# Content Manager

Software Version 9.4

## Web Client Installation and Configuration

**Micro Focus®**

# Legal notices

### Copyright notice

© Copyright 2008-2022 Micro Focus or one of its affiliates.

The only warranties for products and services of Micro Focus and its affiliates and licensors ("Micro Focus") are set forth in the express warranty statements accompanying such products and services. Nothing herein should be construed as constituting an additional warranty. Micro Focus shall not be liable for technical or editorial errors or omissions contained herein. The information contained herein is subject to change without notice.

Adobe™ is a trademark of Adobe Systems Incorporated.

Microsoft® and Windows® are U.S. registered trademarks of Microsoft Corporation.

UNIX® is a registered trademark of The Open Group.

This product includes an interface of the 'zlib' general purpose compression library, which is Copyright © 1995-2002 Jean-loup Gailly and Mark Adler.

## Documentation updates

The title page of this document contains the following identifying information:
- Software Version number, which indicates the software version.
- Document Release Date, which changes each time the document is updated.
- Software Release Date, which indicates the release date of this version of the software.

You can check for more recent versions of a document through the MySupport portal. Many areas of the portal, including the one for documentation, require you to sign in with a Software Passport. If you need a Passport, you can create one when prompted to sign in.

Additionally, if you subscribe to the appropriate product support service, you will receive new or updated editions of documentation. Contact your Micro Focus sales representative for details.

## Support

Visit the MySupport portal to access contact information and details about the products, services, and support that Micro Focus offers.

This portal also provides customer self-solve capabilities. It gives you a fast and efficient way to access interactive technical support tools needed to manage your business. As a valued support customer, you can benefit by using the MySupport portal to:

- Search for knowledge documents of interest
- Access product documentation
- View software vulnerability alerts
- Enter into discussions with other software customers
- Download software patches
- Manage software licenses, downloads, and support contracts
- Submit and track service requests
- Contact customer support
- View information about all services that Support offers

Many areas of the portal require you to sign in with a Software Passport. If you need a Passport, you can create one when prompted to sign in. To learn about the different access levels the portal uses, see the Access Levels descriptions.

# Contents

# Content Manager Web Client

## Introduction

The purpose of the Content Manager Web client is to provide zero-footprint browser-based access to a Content Manager database.

Content Manager Web Client uses the responsive design paradigm. The responsive design feature allows Content Manager Web Client to automatically adjust and layout the content of the application to suite the specific device.

## Audience

This document is for administrators who need to install and configure the Content Manager Web client.

> **IMPORTANT:** For installation and upgrade information, please see **CM9.4_Install.pdf**.

## System requirements

For operating system and browser requirements for Content Manager Web client, see **CM9.4_ Spec.pdf**.

### Pre-requisites

### Web Servers

The requirements for the Content Manager Web Client Server are the same as for a Content Manager Workgroup Server plus the additional requirements below.

See **CM9.4_Spec.pdf** for Content Manager specifications and limitations.

Additional requirements:

- Microsoft Internet Information Server (IIS) 7.5, 8.0 or 8.5 with Microsoft .Net Framework 4.5.

- The correct Roles Services for the respective Web Server under Windows Server 2012 need to be configured and installed.

# Configuration

## Setting up Single Sign On (SSO) authentication for the Web Client

The Content Manager Web Client can be setup to use Single Sign On (SSO) for authentication.

In this instance, SSO is used to refer to the task of identity federation, which is authenticating with a web application located outside your organisation boundary using your organisation credentials. This might be achieved in a variety of ways, such as, a WS-Fed solution like ADFS (or Azure AD) or one of the many public SAML providers (OneLogin, Ping Identity and Okta).

Once the Content Manager Service API has been installed, navigate to http://localhost/CMServiceAPI/help/authentication for an in-depth overview of how to integrate the Web Client with ADFS.

After integrating the Web Client with ADFS, the key **useADFS** needs to be changed to **true** in the file hprmServiceAPI.config.

<!-- Web Client configuration -->

```
<setup databaseId="45" searchAhead="true" workpath="C:\Micro Focus Content
Manager\ServiceAPIWorkpath\Uploads" useADFS="true" />
```

## Configuring PDF Document Viewer

The Content Manager Web Client allows viewing PDF documents using a native PDF viewer. By default, the PDF documents are viewed as HTML documents when the document is viewed. To setup viewing a native version of the PDF document using a PDF viewer:

Users need to enter the file type PDF in the **Bypass Content Manager Viewer for these File Types** in the Content Manager client on the **Settings** dialog - **Document Viewer** tab.

Internet Explorer and Firefox users need to install Adobe Acrobat Reader on their desktop in order to view the PDF version of the document when clicking on the View icon.

## Managing Content Manager Web Client Security

This section assumes you are familiar with the following technologies:

- Microsoft Internet Information Server (7+ and 8+)
- Content Manager Enterprise Studio

## Background

For the Content Manager Web Client to work, it needs to be able to connect to the Content Manager Workgroup Server. The Content Manager Web Client can connect to the local Content Manager Workgroup Server or a remote Content Manager Workgroup Server. To connect, the Content Manager Web Client IIS account must be the same as the Content Manager Workgroup Server service account. Additional configuration will be required if the Content Manager Web Client IIS account is different to the Content Manager Workgroup Server service account.

## Why impersonation delegation?

In Windows, the Content Manager desktop client logs onto a Content Manager Workgroup Server using the user's Windows credentials. The Content Manager Web Client is essentially a Content Manager client that connects the Web Client user to the Content Manager Workgroup Server. The Web Client service account needs to log onto the Content Manager Workgroup, not as itself, but as the Web Client user account. This process is called impersonation.

Security would be compromised if any account were to be allowed to impersonate any other account. In Content Manager Enterprise Studio, configure a list of trusted server accounts. These trusted server accounts are allowed to impersonate other accounts. It is important that you keep this list of accounts as short as possible, and that the accounts listed in it are not compromised. When the Content Manager Web Client attempts to log onto the Content Manager Workgroup Server, the Content Manager Workgroup Server checks that the Web Client is running under a trusted server account.

The authentication settings allow the Web Client user to connect to a Content Manager Workgroup Server in a double hop environment, where the user logs on to the Content Manager Web Client using NTLM. The Windows NT Challenge/Response authentication does not support double-hop impersonations (in that once passed to the IIS server, the same credentials cannot be passed to a back-end server for authentication.

See also http://support.microsoft.com/kb/264921, section Windows NT Challenge / Response.

> **NOTE:** The Content Manager Workgroup Server service account is automatically trusted. If the Web Client runs under the same account as the Content Manager Workgroup Server, it is not be necessary to configure it in Content Manager Enterprise Studio as a trusted server account.

## Content Manager Web Client account is the same as Content Manager Workgroup Server account

In this instance, no additional configuration is needed. If your Content Manager Workgroup Server account is **domain\trimservices**, where **domain** is your organization's domain name and **trimservices** is the domain user name, the account that runs your Content Manager Web Client must also be **domain\trimservices**.

For example, in the deployment scenario below, you will not need to do any additional configuration because the Content Manager Web Client and the Content Manager Workgroup Server are both running as **TRIM\trimservices** account.
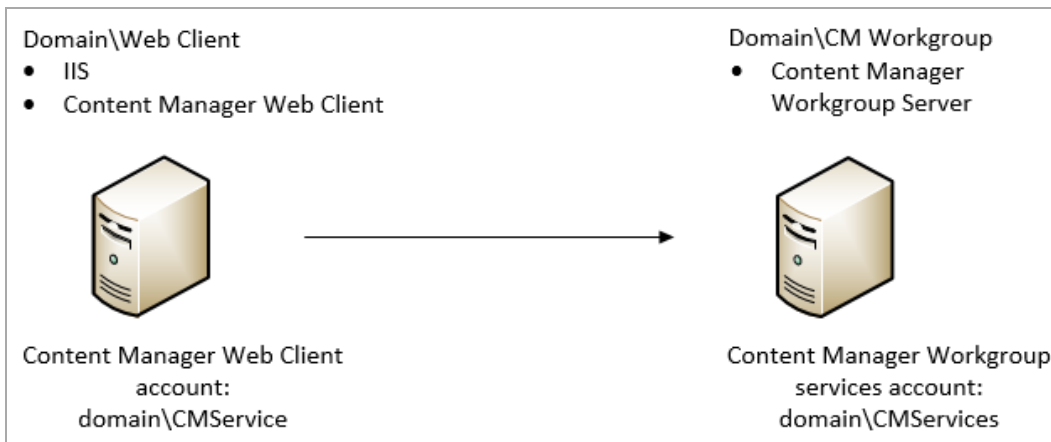
## Content Manager Web Client account is different to Content Manager Workgroup Server account

If the Content Manager Web Client IIS account is different to the Content Manager Workgroup Server service account, the Content Manager Web Client IIS account must be added to the list of authorized user accounts that the Content Manager Workgroup Server will trust with supplying valid user credentials.

For example, if the Content Manager Web Client instance runs as Network Services (**Web Client\Network Services**) and the Content Manager Workgroup Server runs as **TRIM\trimservices**, you will need to add **TRIM\Web Client$** in the trusted user list in Content Manager Enterprise Studio. This is because Network Services will try to log on to Content Manager Workgroup Server as **TRIM\Web Client$ (domainName\computerName$)**.
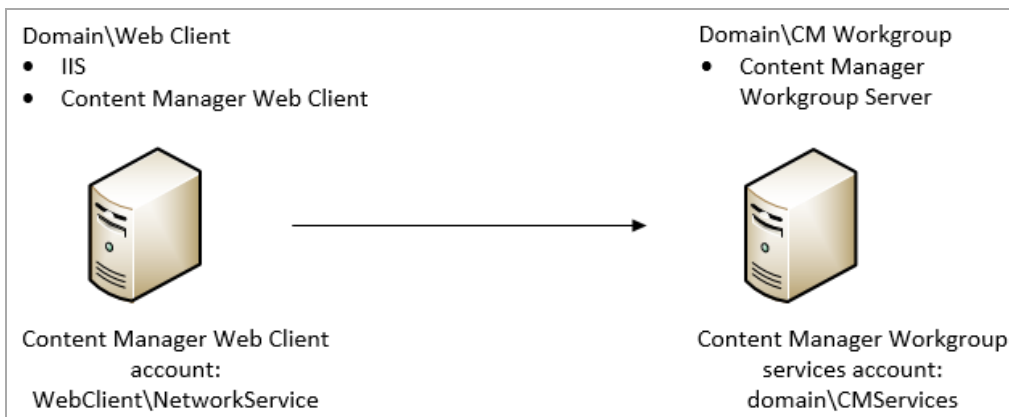


Adding the authorized account that the Content Manager Workgroup Server will trust with supplying valid user credentials:

1. As an Administrator, open **Content Manager Enterprise Studio**, select the dataset name that is the authorized account is to be added to, from the **Home** tab, on **General** group, click **Options**.
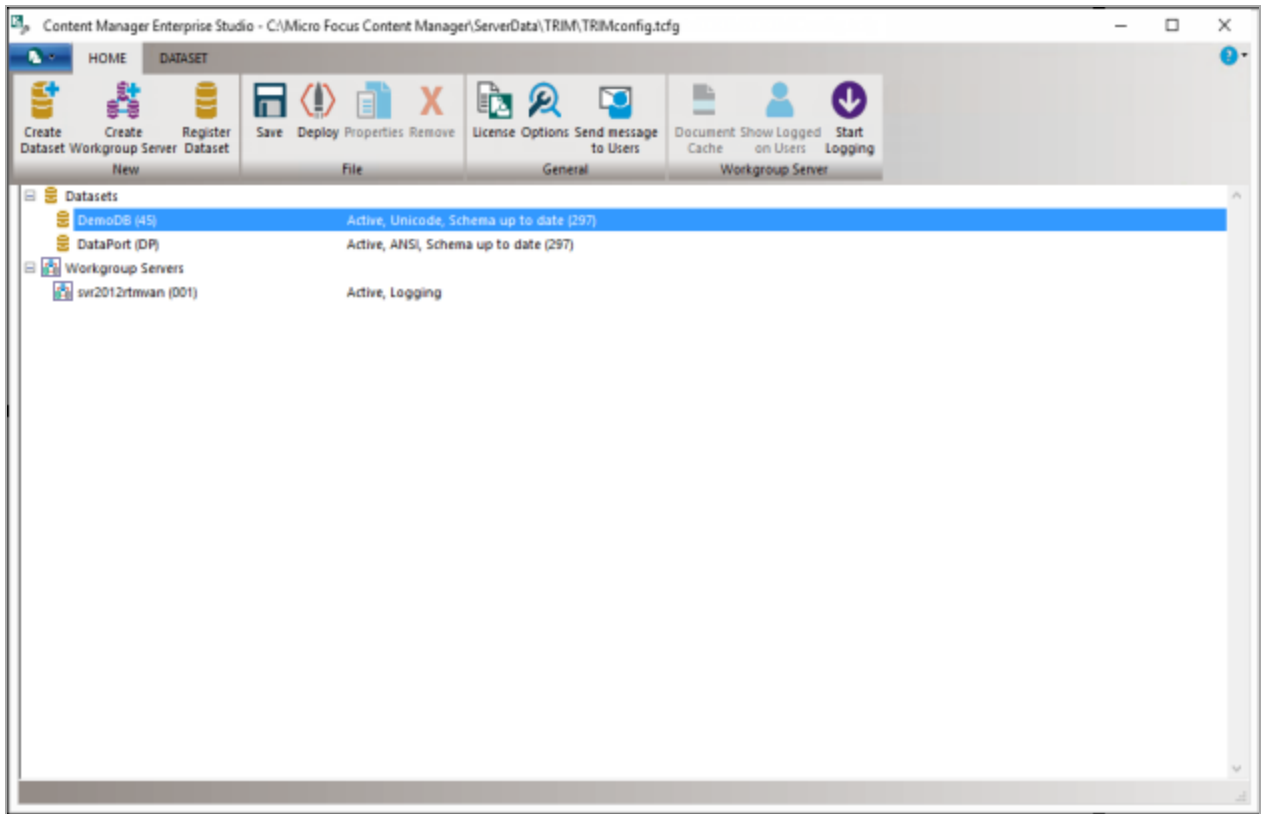
2. On the displayed **Options** dialog, click **Other**.
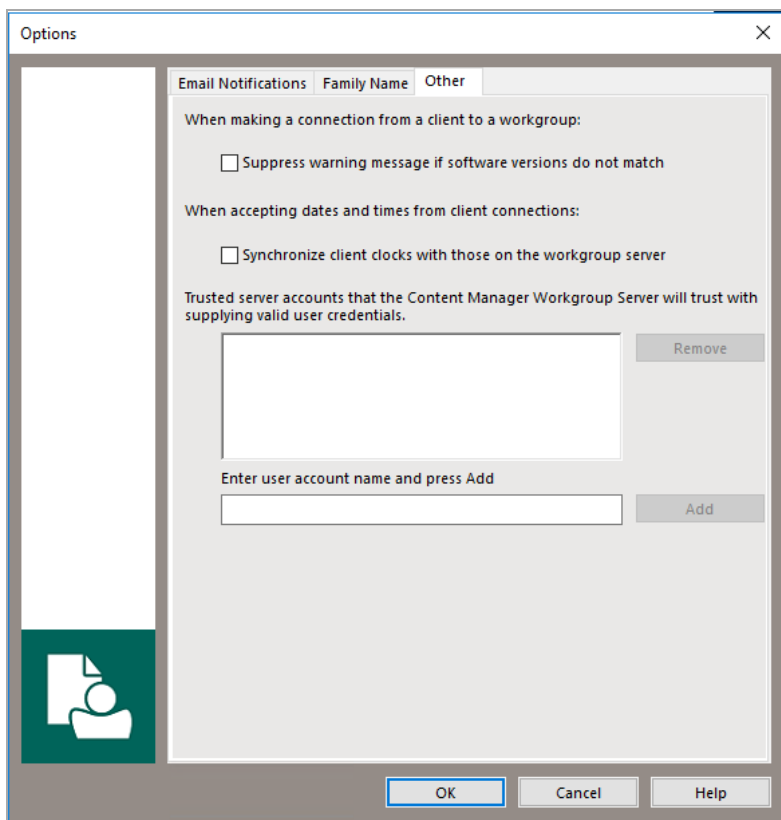
3. In the field **Enter user account name type** the full account name (domain\user) for the Content Manager Web Client and then click **Add**.
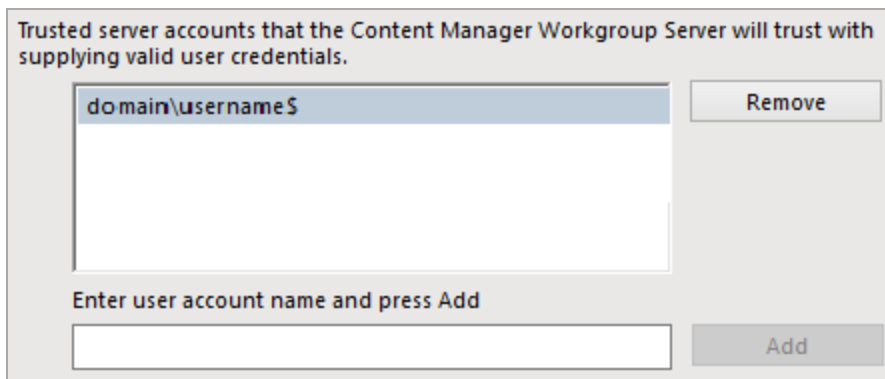


4. Click **OK**.

5. On the **Content Manager Enterprise Studio** dialog, from the **Home** tab, on the **File** group, click **Save** and then click **Deploy**.

By default, the Content Manager Web Client runs under the IIS application pool's identity which is Network Services. When the Content Manager Web Client runs as Network Services, IIS will try to log onto the Content Manager Workgroup Server as **domainName\computerName$**, where domain name is your organization domain and **computerName$** is your computer account. This domain account is created automatically when the computer is joined to the domain.

For the Network Services account to be able to log on to the Content Manager Workgroup Server, change the Content Manager Web Client identity from the Network Services account to the account that runs the Content Manager Workgroup Server . Otherwise, add the computer account of the computer that the Content Manager Web Client is running on (**domainName\computerName$**) to the trusted account list in Content Manager Enterprise Studio.

By default, the Content Manager Web Client runs as Network Services and will try to connect to the Content Manager Workgroup Server as TRIM\WEB$. You can change the Content Manager Web Client application identity by creating a new IIS application pool and set the new application pool identity to a domain service account. If this account is the same as the Content Manager Workgroup Server account, you will not have to add this account to the trusted list in Content Manager Enterprise Studio.

**Example: running Content Manager Web Client as a domain account**

For the Content Manager Web Client to connect to the Content Manager Workgroup Server as domain account, the following conditions must be met:

- Content Manager Web Client identity must be set by changing the application pool identity to the domain account that you want

- The domain account must be in the trusted account list in Content Manager Enterprise Studio
  You will need to change the application pool identity that the Content Manager Web Client is using. It is recommended to create a new application pool and set this pool identity to the domain user account. For information on how to do this, please consult Microsoft's guides on how to create an IIS application pool.

- IIS 7+ http://technet.microsoft.com/en-us/library/cc731784(WS.10).aspx

# Prevent Download

If your organization as a requirement to disable the Download functionality of the Web Client so users cannot create local copies of the documents, this can be done by adding a new attribute to Web Client hprmServiceAPI.config

When this attribute is set to true:

- The **Download** option will not be available on the Preview panel

- The **Generate URL Link** option will not be available on the Preview panel, and

- Users will not be able to use the Check Out option from the record details component.

  **NOTE:** The **Check Out and Edit** functionality is not affected by this attribute.

To disable the Download functionality:

1. On the machine where the Web Client is installed and configured, navigate to the installation directory, by default, this is C:\Program Files\Micro Focus\Content Manager\Web Client.

