

---

CM System

# Implementing WebDAV

Version 7.3

---

## **Copyright and Licensing Statement**

All intellectual property rights in the SOFTWARE and associated user documentation, implementation documentation, and reference documentation are owned by Percussion Software or its suppliers and are protected by United States and Canadian copyright laws, other applicable copyright laws, and international treaty provisions. Percussion Software retains all rights, title, and interest not expressly granted. You may either (a) make one (1) copy of the SOFTWARE solely for backup or archival purposes or (b) transfer the SOFTWARE to a single hard disk provided you keep the original solely for backup or archival purposes. You must reproduce and include the copyright notice on any copy made. You may not copy the user documentation accompanying the SOFTWARE.

The information in documentation is subject to change without notice and does not represent a commitment on the part of Percussion Software, Inc. This document describes proprietary trade secrets of Percussion Software, Inc. Licensees of this document must acknowledge the proprietary claims of Percussion Software, Inc., in advance of receiving this document or any software to which it refers, and must agree to hold the trade secrets in confidence for the sole use of Percussion Software, Inc.

The software contains proprietary information of Percussion Software; it is provided under a license agreement containing restrictions on use and disclosure and is also protected by copyright law. Reverse engineering of the software is prohibited.

Due to continued product development this information may change without notice. The information and intellectual property contained herein is confidential between Percussion Software and the client and remains the exclusive property of Percussion Software. If you find any problems in the documentation, please report them to us in writing. Percussion Software does not warrant that this document is error-free.

No part of this publication may be reproduced, stored in a retrieval system, or transmitted in any form or by any means, electronic, mechanical, photocopying, recording or otherwise without the prior written permission of Percussion Software.

Copyright © 1999-2013 Percussion Software.  
All rights reserved

## **Licenses and Source Code**

uses Mozilla's JavaScript C API. See <http://www.mozilla.org/source.html> (http://www.mozilla.org/source.html) for the source code. In addition, see the *Mozilla Public License* (http://www.mozilla.org/source.html).

Netscape Public License

Apache Software License

IBM Public License

Lesser GNU Public License

## **Other Copyrights**

The installation application was developed using InstallShield, which is a licensed and copyrighted by InstallShield Software Corporation.

The Sprinta JDBC driver is licensed and copyrighted by I-NET Software Corporation.

The Sentry Spellingchecker Engine Software Development Kit is licensed and copyrighted by Wintertree Software.

The Java™ 2 Runtime Environment is licensed and copyrighted by Sun Microsystems, Inc.

The Oracle JDBC driver is licensed and copyrighted by Oracle Corporation.

The Sybase JDBC driver is licensed and copyrighted by Sybase, Inc.

The AS/400 driver is licensed and copyrighted by International Business Machines Corporation.

The Ephox EditLive! for Java DHTML editor is licensed and copyrighted by Ephox, Inc.

This product includes software developed by CDS Networks, Inc.

The software contains proprietary information of Percussion Software; it is provided under a license agreement containing restrictions on use and disclosure and is also protected by copyright law. Reverse engineering of the software is prohibited.

Due to continued product development this information may change without notice. The information and intellectual property contained herein is confidential between Percussion Software and the client and remains the exclusive property of Percussion Software. If you find any problems in the documentation, please report them to us in writing. Percussion Software does not warrant that this document is error-free.

No part of this publication may be reproduced, stored in a retrieval system, or transmitted in any form or by any means, electronic, mechanical, photocopying, recording or otherwise without the prior written permission of Percussion Software.

AuthorIT™ is a trademark of Optical Systems Corporation Ltd.

Microsoft Word, Microsoft Office, Windows®, Window 95™, Window 98™, Windows NT® and MS-DOS™ are trademarks of the Microsoft Corporation.

This document was created using *AuthorIT™, Total Document Creation* (see AuthorIT Home - <http://www.author-it.com>).

Schema documentation was created using XMLSpy™.

### **Percussion Software**

600 Unicorn Park Drive

Woburn, MA 01801 U.S.A.

