

Kofax Customer Communications Manager

5.0

Installation Guide

CCM ComposerUI ASP.NET 5.0



Contents

Introduction.....	3
This Manual.....	3
Copyrights and trademarks	3
Requirements.....	5
Web server	5
Client	5
Microsoft IIS 7 prerequisites.....	6
Microsoft Windows Server 2008: ASP.NET	6
Microsoft Windows Server 2008: Windows authentication.....	7
Microsoft Windows Server 2008: URL Authorization	7
Installing ITP/OnLine ASP.NET	9
Installation Dialogs.....	9
The CCM ComposerUI installation folders.....	10
CCM ComposerUI applications folder	10
Upgrading from an earlier version.....	10
Upgrading the Sample Application	11
Upgrading CCM ComposerUI Server 3.x applications	11
Upgrading to CCM ComposerUI Server 4.2	11
CCM ComposerUI virtual directory	11
Completing the installation	12
Authorization	12
Setting up an CCM ComposerUI administrators group	12
Set Microsoft IIS metabase permissions for the CCM ComposerUI administrators group ...	12
Setting file system authorizations.....	13
Microsoft IIS 6: Setting the correct ASP.NET version	16
Allow ASP.NET 2.0.....	16
Virtual directory ASP.NET version setting	16
Microsoft IIS 6: Improving stability in production environment.....	17
Microsoft IIS 7: Setting the correct application pool	18
Configure the application to use Classic .NET AppPool.....	18
Convert the virtual directory to an application	19
Configure Application pool.....	20
Microsoft IIS 7: Configuring feature delegation	21
Microsoft IIS 7: Improving stability in production environment.....	23
Configuration of CCM ComposerUI.....	24
Configuration of other ITP software	25
Enable CCM Repository model runs.....	25
CCM Core environment.....	25
Text Blocks, Forms, Field Sets, Views, and Content Wizards.....	25
License	27
Troubleshooting	28
ASP.NET write access permissions	28
Temporary file write permissions	28
Configuration fails with error "Could not access the configuration file. You can only access this page if you have write access to the configuration file."	28
503 Service Unavailable error while opening CCM ComposerUI Configuration.....	29

Introduction

CCM ComposerUI is an extension to CCM Core. CCM Core combined with CCM ComposerUI is called CCM ComposerUI Server. CCM ComposerUI Server consists of two components:

1. CCM Core: CCM Core has functionality designed to make it function in an CCM ComposerUI Server setup. This functionality is an integral part of CCM Core, and as such it is installed when CCM Core is installed. For more details refer to the CCM Core Installation Guide.
2. CCM ComposerUI: The CCM ComposerUI front-end web application.

This manual contains instructions on how to install CCM ComposerUI ASP.NET, version 5.0. The CCM ComposerUI version described in this guide is designed for ASP.NET. There is also a Java 2 Enterprise Edition (J2EE) implementation of CCM ComposerUI, which is not covered by this installation guide. There is a separate installation guide available for the installation of J2EE.

All documentation can also be found on the Kofax Customer Communications Manager Knowledge Center (<http://ccmkc.kofax.com>).

This Manual

This installation guide will help you to install ASP.NET correctly. The first thing this guide will help you to do, is check the [requirements](#) (page 5) necessary for a correct installation of ASP.NET. Please make sure you check whether all of these aspects are correctly available before you start this installation.

To guide you through an [installation](#) (page 9) or an [upgrade](#) (page 10), this guide describes all steps and concepts involved in this process. Also, the [steps required after installation](#) (page 12) and the entry of the [license](#) (page 27) are described. The [troubleshooting](#) (page 28) section may provide a solution, should a problem occur after installation.

Copyrights and trademarks

© 1993–2016 Lexmark. All rights reserved.

THIS SOFTWARE CONTAINS CONFIDENTIAL INFORMATION AND TRADE SECRETS OF KOFAX. USE, DISCLOSURE OR REPRODUCTION IS PROHIBITED WITHOUT THE PRIOR EXPRESS WRITTEN PERMISSION OF KOFAX.

Kofax, the Kofax logo, and the Kofax product names stated herein are trademarks or registered trademarks of Kofax in the U.S. and other countries. All other trademarks are the trademarks or registered trademarks of their respective owners.

U.S. Government Rights Commercial software. Government users are subject to the Kofax standard license agreement and applicable provisions of the FAR and its supplements.

You agree that you do not intend to and will not, directly or indirectly, export or transmit the Software or related documentation and technical data to any country to which such export or transmission is restricted by any applicable U.S. regulation or statute, without the prior written consent, if required, of the Bureau of Export Administration of the U.S. Department of Commerce,

or such other governmental entity as may have jurisdiction over such export or transmission. You represent and warrant that you are not located in, under the control of, or a national or resident of any such country.

DOCUMENTATION IS PROVIDED "AS IS" AND ALL EXPRESS OR IMPLIED CONDITIONS, REPRESENTATIONS AND WARRANTIES, INCLUDING ANY IMPLIED WARRANTY OF MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE OR NON-INFRINGEMENT, ARE DISCLAIMED, EXCEPT TO THE EXTENT THAT SUCH DISCLAIMERS ARE HELD TO BE LEGALLY INVALID.

Requirements

This chapter describes the system requirements for the web server on which CCM ComposerUI is installed and for the clients that access this web server.

Web server

CCM Core

CCM ComposerUI requires access to an CCM Core installation. This CCM Core does not have to be installed on the web server but it should be accessible from the web server through TCP/IP. An exact match between the versions of CCM ComposerUI and CCM Core is not required. The level of functionality that is supported by a combination of CCM ComposerUI and CCM Core may be limited by the lowest version.

Operating systems

Deployment of CCM ComposerUI is supported on 32-bit and 64-bit versions of the following server platforms:

- Microsoft Windows Server 2008 R2.
- Microsoft Windows Server 2012.
- Microsoft Windows Server 2012 R2.

Deployment of CCM ComposerUI on client platforms of Microsoft Windows is not supported.

Web servers

CCM ComposerUI ASP.NET requires Microsoft Internet Information Services (IIS) version 7.5 or later.

For IIS 8 follow the IIS7 instructions.

.NET Framework

CCM ComposerUI ASP.NET requires the Microsoft .NET Framework 3.5.

Other software

- Virus scanners installed on the system can negatively impact performance.

Client

Browsers

CCM ComposerUI is supported on:

- Microsoft Internet Explorer version 11,
- Microsoft Edge
- Mozilla Firefox version 38 and up,
- Google Chrome version 43 and up,

Please refer to the Known Issues section of the CCM ComposerUI Server Manual for any issues related to specific browsers or browser versions.

Other software

CCM ComposerUI produces preview documents in PDF format. To view these documents properly a PDF plug-in is required for the client browser. Adobe Reader is one of the PDF viewers which includes such a plug-in.

CCM ComposerUI offers functionality to edit Rich Text Blocks. To be able to edit Rich Text Blocks, you need to have Microsoft Word 2010 or higher and a browser with ActiveX support enabled. Currently, this feature is only available on CCM ComposerUI (ASP.NET). Editing of Rich Text Blocks in the Microsoft Word DOC format is only supported with Microsoft Word 2010.

Virus scanners can negatively influence performance.

Microsoft IIS 7 prerequisites

If Microsoft IIS 7 is not installed, please refer to the documentation of the Microsoft Windows Server TechCenter (<http://technet.microsoft.com>).

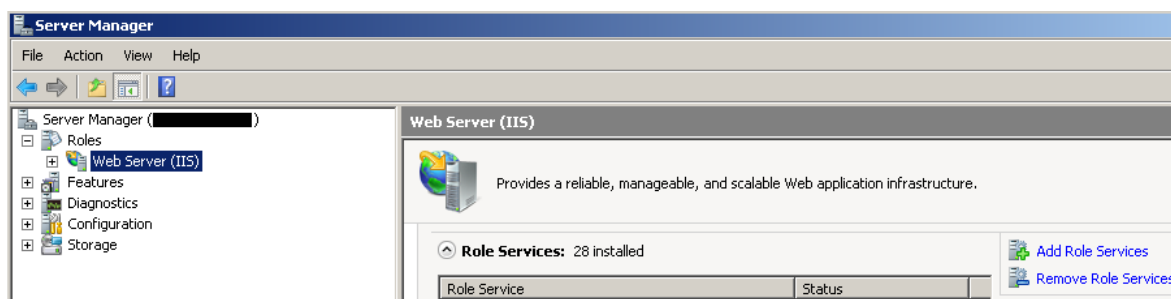
Make sure you also install the following non-default features when installing Microsoft IIS 7, as they are needed to complete the CCM ComposerUI installation. Refer to the sections below to see where these features are located in the tree when installing Microsoft IIS 7.

- ASP.NET; to make the application pool Classic .NET AppPool available.
- Windows Authentication; to make this authentication type available.
- URL Authorization; to make authorization on a per-URL basis available.