2. As an administrator, open **hprmServiceAPI.config** in a text editing application, e.g. Notepad ++

3. To the **<setup>** section, add **disableDownloadDocument="true"**, e.g.

```
<setup databaseId="45" searchAhead="false" workpath="C:\Micro Focus Content
Manager\ServiceAPIWorkpath\Uploads"
```

```
bypassViewerFileTypes="*.JPEG;*.JPG;*.PNG;*.TXT;*.GIF;*.BMP;*.MPG;*.MPEG;*.XML;*.TI
FF;*.TIF;*.PDF" disableDownloadDocument="true"/>
```

4. Save the updated hprmServiceAPI.config

# Web Client Mobile redirection

By default, if the Web Client is accessed on a mobile phone device, the Web Client will be displayed using the Mobile client view.

This redirection can be disabled by adding the **webToMobileRedirection** attribute and setting it to **false**, to the setup tag in the **hprmServiceAPI.config** file.

For example:

```
<!-- phoenix configuration -->

<setup databaseId="45" searchAhead="false" webToMobileRedirection="false"
advancedSearch="false" workpath="C:\Micro Focus Content
Manager\ServiceAPIWorkpath\Uploads"/>
```

This redirection can be re-enabled by changing the **webToMobileRedirection** attribute and setting it to **true**.

# Allowing specific File Types to be uploaded

To allow the uploading of specific File Types to the Web Client, a white list can be added to the **hprmserviceapi.config** file. This list defines what file types can be uploaded to the Web Client, any file types not listed will fail if uploaded.

To create a white list for the Web Client, add the **fileUploadWhiteList** attribute with the accepted file extensions, separate each allowed extension with a comma, to the setup tag in the **hprmServiceAPI.config** file.

For example:

```
<!-- phoenix configuration -->

<setup databaseId="45" searchAhead="false" webToMobileRedirection="false"
advancedSearch="false" workpath="C:\Micro Focus Content
Manager\ServiceAPIWorkpath\Uploads" fileUploadWhiteList="xlsx,docx,pdf"/>
```

If the **fileUploadWhiteList** attribute is not defined, all file types will be accepted.

# Setting a custom logo

You can now customise the logo in Web Client by changing the image file. To set the custom logo in Web Client, follow these steps:

1. Launch the Web Client. The Micro Focus Content Manager logo is displayed on the top left corner of the Web Client.

2.  Navigate to the following location: **C:\Program Files\Micro Focus\Content Manager\Web Client\Content\img\**.

3.  Replace the **cm_68_68.png** image file with the desired logo.

    > **NOTE:** Make sure that the file name of the new image is same as the old one, i.e., **cm_68_68.png**.

4.  Refresh the Web Client system.

    The new logo is displayed on the top left corner of the Web Client.

# Allow file scanning before uploading

To allow file scanning before uploading to the Workgroup server or Document Store, you need to add a configuration attribute. By setting the value of this attribute, you can delay the WebClient record creation process. When you upload the file to the server or document store, a progress message is displayed that the file is being scanned.

To allow file scanning, add the **delayUploadedFileTransferTime** attribute to the setup tag in the **hprmServiceAPI.config** file.

For example,

```
<!--phoenix configuration-->

<setup databaseId="45" searchAhead="false" workpath="C:\Micro Focus Content
Manager\ServiceAPIWorkpath\Uploads"
bypassViewerFileTypes="*.JPEG;*.JPG;*.PNG;*.TXT;*.GIF;*.BMP;*.MPG;*.MPEG;*.XML;*.TI
FF;*.TIF;*.PDF" webToMobileRedirection="true" delayUploadedFileTransferTime="5000"
/>
```

If the **delayUploadedFileTransferTime** attribute is set to 0 or not added in the <setup> tag means the feature is disabled. To enable this feature, set the value to any integer greater than or equal to 2 seconds.

# Blocked Search Methods

If a Content Manager Administrator blocks a search method, it will not be visible in the Web Client advanced search editor. However, user can still perform a string search which sends the query to the Content Manager Web Service. By default, these blocked searches will not be blocked by the Content Manager Web Service and will still be executed from the Web Client string search.

In order to force the Web Service to block the blocked search methods from the string search, **preventBlockedSearchMethod** needs to be set to **true** in the Searching config element in **hprmServiceApi.config** under the Web Client install directory.

For example:

```
<hptrim>

...
```

```
<searching pageSize="30" searchRecursiveOption="_$_"
preventBlockedSearchMethod="true" />

....

</hptrim>
```

If the **preventBlockedSearchMethod** attribute is not present, the default value is false which means blocked search method will still be executed from the string search field from the Web Client.

# Prevent Link Injection

To prevent users from inserting malicious link in to the application, a new attribute can be added to the **hprmServiceAPI.config** file. Enabling this attribute prevents user from modifying the links when viewing the documents.

Add the attribute **disableKeyViewURL** and set the value to **true** in the **setup** tag of **hprmServiceAPI.config** file.

For example:

```
<setup databaseId="45" disableKeyViewURL="true" searchAhead="false"
webToMobileRedirection="false"
advancedSearch="false" workpath="C:\HPE Content Manager\ServiceAPIWorkpath\Uploads"
fileUploadWhiteList="xlsx,docx,pdf"/>
```

When you view the document, the link is now not clickable.

# Prevent downloading/checking out corrupt document with incorrect file size

During upgrade, the document size may be different than the actual size of the document. A corrupted document is displayed when you try viewing or downloading this document. To prevent this, add the **preFetchFiles = "true"** parameter to the **hprmServiceAPI.config** file.

**To add the parameter, perform the following:**

1. Navigate to installation folder. For example, C:\Program Files\Micro Focus\Content Manager\Web Client.

2. Open the **hprmServiceAPI.config** file in a text editor.

3. Add the **preFetchFiles** parameter and set it to **true**.

   For example,

   ```
   <hptrim serviceFeatures="Razor,Html,Json,Xml,PredefinedRoutes" preFetchFiles =
   "true" bulkLoaderPort="7791" uploadedFileLifetime="5" fieldPrefix=""
   uploadBasePath="C:\Micro Focus Content Manager\ServiceAPIWorkpath\Uploads"
   autoPoolClean="true" debugMode="false"
   xmlns="http://HP.HPTRIM.CMIS/hptrimConfig.xsd"
   ```

```
xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
xsi:noNamespaceSchemaLocation="hptrimConfig.xsd">
```

4. Save the configuration file.

# Troubleshooting

## Authentication Prompts

### Issue

Content Manager Web Client prompts users for authentication although their credentials are valid.

### Solution

In **Web.config**, **set clientCredentialType** to **Windows** if it is set to **NTLM**:

```
<transport clientCredentialType="Windows" />
```

# Windows Integrated Authentication and Internet Explorer

Internet Explorer prompts for a password when you are using Windows integrated authentication, also known as Microsoft Windows NT challenge/response or NTCR.

The following conditions must be met for Internet Explorer to automatically authenticate a user's logon and password and maintain security:

- NTCR must be enabled in the Web site properties in IIS. Anonymous authentication is attempted first, followed by NTCR, digest authentication (if applicable), and finally basic (clear text) authentication.

- Both the client and the Web Server must be either in the same Microsoft Windows NT-based or Windows 2000-based domain or in trusted Windows NT-based or Windows 2000-based domains in which the user's account can be granted permissions to resources on the IIS-based computer

- The user's browser must be Internet Explorer. Internet Explorer is the only browser that supports NTCR

- Internet Explorer must consider the requested URL to be on the intranet (local). If the computer name portion of the requested URL contains periods (such as http://www.microsoft.com and http://10.0.0.1), Internet Explorer assumes that the requested address exists on the Internet and does not pass any credentials automatically. Addresses without periods (such as http://webserver) are considered to be on the intranet (local); Internet Explorer passes credentials automatically. The only exception is addresses included in the Intranet zone in Internet Explorer.

- Internet Explorer's Intranet zone security setting must be set to Automatic logon only in Intranet zone. This is the default setting for Internet Explorer. For additional information about Internet Explorer security zones, click the article number below to view the article in the Microsoft Knowledge Base:

Source: See http://support.microsoft.com/kb/258063

# A Note to Administrators Regarding Systems Options

When the system options are changed using the Content Manager client, the IIS application pool associated with the Content Manager Web Client virtual directory or the Web site needs to be recycled.

This will not affect other Web sites or virtual directories on the IIS server. The assumption is that the implementation of the Content Manager Web Client needs to have its own application pool.

1.  Open **Internet Information Services (IIS) Manager**.

2.  Expand the Server Name, click on **Application Pools**.

3.  Right-click on **ContentManagerAppPool**, click **Recycle** from the **Application Pool Tasks**.

# Installer Behavior When Installing to Invalid Web Site

When installing Content Manager Web Client to an invalid Web site, e.g. one that uses port 80 and no host name, the installer installs Content Manager Web Client to the default Web site instead.

If the Web site that the installer used is incorrect, you can use IIS to move the installation. See Moving the installation to another Web site below

# Moving the installation to another Web site

1.  In IIS, right-click the Web site you want the Web Client to work under and using **Add Application**, point it to the **Web Client directory**.

2.  Complete the required details on the **Add Application** dialog and then click **OK**.

3.  Change the port or add the host name under **Bindings**.

4.  To avoid confusion, under **Default Web Site**, remove **Content Manager Web Client** by right-clicking it and clicking **Remove**.

# Editing hprmServiceAPI.config

The file **hprmServiceAPI.config** in the installation folder has the following section, which the installation sets automatically:

```
<?xml version="1.0" encoding="utf-8" ?>

<hptrim>

<setup databaseId="45"  useBrowserPDFViewer="false"

searchAhead="true" workpath="C:\Micro Focus Content
Manager\ServiceAPIWorkpath\Uploads" useADFS="false" />

</hptrim>
```

When the settings are not correct, you should correct them manually.

The parameters refer to:

- **DatabaseId** - database id

- **useBrowserPDFViewer** - the value indicates whether the PDF document is viewed in its native format using a PDF Viewer or HTML

- **SearchAhead** – the value indicates whether the automatic type ahead when searching is turned on or off

- **Workpath** - the path where the Content Manager Web Client stores temporary user uploaded files before the database stores these temporary files

- **useADFS** - the value indicates whether Single Sign On authentication for the Content Manager Web Client is on or off

# Cross-site Request Forgery

**Background**

"A common type of attack on websites is referred to as cross-site request forgery (often abbreviated as CSFR or XSFR). When users visit a malicious website or open a malicious email message or instant message, code can attach to their browser and can secretly submit harmful requests on a site where the users are authenticated. In effect, the malicious site forges ("spoofs") requests so that they appear to come from a legitimate user."

From https://msdn.microsoft.com/en-us/library/system.web.helpers.antiforgery(v=vs.111).aspx

In 9.0, the Content Manager Web Client introduced an Anti-forgery mechanism to protect users from such attack. This Anti-forgery mechanism is turned off by default. This can be turned on by adding the **requireAntiForgeryToken="true"** within the <hptrim> element in **hprmServiceAPI.config** in Web Client installation directory, by default, this is C:\Program Files\Micro Focus\Content Manager\Web Client

For example:

```
<hptrim requireAntiForgeryToken="true" poolSize="1000" indexPagePath="/Home"
notFoundErrorHandler="/APIErrorPages/NotFound"
globalErrorHandler="/APIErrorPages/GlobalErrors" uploadBasePath="C:\Micro Focus
Content Manager\ServiceAPIWorkpath\Uploads" autoPoolClean="true"
serviceFeatures="Html,Json,Razor,Xml"
xmlns="http://HP.HPTRIM.CMIS/hptrimConfig.xsd"
xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
xsi:noNamespaceSchemaLocation="hptrimConfig.xsd">
```

The effect of this is, when user logs in, the Content Manager Web Client will generate an anti-forgery token to be included for all HTTP POST to the server for the current user. POST request will be rejected if it does not include the anti-forgery token.

This applies to all POST request such as : create, update record, file upload.

# Error navigating to the Content Manager Web Client

## Issue

Error received when attempting to navigate to the Content Manager Web Client after installation:
Server Error in '/WebClient' Application.

Could not load type 'System.ServiceModel.Activation.HttpModule' from assembly 'System.ServiceModel, Version=3.0.0.0, Culture=neutral, PublicKeyToken=b77a5c561934e089'.

**Description:** An unhandled exception occurred during the execution of the current web request. Please review the stack trace for more information about the error and where it originated in the code.

**Exception Details:** System.TypeLoadException: Could not load type 'System.ServiceModel.Activation.HttpModule' from assembly 'System.ServiceModel, Version=3.0.0.0, Culture=neutral, PublicKeyToken=b77a5c561934e089'.

**Source Error:**

An unhandled exception was generated during the execution of the current web request. Information regarding the origin and location of the exception can be identified using the exception st below.

## Summary

The error message may appear when trying to navigate to the Content Manager for the first time.

You may encounter this error message when either an earlier version of the .NET Framework has been installed or .NET 3.0 WCF HTTP Activation is enabled after .NET Framework 4.0/4.5 has been installed.

## Solution

Run the following command line:

Aspnet_regiis.exe. /iru

The Aspnet_regiis.exe file can be found in one of the following locations:

- %windir%\Microsoft.NET\Framework\v4.0.30319

- %windir%\Microsoft.NET\Framework64\v4.0.30319 (on a 64-bit computer)

Source: See http://support.microsoft.com/kb/2015129

# Getting 401 unauthorized when using FQDN or custom host header

## Issue

When navigating to the Content Manager Web Client using either the fully qualified domain name (FQDN) or a custom host header, *HTTP 401.1 – Unauthorized: Login Failed* message is displayed.

## Summary

The error message is displayed even when the user enters correct credentials when prompted. This occurs when the CM Web Client uses Integrated Windows Authentication and has a name that is mapped to the local loopback address.

# Changing the Date Format when using the (.) separator

## Issue

If a user changes the Date Separator from either a forward slash (/) or a dash (-) to a period, and then changes the Date Format setting to either dd/mm/yyyy or mm/dd/yyyy, date searches might not return correct results.

## Solution

Wait for the application pool to recycle automatically in due time or manually recycle the application pool.

# Appendix A
# Configuring WebDav

In Content Manager 9.0, a new **Check Out and Edit** option was introduced as an addition to the existing Check Out and Check In options that check the document out to a local directory and requires the user to browse to that location to check the document in after it has been modified. This new option allows users, accessing the Content Manager Web Client via a supported version of Internet Explorer, to check out an electronic document and edit it directly in its authoring application.

This new option utilizes a workpath directory on the Web Client Web Server. When a user checks out a document using this new option, the document is saved as a working copy in a sub-directory on the Web Client Web Server. The user can make edits and save the document, and at their convenience, using the Content Manager Web Client, they can check the document back into the Content Manager database.

This appendix covers the configuration of the Server, Internet Information Services Manager (IIS) and Web Server directory to enable the new WebDAV **Check Out and Edit** option.

> **NOTE:** this feature does not work when using ADFS for Web Client Authentication. If you are using ADFS Authentication **do not** complete the following WebDAV configuration steps in your environment.

## Configuring and Installing WebDav Publishing

> **NOTE:** these instructions are for a Web Server that is already configured for the Content Manager Web Client. If you are a new site, please following the installation and configuration instructions found in Installation and Configuration

To enable the new Check Out and Edit option, the Web Server needs to have the WebDav Publishing role installed.
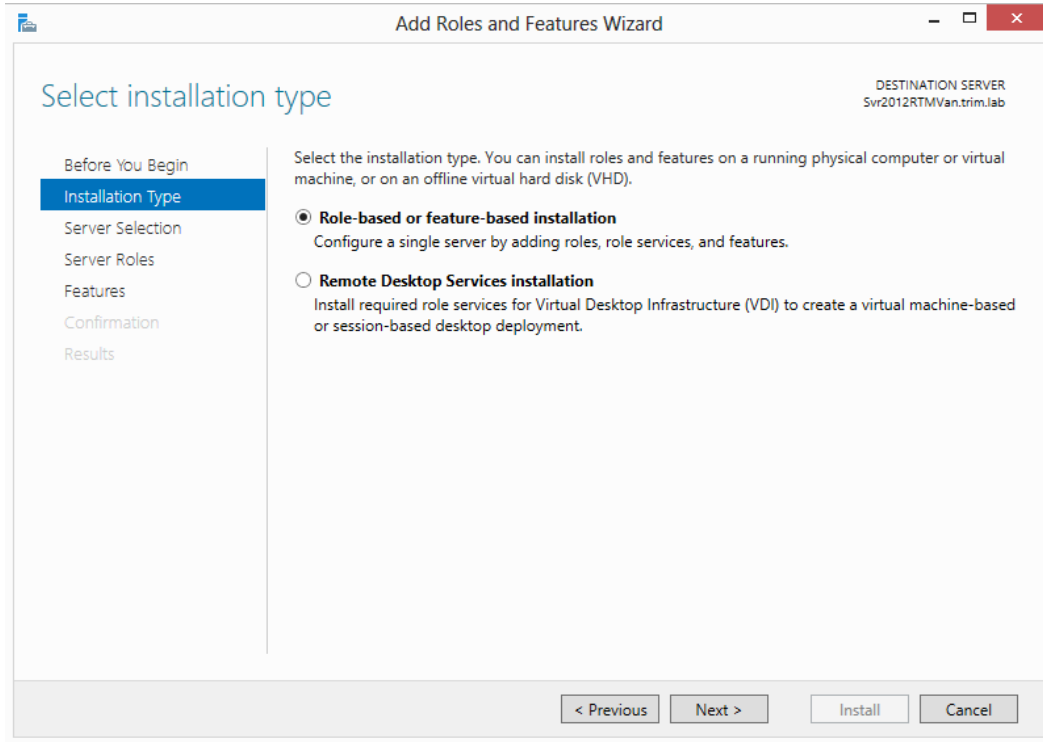
1. Open the Server Manager.

