that consist only of static HTML pages and images in a multi-server configuration by copying the files to multiple Web servers and then configuring a load balancing mechanism to distribute requests between the Web servers.

As the Web site complexity increases, the difficulty of synchronizing files and configurations between the servers also increases. Dynamic sites require multiple servers to have access to a single database and to share state information among them.

It is recommended that there are a minimum of three servers in a multi-server configuration: The State Server and two Web Servers. The File Server can be a separate server as well or can be combined with the State Server.

Note In a multi-server configuration, the state server is used to store the user sessions to enable the user to be served by any of the web servers. This is done to avoid forcing the load balancer to manage session affinity (same session always goes to the same web server). If you wish to set up load balancing with session affinity then you can use in-proc sessions, however this means that if a web server goes down, all sessions it was serving will be dropped. Using in-proc sessions also limits the load balancer's ability to redistribute the load for optimal performance as it cannot push requests to a different server as soon as the user has a session.

The state server will be a single point of failure but it is generally regarded as more reliable than having a single web server as the point of failure. The state server does not do much more than store key (session id) value (session content) pairs with expiration times and push them to/from the web servers so it is far less likely to fail than a more complex web server.

In order to configure multiple application servers, you need to finish the following tasks:

On the State Server:

1. Start and setup State Server service

On the File Server (could be the same as the State Server):

2. Setup an Impersonation account
3. Create and share Shared Sites folder

On each Web Server:

4. Setup IIS
5. Setup web.config

Setting up State Server Service

Follow these instructions on the server that will run the ASP.NET State Server. It is recommended that the service not be set up on a server that is operating as a web server.

1. Install .NET 3.5. This may be obtained from http://www.microsoft.com/downloads/details.aspx?familyid=AB99342F-5D1A-413D-8319-81DA479AB0D7&displaylang=en.
2. Configure the state server to allow connections from remote machines by setting the following registry key to 1: HKEY_LOCAL_MACHINE\SYSTEM\CurrentControlSet\Services\aspnet_state\Parameters\AllowRemoteConnection
3. In the services management console, locate the service "ASP.NET State Service".
   a. Start the service
   b. Set the service startup type to automatic

Note By default, the State Service listens on port 42424. This can be changed in the registry at the following loction: HKEY_LOCAL_MACHINE\SYSTEM\CurrentControlSet\Services\aspnet_state\Parameters\Port

Setting up the ACM Cache Invalidation Service

Follow these instructions on the server that will run the ACM Cache Invalidation Service. The instructions use the location C:\ACMRoot for the ACM installation. It is recommended that the service not be set up on a server that is operating as a web server.

Note The CacheValidationService.lic license file is required in order to use the cache invalidation service. It must be placed in the /bin directory. Please contact your Active Network account representative for this license.

1. Install .NET 3.5. This may be obtained from http://www.microsoft.com/downloads/details.aspx?familyid=AB99342F-5D1A-413D-8319-81DA479AB0D7&displaylang=en.
2. Copy an ACM build to C:\ACMRoot. This must be the same version the web servers are running and must be updated when the web servers are updated.
3. Make sure the connection information specified in C:\ACMRoot\bin\IronPoint.DataAccess.dll.config is valid
4. Register the ACM Cache Invalidation Service
   a. Open a command prompt
   b. Run the following command:

```
C:\WINDOWS\Microsoft.NET\Framework\v2.0.50727\InstallUtil.exe /i C:\ACMRoot\bin\
CacheInvalidationService.exe
```

Note Even though ACM 10.0 requires .Net 3.5, the installutil.exe referenced above is still found in the .Net v2.0 directory. This is because .Net3.5 is only an extension of the .Net 2.0 runtime.

c. Verify that the final lines written to the console by InstallUtil are:

```
The Commit phase completed successfully.

The transacted install has completed.
```

d. If this is not the case contact ACM support for assistance

5. Open the services management console and locate the service "Active CM Cache Service"
   a. Start the service
   b. Set the service startup type to automatic
6. If a web server is available, verify that a System Event Log message exists saying that "The cache invalidation service has been started." in the Application log.

Note The Cache Validation Service listens on port 6642, and the web servers listen on 6643 for the cache invalidation service's messages. These cannot be changed so the appropriate ports must be opened in the Firewall.

Note If this is not a web server, no further modifications need to be made to the Ironpoint.config or Web.config files.