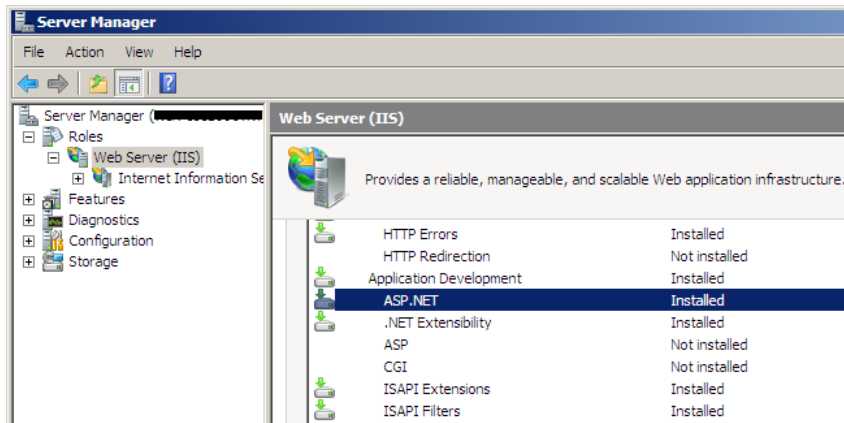
When Microsoft IIS 7 is already installed you need to verify that the above components are installed.

Microsoft Windows Server 2008: ASP.NET

1. Start the option **Programs and Features** from the Control Panel.
2. Select the option **Turn windows features on or off**.
3. Open the following path in the Server Manager tree: *Roles > Web Server (IIS)*.



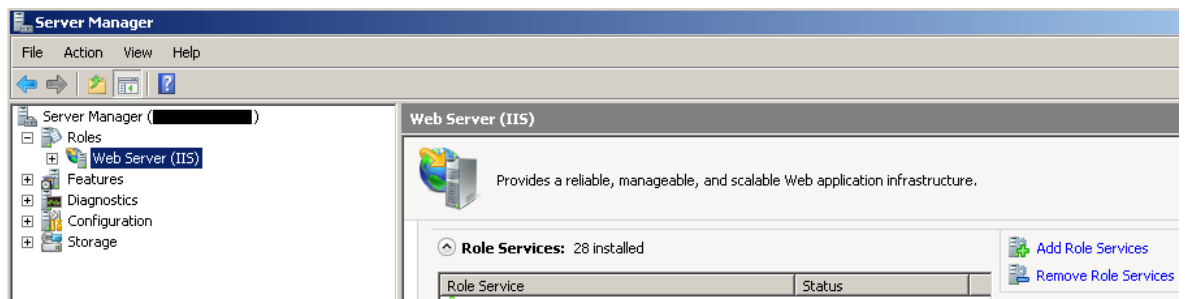
4. Check if the Role Service **Static Content** is installed. This is located under *Web Server > Common Http Features*.
5. Check if the Role Service **ASP.NET** and its associated Role Services are installed. These are located under *Web Server > Application Development*.



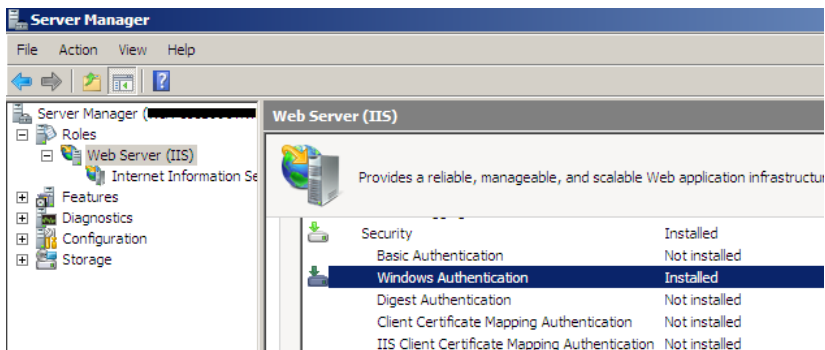
6. If these Role Service are not installed, install them using the button **Add Role Service** at the top of the list. When selecting the ASP.NET Role Service other Role Services will be selected, these need to be installed as well.

Microsoft Windows Server 2008: Windows authentication

1. Start the option **Programs and Features** from the Control Panel.
2. Select the option **Turn windows features on or off**.
3. Open the following path in the Server Manager tree: *Roles > Web Server (IIS)*.



4. Check if the Role Service **Windows Authentication** is installed. This is located under *Web Server > Security*.

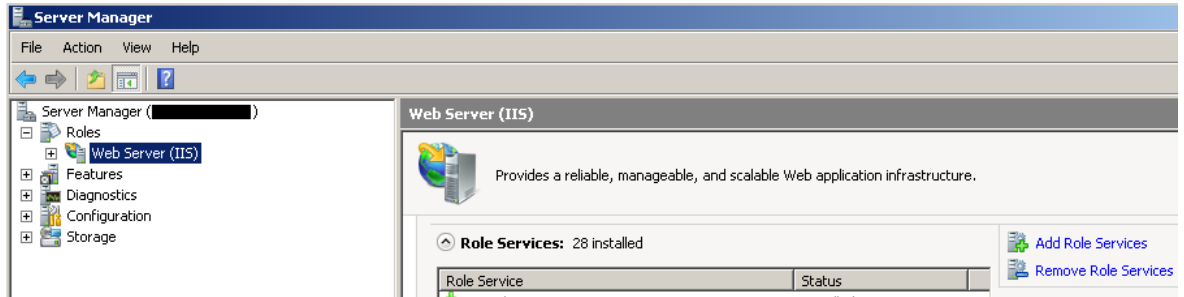


5. If this Role Service is not installed, install it using the button **Add Role Service** at the top of the list.

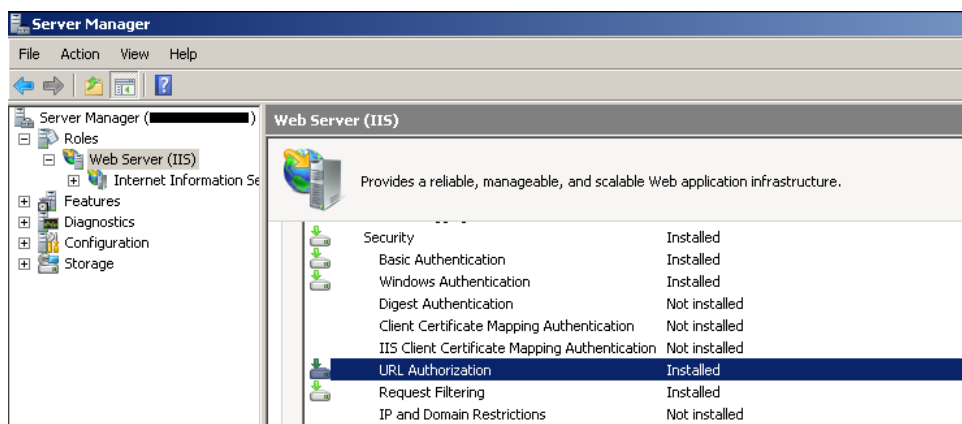
Microsoft Windows Server 2008: URL Authorization

1. Start the option **Programs and Features** from the Control Panel.
2. Select the option **Turn windows features on or off**.

- Open the following path in the Server Manager tree: *Roles > Web Server (IIS)*.



- Check if the Role Service **URL Authorization** is installed. This is located under *Web Server > Security*



- If this Role Service is not installed, install it using the button **Add Role Service** at the top of the list.

Installing ITP/OnLine ASP.NET

Download CCM ComposerUI from the support section of the [Aia Software website](#). Select the appropriate zip file and run the installer named ITPOnLineASP.NET-[version number]_Setup.exe.

Installation Dialogs

During a normal first-time installation, the installer shows the following dialogs. Upgrade installations are described later in this chapter, in the section [Upgrading from an earlier version](#) (page 10).

Welcome

The Welcome dialog gives information on the software that you are going to install.

Applications Destination Folder

In this dialog you should specify in which folder the CCM ComposerUI sample applications (custom web applications) will be installed. By default this is `C:\ITPOnLineApps`. By default, this folder will also contain the folders for the CCM ComposerUI activity logs and for temporary storage.

IIS Virtual Directory

CCM ComposerUI ASP.NET requires a virtual directory in Microsoft Internet Information Services (IIS). This virtual directory is the location where the CCM ComposerUI ASP.NET web application is exposed to the web. The virtual directory is automatically created by the CCM ComposerUI ASP.NET installer. By default the virtual directory name is "itp" and the corresponding physical path is `<wwwroot>\itp`, where `<wwwroot>` is the directory of the default web site.