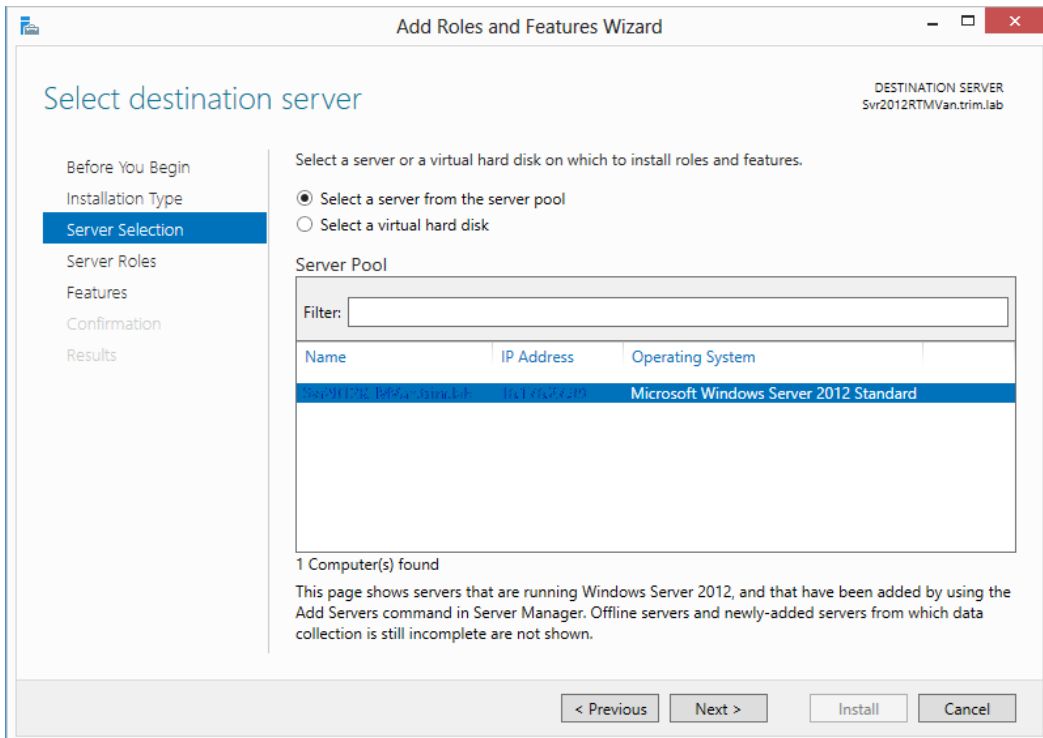2. On the Server Manager Dashboard, click **Add roles and features**.
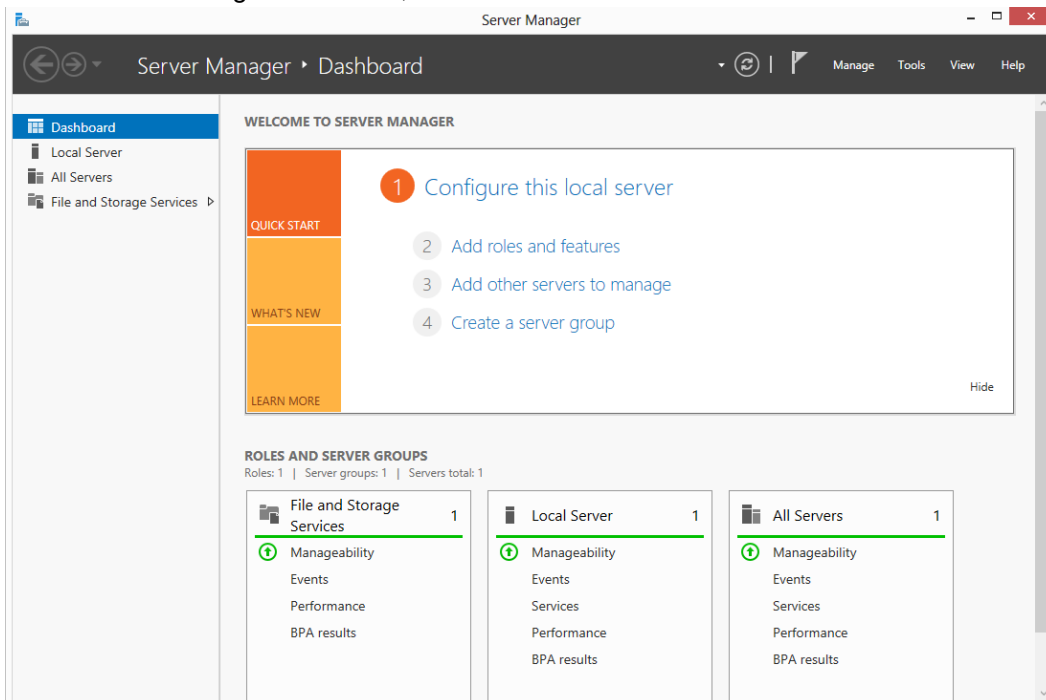


3. Click **Next** on the **Add Roles and Features Wizard**.
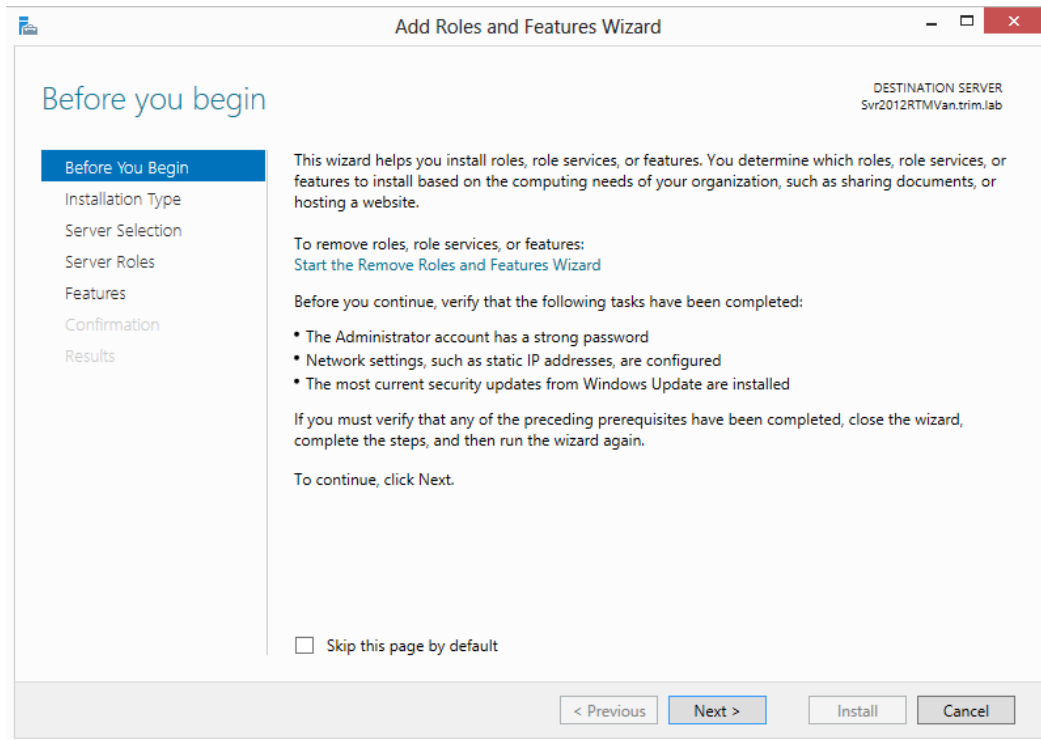
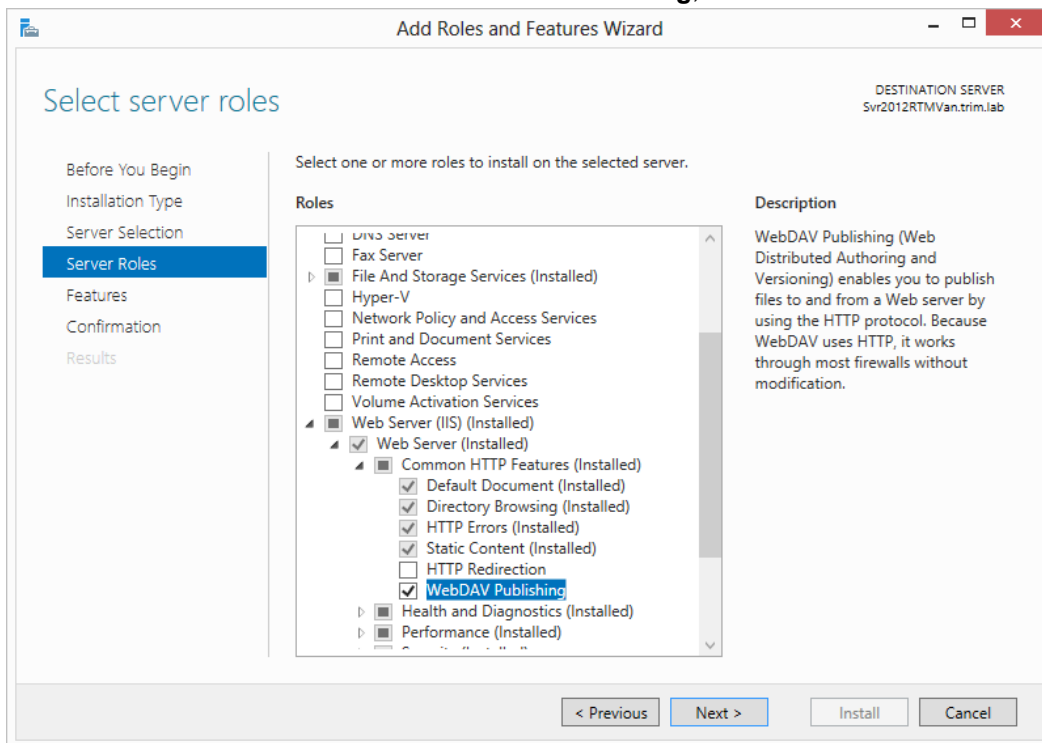4. Select Role-based or feature-based installation and then click **Next**.



5. Select the Server that the Roles and Features are to be installed on, and then click **Next**.

6. From the displayed Roles list, expand **Web Server (IIS)**, navigate to **Web Server >
Common HTTP Features** and select **WebDav Publishing**, and then click **Next**.



7. Click **Next** on the **Select features** dialog. The **Confirmation installation selections** dialog will
be displayed.

8. Click **Install**. Once the installation is complete, click **Close**.

# Enabling and Configuring WebDAV in Internet Information Services (IIS)

## Enabling and Configuring the WebDAV Settings for IIS

1. Open Internet Information Services (IIS) Manager.

2. Under the defined Web Server name, expand the **Site** node and select the site where the Web
Client is installed, e.g. **Default Web Site**.

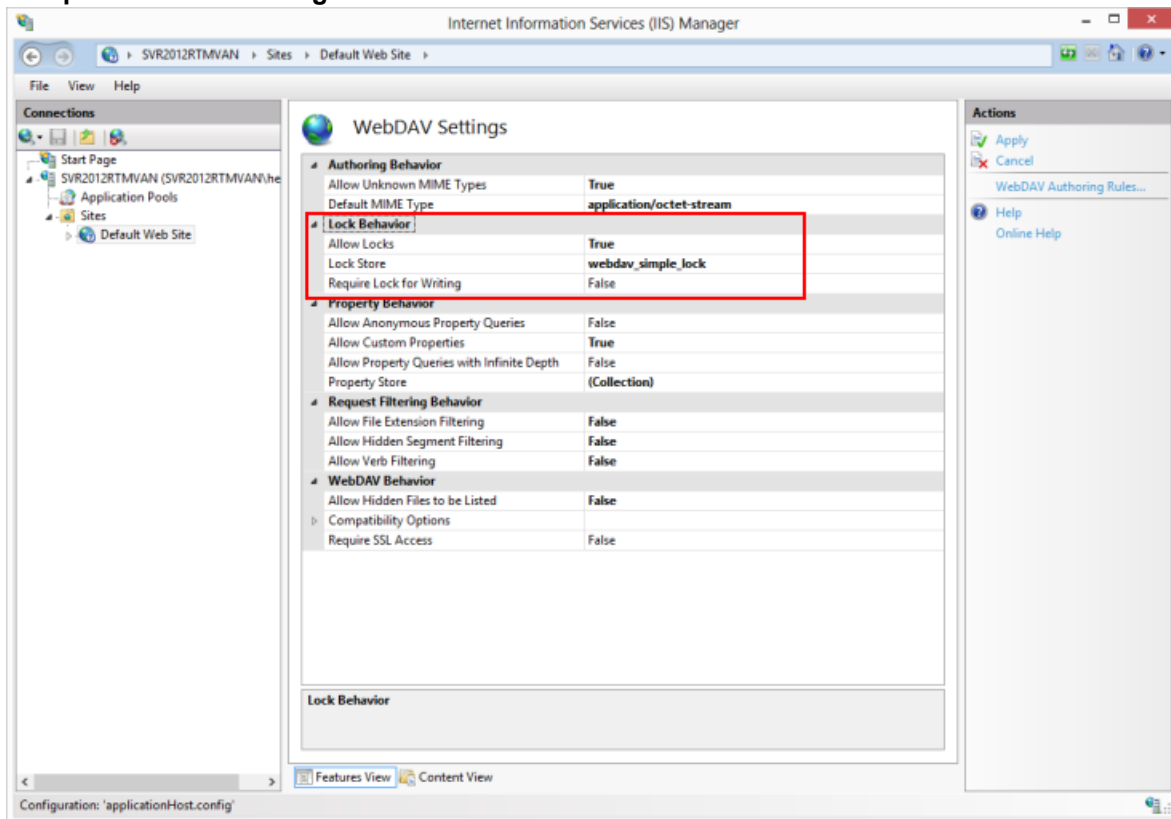3. From the **IIS** group, select and open **WebDAV Authoring Rules**.
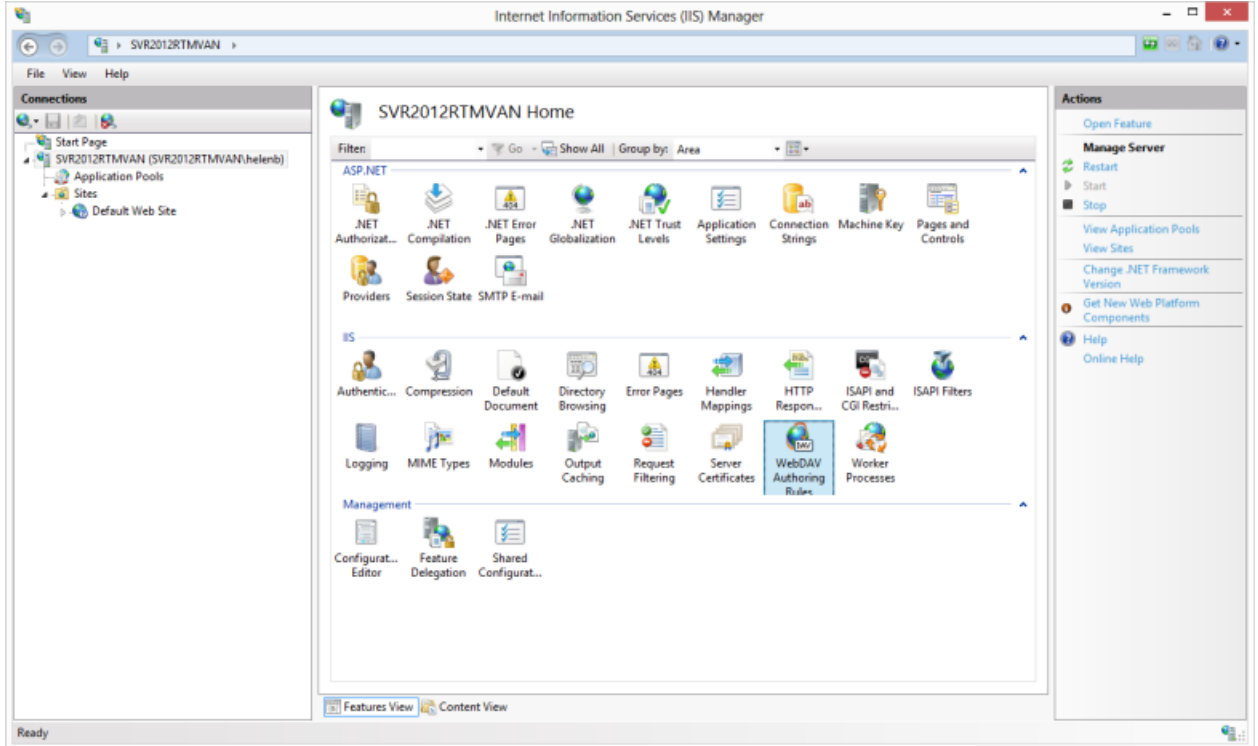


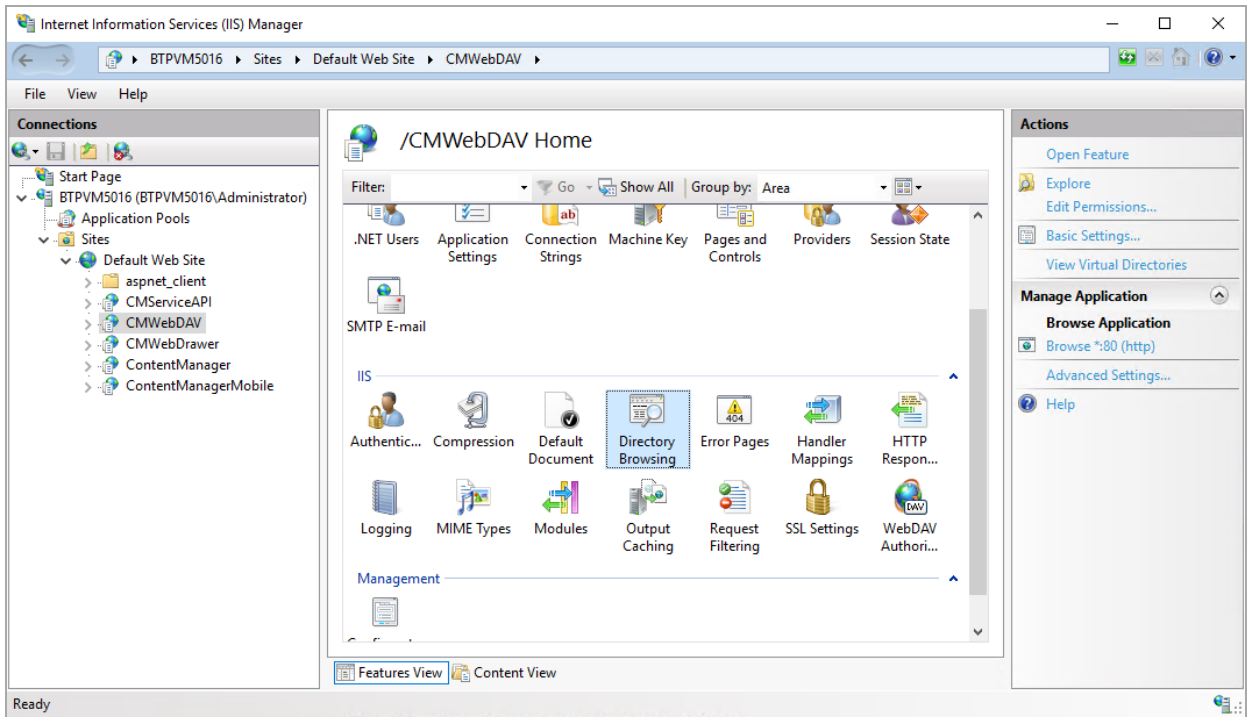4. On the **WebDAV Authoring Rules** dialog, click **Enable WebDAV**.
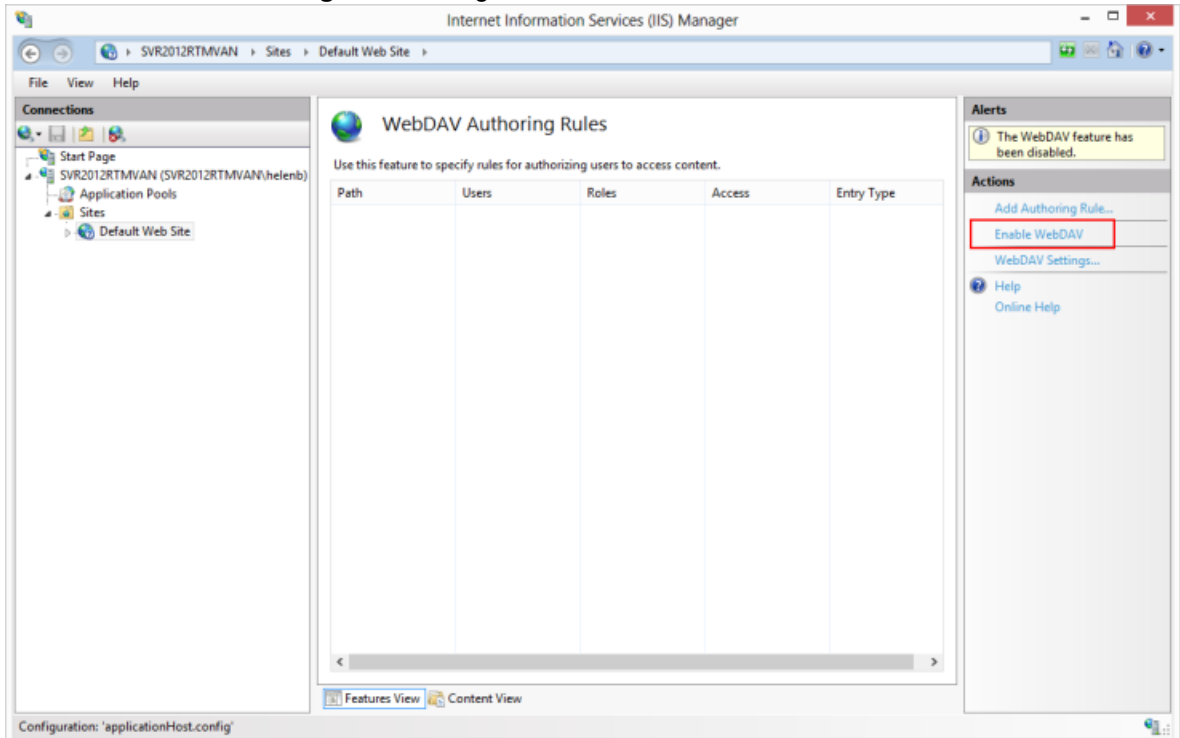