781.438.9900

Internet E-Mail: [technical\\_support@percussion.com](mailto:technical_support@percussion.com)

Website: <http://www.percussion.com>

# Contents

<b>Introduction.....</b>	<b>6</b>
Architecture of WebDAV Components in.....	7
CM System .....	7
Prerequisites and Recommendations.....	8
Supported WebDAV Clients and Application.....	9
Servers .....	9
<b>Implementing WebDAV .....</b>	<b>11</b>
Implementing a New WebDAV Configuration .....	12
Editing a Folder Configuration File.....	16
Simple Example Implementation of WebDAV .....	18
Webdav for Content Types with hash fields.....	27
Turning off validation for Items uploading through WebDav .....	28
Troubleshooting WebDAV .....	28
<b>Working with WebDAV .....</b>	<b>29</b>
Opening Web Folders in Windows Explorer .....	30
Opening Web Folders in Internet Explorer.....	31
Opening Web Folders in Mac OS X .....	33
Working with CM System Content in Web .....	34
Folders .....	34
WebDAV and CM System Workflow.....	35
<b>Appendices .....</b>	<b>37</b>
Default Folder Configuration File.....	37
WebDAV Methods .....	41

## CHAPTER 1

# Introduction

WebDAV is an extension to HTTP that lets users work with files on a remote Web Server. When content contributors use CM System with WebDAV they can create, modify, and save CM System Content Items while working in third-party applications, such as DreamWeaver and PhotoShop. Using CM System with WebDAV streamlines content creation in third-party applications because users do not have to log in to CM System in addition to opening the third-party application to create and modify content. To perform other CM System actions, such as making a Workflow Transition or viewing the Audit Trail, users must still log in to CM System.

To use CM System with WebDAV, you configure CM System Folders to associate CM System Content Types with file MIME types. Then you associate the CM System Folders with a virtual Web site, which is specified by a WebDAV URL, and point third-party applications at the same URL. The CM System Folder (and any sub-Folders) appear in the explorer dialog of the third-party applications. When a user saves a file created in a third-party application in one of these WebDAV-enabled Folders, WebDAV uploads it into CM System as the Content Type associated with its MIME type.

In addition to configuring CM System to use WebDAV, you must determine if your third-party applications require special configuration. Some applications, such as Internet Explorer, are already WebDAV-enabled, and you can use them as they are to access CM System content. Other applications, such as Adobe Photoshop and MacroMedia DreamWeaver, need to be installed with a WebDAV option or may require extra configuration to use WebDAV. Check your application's documentation for information about configuring WebDAV.

## Architecture of WebDAV Components in CM System

A WebDAV Servlet communicates with the CM System Server in the following manner:

The CM System WebDAV servlet communicates with the WebDAV client (WebDAV-enabled third-party application). It authenticates and passes requests from the client to the CM System servlet located in the CM System application server. The CM System servlet sends responses to the WebDAV servlet, which sends the responses to the WebDAV client.

You can create multiple WebDAV folder configurations in CM System's WebDAV servlet. Each folder may be available to different Communities.

The following graphic illustrates the architecture of the WebDAV components in CM System