IIS Public Directory

CCM ComposerUI ASP.NET requires a virtual public directory in Microsoft Internet Information Services (IIS). This directory is used by CCM ComposerUI ASP.NET to publish dynamically generated static web content, including HTML previews for Rich Text Blocks from the CCM Repository. The public directory is automatically created by the CCM ComposerUI ASP.NET installer. By default the public directory name is "itppublic" and the corresponding physical path is `<wwwroot>\itppublic`, where `<wwwroot>` is the directory of the default web site.

Completing the Installer

Click **NEXT** to start the installation.

Installation Complete

When the installation is completed, this dialog will appear: "Clicking the button **Finish** will end the installation." After the installation is complete, the operator should manually apply the authorizations described in chapter [Authorization](#) (page 12).

The CCM ComposerUI installation folders

There are three folders in which files will be installed. These folders are described here. After the installation is complete, the operator should manually apply authorizations and verify some settings. These actions are described in the chapter [Completing the installation](#) (page 12).

CCM ComposerUI applications folder

The applications folder for CCM ComposerUI is the folder in which custom CCM ComposerUI web applications are developed. Applications that are developed in this location can later be deployed to the virtual directory of CCM ComposerUI. By default the applications folder is `C:\ITPOnLineApps`.

When the installation is completed, the applications folder for CCM ComposerUI will contain the following subfolders:

- **itplog**, this folder is the default location for CCM ComposerUI log files.
- **sessiondata**, this folder is the default location for temporary storage for CCM ComposerUI web sessions.
- **Sample**, this is a sample custom CCM ComposerUI ASP.NET web application.
- **Sample2**, this is another sample custom CCM ComposerUI ASP.NET web application.
- **SecureSample**, this is a sample custom CCM ComposerUI ASP.NET web application intended for use in Secure Mode.

Upgrading from an earlier version

The ITP/OnLine installer supports upgrading from any earlier version, starting from version 3.2.6. Upgrading from versions before 3.2.6 is not supported. During the upgrade process, the following dialogs are shown.

Welcome

The Welcome dialog gives information on the software that you are going to install.

IIS Public Directory

CCM ComposerUI ASP.NET requires a virtual public directory in Microsoft Internet Information Services (IIS). This virtual public directory is automatically created by the CCM ComposerUI ASP.NET installer if it does not already exist. If the virtual public directory was already created by an earlier upgrade, this dialog will not be shown. By default the virtual public directory name is "itppublic" and the corresponding physical path is `<wwwroot>\itppublic`, where `<wwwroot>` is the directory of the default web site.

Upgrade existing installation

Click **Next** to start the upgrade.

Open main configuration page

At the end of the upgrade process, the installer will offer to open the main CCM ComposerUI ASP.NET configuration page. To finalize the upgrade, the operator should manually deploy all custom CCM ComposerUI ASP.NET web applications that are listed on this page. If there are problems during deployment, check if the authorization settings (as described in [Authorization](#) (page 12)) are still configured correctly.

Upgrading the Sample Application

The CCM ComposerUI upgrade installation includes a zip file that contains an up-to-date sample application. This allows the administrator to manually upgrade the sample application. An upgrade installation does not automatically upgrade the sample application as they may have been modified during the integration of CCM ComposerUI. The zip file containing the sample application can be found in the CCM ComposerUI application folder.

Note

When extracting a zip file to the corresponding application folder, all functionality that has been added will be lost. **Before upgrading**, we recommend to make a backup of the application.

To upgrade a sample application follow these steps:

1. Extract the zip file of the application you wish to update inside the corresponding folder in the CCM ComposerUI application folder. This overwrites all existing files.
2. Click the link **Deploy** in the Main Configuration page to finish the update of the application.

Upgrading CCM ComposerUI Server 3.x applications

From CCM ComposerUI 4.4 and onwards, support for table oriented output has been dropped. When upgrading to CCM ComposerUI Server 4.4, existing table oriented applications should be upgraded to div oriented applications.

Upgrading to CCM ComposerUI Server 4.2

Between CCM ComposerUI Server 3.5 and CCM ComposerUI Server 4.2 there is one relevant breaking change. Letterbooks are no longer retrieved from the file system; they are configured in and retrieved from ITP Base instead.

The new Letterbooks are expressed in XMLs with the namespace

`xmlns:itp='http://www.aia-ity.com/4.2/listmodels'` (rather than

`xmlns:itp='http://www.aia-ity.com/3.1/interact'`). This impacts all CCM ComposerUI

applications that contain customizations to the Letterbook functionality. The following files in your CCM ComposerUI applications are suspect to contain references to the old namespace and have to be modified to refer to the new namespace:

- `\xslt\custom\defaults_letterbook.xsl`
- `\xslt\custom\letterbook.xsl`

Note:

All other files in `\xslt\custom` folder do not apply to Letterbooks and should therefore not be subject to this change.

CCM ComposerUI virtual directory

The virtual directory for CCM ComposerUI is created and added as virtual directory in Microsoft IIS during installation. This folder contains all web content and deployed custom applications. These custom applications are deployed from the CCM ComposerUI applications folder.

Completing the installation

Authorization

After CCM ComposerUI ASP.NET has been installed additional steps are required to configure CCM ComposerUI ASP.NET. This section describes the authorizations that should be applied to file system objects and the Microsoft IIS metabase in order to get a working CCM ComposerUI ASP.NET installation.

Setting up an CCM ComposerUI administrators group

By default access to the configuration pages of CCM ComposerUI is restricted to members of the local Administrators group of the server that runs CCM ComposerUI. In addition, access can be granted to users that do not need to be local administrators on the server. This step is required on Microsoft Windows platforms that support User Account Control (UAC) such as Microsoft Windows Server 2008, when User Account Control is enabled. Without a separate CCM ComposerUI administrators group, it is not possible to administer CCM ComposerUI on these platforms. On all other platforms, this step is optional.

To permit users to access the CCM ComposerUI configuration pages, without making them a member of the local Administrators group, an alternative CCM ComposerUI administrators group should be created and given the appropriate permissions. Users can then be given CCM ComposerUI administration permissions by adding them to this group.

The CCM ComposerUI administrators group should be granted the following permissions:

1. Permissions on the Microsoft IIS metabase. How to set these rights for the CCM ComposerUI administrators group is explained in section [Set Microsoft IIS metabase permissions for the CCM ComposerUI administrators group](#) (page 12).
2. The same file system authorization rights as the local Administrators group. These authorizations rights are mentioned in section [Setting file system authorizations](#) (page 13).

Set Microsoft IIS metabase permissions for the CCM ComposerUI administrators group

The tool Microsoft MetaAcl can be used to grant the CCM ComposerUI administrators group write permissions for the Microsoft IIS metabase. These permissions are needed to perform the actions Deploy on the main CCM ComposerUI configuration page. This action is required only for Microsoft IIS versions earlier than 7.0.

Follow the instructions below for using this tool:

1. Download the tool MetaAcl from the Microsoft support site:
<http://support.microsoft.com>
2. Extract the tool by running the downloaded executable.
3. After extracting the tool, open a Microsoft Windows Command Prompt.
4. Go to the folder where the tool has been extracted, using the command "CD <folder>".
5. Then execute the following command: `cscript.exe Metaacl.vbs "IIS://localhost/W3SVC/1/ROOT/itp" <domain or machine name>\<name of the CCM ComposerUI administrators group> RWS`

6. In this command, replace "itp" by the name of the virtual directory that has been created for CCM ComposerUI ASP.NET. Note that "itp" is the default name of the virtual directory.

Setting file system authorizations

The Microsoft IIS web server and the CCM ComposerUI web application perform some tasks under special system user accounts. These system user accounts should already exist on the system. For correct operation of CCM ComposerUI ASP.NET, these user accounts should be granted certain permissions for the CCM ComposerUI virtual directory and the CCM ComposerUI applications folder. These permissions should be configured manually by the operator after the initial installation of CCM ComposerUI ASP.NET has completed. For security reasons it is also advisable to remove permissions for certain groups of users from these folders.

The following tables show which authorizations should be set for each folder for a specific OS. Take note of the following:

- All account and group names are machine local user accounts and groups.
- Accounts not listed here should not normally be granted access to the folders. Of course, unless there is a good reason to do so.
- If an additional CCM ComposerUI administrators group has been created, this group should be granted the same permissions for CCM ComposerUI folders as the local Administrators group. Note that the CCM ComposerUI administrators group permissions are not listed in the tables below.
- The permissions listed in the tables below include the group Users. These are the users that are allowed to access the web pages of CCM ComposerUI for applications where Secure Mode is disabled. In some cases the set of allowed CCM ComposerUI users should not be the same as the local users of the server. In such situations it is possible to create a user group for the CCM ComposerUI users, similar to the CCM ComposerUI administrators group described earlier. When such an CCM ComposerUI users group is used, the permissions granted to the group Users in the tables below should be granted to the CCM ComposerUI users group instead, and the group Users should not be granted any permissions.
- Before setting permissions for these folders using the dialog Properties in Windows Explorer, you must **always** do the following, for each of these folders:
 - In the tab Security of the dialog Properties, click the button **Advanced** to open the dialog Advanced Security Settings. Close this dialog before setting the permissions.
 - In the tab Permissions, disable the option **Inherit from parent the permission entries that apply to child objects. Include these with entries explicitly defined here**. Make sure you select **Copy** from the dialog that pops up after disabling this option. This ensures that you set the permissions for the folders exactly as mentioned below, and that they do not include permissions inherited from the parent folder.