5. On the **WebDAV Authoring Rules** dialog, click **WebDav Settings**.

6. On the **WebDAV Settings** dialog, set the **Lock Behavior** options as follows, and the click **Apply**:
   - **Allow Locks** - True
   - **Lock Store** - webdav_simple_lock
   - **Require Lock for Writing** - False



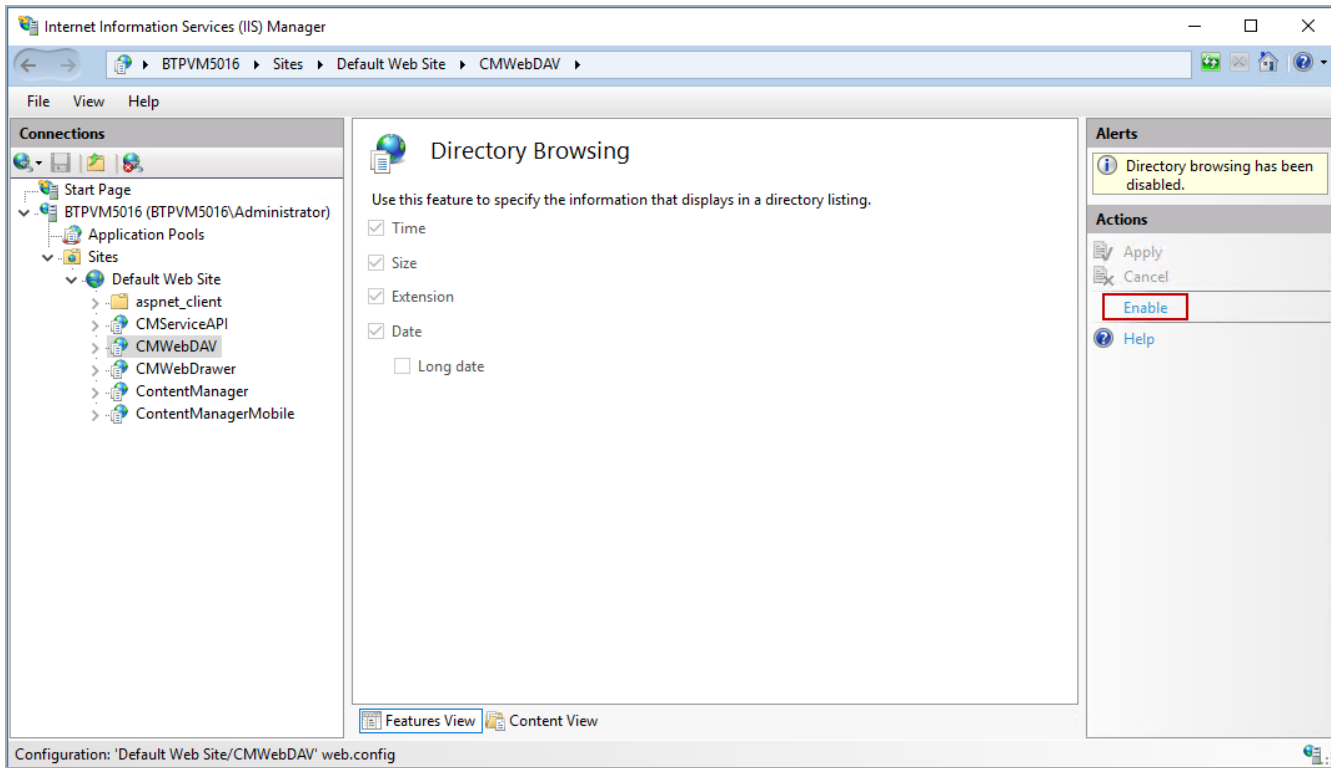## Enabling and Configuring Directory Browsing for CMWebDav

1. Once the WebDAV settings have been applied, on the **Connections** panel, expand the **Sites** node, then expand the site node where the Web Client is installed, e.g. **Default Web Site** and select the **CMWebDAV** site.

2. From the **IIS** group, select and open **Directory Browsing**.

3. On the **Directory Browsing** dialog, click **Enable**.

# Adding WebDAV Authoring Rules for CMWebDAV

1. Once **Directory Browsing** has been enabled, on the **Connections** panel, expand the **Sites** node, then expand the site where the Web Client is installed, e.g. **Default Web Site** > **CMWebDAV** nodes and select **DAVDir**.

2. From the **IIS** group, select and open **WebDAV Authoring Rules**.
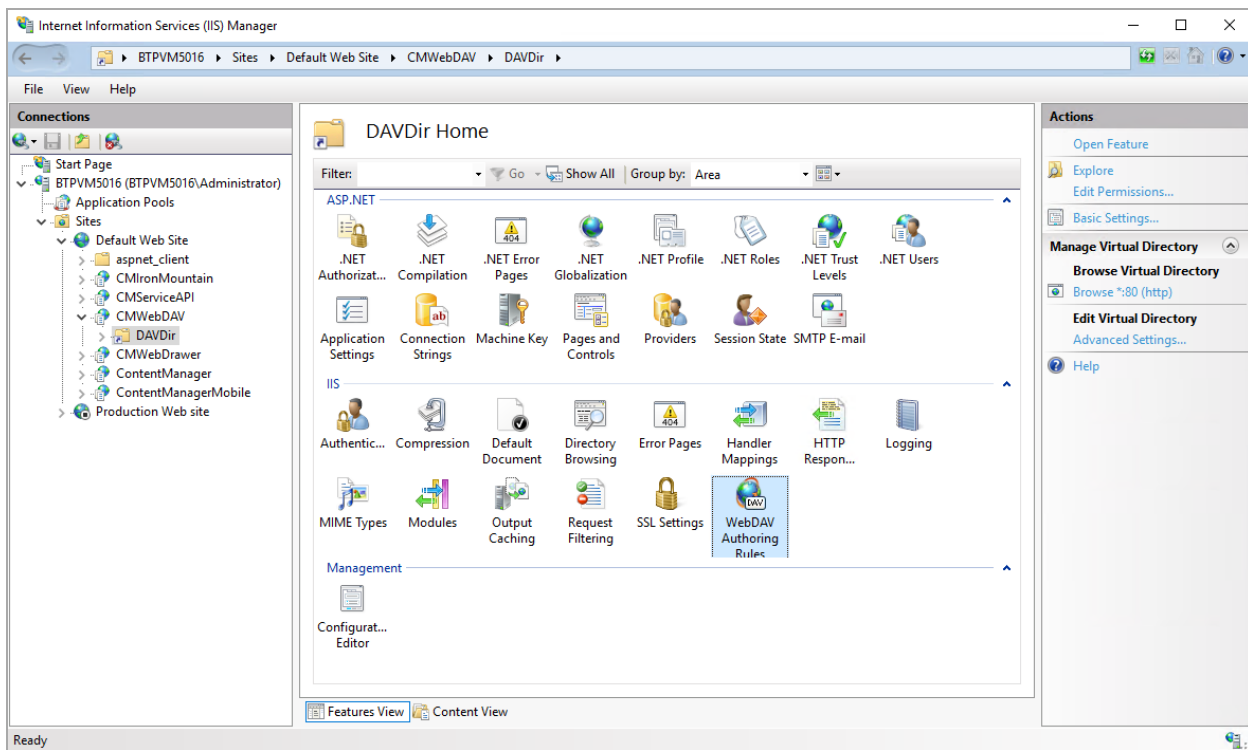


3. On the **WebDAV Authoring Rules** dialog, click **Add Authoring Rule**. The **Add Authoring Rule** dialog will appear.

4. Create a new rule with the following options selected:
   - **Allow access to** - All content
   - **Allow access to this content to** - All users

- **Permissions** - select Read, Source and Write
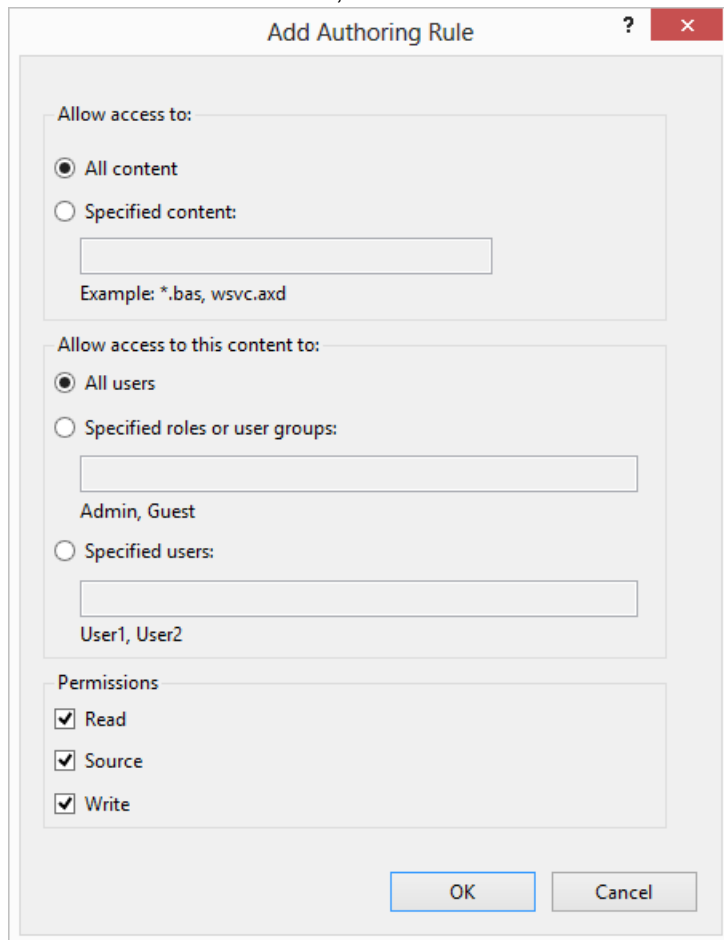


5.  Click **OK**.

# Configuring the CMWebDAV Workpath Folder

On the Web Server, in the Content Manager Web Client workpath folder, the **CMWebDAV** folder, the Windows File Sharing settings must be updated so all users of the Web Client have Change and Read permissions to this directory. Each document a user checks out and edits using the **Check Out and edit** option in the Web Client, will be saved as a working copy to a sub-directory beneath this directory. Each sub-directory and working copy files are restricted to the Content Manager user who checked them out.

1.  On the Content Manager Web Client Web Server, using Windows Explorer, navigate to and open the WebClientWorkPath folder, by default this will be installed to C:\Micro Focus Content Manager\WebClientWorkpath.

2.  On **CMWebDAV** folder, right-click and click **Properties**.

Alternatively, in IIS, on the **Connections** panel, expand the **Sites** node, then expand the site node where the Web Client is installed, e.g. **Default Web Site** and select the **CMWebDAV** site, then click **Edit Permissions** from the **Actions** panel.

The **CMWebDAV Properties** dialog will appear.

1. On the **Sharing** tab, click **Advanced Sharing**.
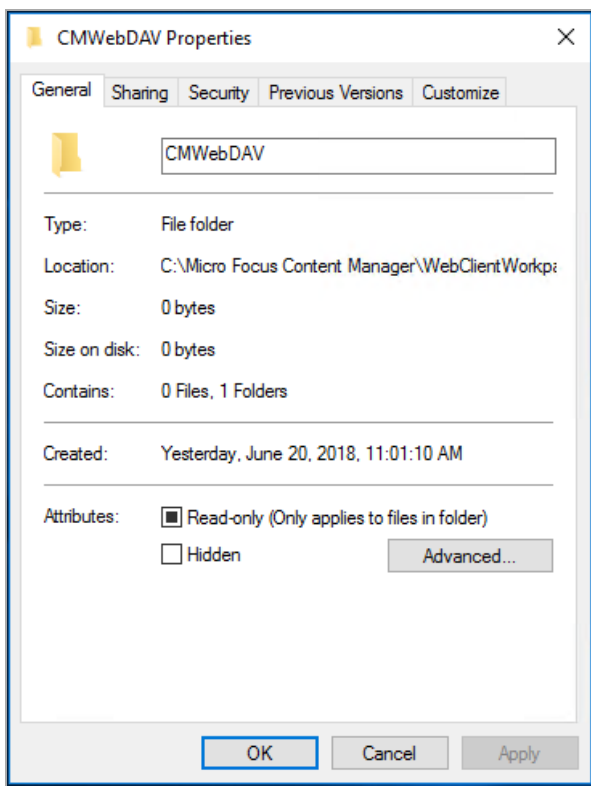


2. On the **Advanced Sharing** dialog, select **Share this folder** and then click **Permissions**.
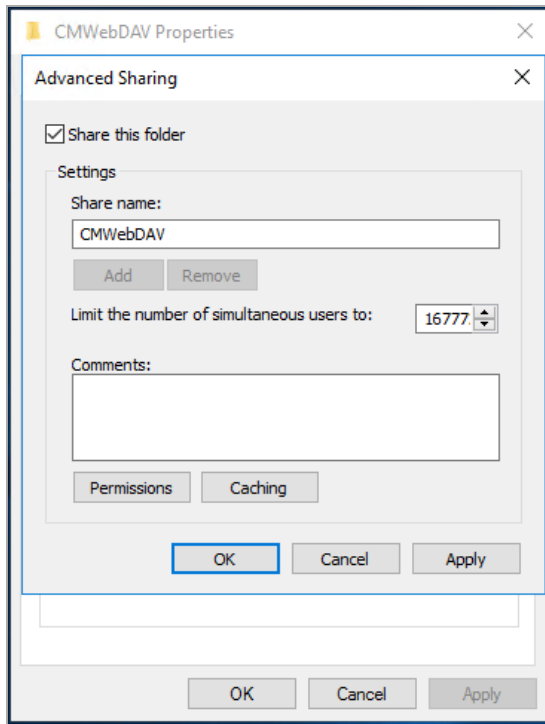
3. On the **Permissions for CMWebDav** dialog, from the list of **Group or user names**, select **Everyone**.

4. From the displayed list of **Permissions for Everyone**, select **Change**, and if not already selected, **Read**.

5. Click **Apply** and then click **OK** until you're back to the **CMWebDAV Properties** dialog, then click **Close** to save the new Sharing permissions.
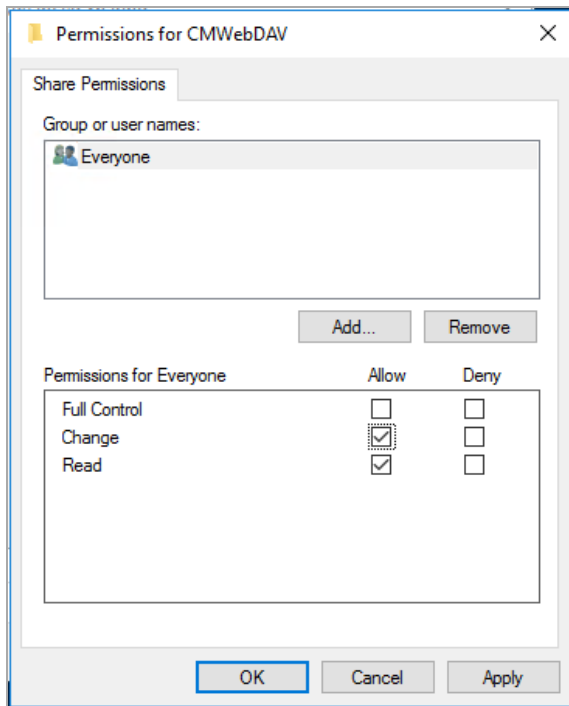
# Configuring hprmServiceAPI.config

When installed the **hprmServiceAPI.config** file will contain the relevant section enabling it to load the WebDAV plugin. This section looks like:

```
<pluginAssemblies>

        <add name="HP.HPTRIM.WebClient.WebDAV"/>

</pluginAssemblies>
```

> **NOTE:** If you have upgraded from an earlier version of Records Manager you will need to navigate to the installation directory for the Web Client, by default, for a new installation, this is installed to C:\Program Files\Micro Focus\Content Manager\Web Client and open the **hprmServiceAPI.config** file and edit the <add name> property to uncomment <add name="HP.HPTRIM.WebClient.WebDAV"/> before saving the changes. If you have made any customizations to the **hprmServiceAPI.config** file you will need to manually copy the customizations from the **hprmServiceAPI.config**file that is copied to the WebClientWorkpath directory and paste them into the installed **hprmServiceAPI.config**
>
> **Note:** If you've enabled the WebDAV module in **hprmServiceAPI.config** and the **Check out and Edit** option is not available in the Web Client, check the logs for errors stating why WebDAV can't be enabled. By default, the error log files are found C:\Micro Focus Content Manager\ServiceAPIWorkpath\logs

# Client Configuration

When a user checks out and edits a document from Content Manager Web Client using the WebDAV check out and edit option, a warning message will appear:
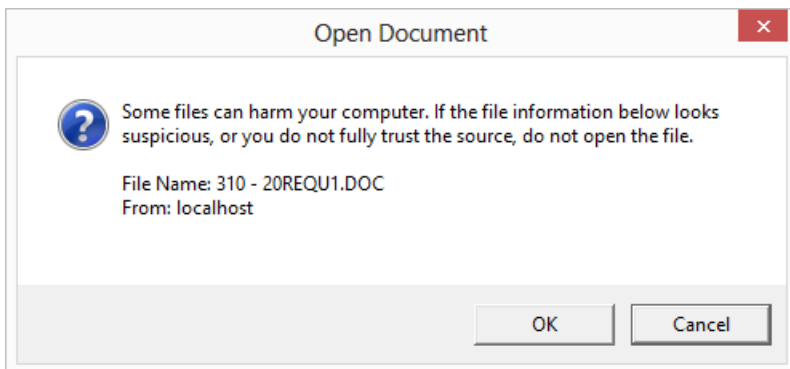


This message can be suppressed for MS Office documents, but cannot be stopped for non-Office files, for example text files.

> **NOTE:** If the Content Manager Web Client has been set up on an IIS site that uses a port different than the standard Port 80, then users may need to dismiss this warning twice when editing non office files.

To suppress this warning message:

1. On the Client machine, open the **Internet Options** control panel.

2. On the **Security** tab, select **Trusted sites** from the list of zones to view or change security settings and then click **Sites**.

3. On the displayed **Trusted sites** dialog, type in the IP address or host name of the Content Manager Web Server to the **Add this website to the zone** field, and then click **Add**.

4. Click **Close** to add the site to the Trusted sites for the client machine.

If you choose to add the Content Manager Web Client to the Trusted sites, you must also ensure that the User Authentication settings for the Trusted sites zone is set to **Automatic Logon with current user name and password**.

To set this authentication:

1. On the Client machine, open the **Internet Options** control panel.

2. On the **Security** tab, select **Trusted sites** from the list of zones to view or change security settings and then click **Custom level**.

3. On the displayed **Security Settings - Trusted Sites Zone** dialog, scroll through the list of Settings to the **User Authentication** options and select **Automatic logon with current user name and password**.

4. Click **OK** to save these changes.

> **NOTE:** Due to this authentication requirement, the WebDAV Check out and edit option does not support Guest account access to the Content Manager Web Client. All users must have a valid Windows Domain account that can automatically authenticate to the Web Server.

# WebDAV with Load Balancing

Organizations who want to use the WebDAV **Check Out and edit** option in an environment that uses Load Balanced servers will need to modify their WebClient configuration to enable this to work.

If you need multiple server instances, then each instance needs to store and access working copies in the same WebDAV share, and direct the Web browser clients to that share.

The easiest way to do this is to designate one of the instances to store all the working copies. In Content Manager 9.2 or later this can be done by placing the real name of the designated instance into the Application Settings key named 'WebDAVHost'.

You can also use a WebDAV share on an independent host that isn't part of load balancing, but you will need to create the WebDAV share with exactly the same name, path, protocol & port bindings and settings as would be used by the WebDAV share on a stand alone Content Manager Web server setup.

To configure the Content Manager Web Client to use WebDAV with Load Balancing:

1. As an administrator, open the Web Client **web.config** file using a text editor, by default this is installed to C:\Program Files\Micro Focus\Content Manager\Web Client.