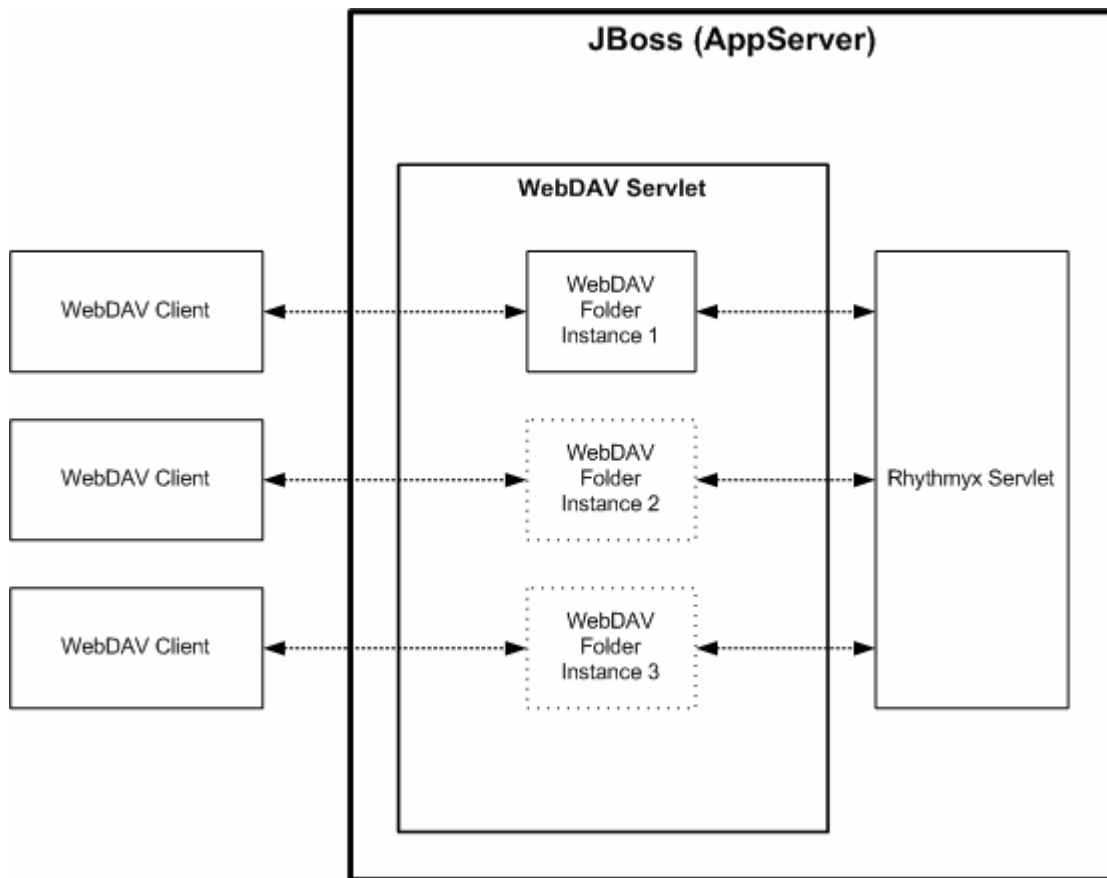


Figure 1: WebDAV Architecture in CM System

Each WebDAV folder instance supports a combination of a CM System Folder (and any sub-Folders of the specified Folder) with a set of Content Types. You will need several WebDAV folder instances if you want to access multiple root-level Folders.

---

## Prerequisites and Recommendations

- Each Content Type that you associate with a WebDAV-enabled folder must include fields for storing the following:
  - user with lock on Content Item;
  - size of uploaded file;
  - MIME type of uploaded file;
  - content of uploaded file.

These fields are mapped to WebDAV values in each folder configuration file.

See the topic *Editing a Folder Configuration File* (see page 16) for information about how these fields are formatted.

- When accessing a .pdf file in a WebDAV-enabled folder from outside of CM System, double-clicking on the file does not open it. If you are using Adobe Acrobat or Acrobat Reader, you can access the file by using the *File > Open* option in Acrobat or Reader. Acrobat and Acrobat Reader Version 7.0, however, cannot access files in a WebDAV Folder.



---

## Supported WebDAV Clients and Application Servers

CM System WebDAV supports the following WebDAV clients:

- Microsoft's Word 2002, Excel 2002, PowerPoint 2002, FrontPage 2002, and Project 2003. These products use Windows Explorer Web folders to access WebDAV in CM System.
- Adobe Acrobat Reader 6.0 uses Windows Explorer Web folders to access WebDAV in CM System.
- Macromedia Dreamweaver MX 2004 communicates with WebDAV in CM System directly; it does not require Windows Explorer Web folders.
- Adobe Photoshop 7.0.1 communicates with WebDAV in CM System directly; it does not require Windows Explorer Web folders.
- Windows Explorer on Windows XP, Windows 2003, and Windows 7  
Note: For Windows 7, WebDAV only communicates via HTTPS (not HTTP).
- MacOS WebDAV client that is shipped with the OS.