Microsoft Windows Server 2003		
Location	Account	Permission type
CCM ComposerUI virtual directory	Administrators	Full Control
	IIS_WPG	Read & Execute
	IUSR_<machine name>	Read & Execute

Microsoft Windows Server 2003		
Location	Account	Permission type
	NETWORK SERVICE	Read & Execute
	Users	Read & Execute
CCM ComposerUI public directory	Administrators	Full Control
	IIS_WPG	Read & Execute
	IUSR_<machine name>	Read & Execute
	NETWORK SERVICE	Modify
	Users	Read & Execute
CCM ComposerUI applications folder	Administrators	Full Control
	Users	NONE
CCM ComposerUI log folder; by default this is the subfolder itplog of the CCM ComposerUI applications folder	Administrators	Full Control
	NETWORK SERVICE	Modify
	Users	NONE
CCM ComposerUI session data folder; by default this is the subfolder sessiondata of the CCM ComposerUI applications folder	Administrators	Full Control
	NETWORK SERVICE	Modify
	Users	NONE

Microsoft Windows Server 2008		
Location	Account	Permission type
CCM ComposerUI virtual directory	Administrators	Full Control
	IIS_IUSRS	Read & Execute
	IUSR	Read & Execute
	NETWORK SERVICE	Read & Execute
	Users	Read & Execute
CCM ComposerUI public directory	Administrators	Full Control
	IIS_IUSRS	Modify
	IUSR	Read & Execute

Microsoft Windows Server 2008		
Location	Account	Permission type
	NETWORK SERVICE	Modify
	Users	Read & Execute
CCM ComposerUI applications folder	Administrators	Full Control
	Users	NONE
CCM ComposerUI log folder; by default this is the subfolder itplog of the CCM ComposerUI applications folder	Administrators	Full Control
	NETWORK SERVICE	Modify
	Users	NONE
CCM ComposerUI session data folder; by default this is the subfolder sessiondata of the CCM ComposerUI applications folder	Administrators	Full Control
	NETWORK SERVICE	Modify
	Users	NONE

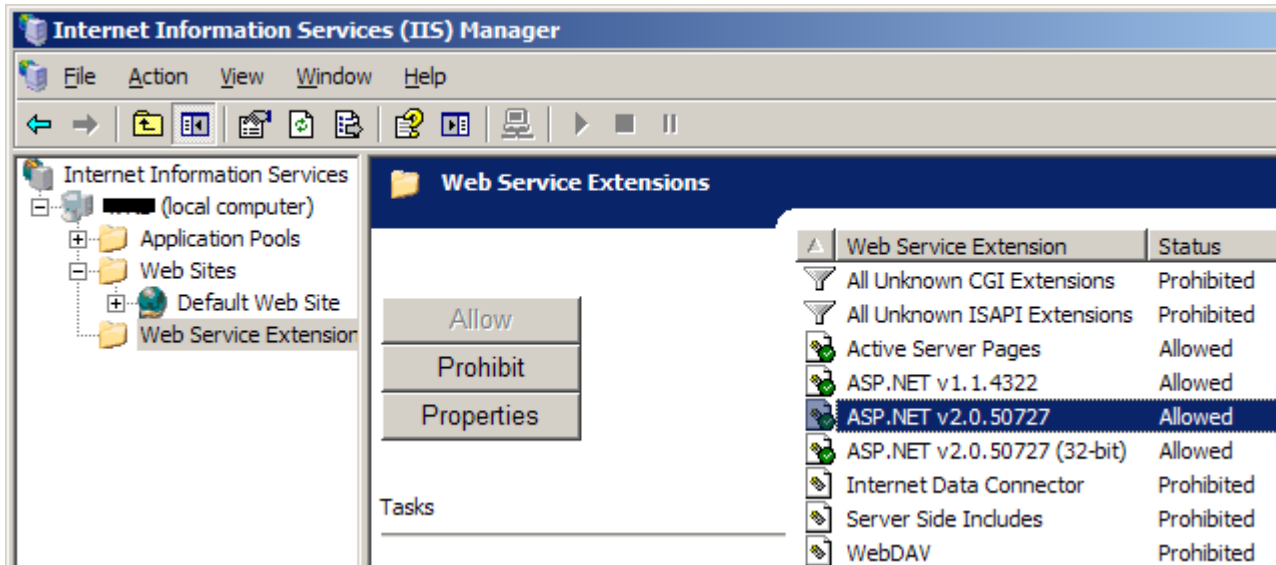
After setting the permissions on a folder, we advise to verify that the security settings have been applied correctly. This can be done as follows:

1. Open the dialog Properties of the folder and select the tab **Security**.
2. Click the button **Advanced** to open the Advanced Security Settings dialog.
3. Select the tab **Effective Permissions**.
4. Click the button **Select** and then select a non-privileged user account that has no rights to access the folder. Now verify that this regular user has no effective permissions on the folder, except for folders where the group Users should have been granted permissions. In the latter case, verify that the effective permissions are Read & execute.
5. Then, one by one, check the user accounts and groups that have been explicitly granted a certain level of access. For each user account and group, check that the effective permissions are not broader than those that were configured.

Microsoft IIS 6: Setting the correct ASP.NET version

Allow ASP.NET 2.0

If you use Microsoft IIS 6, make sure that ASP.NET v2.0.50727 is set to **Allowed**. For this, check the Web Service Extensions page, and click **Allow** if this extension was set to Prohibited. On 64-bits Microsoft Windows, select either the 32-bit or 64-bit version of ASP.NET v2.0.50727 depending on whether Microsoft IIS is running in 32-bits or 64-bits mode.



In some cases version 2.0.50727 is not available in Microsoft IIS, even though Microsoft .NET Framework 2.0 has been installed correctly. This can be corrected by using the ASP.NET IIS Registration Tool (`aspnet_regiis.exe`), which is installed with Microsoft .NET Framework 2.0. Depending on whether the Microsoft Windows version is 32bit or 64bit, `aspnet_regiis.exe` can be found in the folder `C:\WINDOWS\Microsoft.NET\Framework\v2.0.50727`, or `C:\WINDOWS\Microsoft.NET\Framework64\v2.0.50727`.

For more information on the ASP.NET IIS Registration Tool, refer to the Microsoft documentation at MSDN.

Example

Using `aspnet_regiis.exe` with `-i` key, installs the version of ASP.NET that is associated with `aspnet_regiis.exe` and updates the script maps at the IIS metabase root and below. Only the script maps for applications that use an earlier version of ASP.NET are updated. Applications that use a later version are not affected.

Note

Although in most cases this will resolve the issue, care should be taken and the above referred Microsoft documentation should always be consulted.

Virtual directory ASP.NET version setting

After the installation of CCM ComposerUI ASP.NET a Virtual Directory is created in Microsoft IIS. This virtual directory should be managed by ASP.NET 2.0. The installation program will set the correct version in Microsoft IIS.

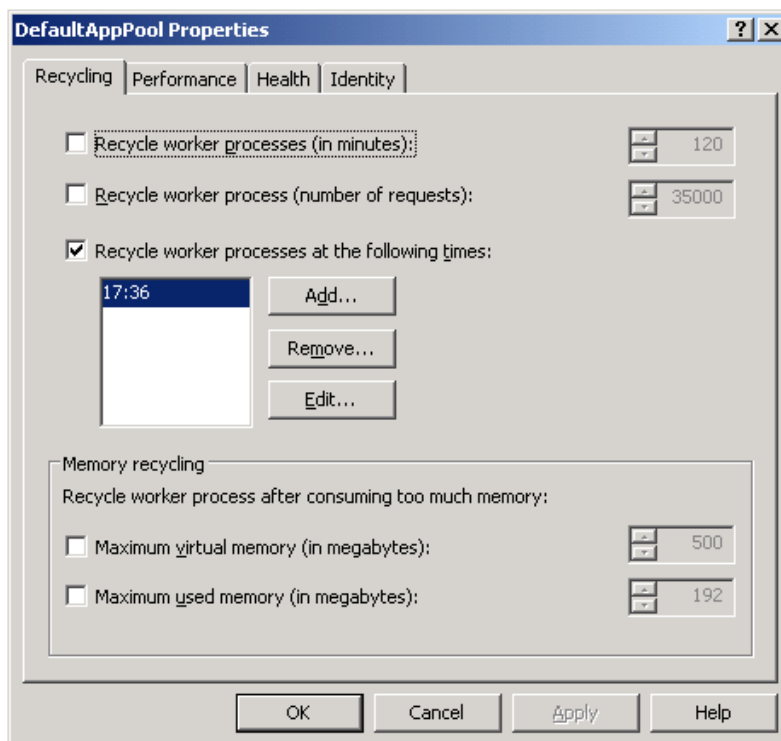
To verify if Microsoft IIS is indeed configured to manage this virtual directory with ASP.NET 2.0, take the following steps:

1. Open the Microsoft IIS Manager.
2. Browse to the virtual directory for CCM ComposerUI in the IIS Manager. The name of the virtual directory, by default "itp", is chosen during installation.
3. Select Properties from the context menu of the virtual directory for CCM ComposerUI.
4. On the tab ASP of the dialog Properties, check that the ASP.NET version is set to 2.0.50727.