2. In the **<appSettings>** section, modify the **<add key="webDAVHost" value=""/>** so the value string contains the name of the server that holds the WebDAV share to be used by all of the Load Balanced Content Manager Web servers, e.g.

```
<appSettings>

<add key="webpages:Version" value="2.0.0.0"/>

<add key="webpages:Enabled" value="false"/>

<add key="PreserveLoginUrl" value="true"/>

<add key="ClientValidationEnabled" value="true"/>

<add key="UnobtrusiveJavaScriptEnabled" value="true"/>

<add key="webDAVHost" value="CMweb1.loadbalance.com"/>

</appSettings>
```

> **IMPORTANT:** replace the "webDAVHost" value in the highlighted example with your own value.

3. Save and close the **web.config** file.

# Appendix B
# Adding Custom Record Add-ins

Introduced in Records Manager Web Client 8.3, is an option for developers and business partners to add custom code via the record add-in infrastructure. This custom functionality is exposed under the **More** drop-down option on a Record object.
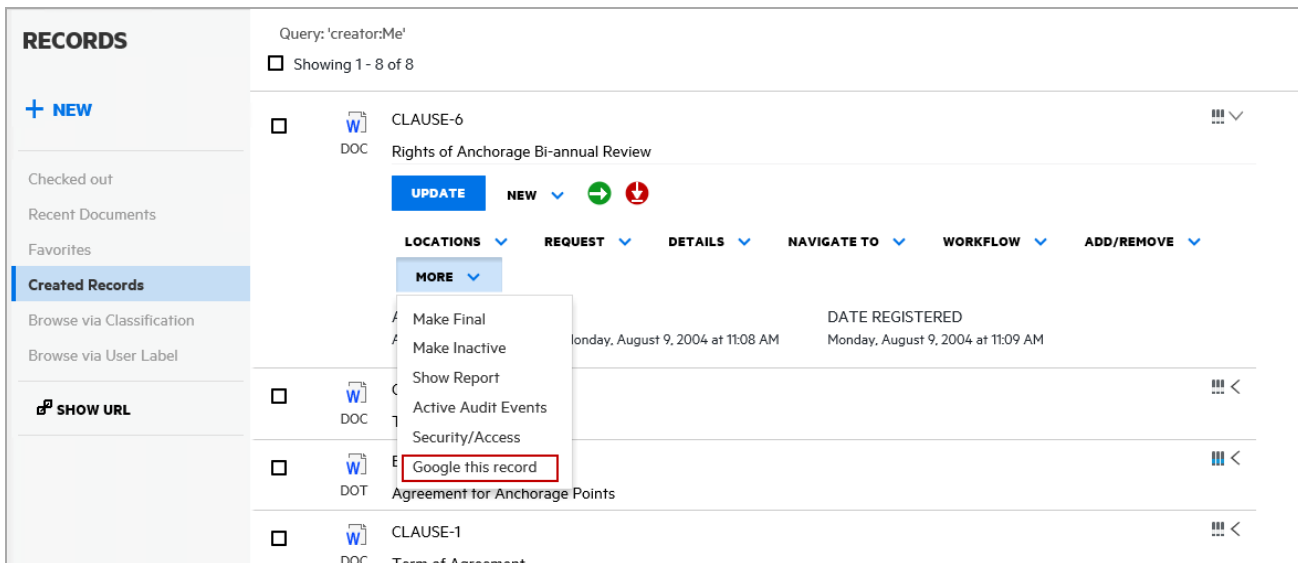


To expose the custom record add-in:

1. In the installation directory for the Web Client, by default this is C:\Program Files\Micro Focus\Content Manager\Web Client, create a folder called **CustomScripts**.

2. Copy the custom code JS file to the **CustomScripts** folder.

3. Open Content Manager Web Client , open a Record and then click **More**. Your new Record Add-in option will be available from the available options.

## Example Simple Record Add-in

The following example shows how to create a custom record add-in that will, when a user clicks on the option, perform a Google search with the selected record's title.

```
//Wrap your code in a function to avoid conflict

var RMGoogleButtonAddon = function(){

var buttonCaption = "Google this record";

//1 - Create a new instance of a RecordAddonButton

var googleSearchButton  = new HP.HPTRIM.Addon.RecordAddonButton ({

caption : buttonCaption,

clickHandler : function(){

console.log(this.context);

window.open("http://google.com/search?q=" + this.context.RecordTitle.Value)

}

});

//Optional - Perform custom checking before the button is rendered

googleSearchButton.preRender = function(){

var record = this.context;

if(record.RecordTitle.Value == "Microsoft") {

this.setVisible(false);

}

};

//2 - Register the addon button with CustomScriptManager

HP.HPTRIM.Addon.CustomScriptManager.register(googleSearchButton);

}();
```

## RecordContext Properties

The clickHandler parameter in the HP.HPTRIM.Addon.RecordAddonButton constructor will be executed when the button is clicked. In this function, you will have access to the current context which is the current record. The properties of the current record object are:

```
export interface RecordContext {
    TrimType: string;
    RecordAssignee :Object;
    RecordAuthor :Object;
    RecordCheckedOutTo :Object;
    RecordContainer :Object;
    RecordDateAssigned :Object;
    RecordDateCreated :Object;
    RecordDateRegistered :Object;
    RecordDocumentStatus :Object;
    RecordDocumentType :Object;
    RecordExtension :Object;
    RecordHomeLocation :Object;
    RecordIsContainer :Object;
    RecordIsElectronic :Object;
    RecordLastPartRecord :Object;
    RecordNumber :Object;
    RecordOwnerLocation :Object;
    RecordRecordType :Object;
    RecordSpURL :Object;
    RecordTitle :Object;
    IsInFavorites: boolean;
    EnabledCommandIds: Array<string>;
    Uri: number;
}
```

**HP.HPTRIM.Addon.RecordAddonButton Methods/Properties**

RecordAddonButton supports the following methods and properties:

- **preRender**: this method can be assigned to your custom code when you want to perform any custom logic before the button is rendered

- **setVisible**: this method takes in a Boolean value which will set the visibility of the button.

- **caption**: this property represents the caption of the button.

**Deployment**

Once the code is written, it will need to be deployed into the{WebClientInstallDir}/CustomScripts folder where WebClientInstallDir is the Content Manager Web Client Installation directory. The CustomScripts folder is not created as a part of the installation process, you will need to manually create the folder before deploying your code.

# Record Context Property types

The record context properties are on of a number of different object types, the types of these objects are:

| Property | Type |
| --- | --- |
| RecordAssignee | Location |
| RecordAuthor | Location |
| RecordCheckedOutTo | Location |
| RecordContainer | Record |
| RecordDateAssigned | Date |
| RecordDateCreated | Date |
| RecordDateRegistered | Date |
| RecordDocumentStatus | String |
| RecordDocumentType | String |
| RecordExtension | String |
| RecordHomeLocation | Location |
| RecordIsCheckedOut | Boolean |
| RecordIsContainer | Boolean |
| RecordIsElectronic | Boolean |
| RecordLastPartRecord | Record |
| RecordNumber | String |
| RecordOwnerLocation | Location |
| RecordRecordType | Record Type |
| RecordRequests | String |
| RecordSpURL | String |
| RecordTitle | String |
| IsInFavorites | Primitive Boolean |
| EnabledCommandIds | Array of primitive string |
| Icon | Icon |
| Uri | Primitive number |

# Example Object Types

## Example object types

The following objects are JSON examples of the types found in the Record context.

**Location**

```
{
"TrimType": "Location",
"LocationFormattedName": {
"Value": "Full name of Location"
},
"Uri": 9000000000,
"StringValue": "Short name of location"
}
```

**Record**

```
{
"TrimType": "Record",
"RecordExtension": {
"Value": "PDF"
},
"RecordIsElectronic": {
"Value": true
},
"RecordNumber": {
"Value": "REC_411"
},
"RecordTitle": {
"Value": "My Record Title"
},
"Uri": 9000000378,
"StringValue": "REC_411",
"Icon": {
"IsFileTypeIcon": true,
"IsInternalIcon": false,
"IsValid": true,
"FileType": "PDF",
"Id": "Unknown"
}
}
```

**Date**

```
{
"IsClear": false,
"IsTimeClear": false,
"DateTime": "2015-11-08T21:48:12.0000000Z",
"StringValue": "9/11/2015 8:48 AM"
}
```

**String**

```
{
"Value": "A string",
"StringValue": "A string"
}
```

**Boolean**

```
{
"Value": false,
"StringValue": "Checked In"
}
```

### Record Type

```
{
"TrimType": "RecordType",
"RecordTypeAllowParts": {
"Value": true
},
"RecordTypeAllowReplace": {
"Value": true
},
"RecordTypeAllowVersions": {
"Value": true
},
"RecordTypeLevel": {
"Value": 2
},
"RecordTypeMoveWhenReadOnly": {
"Value": false
},
"RecordTypeName": {
"Value": "Document"
},
"RecordTypeStoreType": {
"Value": "UseStore",
"StringValue": "Use A document store"
},
"RecordTypeTitlingMethod": {
"Value": "FreeText",
"StringValue": "Free Text"
},
"RecordTypeUsualBehaviour": {
"Value": "Document",
"StringValue": "Document"
```

```
    },
    "Uri": 2,
    "StringValue": "Document",
    "Icon": {
    "IsFileTypeIcon": false,
    "IsInternalIcon": true,
    "IsValid": true,
    "FileType": "",
    "Id": "YellowDoc"
    }
    }
```

**Icon**

```
    {
    "IsFileTypeIcon": true,
    "IsInternalIcon": false,
    "IsValid": true,
    "FileType": "PDF",
    "Id": "Unknown"
    }
```

## Complete Example Record Context

This example shows the properties available on each of the objects in the Record context (summarized earlier in this document). As can be seen most properties are not simple objects but, depending on their type contain a number of properties.

```
{
"TrimType": "Record",
"RecordAssignee": {
"TrimType": "Location",
"LocationFormattedName": {
"Value": "David"
},
"Uri": 1,
"StringValue": "David",
"Icon": {
"IsFileTypeIcon": false,
"IsInternalIcon": true,
"IsValid": true,
"FileType": "",
"Id": "LocPerson"
}
},
"RecordAuthor": {
"TrimType": "Location",
"Uri": 123,
"StringValue": ""
},
"RecordCheckedOutTo": {
"TrimType": "Location",
"Uri": 0,
"StringValue": ""
},
```

```
"RecordContainer": {
"TrimType": "Record",
"Uri": 0,
"StringValue": ""
},
"RecordDateAssigned": {
"IsClear": false,
"IsTimeClear": false,
"DateTime": "2015-11-08T21:48:12.0000000Z",
"StringValue": "9/11/2015 8:48 AM"
},
"RecordDateCreated": {
"IsClear": false,
"IsTimeClear": false,
"DateTime": "2015-08-12T15:22:26.0000000Z",
"StringValue": "13/08/2015 2:22 AM"
},
"RecordDateRegistered": {
"IsClear": false,
"IsTimeClear": false,
"DateTime": "2015-11-08T21:48:12.0000000Z",
"StringValue": "9/11/2015 8:48 AM"
},
"RecordDocumentStatus": {
"Value": "Checked In",
"StringValue": "Checked In"
},
"RecordDocumentType": {
"Value": "Adobe Acrobat Document",
"StringValue": "Adobe Acrobat Document"
},
"RecordExtension": {
"Value": "PDF",
```

```
"StringValue": "PDF"
},
"RecordHomeLocation": {
"TrimType": "Location",
"LocationFormattedName": {
"Value": "Adelaide"
},
"Uri": 9000000000,
"StringValue": "Adelaide",
"Icon": {
"IsFileTypeIcon": false,
"IsInternalIcon": true,
"IsValid": true,
"FileType": "",
"Id": "LocUnit"
}
},
"RecordIsCheckedOut": {
"Value": false,
"StringValue": "Checked In"
},
"RecordIsContainer": {
"Value": false,
"StringValue": "No"
},
"RecordIsElectronic": {
"Value": true,
"StringValue": "Yes"
},
"RecordLastPartRecord": {
"TrimType": "Record",
"RecordExtension": {
"Value": "PDF"
```

```
        },
        "RecordIsElectronic": {
        "Value": true
        },
        "RecordNumber": {
        "Value": "REC_411"
        },
        "RecordTitle": {
        "Value": "15-532471"
        },
        "Uri": 9000000378,
        "StringValue": "REC_411",
        "Icon": {
        "IsFileTypeIcon": true,
        "IsInternalIcon": false,
        "IsValid": true,
        "FileType": "PDF",
        "Id": "Unknown"
        }
        },
        "RecordNumber": {
        "Value": "REC_411",
        "StringValue": "REC_411"
        },
        "RecordOwnerLocation": {
        "TrimType": "Location",
        "LocationFormattedName": {
        "Value": "Adelaide"
        },
        "Uri": 9000000000,
        "StringValue": "Adelaide",
        "Icon": {
        "IsFileTypeIcon": false,
```

```
"IsInternalIcon": true,

"IsValid": true,

"FileType": "",

"Id": "LocUnit"

}

},

"RecordRecordType": {

"TrimType": "RecordType",

"RecordTypeAllowParts": {

"Value": true

},

"RecordTypeAllowReplace": {

"Value": true

},

"RecordTypeAllowVersions": {

"Value": true

},

"RecordTypeLevel": {

"Value": 2

},

"RecordTypeMoveWhenReadOnly": {

"Value": false

},

"RecordTypeName": {

"Value": "Document"

},

"RecordTypeStoreType": {

"Value": "UseStore",

"StringValue": "Use A document store"

},

"RecordTypeTitlingMethod": {

"Value": "FreeText",

"StringValue": "Free Text"
```

```
    },
    "RecordTypeUsualBehaviour": {
    "Value": "Document",
    "StringValue": "Document"
    },
    "Uri": 2,
    "StringValue": "Document",
    "Icon": {
    "IsFileTypeIcon": false,
    "IsInternalIcon": true,
    "IsValid": true,
    "FileType": "",
    "Id": "YellowDoc"
    }
    },
    "RecordRequests": {
    "Value": "",
    "StringValue": ""
    },
    "RecordSpURL": {
    "Value": "",
    "StringValue": ""
    },
    "RecordTitle": {
    "Value": "15-532471",
    "StringValue": "15-532471"
    },
    "IsInFavorites": false,
    "EnabledCommandIds": [
    "Properties",
    "RecCheckOut",
    "RecCheckIn",
    "RecNewPart",
```

```
    "RecNewVersion",

    "RecAddRetrieveTemporaryRequest",

    "RecContainer",

    "RecOwnerLoc",

    "RecAddContact",

    "ShowContacts",

    "RecCurrentLoc",

    "RecHomeLoc",

    "RecRemoveContact",

    "RecShowRequests",

    "RecAddRetrieveTemporaryRequest",

    "RecAddRetrieveRecurrentRequest",

    "RecAddRetrievePermanentRequest",

    "RecAddPickupTemporaryRequest",

    "RecAddPickupPermanentRequest"

    ],

    "Icon": {

    "IsFileTypeIcon": true,

    "IsInternalIcon": false,

    "IsValid": true,

    "FileType": "PDF",

    "Id": "Unknown"

    },

    "Uri": 9000000378
```

# Appendix C
# Active Directory Federation Services (ADFS)

## Scenario

In this scenario we have the following domains:

- TRIM

- TESTTEAM

The Content Manager Workgroup Server and Content Manager WebClient server (IIS) are part of TRIM domain

The Content Manager users should be able to log on to the Content Manager WebClient server with their TESTTEAM domain credential.

## Content Manager Web Client Configuration

**Prerequisite**

Content Manager Web Client must be configured to run on Secure Sockets Layer (SSL).

## Editing hprmServiceAPI.config

Set useADFS attribute under <setup> key to *true*. This will add a log out button to the Content Manager Web Client.

**Configuration example**

```
<setup databaseId="G1"  useADFS="true"  workpath="C:\Micro Focus Content
Manager\ServiceAPIWorkpath"/>
```

## Editing Web.config

Under **<configSections>** add the following lines:

```
  <section name="system.identityModel"
type="System.IdentityModel.Configuration.SystemIdentityModelSection,
System.IdentityModel, Version=4.0.0.0, Culture=neutral,
PublicKeyToken=B77A5C561934E089" />
  <section name="system.identityModel.services"
type="System.IdentityModel.Services.Configuration.SystemIdentityModelServicesSectio
n, System.IdentityModel.Services, Version=4.0.0.0, Culture=neutral,
PublicKeyToken=B77A5C561934E089" />
```

Under **<appSettings>** add the following lines:

```
<add key="ida:FederationMetadataLocation"
value="https://adfs1.testteam.local/FederationMetadata/2007-
06/FederationMetadata.xml" />
```

> **NOTE:** replace the highlighted value with your correct value.

Under **<system.web>** (this is under **<configuration>**) add the following lines:

```
<authorization>
    <deny users="?" />
</authorization>
 <authentication mode="None" />
```

> **NOTE:** replace the existing <authentication> element.

Under **<configuration>** add the following lines:

```
<system.identityModel>
    <identityConfiguration>
      <audienceUris>
        <add value="https://MyServer/MyWebClient/" />
      </audienceUris>
      <securityTokenHandlers>
        <add
type="System.IdentityModel.Services.Tokens.MachineKeySessionSecurityTokenHandler,
System.IdentityModel.Services, Version=4.0.0.0, Culture=neutral,
PublicKeyToken=b77a5c561934e089" />
        <remove type="System.IdentityModel.Tokens.SessionSecurityTokenHandler,
System.IdentityModel, Version=4.0.0.0, Culture=neutral,
PublicKeyToken=b77a5c561934e089" />
      </securityTokenHandlers>
      <certificateValidation certificateValidationMode="None" />
      <issuerNameRegistry
type="System.IdentityModel.Tokens.ValidatingIssuerNameRegistry,
System.IdentityModel.Tokens.ValidatingIssuerNameRegistry">
        <authority name="http://ADFS1.testteam.local/adfs/services/trust">
          <keys>
            <add thumbprint="538B9CF4D293995C9406D7EA74C3F7353C5DD62A" />
          </keys>
          <validIssuers>
            <add name="http://ADFS1.testteam.local/adfs/services/trust" />
          </validIssuers>
        </authority>

      </issuerNameRegistry>
    </identityConfiguration>
  </system.identityModel>
```

> **NOTE:** replace the highlighted value with your correct value.

> **NOTE:** You can get the value for the *thumbprint* from ADFS Token Signing Certificate. To get the *thumbprint*, run the following Power Shell command from ADFS server:
> ```
> Get-ADFSCertificates
> ```

Under **<configuration>** add the following lines:

```
<system.identityModel.services>
    <federationConfiguration>
      <cookieHandler requireSsl="true" />
      <wsFederation passiveRedirectEnabled="true"
issuer="https://adfs1.testteam.local/adfs/ls" realm="https://MyServer/MyWebClient/"
requireHttps="true" />
    </federationConfiguration>
  </system.identityModel.services>
```

> **NOTE:** replace the highlighted value with your correct value.

Under **<configuration>** add the following lines:

```
<location path="adfs">
<system.web>
<authorization>
<allow users="?" />
</authorization>
</system.web>
</location>
```

Under **<system.webServer>** (under **<configuration>**) add the following modules:

```
<modules>
      <add name="WSFederationAuthenticationModule"
type="System.IdentityModel.Services.WSFederationAuthenticationModule,
System.IdentityModel.Services, Version=4.0.0.0, Culture=neutral,
PublicKeyToken=b77a5c561934e089" preCondition="managedHandler" />
      <add name="SessionAuthenticationModule"
type="System.IdentityModel.Services.SessionAuthenticationModule,
System.IdentityModel.Services, Version=4.0.0.0, Culture=neutral,
PublicKeyToken=b77a5c561934e089" preCondition="managedHandler" />
    </modules>
```

> **IMPORTANT:**
> To restrict anonymous file upload to the Web Server, in the **<!-- Service API Configuration -->** section of the **web.config** file, within the **<location path="serviceapi">** and **<location path="Api">** attributes, modify the **<allow users="?" />** element to be **<deny users="?" />**.