## CHAPTER 2

# Implementing WebDAV

This chapter describes how to implement WebDAV in CM System, and includes a simple example of WebDAV implementation. The final section provides troubleshooting hints for your implementation.

In general, we refer to the Folders configured for WebDAV in CM System as WebDAV-enabled Folders and the folders configured for WebDAV in third-party applications as WebDAV folders.

---

# Implementing a New WebDAV Configuration

CM System includes a default WebDAV Web application. You must customize the files in the default application to implement your system's WebDAV setup.

This topic lists the procedure for implementing WebDAV in CM System. For specific details about performing each step, see *Simple Example Implementation of WebDAV* (on page 17).

---

## NOTES:

The default implementation of CM System's WebDAV specifies Folders installed with FastForward. If you have installed FastForward, you can access the default Corporate Investment Web Folders with users in the Roles CI\_Members or CI\_Admin\_Members. You can access the default Enterprise Investment Web Folders with users in the Roles EI\_Members or EI\_Admin\_Members.

For information about installing and using FastForward, see the FastForward documentation.

---

To implement WebDAV in CM System:

- 1 Create the folders that you want to associate with WebDAV under the Sites and Folders nodes in the Content Explorer navigation pane. In the Create Folder dialog **Folder Community** field, choose the value of the Community that you want to have access to that Folder.
  - a) Choose *All Communities* or a specific Community (which must be the same as the Community entered in the folder configuration file).
  - b) Give the Roles that you want to have access to the Folder in CM System Write or Admin permission to the Folder.
- 2 For each of your WebDAV folders, associate the Roles that should have access to the WebDAV folder with the Community that you want to have access to the WebDAV-enabled Folder. In addition add the Roles to the Workflows assigned to the Content Types that you want to create in the folders. Assign each Role to all States in its Workflows as an assignee.
- 3 Create a WebDAV configuration file:
  - c) Back up the WebDAV config file, `<Rhythmyx root>\AppServer\server\rx\deploy\rxapp.ear\rxapp.war\WEB-INF\config\user\webdav\rxwebdav-servlet.xml` and open the current copy. There are four `<bean>` nodes that contain the default folder configuration files, *CIFiles*, *CIImages*, *EIFiles* and *EIImages*.
  - d) In one of the default `<bean id="name">` nodes, change "name" to a custom name for one of your folder configurations. Then in the sub-element `<prop key="RxWebDAVConfig">` change the name of the folder configuration file to a custom folder configuration filename.
  - e) At the bottom of the file, change the name in one of the `<prop>` elements in the `<bean id="urlMapping">` element to the custom name.
  - f) Repeat steps b and c for all of the WebDAV folders you have set up in Content Explorer.
  - g) Delete `<bean>` elements for folder config files that you are not using, and delete `<prop>` elements with URLs that you are not using from the `<bean id="urlMapping">` element.
- 4 Configure your custom folder configuration files. The default folder configuration files are:

```
<Rhythmyx root>\AppServer\server\rx\deploy\rxapp.ear\rxapp.war\WEB-  
INF\config\user\webdav\RxWebdavConfigEIIImages.xml  
<Rhythmyx root>\AppServer\server\rx\deploy\rxapp.ear\rxapp.war\WEB-  
INF\config\user\webdav\RxWebdavConfigEIFiles.xml  
<Rhythmyx root>\AppServer\server\rx\deploy\rxapp.ear\rxapp.war\WEB-  
INF\config\user\webdav\RxWebdavConfigCIImages.xml  
and  
<Rhythmyx root>\AppServer\server\rx\deploy\rxapp.ear\rxapp.war\WEB-  
INF\config\user\webdav\RxWebdavConfigCIFiles.xml
```

For each of your WebDAV folders:

Back up one of these files. Rename the current copy with the folder configuration file name you specified in your WebDAV configuration file. Edit the file to specify the CM System WebDAV Folder and its Community and Content Types.