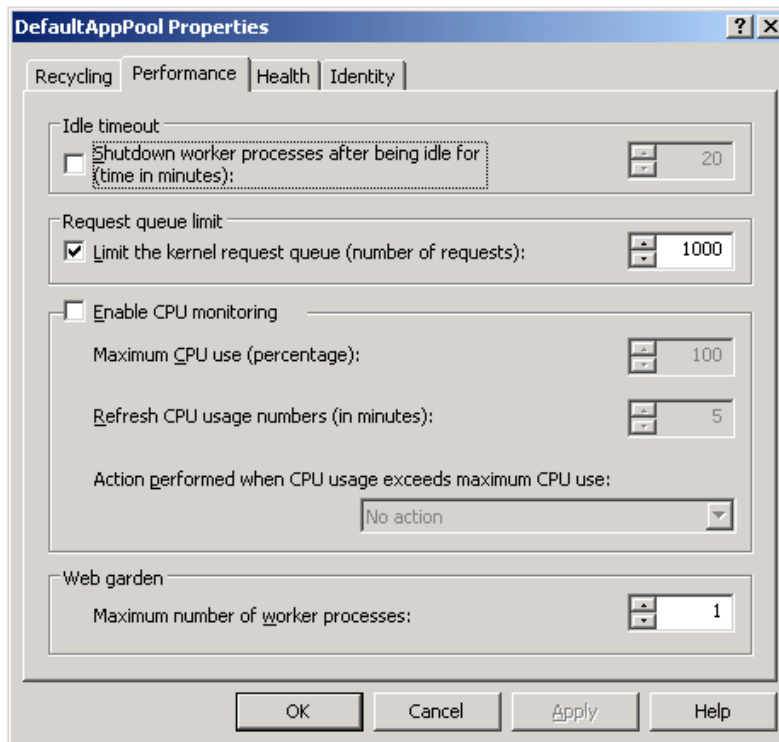
Microsoft IIS 6: Improving stability in production environment

To improve stability in production environment take the following steps:

1. Open Microsoft IIS Manager
2. Open the dialog **Properties** of the application pool, that you are using
3. You should disable the following properties:
 - a. On the tab **Recycling**:
 - i. Recycle worker processes (in minutes)
 - ii. Recycle worker process (number of requests)
 - iii. Maximum virtual memory (in megabytes)
 - iv. Maximum used memory (in megabytes)



- b. On the tab **Performance**
 - i. Shutdown worker processes after being idle for (time in minutes):

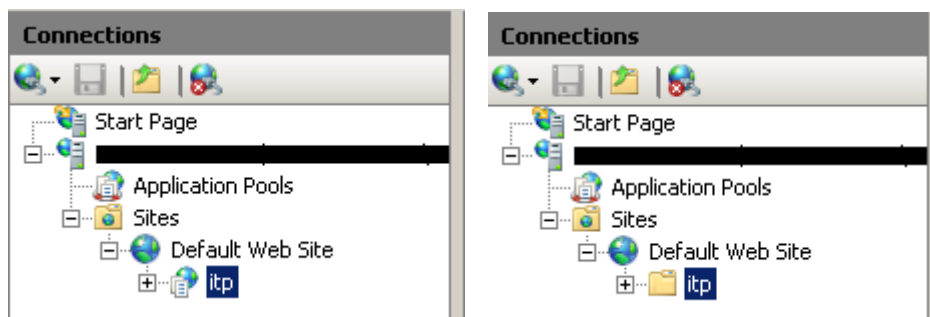


4. On the tab **Performance** change the setting Maximum number of worker processes to 1.
5. On the tab **Recycling** enable the property Recycle worker processes at the following times, and schedule the times to occur during system maintenance hours.

Microsoft IIS 7: Setting the correct application pool

Take the following steps to set the correct application pool:

1. In the Microsoft IIS Manager, look up the CCM ComposerUI Server ASP.NET application, which default name is itp.
2. Check if the CCM ComposerUI Server ASP.NET installation already is an application (left image), or if it is a folder (right image).

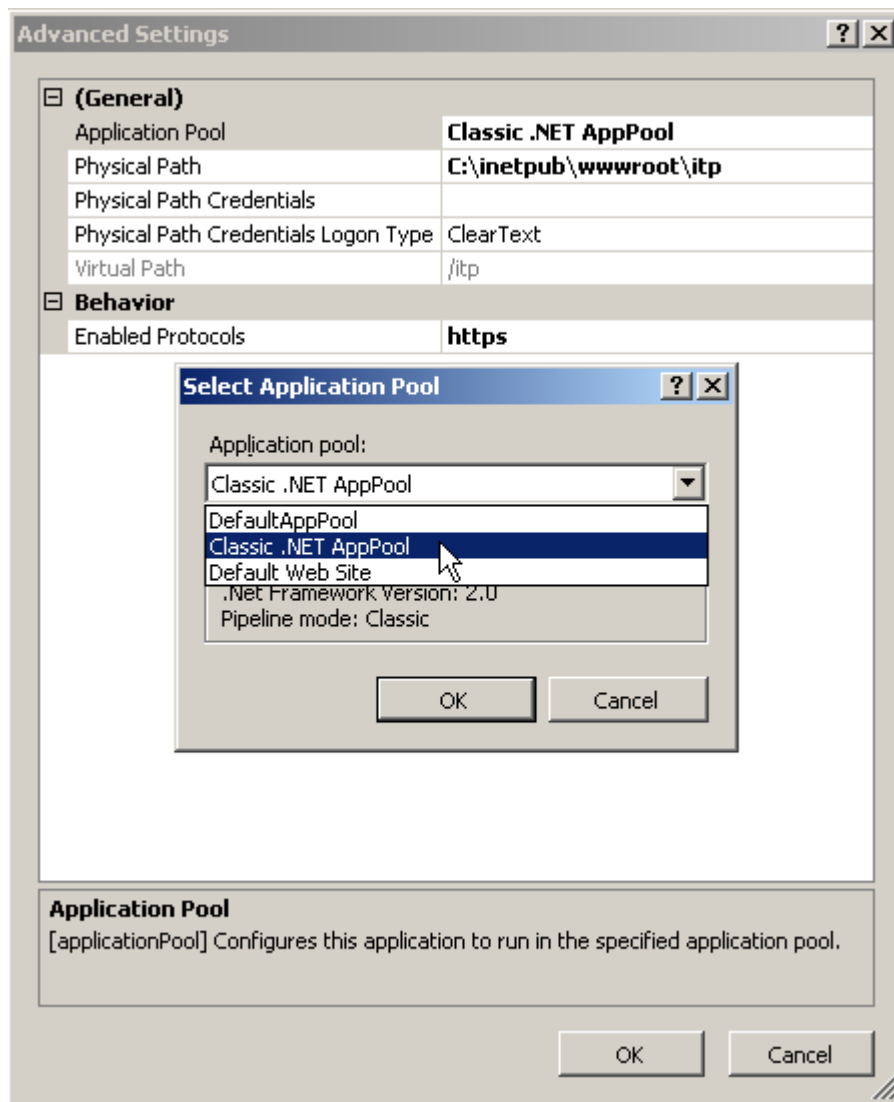


If the CCM ComposerUI Server ASP.NET installation is an application (left image), refer to chapter [Configure the application to use Classic .NET AppPool](#) (page 18) for more information. If the CCM ComposerUI Server ASP.NET installation is a folder (right image), refer to chapter [Convert the virtual directory to an application](#) (page 19) for more information.

Configure the application to use Classic .NET AppPool

Take the following steps to configure the application to use Classic .NET AppPool:

1. In the Microsoft IIS Manager select the itp application, by default itp.
2. Select the option **Advanced settings** from Manage Application, or from the context menu. The path depends on the used operating system.
3. Set the setting Application Pool to **Classic .NET AppPool** and click **OK**.

**Note**

If Application pool "Classic .NET AppPool" is not available, refer to section [Microsoft IIS 7 prerequisites](#) (page 6) for more information.