Creating Shared Sites folder

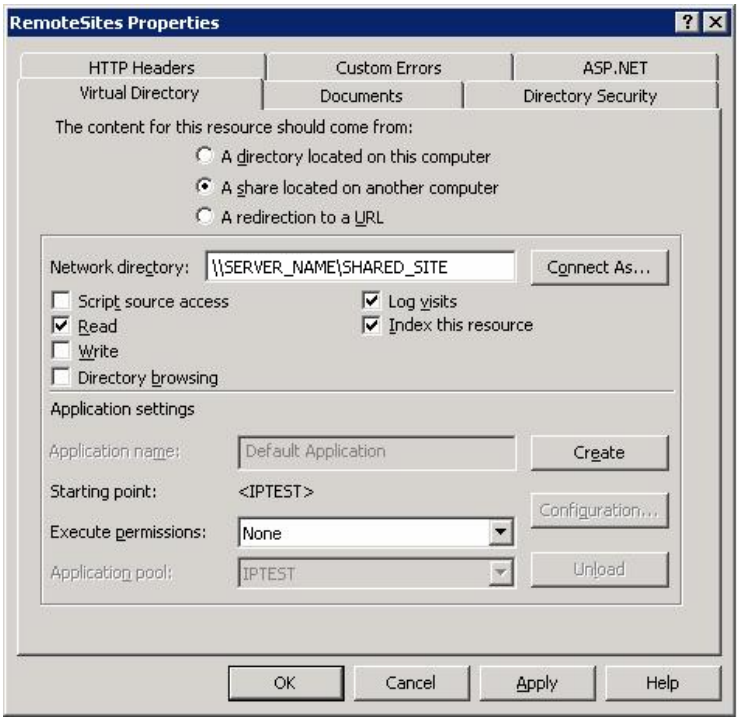
Now you need to create a folder in which you will store all data from the Sites folder. It would be ideal if this could be away from either Application server so that in case one of the servers needs to be shut down (redundancy), access to the Sites folder will not be affected. This server will be used to host a shared directory to allow a common set of digital assets and other files that are not stored in the database.

1. Create a UNC path accessible share.
2. Create an account that can add, remove, and modify folders and files from this directory. This account must be a domain user.
3. Move the Sites folder from one of the web servers to this share.

Setting up IIS

The following steps should be repeated on each of the web servers to be used in the multi-server configuration.

1. Start Internet Information Server Manager
2. Expand Web Sites
3. Expand web site you're working with
4. Right click on it to create new virtual folder (e.g. SHAREDSITES)
5. Type UNC path (e.g. \\SERVER_NAME\SHARED_SITE)



6. Click on "Connect as…"
7. Type user name (impersonation account) (e.g. DOMAIN_NAME\USER NAME)
8. Type password for this account.
9. Clear check mark before "Always use the authenticated user's credentials…"
10. Click OK button
11. Repeat these steps on all web servers

Setting up Web.Config

Follow these instructions on every ACM web server. These steps assume that the web server has been correctly configured for single-server ACM operation with the ACM files installed in C:\ACMRoot. The user specified for impersonation in Web.config must be a user who has access to the file server containing the Sites folder.

Open C:\ACMRoot\Web.config

1. In the <system.web> section, add the following line as the first item in this section:
   ```
   <identity impersonate="true"
   userName="DOMAIN\UserAccount" password="Password"
   />
   ```
   The username and password must be valid to access the SHARED_SITE folder on the file server.
2. Immediately after the `<system.web>` element specify the encryption key to use for ViewState:

```
<machineKey
validationKey="0123456789ABCDEF0123456789ABCDEF0123
456789ABCDEF"
decryptionKey="FEDCBA9876543210FEDCBA9876543210FEDC
BA9876543210"
validation="SHA1"
decryption="AES" />
```

The validationKey and decryptionKey should be set to 48 random hexadecimal characters – do NOT use the values shown above.

3. Find the sessionState Mode setting and change it from this:
```
<sessionState mode="InProc"
stateConnectionString="tcpip=127.0.0.1:42424"
sqlConnectionString="data source=127.0.0.1;user
id=sa;password=" cookieless="false" timeout="60" />
```
To this:
```
<sessionState mode="StateServer"
stateConnectionString="tcpip=XXX.XXX.XXX.XXX:42424"
sqlConnectionString="data source=127.0.0.1;user
id=sa;password=" cookieless="false" timeout="60" />
```
This will allow your session to persist across multiple servers (If the web server you're on restarts, it will not kick you off). The default is InProc which stores your session in memory. StateServer stores your session in the ASP.NET State Service. Other options SQLServer, but StateServer is much easier to set up so we recommend that method. The XXX.XXX.XXX.XXX is the IP address of the server on which the State Service is installed.

Note :42424 is the port number used by the ASP.NET State Service. Make sure this matches the port number configured during the State Service install

4. Enable multi-server cache by setting the following element to true:
```
<add key="EnableMultiServerCache" value="True" />
```
5. Change the PhysicalPathToSitesFolder to specify the remote share. For example, if the site folder was shared as SHARED_SITE from the machine SERVER_NAME:
```
<add key="PhysicalPathToSharedFolder"
value="\\SERVER_NAME\SHARED_SITE\" />
```
6. Change VirtualPathToSharedFolder to point to the RemoteSites virtual directory created in the previous section:
```
<add key="VirtualPathToSharedFolder"
value="/SharedSites" />
```
7. Locate the following entry and change the value to the IP address of the Cache Invalidation Service host.
```
<add key="CacheInvalidationServiceAddress" value=""
/>
```
8. Locate the following entry and change the value to the IP address of the web server that the Cache Invalidation Service can use to contact it. Note that this will be different on each web server in a cluster – do NOT specify the cluster IP.
```
<add key="CacheInvalidationWebServerAddress"
value="" />
```
9. If the site will use export, specify the same user used for impersonation (step 1) for asynchronous export authentication:
```
 <add key="AsyncExportUserName"
value="DOMAIN\UserAccount" />
<add key="AsyncExportPassword" value="Password" />
```