For each additional WebDAV folder, make a copy of a folder configuration file, give it the name you specified in the sub-element `<prop key="RxWebDAVConfig">`

in the element `<bean id="customname">` and modify the CM System Folder, Community and Content Types.

For more information about editing the folder configuration file, see *Editing a Folder Configuration File* (see page 16).

- 5 After you make changes to your WebDAV configuration, restart your CM System server to apply them.
- 6 To confirm that implementation is correct, for each WebDAV folder configured, in your Web browser:

- Enter  
`http://hostname:9992/Rhythmyx/rxwebdav/[URLPattern]?sys_command=getConfig`. CM System will prompt you to log in. Log in with a user whose Role has access to the WebDAV folder.  
 If WebDAV is configured correctly, the browser displays the contents of the folder configuration file.

```
<?xml version="1.0" encoding="UTF-8" ?>
- <!--
  The WebDAV configuration information

  "root" - the virtual root path of the Rhythmyx for all resource
           specified in the request. It must start from "Site" or "Folder"
           For example, root="//Site/mysite", root="//Folder/myfolder".
           It must be an existing folder.

  "communityid" - the name of the community, which is used to communicate
                  with Rhythmyx Server.

  "communityname" - the name of the community, this is only a reference
                   to the communityid, but is not used by the WebDAV.

  "locale" - the locale that is used for the created items

  "deleteas" - an optional attribute, indicating the behavior of the
               DELETE operation (in DELETE, COPY and MOVE method).
               The possible values are:

               "purge": indicate the DELETE operation will purge the
                       target items and/or folders. The purged items
                       will not be able to recovered afterwards.

               "remove": indicate the DELETE operation will remove the
                       folder relationships with the target items. The
                       removed items can be recovered afterwards. This
                       is the default behavior if this attribute is not
                       specified.
```

Figure 2: Get Configuration screen

- Enter  
`http://hostname:9992/Rhythmyx/rxwebdav/[URLPattern]?sys_command=validateConfig`.  
 If the folder configuration's references to CM System elements are correct, the screen displays the message "WebDAV configuration successfully validated".

### Rhythmyx WebDav Configuration Validator

---

**Warnings found:**

- In content type 'rffImage', the file name will be set to both `displaytitle` and `'sys_title'` fields, but the `displayname` property always maps to `'sys_title'` field.
- In content type 'rffFile', the file name will be set to both `displaytitle` and `'sys_title'` fields, but the `displayname` property always maps to `'sys_title'` field.
- Field rules `isValidFile` must be removed from field `item_file_attachment` in content type 'rffFile' if there is no default value or other ways to set the value during insert or update process.

WebDav configuration successfully validated.

Figure 3: Configuration validation

The first two warnings displayed on the screen above appear if WebDAV is uploading a file into a Content Type with a required displaytitle field to avoid confusion over which field WebDAV displays as the item's title. If displaytitle is required, CM System WebDAV does the following when it uploads a file into CM System:

- It sets the CM System Content Item's displaytitle field to the file name.
- It sets the CM System Content Item's sys\_title field to the file

name. During processing after upload, WebDAV:

- Does not use the CM System Content Item's displaytitle field.
- Maps the CM System Content Item's sys\_title field to WebDAV's displayname property. When the value in one changes, the other sets itself to the new value.

Therefore, changing the Content Item's displaytitle field does not change the displayname that WebDAV shows to users. Explain this to your users who work with CM System WebDAV.

The third warning indicates that WebDAV does not fill the item\_file\_attachment field with any data, and because the field has a validation rule attached to it, an error may result. For example, if the validation rule requires that the field be filled, an error will result unless some method other than WebDAV is used to fill the field.

---

Note: In the folder configuration file, the URL pattern is the value of the *root* parameter in the <PSXWebdavConfigDef> element.