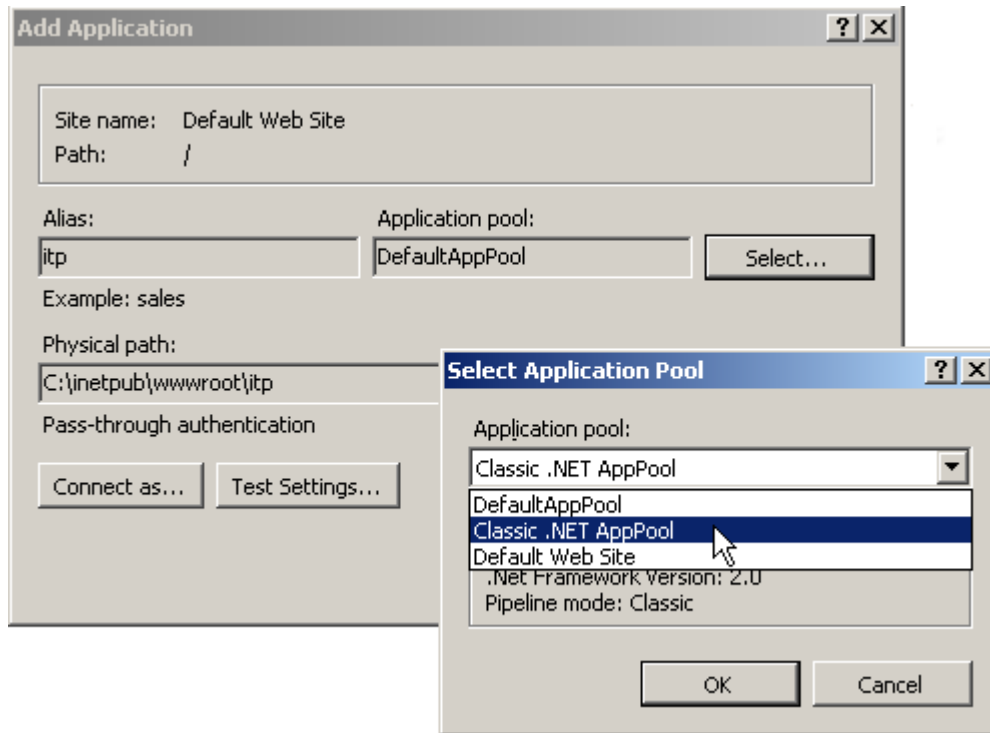
Note

The identity of the Application pool should be NetworkService. Refer to section [Configure Application pool](#) (page 20) for more information.

Convert the virtual directory to an application

Take the following steps to convert the virtual directory to an application:

1. In the Microsoft IIS Manager, select the itp virtual directory, by default itp.
2. Select the action **Convert to Application** from the context menu.
3. Select **Classic .NET AppPool** as Application pool and click **OK**.

**Note**

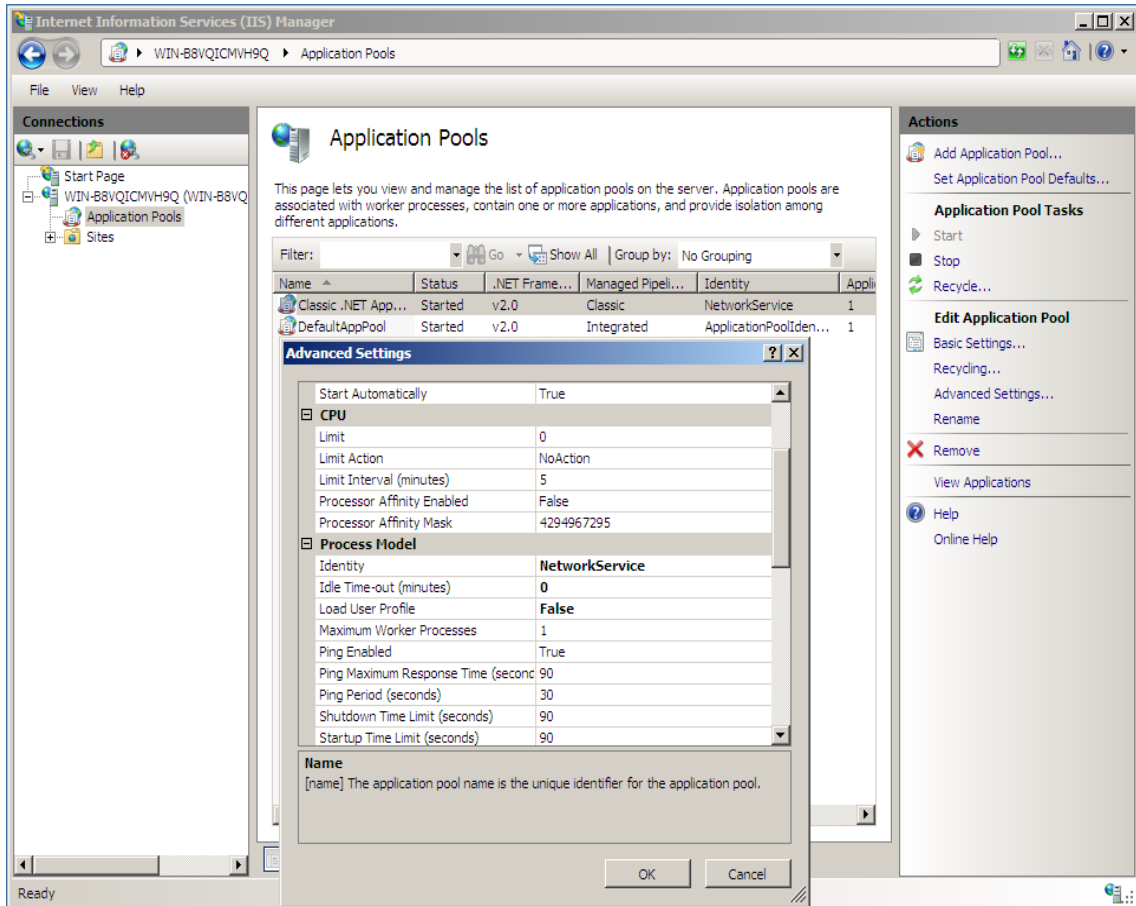
If Application pool "Classic .NET AppPool" is not available, refer to section [Microsoft IIS 7 prerequisites](#) (page 6) for more information.

Configure Application pool

Take the following steps to configure the Application pool

1. Open Microsoft IIS Manager
2. On the pane **Application Pools** open the dialog **Advanced Settings...** of the application pool, that you are using.

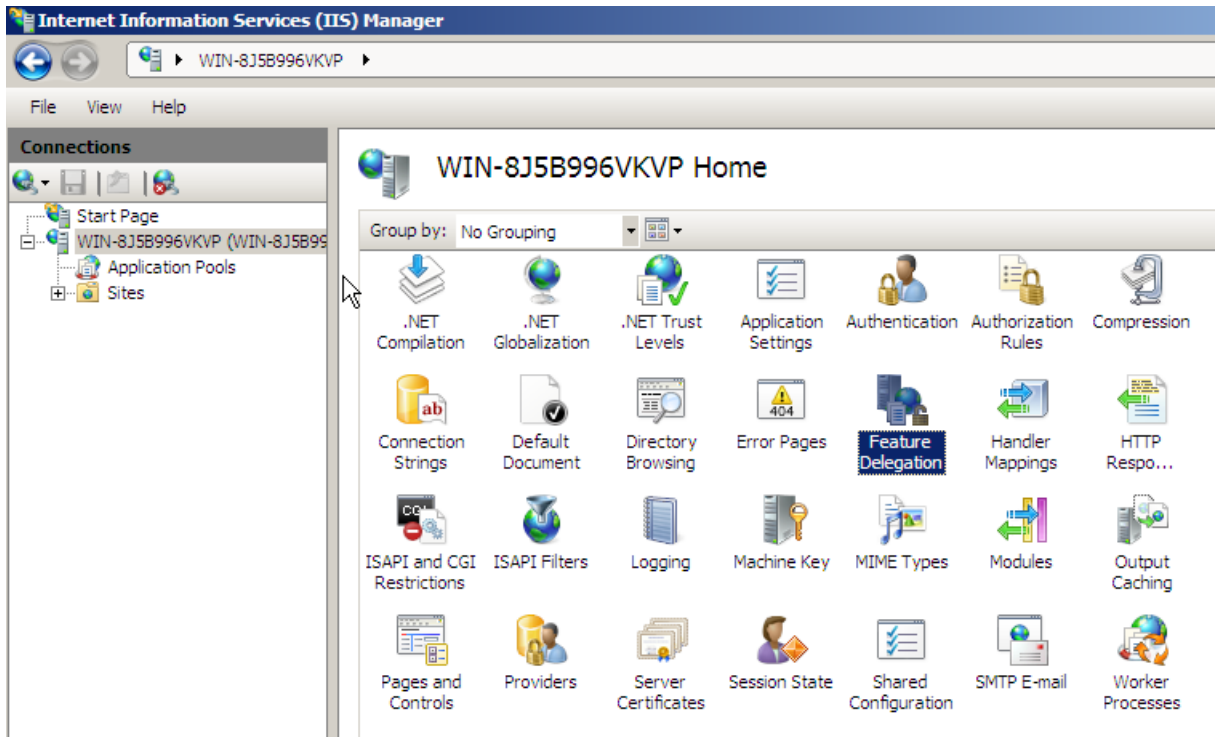
- Under the section **Process Model** change the setting Identity to NetworkService.