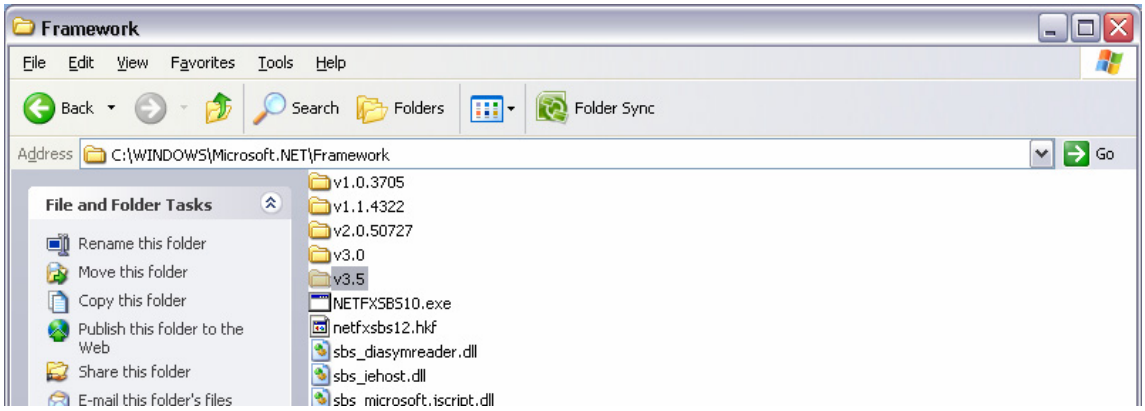
10. In Windows Explorer, locate: C:\WINDOWS\Microsoft.NET\Framework\v2.0.50727.
11. Grant Modify on the “Temporary ASP.NET Files” directory to the impersonation user specified in step 1 above.

Note Even though ACM 10.0 requires .Net 3.5, the folder referenced above is still found in the .Net v2.0 directory. This is because .Net3.5 is only an extension of the .Net 2.0 runtime.

Setting Permissions for the Impersonation User Account

This section explains how to set the correct permissions on the .NET directory for the impersonation user account.

1. Open Windows Explorer and browse to: %systemroot%\Microsoft.NET\Framework.



2. Right-click on the v3.5 (or whatever version of .NET ACM is using) and select Properties.
3. Select the Security tab.
4. Add the impersonation user account and grant it Modify permissions on this folder.

Backup Procedures

The Active CM stores data in (up to) three locations: the database instance, the public shared folder, and the private shared folder (note that in a simple deployment, the public shared folder and the private shared folder will be the same location). To backup all data, each of these data repositories must be addressed.

Backups should be scheduled to occur at a time when there are not content contributors changing content.

- Database – use standard SQL Server or Oracle database backup tools.
- Shared (private and public) folders – use standard file backup tools.

Installation Checklist

1. Create installation home directory. For example, C:\Inetpub\cmsroot'.
2. Copy all files into the CMS root
3. Decide if using full Digital Asset security. Configure Shared (and Private Shared) folders as appropriate.
4. Apply 'aspnet' or 'Network Service' user account to the Shared (normally 'Sites') folder with all but Full permission.
5. Create the database by running SQL Scripts
    - Schema script
    - Data script
    - Update scripts
6. Enter the correct database connection string in the /cmsroot/bin/IronPoint.DataAccess.dll.config file.
7. Copy license (.lic) files from extracted License.zip file
8. Configure indexing service.
    - If pdf search is required, install ifilter60.exe from www.adobe.com
9. Configure DNS, default.aspx, host headers, etc. for multiple websites (as required)
10. Configure IIS
    - Server Side Includes Setup and enabled?
    - Application Pool?
11. Configure web.config application settings
    - Exception management – Enter in email address for user who is going to receive application exception notifications.
    - Custom Error Message – Enabled
    - Max Request Length – Max size of files allowed to be uploaded
    - Setup paths to Shared Data folders
    - E-mail relaying Server (SMTP) – Setup network accessible mail server for sending system emails.
    - Digital Asset Security – Enabled or Not?
12. Install and configure Health Monitor service

Reference

DNS

Frequently Asked Questions About Windows 2000 DNS and Windows Server 2003 DNS (291382)
http://support.microsoft.com/default.aspx?scid=kb;en-us;291382

How To Integrate Windows Server 2003 DNS with an Existing DNS Infrastructure in Windows Server 2003 (323417)
http://support.microsoft.com/default.aspx?scid=kb;en-us;323417

IIS

IIS 6.0 Support Center
http://support.microsoft.com/default.aspx?scid=fh;en-us;iis60

How to use host header names to configure multiple Web sites in Internet Information Services 6.0 (324287)
http://support.microsoft.com/default.aspx?kbid=324287

Security

Microsoft Security Guidance Center
http://www.microsoft.com/canada/security/default.aspx