## Modifying the Authentication Setting in IIS

Using Internet Information Services (IIS) Manager, for the Content Manager Web Client site, enable **Anonymous Authentication** for:

- Content folder

- Scripts folder
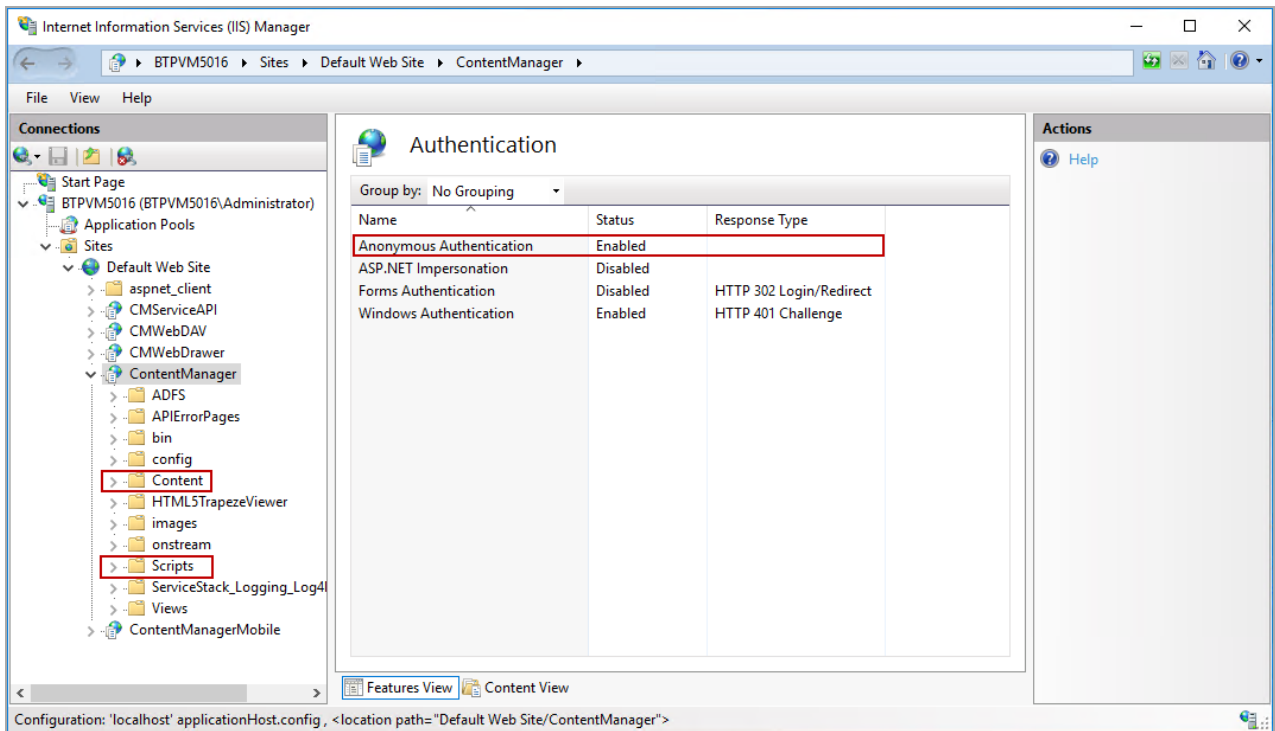


# Configuring the Office/Outlook Addins

## Background

Authentication from a client side application, such as the thin Office and Outlook add-ins, requires that the application be registered in ADFS and that the registration information be supplied to Azure AD by each client. This section details how to configure this.

## Creating the Client

Create an ADFS client using the powershell command 'Add-ADFSClient'. For example:

```
Add-ADFSClient -Name "MyCMWebClient" -ClientId "3A69D33C-7ED4-4D16-82F2-
48245F0B2C05" -RedirectUri https://rmcloud/WebClient/
```

The Client ID can be generated using the guidgen utility, RedirectUri must be a valid URI but does not need to point to an actual end point.

## Linking from the Content Manager Web Client

Native client integration must be enabled in the hprmServiceAPI.config file. This is done via the **<authentication>** element.

In the example below the audience is the identifier from your Relying Party Trust and the metadataEndpoint is the same as that used in the ida:FederationMetadataLocation appSetting in the web.config file.

```
<authentication allowAnonymous="false">
  <activeDirectory>
    <add name="adfs" audience="https://MyServer/MyWebClient/"
metadataEndpoint="https://adfs1.testteam.local/FederationMetadata/2007-
06/FederationMetadata.xml"/>
  </activeDirectory>
</authentication>
```

## Configuring the Client Details

Once the Web Client is configured to talk to ADFS we need to configure the clients (e.g. the thin Office and Outlook add-ins) to also connect to ADFS. The method we use to do this is to place the required information in an XML file beneath the Web Client folder on the web server, the thin Office and Outlook add-ins will look for this information and if it is found attempt to use ADFS to authenticate. To configure this:

1. Open the Content Manager Web Client install folder and find the folder ADFS (e.g. C:\Program Files\Micro Focus\Content Manager\Web Client\ADFS)

2. Open the file config.xml file in Edit mode.

3. Set the values of the contained elements as follows:

| Element | Source |
| --- | --- |
| clientAuthority | The address of your adfs instance. Example: https://adfs1.testteam.local/adfs |
| clientResourceUri | The identifier of your relying party trust. Example https://MyServer/MyWebClient/ |
| clientID | The guid used in the client created by Add-ADFSClient. |
| clientReturnUri | The RedirectUri specified when running Add-ADFSClient. |

## Frequently Asked Questions

Delete this text and replace it with your own content.

## How do I enable more logging?

In order to make it easier to troubleshoot problems, it will be useful to enable additional logging. In the Web.config, before the end of Configuration, insert the following config. Change the path in the initializeData attribute under *sharedListeners* to where you want the log to be written to.

```
<system.diagnostics>
    <trace autoflush="true" />
    <sources>
      <source name="System.Net">
        <listeners>
          <add name="System.Net" />
        </listeners>
      </source>
      <source name="System.Net.HttpListener">
        <listeners>
          <add name="System.Net" />
        </listeners>
      </source>
      <source name="System.Net.Sockets">
        <listeners>
          <add name="System.Net" />
        </listeners>
      </source>
      <source name="System.Net.Cache">
        <listeners>
          <add name="System.Net" />
        </listeners>
      </source>
    </sources>
    <sharedListeners>
      <add name="System.Net" type="System.Diagnostics.TextWriterTraceListener"
initializeData="C:\mylogs\System.net.trace.log" traceOutputOptions="ProcessId,
DateTime" />
    </sharedListeners>
    <switches>
      <add name="System.Net" value="Verbose" />
      <add name="System.Net.Sockets" value="Verbose" />
      <add name="System.Net.Cache" value="Verbose" />
      <add name="System.Net.HttpListener" value="Verbose" />
    </switches>
  </system.diagnostics>
```

You should then see the logs created. If you don't see the log files generated, manually create the mylogs folder.

# The remote certificate is invalid according to the validation procedure



This is most likely caused by a certificate error. By enabling logging, the error may be something along these lines below.

*System.Net Information: 0 : [9504] SecureChannel#3908756 - Remote certificate has errors:*

*System.Net Information: 0 : [9504] SecureChannel#3908756 - A certificate chain could not be built to a trusted root authority.*

*System.Net Information: 0 : [9504] SecureChannel#3908756 - Remote certificate was verified as invalid by the user.*

*System.Net Error: 0 : [9504] Exception in HttpWebRequest#2237113:: - The underlying connection was closed: Could not establish trust relationship for the SSL/TLS secure channel.*

**Solution**

To resolve this problem, the root certificate from the ADFS server must be trusted by the client. Log on to the ADFS Server, export the required certificate from a trusted root authority and import it on the client's machine into the trusted root authority.

To export certificate, on the AD FS Server, please refer to the following screenshot:
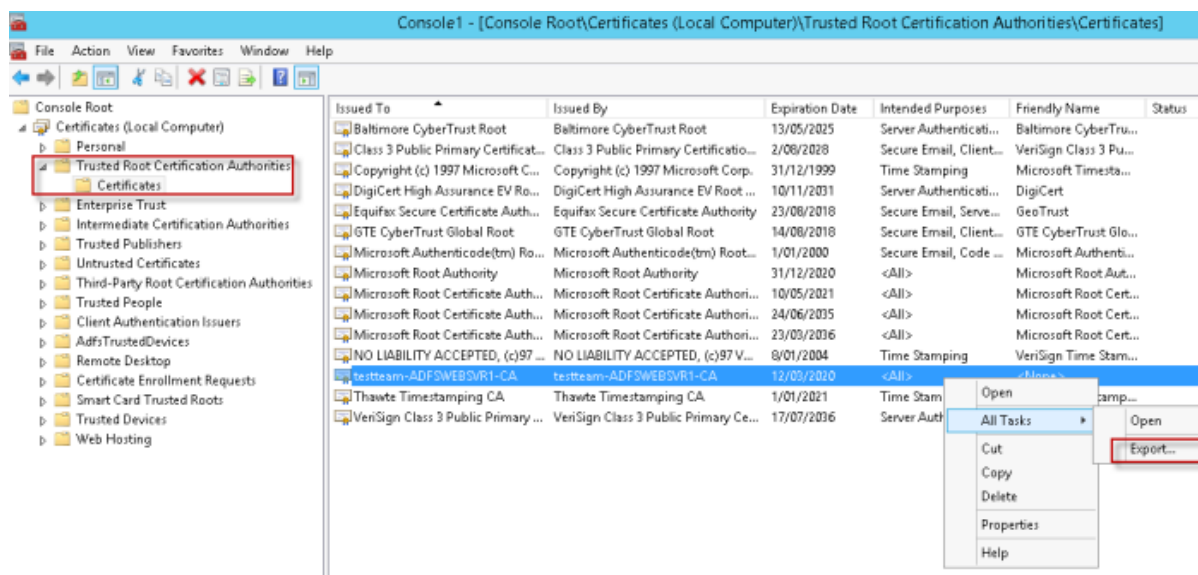


After the export is successful, you can import this certificate to the client's machine (into the Trusted Root Certificate Authorities).

# Appendix C1
# Setup Relying Party Trust in ADFS

1.  On the AD FS Management console, under **Trust Relationships**, select **Relying Party Trusts**.

2.  On the **Actions** panel, click **Add Relying Party Trust**.

The Add Relying Party Trust Wizard dialog is displayed.
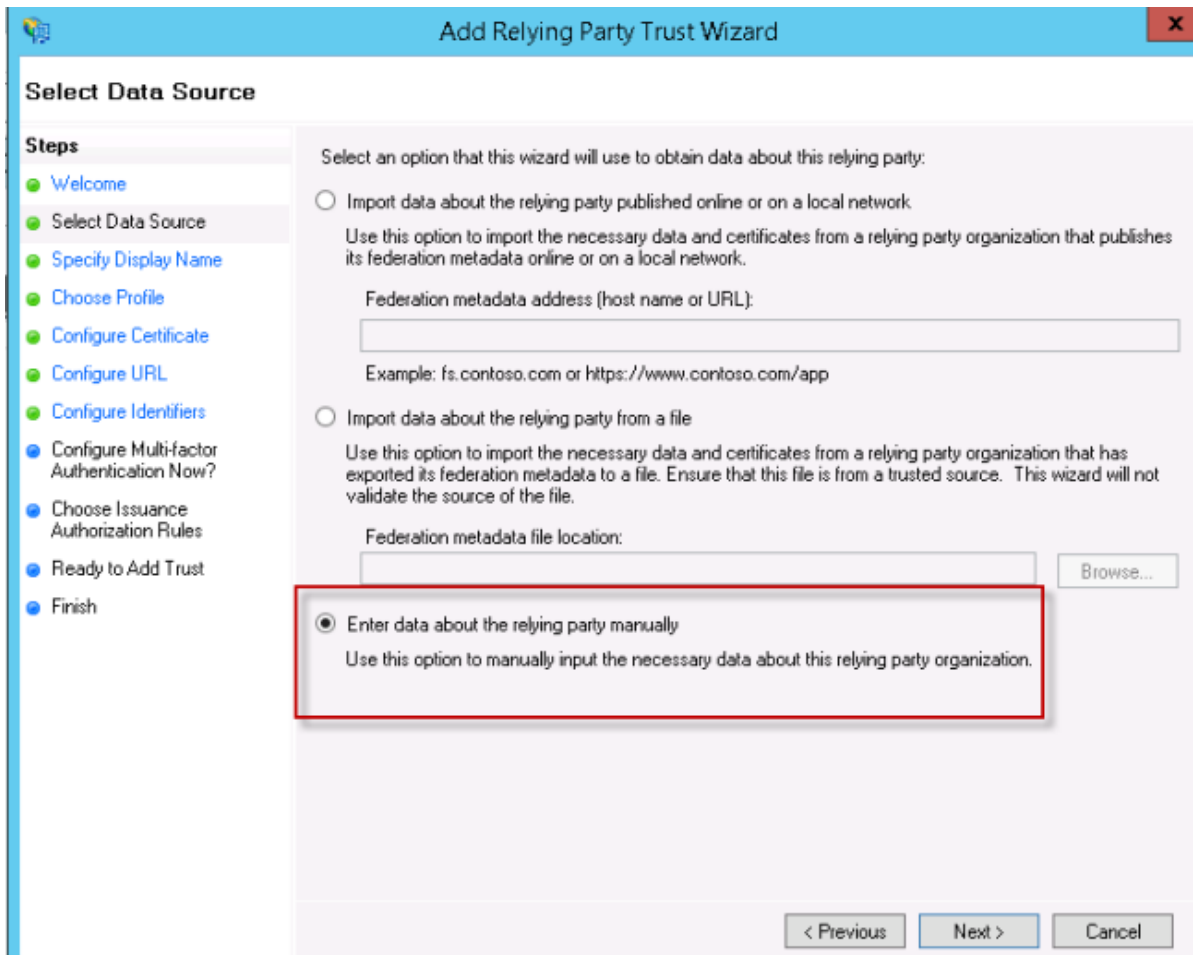
3. On the **Select Data Source** step, select **Enter data about the relying party manually**, click **Next**.

4. On the **Specify Display Name** step, type (or copy and paste) the name into the **Display Name** field, click **Next**.

5. On the **Choose Profile** step, select **AD FS profile**, click **Next**.

6. On the **Configure URL** step, select **Enable support for the WS-Federation Passive protocol** and type in the URL for the **Relying party WS-Federation Passive protocol URL**; click **Next**.

7. On the **Configure Identifiers** step, leave the screen as the default, click **Next**.

8. On the **Configure Multi-factor Authentication Now?** step, select **I do not want to configure multi-factor authentication settings for this relying party trust at this time**; click **Next**.

9. On the **Choose Issuance Authorization Rules** step, select **Permit all users to access this relying party**; click **Next**.

10. On the final step, select **Open the Edit Claim Rules dialog for this relying party trust when the wizard closes**; click **Close**.

**NOTE:** Ensure that Hash Algorithm is SHA-256 (see Properties - Advanced in the Relying Party Trust').

## Adding Claim Rules

1. To add Claim Rules, on the displayed **Edit Claim Rules** dialog, click **Add**.
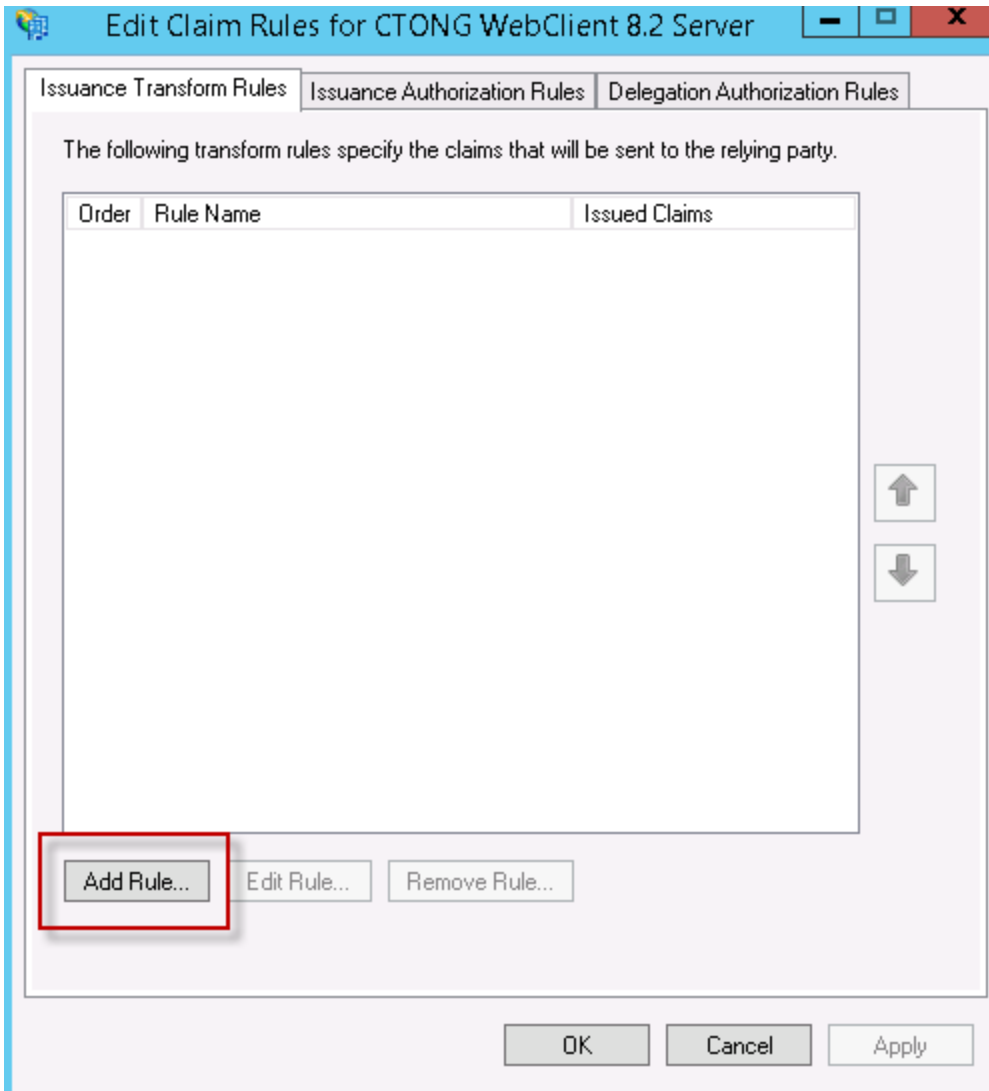


The **Add Transform Claim Rule Wizard** dialog is displayed.