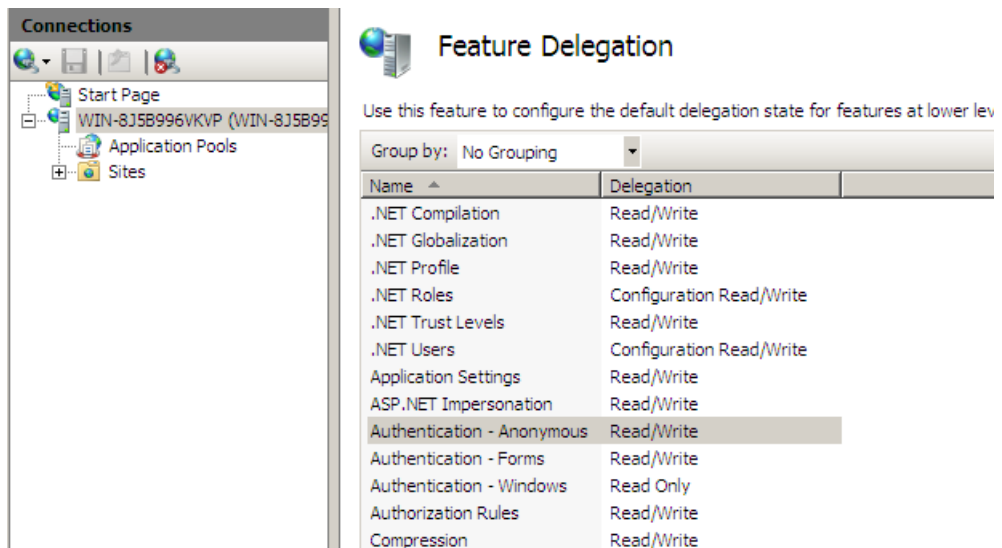
Microsoft IIS 7: Configuring feature delegation

The CCM ComposerUI web application controls its own authentication and authorization settings. This must be explicitly allowed in the Microsoft IIS 7 configuration, using the following procedure:

- Open Microsoft IIS Manager.
- In the Connections panel on the left-hand side, select the element that represents the entire web server.
- In the middle panel, double click **Feature Delegation** to open the Feature Delegation configuration page.



4. On the configuration page, select **Authentication - Anonymous**, and then select **Read/write** from the actions pane. If the link Read/Write in the Actions pane is disabled, then the setting is already configured correctly.



5. Repeat the last step for Authentication - Forms, Authentication - Windows, and Authorization Rules (and in Microsoft IIS 7.5 for Authentication - Basic and Authentication - Digest).

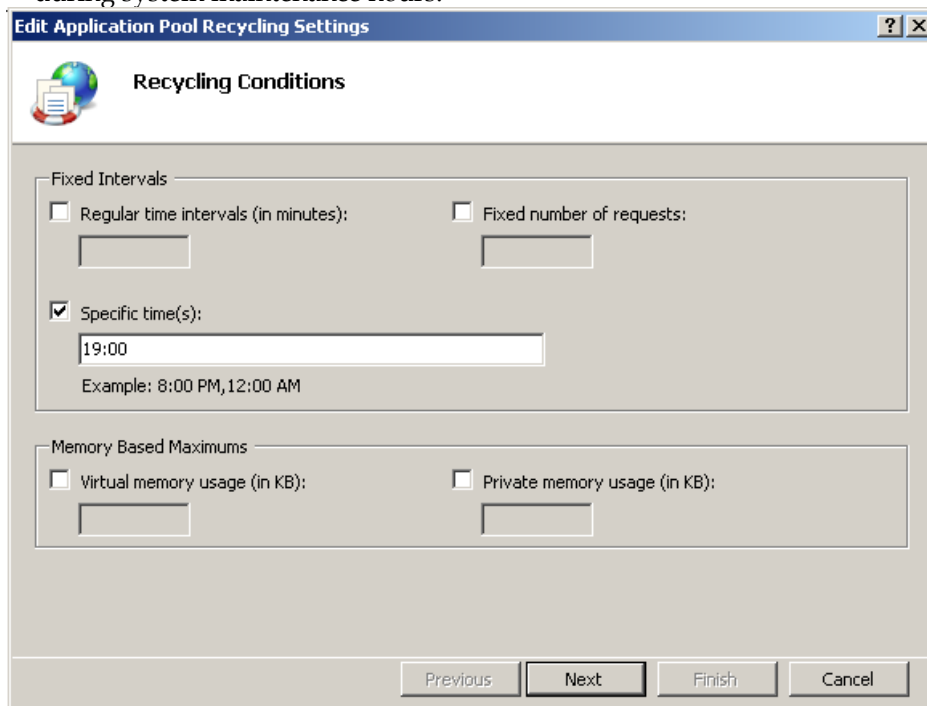
.NET Users	Configuration Read/Write
Application Settings	Read/Write
ASP.NET Impersonation	Read/Write
Authentication - Anonymous	Read/Write
Authentication - Forms	Read/Write
Authentication - Windows	Read/Write
Authorization Rules	Read/Write
Compression	Read/Write
Connection Strings	Read/Write
Default Document	Read/Write

6. If other Authentication Features are installed in IIS, either at the time CCM ComposerUI is installed or when these Features are added at a future time, Feature Delegation must also be set to Read/Write for these Features.

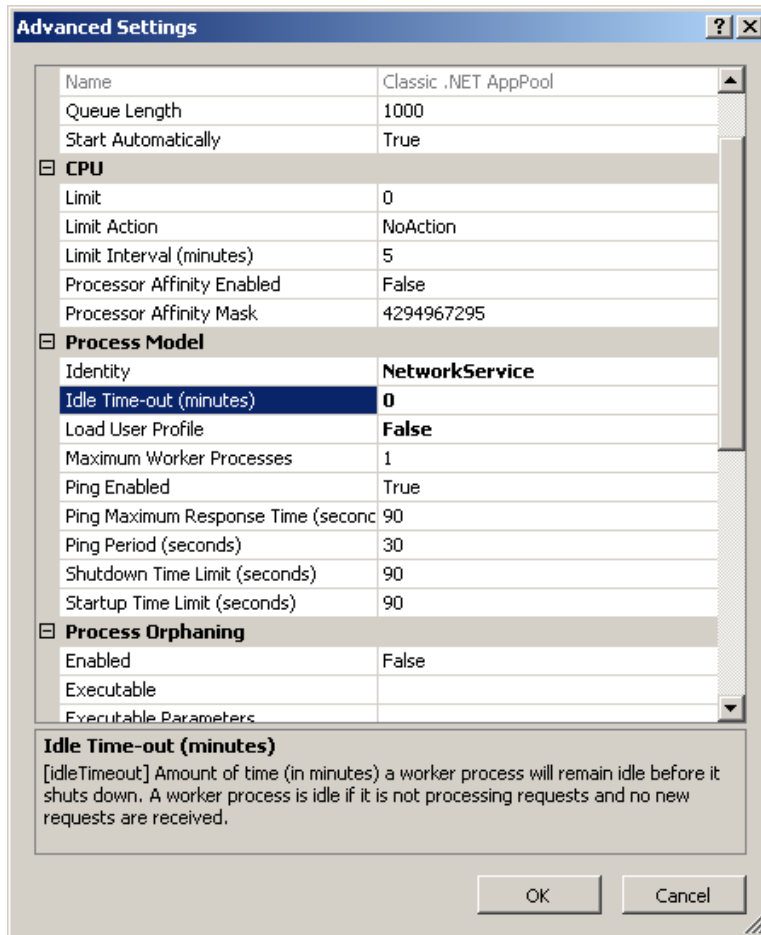
Microsoft IIS 7: Improving stability in production environment

To improve stability in production environment take the following steps:

1. Open Microsoft IIS Manager
2. On the pane **Application Pools** open the dialog **Recycling...** of the application pool, that you are using.
3. On the screen **Recycling Conditions**:
 - a. You should disable the following properties:
 - i. Regular time intervals (in minutes)
 - ii. Fixed number of requests
 - iii. Virtual memory usage (in KB)
 - iv. Private memory usage (in KB)
 - b. On the same screen enable the property **Specific time(s)**, and schedule the times to occur during system maintenance hours.