2. On the Choose Rule Type, from the drop-down **Claim rule template** menu, select **Send Claims Using a Custom Rule**.
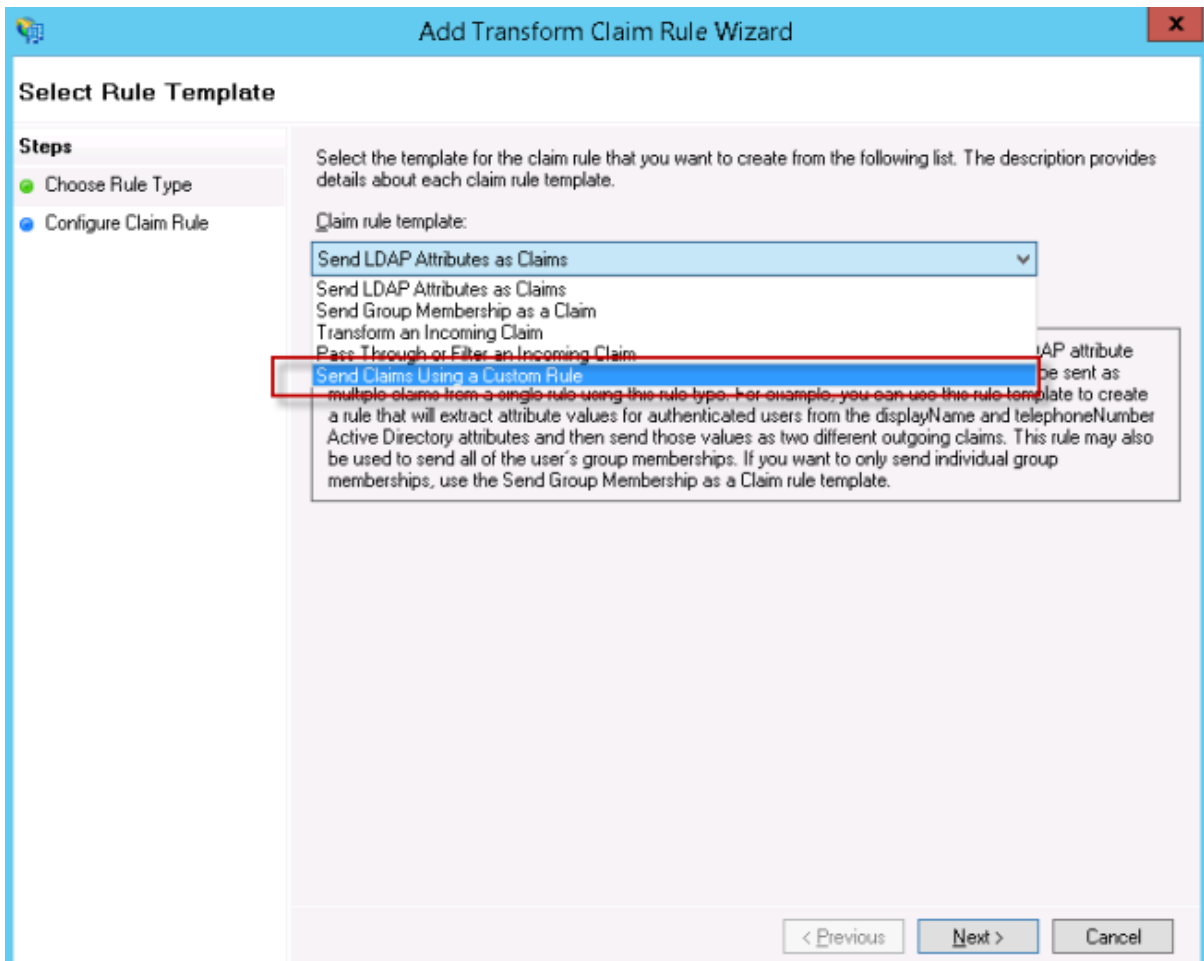
3. Complete the **Edit Rule - Simple Claim** dialog as below, then click **OK**.
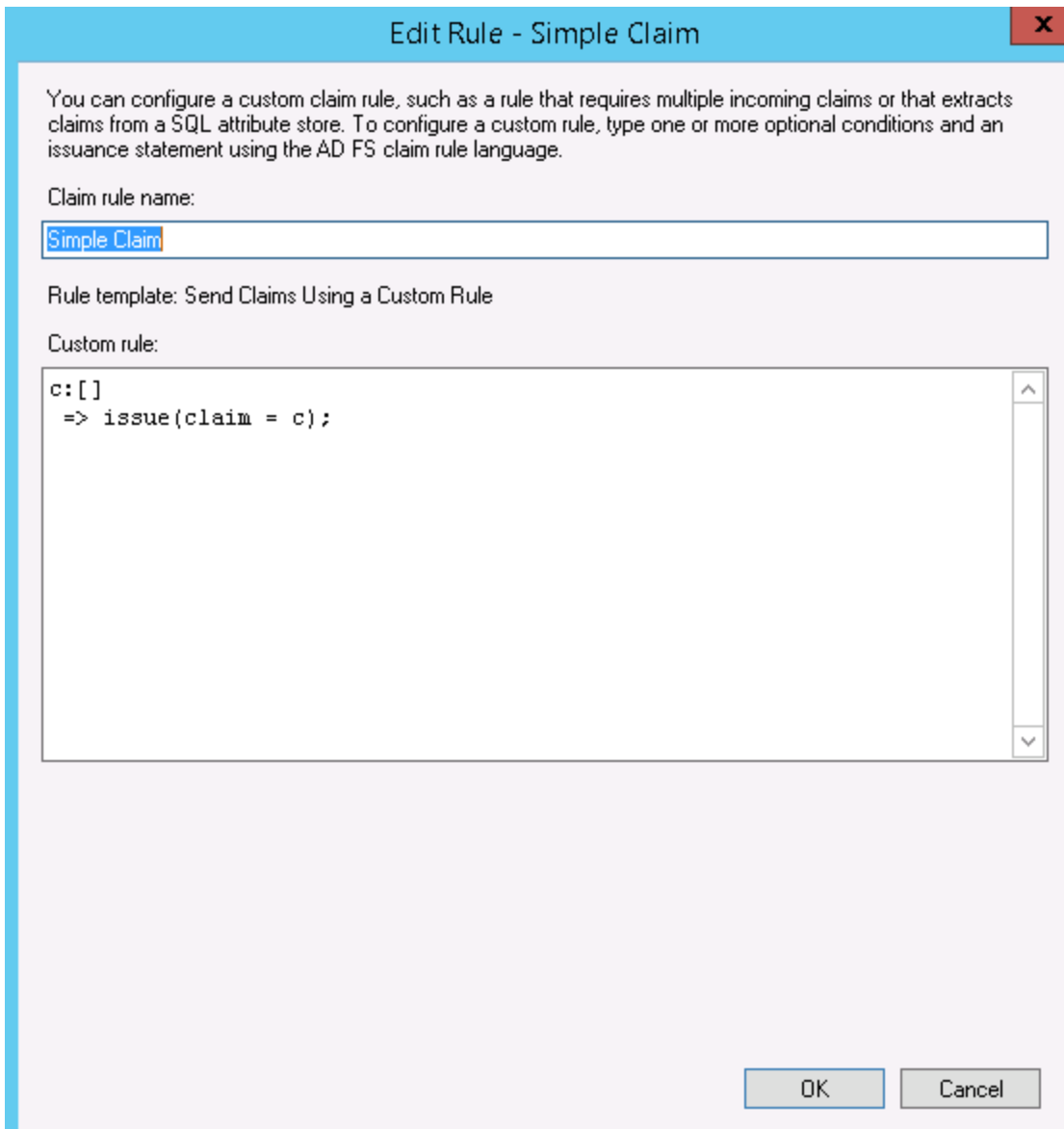
The new Relying Party Trust will be listed in the Relying Party Trusts list.
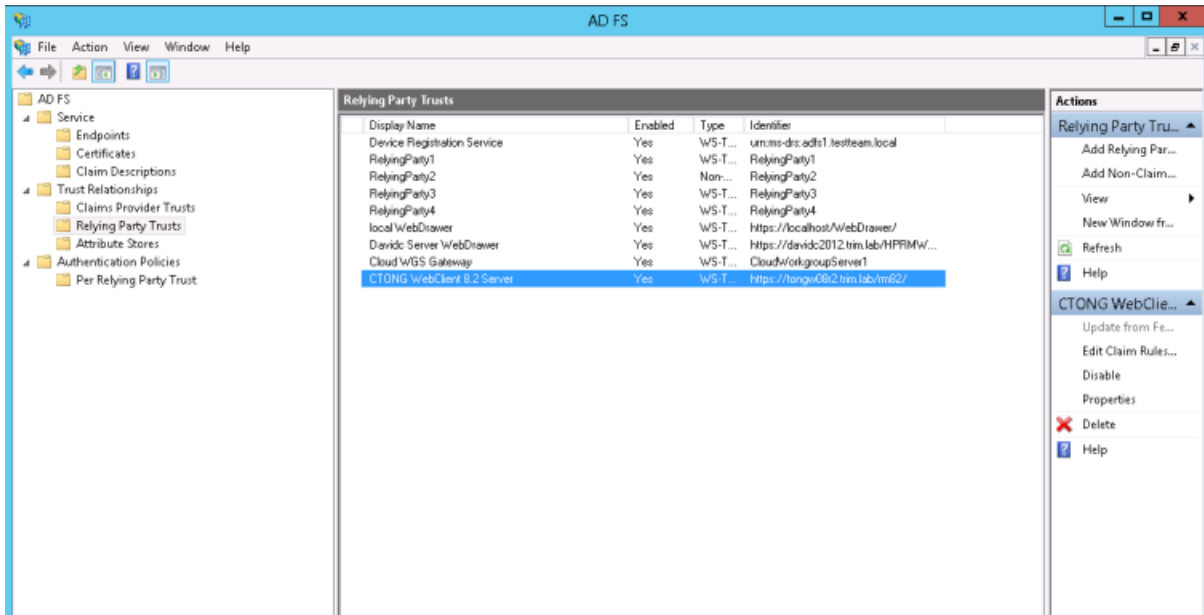
# Configure Relying Party Trust with custom authentication settings

To redirect the user to ADFS login page, make sure that the **Users are required to provide credentials each time at sign in** option is checked.

To enable the option, follow these steps:

1. On the ADFS server, open the server manager and navigate to **Tools > AD FS Management**.

   The **AD FS** window is displayed.

2. On the left pane, click **Authentication Policies > Per Relying Party Trust**.

3. From the right pane, select the appropriate Relying Party Trust from the list under **Replying Party Trusts with global authentication settings only**, right-click and open its **Properties**.

   The **Edit Authentication Policy** window is displayed.

4. In the **Primary** tab, check the **Users are required to provide credentials each time at sign in** option.
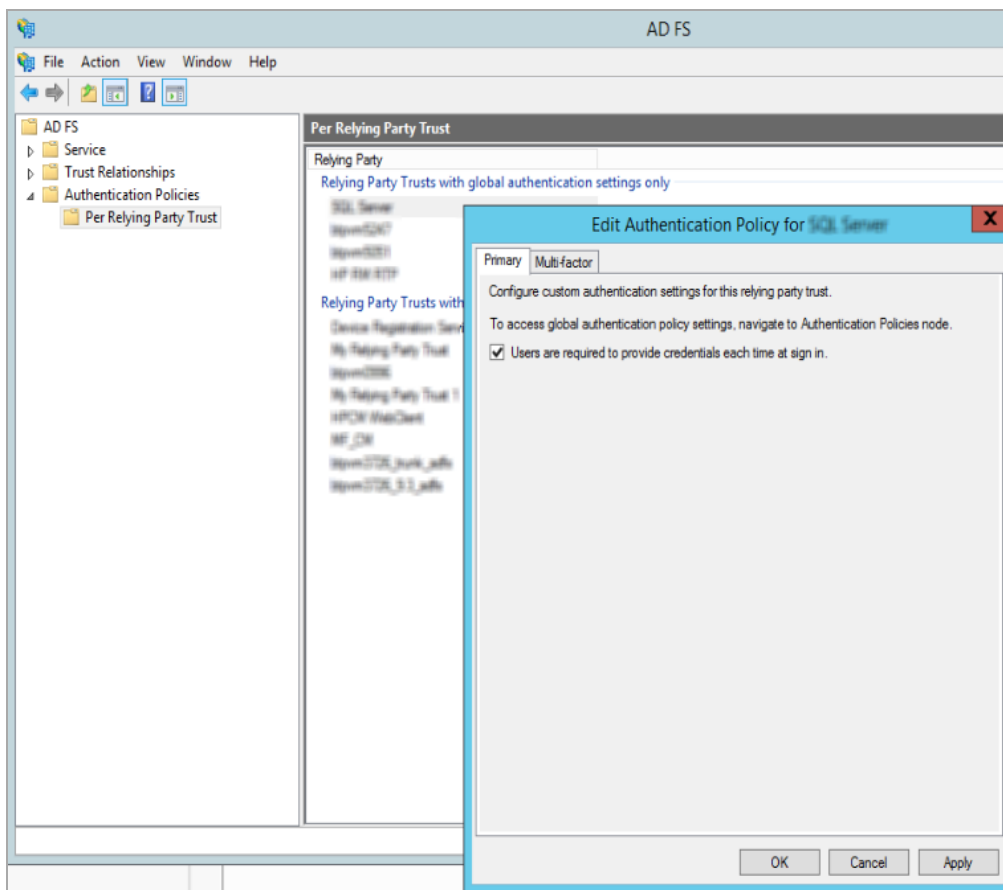
5. Click **Apply** and then click **OK**.

   The Relying Party Trust for which the option is enabled will now be listed under **Relying Party Trusts with custom authentication settings**.

   The user will be redirected to ADFS login page.

# Appendix D
# Azure AD Authentication for Content Manager Web Client

> **IMPORTANT:** Azure authentication needs outgoing access on ports 80 and 443 from the IIS server where the Content Manager Web Client is hosted.

> **NOTE:** The configuration described in this section also applies to Azure AD authentication for Content Manager Web Client Mobile.

## Configuring Web Browser Access

### Disabling IIS Administration

Given that IIS windows integrated authentication will no longer be used disable it in IIS Manager and enable Anonymous, as seen here:
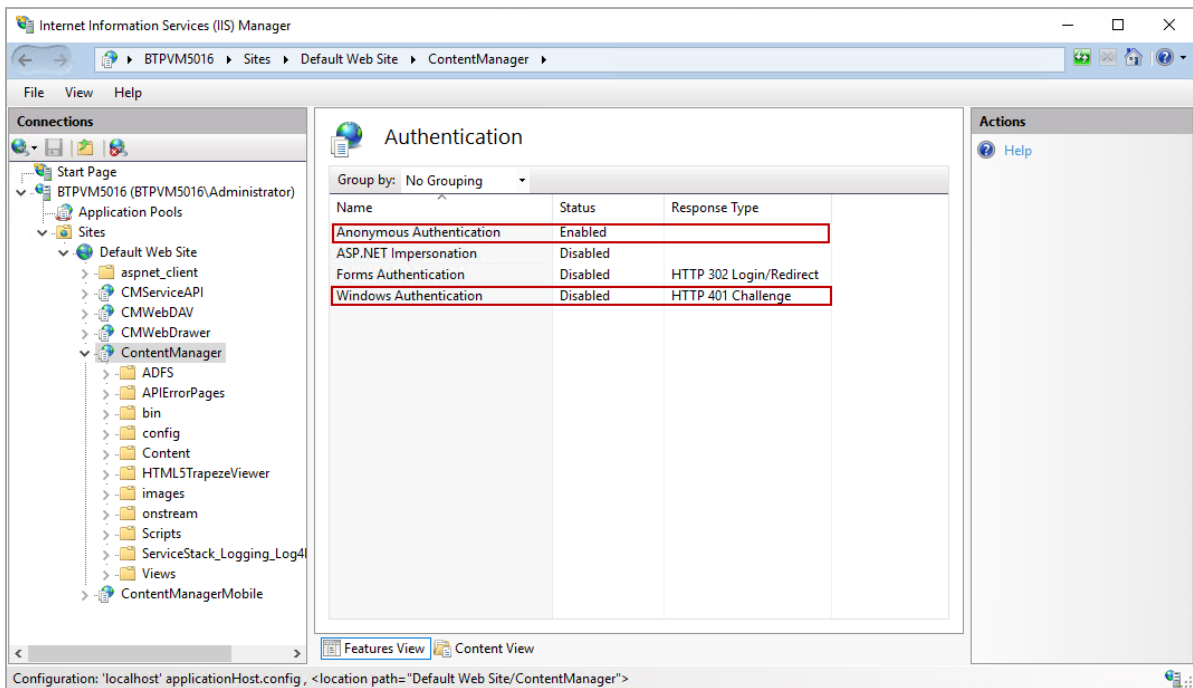


### Creating an Active Directory Web Application

To allow authentication of the Content Manager Web Client you must first configure an Active Directory application.

To do this:

1. Open your Active Directory instance in Azure.

2. Click **Azure Active Directory > App registrations > New registration**.

   The **Register an Application** page is displayed.

3. Type a name for your new application in the **Name** field.

4. Select the **Supported account types**.

   For more information on each option of **Supported account types**, click **Help me choose**.

5. Type (or copy and paste) the URL to your Web Client instance in the **Redirect URI** field and click **Register**.

6. In IIS, enable **Anonymous Authentication** and disable **Windows Authentication** for your Web Client instance.

## Updating the web.config

The Content Manager Web Client web.config must be updated to link it to Azure AD for authentication; there are a number of elements that need to be added.

Within the <**configSections**> element add the following two section elements:

```
<section name="system.identityModel"
type="System.IdentityModel.Configuration.SystemIdentityModelSection,
System.IdentityModel, Version=4.0.0.0, Culture=neutral,
PublicKeyToken=B77A5C561934E089" />
<section name="system.identityModel.services"
type="System.IdentityModel.Services.Configuration.SystemIdentityModelServicesSectio
n, System.IdentityModel.Services, Version=4.0.0.0, Culture=neutral,
PublicKeyToken=B77A5C561934E089" />
```

Within the <**appSettings**> element add this key, replace the value with the value found in the 'Federation metadata document url' field in your AD application.

```
<add key="ida:FederationMetadataLocation"
value="https://login.windows.net/5321c511-aa9d-4958-9857-
b59e08fadaa9/FederationMetadata/2007-06/FederationMetadata.xml" />
```

Within the <s**ystem.Web**> element replace the existing <**authentication**> element with the following two elements.

```
<authorization>
  <deny users="?" />
</authorization>
<authentication mode="None" />
```

After the <**handlers**> element (inside <**system.webServer**> ) add these modules:

```
<modules>
  <add name="WSFederationAuthenticationModule"
```

```
type="System.IdentityModel.Services.WSFederationAuthenticationModule,
System.IdentityModel.Services, Version=4.0.0.0, Culture=neutral,
PublicKeyToken=b77a5c561934e089" preCondition="managedHandler" />
  <add name="SessionAuthenticationModule"
type="System.IdentityModel.Services.SessionAuthenticationModule,
System.IdentityModel.Services, Version=4.0.0.0, Culture=neutral,
PublicKeyToken=b77a5c561934e089" preCondition="managedHandler" />
</modules>
```

At the bottom of the web.config file (before the **</configuration>** tag) add the following:

```
<system.identityModel>
  <identityConfiguration>
    <audienceUris>
      <add value="[APP ID URI]" />
    </audienceUris>
    <securityTokenHandlers>
      <add
type="System.IdentityModel.Services.Tokens.MachineKeySessionSecurityTokenHandler,
System.IdentityModel.Services, Version=4.0.0.0, Culture=neutral,
PublicKeyToken=b77a5c561934e089" />
      <remove type="System.IdentityModel.Tokens.SessionSecurityTokenHandler,
System.IdentityModel, Version=4.0.0.0, Culture=neutral,
PublicKeyToken=b77a5c561934e089" />
    </securityTokenHandlers>
    <certificateValidation certificateValidationMode="None" />
    <issuerNameRegistry
type="System.IdentityModel.Tokens.ValidatingIssuerNameRegistry,
System.IdentityModel.Tokens.ValidatingIssuerNameRegistry">
      <authority name="https://sts.windows.net/[Tenant ID]/">
        <validIssuers>
          <add name="https://sts.windows.net/[Tenant ID]/" />
        </validIssuers>
      </authority>
    </issuerNameRegistry>
  </identityConfiguration>
</system.identityModel>
<system.identityModel.services>
  <federationConfiguration>
    <cookieHandler requireSsl="true" />
    <wsFederation passiveRedirectEnabled="true" issuer="https://login.windows.net/
[Domain]/wsfed" realm="[APP ID URI]" requireHttps="true" />
  </federationConfiguration>
</system.identityModel.services>

<location path="adfs">
    <system.web>
      <authorization>
        <allow users="?" />
      </authorization>
```
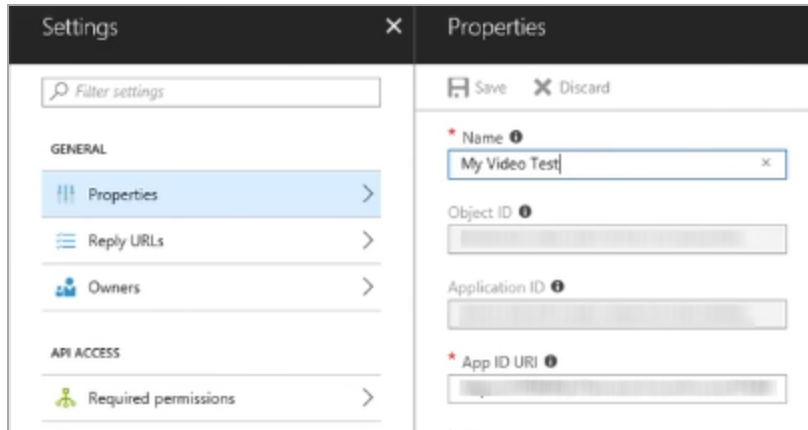
```
        </system.web>
    </location>
```

The XML above contains placeholders, replace them as follows:

- **[APP ID URI]** – from the **Setting > Properties** of your newly created application in Azure Active Directory.
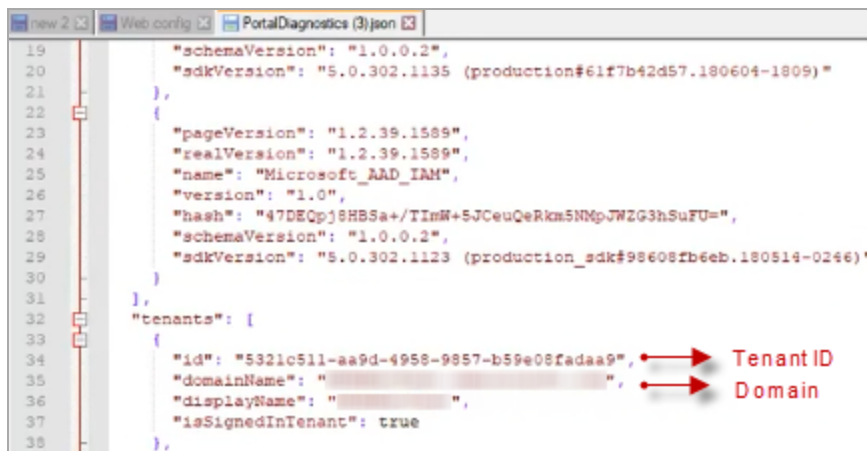


- **[Tenant ID]** – The GUID portion of the federation metadata document URL in the Azure application or
- **[Domain]** – From the domains section of the Azure AD directory

You can also get the **Tenant ID** and **Domain** from the portal diagnostics file.