4. On the pane **Application Pools** open the dialog **Advanced Settings...** of the application pool, that you are using.
5. Under the section **Process Model**:
 - a. Change the setting **Idle Time-out (minutes)** to 0
 - b. Change the setting **Maximum Worker Processes** to 1



Configuration of CCM ComposerUI

Finish the configuration of CCM ComposerUI using the main configuration page at *http://<web server>/<virtual directory>/configure.aspx*, using the name of the web server, or local host on the web server itself, and the name of the virtual directory, by default itp, as specified during the installation of CCM ComposerUI.

Note

Opening the configuration page may take some time on Microsoft IIS 7 (Microsoft Windows Server 2008). In some situations, the configuration page may fail to load with the error message "Could not access the configuration file. You can only access this page if you have write access to the configuration file.". Please refer to the Section [Troubleshooting](#) (page 28) for information on how to resolve this issue.

On the CCM ComposerUI main configuration page, click the link Deploy for the SecureSample application. This will allow CCM ComposerUI to apply the web server security settings that are required for the SecureSample application.

After deploying the SecureSample application, continue by configuring the other configuration settings for CCM ComposerUI. Refer to the CCM ComposerUI Manual chapter Configuration for more information on configuring CCM ComposerUI Server.

Configuration of other ITP software

Enable CCM Repository model runs

It is possible to run models on CCM ComposerUI Server 3.1.14 or later from within the CCM Repository. Refer to the section Test Model tab for more information on configuring the CCM Repository project. The CCM ComposerUI Server itself also needs some configuration for this to work, which is described here.

1. Only perform this step when using an CCM Core version prior to 3.2.4. Version 3.2.4 and later already contain these scripts.
Copy the ITPOLSStartModel.dss script from the Resources folder in the CCM Repository Server installation to the scripts/system subfolder in the work folder of the CCM Core installation. Also copy the CheckRepositoryModelAccess.dss script from the Resources folder to the scripts/user Library folder of CCM Core.
2. In the CCM Core Administrator, select the services node and the Constants tab. Set the AllowRepositoryModelRun constant to the value Y, add it if it is not already present.
3. Recompile the scripts from within CCM Core Administrator, and click Save & Apply.
4. (Only perform this step when using CCM Core versions prior to 3.2.4, or when using an CCM Repository version 3.5.9 or earlier)
Copy the GetModel.exe from an CCM Repository Client installation to the bin subfolder of the CCM Core installation.

The AllowRepositoryModelRun constant determines if CCM ComposerUI Server will accept model run requests from the CCM Repository. If not set to 'Y', CCM Core will deny running models from within the CCM Repository. By explicitly having to configure an CCM ComposerUI Server installation to accept CCM Repository runs, it can be avoided that CCM Repository runs are run accidentally or unauthorized on a particular CCM ComposerUI Server installation.

The user exit script CheckRepositoryModelAccess.dss is run to check whether a specific user is allowed to run a specific model on CCM ComposerUI Server. The default implementation allows all. If you want more restrictive checking, you can implement it by changing this script.

CCM Core environment

If a model is run using CCM ComposerUI Server, the CCM ComposerUI application decides which CCM Core environment should be used. If no environment is passed by the CCM ComposerUI application, the CCM Core default environment will be used. It is possible to override these configurations by passing an CCM Core environment on the CCM ComposerUI Server URI. In this way it is possible to configure an alternate environment for a project.

In the following example the environment 'live' is used to run the ITP Models.

```
http://itponlineserver/itp/app/sample/modelbegin.aspx?env=live
```

Text Blocks, Forms, Field Sets, Views, and Content Wizards

Dynamic objects -- a Data Backbone, Text Blocks, Forms, Field Sets, Views, and Content Wizards -- are retrieved from the CCM Content Publication Database or Editorial CCM Repository at the

moment the ITP Model first references the object during execution. Which CCM Content Publication Database installation is used to retrieve these objects from is configured by a setting in the CCM Core environment. The dynamic objects are always retrieved from the same Project as the Model.

If a Model is tested from the CCM Repository Editorial the dynamic objects will be retrieved from the same Editorial CCM Repository from which the test run was initiated.

License

The CCM ComposerUI license needs to be added to the CCM ComposerUI Server installation using the CCM Core license manager. Refer to the CCM Core Installation Guide for more details.

The CCM Core connection information must be configured in the CCM ComposerUI ASP.NET configuration. In a default installation, the CCM ComposerUI ASP.NET configuration page can be accessed using the following URL: <http://localhost/itp/configure.aspx>. If the IIS virtual directory was changed during the installation, the configuration page can be accessed using the URL <http://localhost/<virtual directory>/configure.aspx>.

Troubleshooting

ASP.NET write access permissions

In some rare cases, the NETWORK SERVICE or ASPNET account, which is used to execute the CCM ComposerUI ASP.NET Application, has insufficient permissions, which results in an error message like: "The current identity does not have write access to "C:\WINDOWS\Microsoft.NET\Framework\v2.0.50727\Temporary ASP.NET Files." This can be solved by granting the necessary permissions using the ASP.NET IIS Registration Tool (aspnet_regiis.exe), which is installed with Microsoft .NET Framework 2.0.

Using aspnet_regiis.exe with the `-ga` switch followed by the account name, grants the following permissions to the specified account:

- Access to the Microsoft IIS metabase.
- Write permissions to the folder:
`C:\WINDOWS\Microsoft.Net\Framework\v2.0.50727\Temporary ASP.NET Files`

The `-ga` switch makes a number of global changes. If you want to restrict access to specific folders, manually adjust the permissions on those folders. For more information on the ASP.NET IIS Registration Tool, refer to the Microsoft documentation at MSDN.

Use the following command to grant the necessary permissions:

```
aspnet_regiis.exe -ga "<NETWORK SERVICE or ASPNET>"
```

Temporary file write permissions

In some rare cases on a Microsoft Windows Server 2003 system, the system user account NETWORK SERVICE has insufficient permissions to write files in the temporary files folder (by default this is `%SystemRoot%\Temp`). ASP.NET uses this location to save files during compilation of web pages. Due to insufficient permissions compilation errors may occur. To prevent this, make sure the account NETWORK SERVICE has write permissions at this location.

Configuration fails with error "Could not access the configuration file. You can only access this page if you have write access to the configuration file."

This error message may occur in either of the following situations:

- The user account used to access the configuration page does not have sufficient permissions to write to the Virtual Directory.
- The user account has sufficient permissions, but Microsoft Internet Explorer is run on the same computer that hosts CCM ComposerUI, and it is running in Protected Mode.
- The user account has sufficient permissions, but Microsoft Internet Explorer is run on the same computer that hosts CCM ComposerUI, UAC is enabled, and the file system permissions are

not configured using a separate ITP/OnLine administrators group.

These situations will be discussed separately below.

Protected Mode

When Microsoft Internet Explorer is running in Protected Mode, the status bar will display the text "Protected Mode: On". This situation can only occur on Microsoft Windows Server 2008 or up, since Microsoft Internet Explorer supports Protected Mode only on these platforms. To resolve this situation, Protected Mode should be disabled for Intranet sites:

1. Open Microsoft Internet Explorer as Administrator. This is done by right clicking on the Internet Explorer icon and selecting "Run as Administrator".
2. Go to the Tools menu and select "Internet Options".
3. In the Internet Options dialog, go to the tab "Security".
4. Select the "Local intranet" zone.
5. Turn off the check box "Enable Protected Mode."
6. Press the "OK" button to close the Internet Options window.
7. Restart Microsoft Internet Explorer.

Note

By default, Microsoft Internet Explorer 7 and 8 disable the "Local intranet" security zone. The fix described above only works when the "Local intranet" security zone is enabled.

Permissions

If the error message occurs when Protected Mode is disabled, then the user does not have enough permissions to write to the file "config.xml" in the Virtual Directory. There are two common causes for this problem. The first cause is that the user is not a member of the local Administrators group of the server computer, and is also not in the CCM ComposerUI administrators group. This can be resolved by using a different user account, or by adding the user account to the appropriate group. The second cause is that the file system permissions have been misconfigured. Verify that the file system permissions are configured as specified in Section [Setting file system authorizations](#) (page 13).

User Account Control (UAC)

When the Microsoft Windows installation has User Account Control (UAC) enabled, members of the local Administrators group are not automatically allowed to configure CCM ComposerUI. In this situation, it is required that CCM ComposerUI administration is configured using a separate CCM ComposerUI administrators group, as described in section [Setting up an CCM ComposerUI administrators group](#) (page 12).

503 Service Unavailable error while opening CCM ComposerUI Configuration

For Microsoft Windows Server 2008 one possible cause for this error could be insufficient permissions for reading a configuration file.

The event viewer on the machine where CCM ComposerUI is installed would show the following error:

The worker process for application pool encountered an error 'Cannot read configuration file due to insufficient permissions

It would indicate that machine.config could not be read.

To solve this problem make sure that the authorization for the machine.config indicated in the event viewer message includes the IIS_IUSRS account with Read & Execute permission.