1. Click **help Icon > Show Diagnostics > Save > Open Folder**.

The **PortalDiagnostics.JSON** is displayed.



## Starting the Web Client

At this point, you should be ready to run the Content Manager Web Client by navigating to the appropriate URL (e.g. https://mywebserver/ContentManager/). Two things to note:

- Azure AD authentication will only work when you including the trailing slash (/) in the URL

- The first time you access the Web Client URL after following these instructions will probably result in an error, this is part of the setup process. Refresh the page and this error should go away.

# Configuring Office/Outlook Addins

Authentication from a client side application, such as the Office and Outlook add-ins, requires that the application be registered in Azure AD and that the registration information be supplied to Azure AD by each client. This section details how to configure this.

## Background

## Creating the Azure Native Client Application

To create the native client application:

1. Go to your Azure AD directory.

2. From **Applications**, click **Add**.

3. Select **Add an application my organization is developing**

4. Type (or copy and paste) the application name, e.g. My CM Web Client for Office.

5. Select **native client application** as the **Type**.

6. Type (or copy and paste) a valid URI in **Redirect URI**, e.g. http://MyCMWebClient

### Give Permission to the Web Application

For the native application (created above) to work you must give it permission to access the AD web application created in the first part of this document. To do this:

1. Go to the **Configure** tab of the native application

2. Find the **Permission to other applications** section

3. Add the AD web application.

For example:
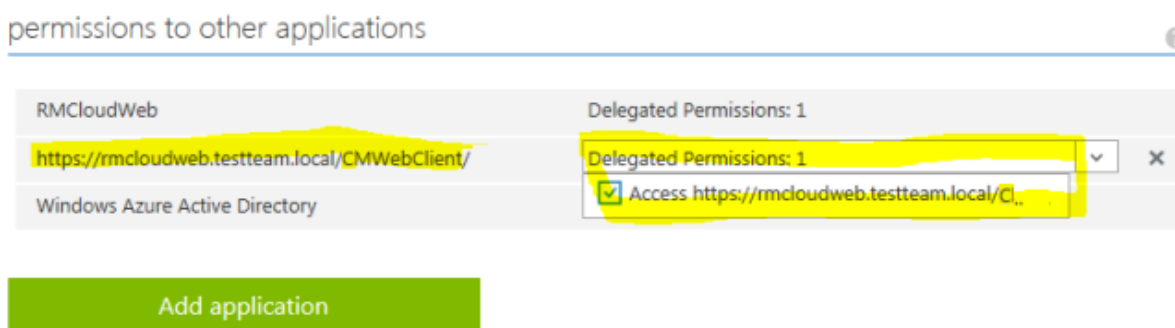


## Updating hprmServiceAPI.config

Set useADFS attribute under <setup> key to *true*. This will add a log out button to the Content Manager Web Client.

**Configuration example**

```
<setup databaseId="G1"  useADFS="true"  workpath="C:\Micro Focus Content
Manager\ServiceAPIWorkpath"/>
```

Copy the following XML into the hprmServiceAPI.config and copy and paste the App ID URI from the AD web application created in the first part of this document into the audience, copy the Federation metadata document URL into the metadataEndpoint.

```
  <authentication allowAnonymous="false">
    <activeDirectory>
      <add name="adfs" audience="[ APP ID URI for AzureAD Web App created for Azure
AD Web CLient ]" metadataEndpoint="[Azure AD app registration Metadata Endpoint]"/>
    </activeDirectory>
  </authentication>
```

### Configure Client Details

Once the Content Manager Web Client is configured to talk to Azure AD we need to configure the clients (e.g. the thin Office and Outlook add-ins) to also connect to Azure AD. The method we use to do this is to place the required information in an XML file beneath the Web Client folder on the web server,

the Office and Outlook add-ins will look for this information and if it is found attempt to use Azure AD to authenticate. To configure this:

1. Open the Content Manager Web Client install folder and find the folder ADFS (e.g. C:\Program Files\Micro Focus\Content Manager\Web Client\ADFS)

2. Open the file config.xml file in Edit mode.

3. Set the values of the contained elements as follows:

| Element | Source |
|---|---|
| clientAuthority | The following URL replacing [Domain] with your Azure Ad domain: https://login.windows.net/[Domain] |
| clientResourceUri | The App Id URI from the Azure AD web application. |
| clientID | Client Id from the Azure AD native client application. |
| clientReturnUri | A Redirect Uri from the Azure AD native client application (in the configuration section. |

**Example**

```
<adfsClient>

<clientAuthority>https://login.windows.net/cmdevtest.onmicrosoft.com</clientAuthority>
        <clientResourceUri>https://login.microsoftonline.com/
[yourDomain]</clientResourceUri>
        <clientID>6ee05ae2-fc8e-4120-a0d1-b43f31263d8d</clientID>
        <clientReturnUri>http://RMCloudWeb</clientReturnUri>
</adfsClient>
```

# Appendix E
# Office Online Integration

## Introduction

The Office Online integration is built into the Content Manager Web Client using Web Application Open Platform Interface (WOPI).

From the Web Client interface, it allows online editing of supported Microsoft Office documents to users with an Office Online subscription that includes editing ability.

## Overview

### Supported Versions of Office Online

The supported version of Office Online Server is 16.0.7766.8550 and later.

If you are installing your own Office Online instance on a local network, then configure to it use either HTTP or HTTPS, but not both. It is recommended you configure it to use HTTPS. Microsoft's own Office 365 online servers only support HTTPS communication

### Supported File Types

Office Online Server delivers browser-based versions of Word, PowerPoint, Excel, and OneNote. Editing is supported for the newer XML style documents (i.e. .docx, .pptx, .xlsx). Older formats such as .doc can be converted to the newer native formats so that they may be edited. This conversion process is usually lossless, but is not guaranteed to be so, and may produce a newer version that is missing certain content or formatting.

The Office Online server broadcasts its capabilities about which actions may be performed on which formats as an XML feed via a HTTP request. The process of determining a server's capabilities via the XML feed is known as 'Discovery'. These capabilities are subject to change - and are highly likely to change - with successive revisions. The Content Manager integration will regularly try to rediscover the capabilities of the Office Online instance, determine which can be used and make them available to Content Manager users when the Office Online instance is upgraded.

### Security

Users are issued tokens by the integration that grant them access to view/edit a document from Content Manager. When the Office server communicates with Content Manager on behalf of the user, it presents this token which is decrypted and verified. This token provides all the information we need to manipulate records in Content Manager. The encryption key is configured in one of the web server's

configuration files. It is recommended that this is changed frequently to minimize the chance of security being compromised.

# Configuring Office Online integration

## Content Manager Web Server Configuration

> **NOTE:** If configuring multiple instances of the Content Manager Web Server in order to use with a load balancer, then each instance within the load balanced cluster must be configured with the same values following the steps below.

- Once you have installed the Content Manager Web Client, from the Web Client installation directory, open the **HPRMServiceAPI.config** file using a text editor and find the section that includes:

```
<pluginAssemblies>

<!-- <add name="HP.HPTRIM.WebClient.WOPI"/> -->

<!-- <add name="HP.HPTRIM.WebClient.WebDAV"/> -->

</pluginAssemblies>
```

To enable the WOPI service handler and required buttons within the Web Client, uncomment the **<add name="HP.HPTRIM.WebClient.WOPI"/>** line in **HPRMServiceAPI.config**, and save the changes.

- In **HPRMServiceAPI.config**, find the section with the below attributes:

```
<officeOnlineServer host="" useSSL="true" allowSelfSignedSSL="false"
tokenEncKey="" tokenEncIV=""/>
```

Change the values of the attributes above to reflect your environment, so that 'host' is the **host name** of the Office Online Server, e.g. "myofficeonline.mynetwork.com".

Change the value of 'useSSL' to "false" if you have configured your Office Online Server instance to Allow HTTP and do not wish to use SSL.

Change the value of 'allowSelfSignedSSL' to "true" if you have configured your Office Online Server to use SSL with a Self Signed Certificate - Self Signed Certificates will otherwise be rejected.

Change the value of 'tokenEncKey' to be a 32 byte/64 character hexadecimal string which will be used as an encryption key to encrypt security tokens. As an example, you could use 'b12df00db12df00db12df00db12df00db12df00db12df00db12df00db12df00d'.

Change the value of 'tokenEncIV' to be a 16 byte/32 character hexadecimal string which will be used as an initialization vector for security token encryption. As an example, you could use 'b12df00db12df00db12df00db12df00d'.

Save the changes.

- Open the **web.config** file in the Content Manager Web Client directory using a text editor and find the below section:

```
<modules runAllManagedModulesForAllRequests="true">
```

```
<!-- <add name="WOPIModule" type="HP.HPTRIM.WebClient.WOPI.WOPIModule" /> -->

</modules>
```

To authenticate the requests from the Office Online server on behalf of the Content Manager user, uncomment **<add name="WOPIModule" type="HP.HPTRIM.WebClient.WOPI.WOPIModule" />** in **web.config**, and save the changes.

## User Client Configuration

- The client PC web browsers must be configured to allow pop up windows from the Content Manager Web Client. If pop up windows are not enabled users will be prompted to allow them each time they edit a document, or depending on the settings, they will not be prompted at all, and will not be able to edit their documents using the Office Online editing functionality.

  To configure Internet Explorer to allow pop windows from the Content Manager Web Client, add the Content Manager Web Client address to the list of trusted sites in the Pop-up Blocker Settings Exceptions list.

- If the Office Online Server is using SSL, the certificate must be trusted by client PCs.

# Appendix F
# Multi Tenancy Configuration

## Scope

This document outlines the steps required to configure Content Manager (CM) 9.2 or later Web Client to support multiple CM datasets.

This document is aimed at system administrators who wish to allow access to multiple CM datasets from a single server running the CM Web Client.

For the purpose of this document, a tenant consists of an application running in Internet Information Services (IIS) connecting to an existing CM dataset provided by a Workgroup server. Each CM Web Client application on IIS can access exactly one dataset.

> **NOTE:** A CM dataset may be accessed by multiple IIS applications if required.

This document is presented as a guide showing how to set up and configure a new instance of a CM Web Client IIS application connecting to a specific dataset. This process may be repeated to create any number of applications.

This document assumes that the CM Web Client has been installed successfully and is working.

Advanced configuration of IIS applications is out of the scope of this document; for more information please consult the IIS documentation https://www.iis.net/

## Configuration Steps

This section will outline the steps to create a new tenant.

Before beginning, ensure that the following information has been gathered:

- Name for the tenant (tenant name)

- Dataset Id of the CM dataset that the tenant will be accessing

- Connection information for the CM workgroup server that the tenant will be connecting to

- The name of the user account on the server that is trusted to connect to the selected CM workgroup server and dataset

- The IIS web site which will host the tenant

- The folder on the web server that will be used as the 'upload path' for the tenant. The upload path is the folder used to store files that are being uploaded to the server to be checked into CM.

### Create Tenants folder

Underneath the installation folder of the CM web client, which is installed to by default,

C:\Program Files\Micro Focus\Content Manager\Web Client

Create a folder named **Tenants**:

C:\Program Files\Micro Focus\Content Manager\Web Client\Tenants

## Create configuration file using sample

Copy the sample configuration file, **Tenant-sample.config** from the Web Client installation directory and paste it into the newly created **Tenants** folder.

Located in the installation folder into the tenants folder underneath the installation folder.

Rename the **Tenant-sample.config** file in the **Tenants** folder to be the name of the Tenant, e.g. ABCorp.config

## Update the configuration file

The following sections **must** be updated as per installation of a standard CM Web Client:

- hptrim
- Setup
- Workgroupserver

In the **hptrim** section of the **hprmServiceAPI.config** file, the upload base path must be set to the folder that will be used as the upload path for the tenant.

In the **setup** section of the **hprmServiceAPI.config** file, ensure that the databaseId is set correctly, and the workpath has been set to a unique folder for this tenant.

In the **workgroupserver** section of the **hprmServiceAPI.config** file, ensure that the connection details, including port, workpath and name have all been set correctly.

> **NOTE:** the workpath attribute in the WorkgroupServer section and the uploadBasePath in the hptrim section must be the same.

```
<hptrim
    poolSize="1000" trace="true" indexPagePath="/Home" notFoundErrorHandler="/APIErrorPages/NotFound" globalErrorHan
    uploadBasePath="C:\Micro Focus Content Manager\ServiceAPIWorkpath\Uploads" autoPoolClean="true" serviceFeatures=

<pluginAssemblies>
  <!-- <add name="HP.HPTRIM.WebClient.WOPI"/> -->
  <!-- <add name="HP.HPTRIM.WebClient.WebDAV"/> -->
</pluginAssemblies>

    <!--phoenix configuration-->
    <setup databaseId="45" searchAhead="false" workpath="C:\Micro Focus Content Manager\ServiceAPIWorkpath\Uploads"
```

## Create a web application

Using IIS Manager, under the selected web site for this **tenant**, add an application. The default web site is acceptable to use. Consult the IIS documentation for more information about creating and managing multiple web sites within IIS.
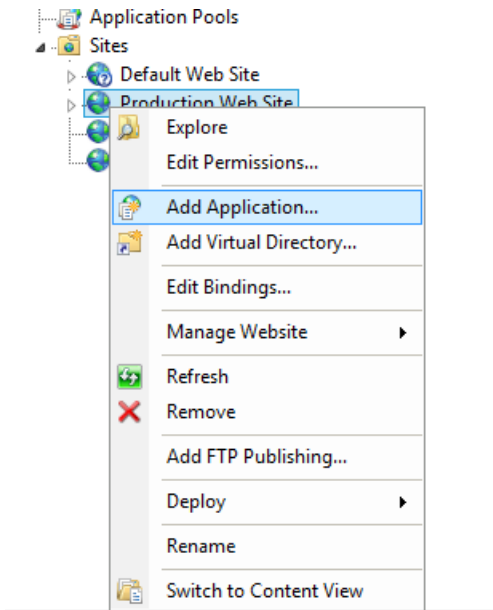
In the Add application dialog, set the alias, physical path and application pool for the tenant.

- Set the Alias of the application to be the Tenant name.

- The physical path of the application should be set to the location that binaries of the CM Web Client were installed, by default:

  C:\Program Files\Micro Focus\Content Manager\Web Client

- Select an Application pool that has an identity that is trusted to connect to the dataset that this instance will be accessing. For information on setting up Application pools in IIS please consult the IIS documentation.
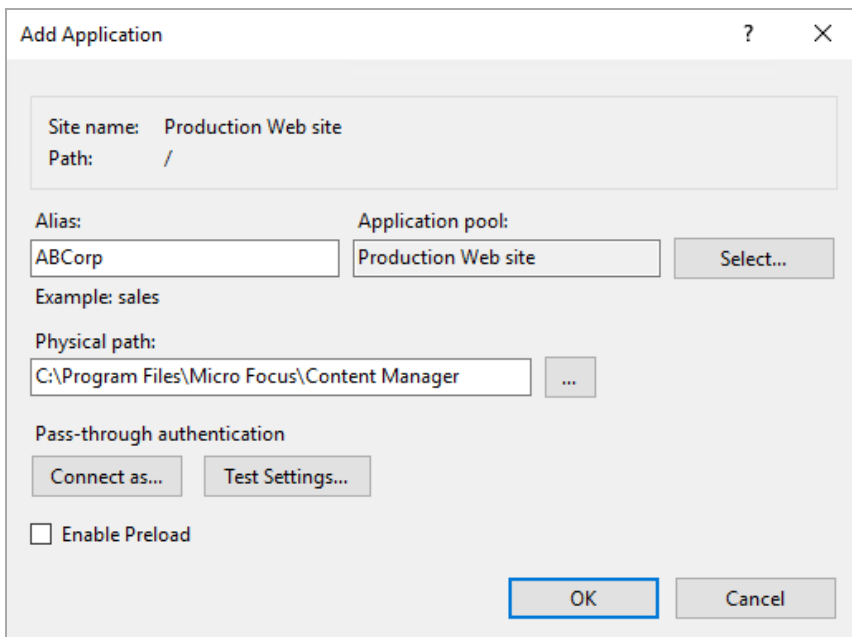
> **NOTE:** The application name **must be the same** as the configuration file name in the Tenants folder (excluding the .config) Example – tenant named 'ABCorp':
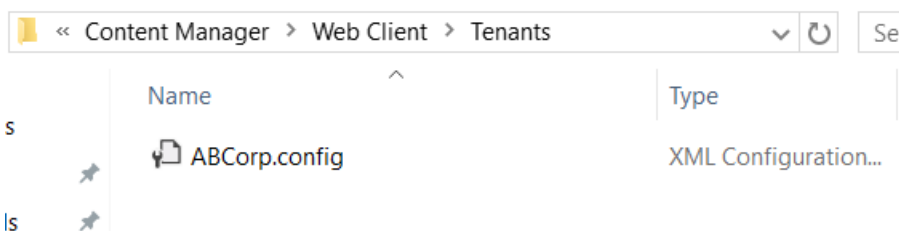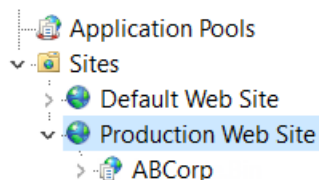




# Update logging

By default only one process can write to the log file. When setting up multiple tenants it is recommended that this is changed by changing the 'LockinModel' on Log4Net (minimalLock). Add the highlighted line to the **web.config** file in the directory containing the CM Web Client binaries.

```
<log4net>
    <appender name="RollingFileAppender"
type="log4net.Appender.RollingFileAppender">
        <lockingModel type="log4net.Appender.FileAppender+MinimalLock "/>
        <file value="C:\HPTRIM\ServiceAPIWorkpath\logs\log-file.txt" />
        <appendToFile value="true" />
        <rollingStyle value="Date" />
        <maximumFileSize value="1MB" />
        <staticLogFileName value="true" />
        <maxSizeRollBackups value="10" />
        <layout type="log4net.Layout.PatternLayout,log4net">
          <param name="ConversionPattern" value="%date [%thread] %-5level %logger -
%message%newline" />
        </layout>
    </appender>
    <root>
      <level value="ERROR" />
      <appender-ref ref="RollingFileAppender" />
    </root>
  </log4net>
```