For example, in one of the sample folder configuration files, the <PSXWebdavConfigDef> element appears as:

```
<PSXWebdavConfigDef root="//Sites/EnterpriseInvestments/Files"
communityname="Enterprise Investments" communityid="1002" locale="en-us"
deleteas="remove" publicValidTokens="y" QEValidTokens="i">
```

To use the getConfig command using this URL pattern, you would enter:

```
http://hostname:9992/Rhythmyx/rxwebdav/Sites/EnterpriseInvestments/Files?sys_command=getConfig.
```

---

- 7** Open the folders in Windows Explorer or Internet Explorer, and if you plan to access the folders through a third-party application, configure the application as specified in its documentation.

WebDAV is now configured in your CM System system. You may now create CM System Content Items in Windows Explorer and your other third-party applications.

## Editing a Folder Configuration File

In CM System's WebDAV, a folder configuration file specifies the CM System Folder and Community the WebDAV configuration makes available, and the Content Types that users can manage in the folder through WebDAV. The folder configuration file DTD is `RxWebDAVconfig.dtd`, and the default folder configuration files in CM System are `RxWebDAVConfigEIIImages.xml`, `RxWebDAVConfigEIFiles.xml`, `RxWebDAVConfigCIIImages.xml` and `RxWebDavConfigCIFiles.xml`. They are located in:  
<Rhythmyx root>\AppServer\server\rx\deploy\rxapp.ear\rxapp.war\WEB-INF\config\user\webdav\.

A WebDAV configuration file consists of the following nodes:

`PSXWebdavConfigDef` - specifies the CM System WebDAV Folder and its Community. This node includes the following attributes:

`root` - specifies the CM System Folder. The value of this attribute must begin with either `“//Sites/”` (if the Folder you want to make available is a Site Folder) or `“//Folders/”` (if the Folder you want to make available is a child of the Folders node).

`communityname` - specifies the name of the Community you want to make available. NOTE: The Community property of the root folder can be either *All Communities* or the Community you specify here.

`communityid` - specifies the ID of the Community (CM System assigns the ID when you save the Community. To find the ID, view the Community in the Properties view or the Object Sorter in the Rhythmyx Workbench. The ID is the third number in the 3-part ID value.).

`locale` – Locale for Content Items created in this Folder.

`deleteas` – How the delete function for items and Folders behaves. Possible values are *remove* and *purge*. Note that if a Content Item is in either Public or Quick Edit State, CM System does not perform any action on that item. If the Content Item is in any other state, CM System executes the action normally.

- `remove` – Default. If a WebDAV-enabled Folder is deleted in CM System or a third-party application, permanently delete the Folder in CM System and remove the Content Items from the Folder, but save them in CM System.
- `Purge` – If a WebDAV-enabled Folder is deleted in CM System or a third-party application, permanently delete the Folder and the Content Items in CM System.

`publicValidTokens` - specifies a list of flags for Public States in the Workflow. Used with the attribute `QValidTokens` to implement automatic checkout of Content Items in the WebDAV Folder. Default value is *y*.

`QValidTokens` - specifies a list of flags for Quick Edit States in the Workflow. Used with the `publicValidTokens` attribute to implement automatic checkout of Content Items in the WebDAV Folder. Default value is *i*.



PSXWebdavContentType – One or more of these nodes may be present. Each node specifies a Content Type that users can maintain in the Folder.

id - specifies the ID of the Content Type. To find the ID, view the Content Type in the Properties view or the Object Sorter of the Workbench. The ID is the third number in the 3- part ID value.

name - specifies the name of the Content Type.

contentfield - specifies the Content Type field that stores the data file for the Content

Type. ownerfield - specifies the Content Type field that stores the lock owner data from

WebDAV. default - specifies whether the Content Type is the default Content Type for the servlet. The

default Content Type is the Content Type that processes any MIME Types not specified for another Content Type in the configuration. Only one Content Type in the configuration can be specified as the default Content Type. Values are *true* or *false*.

MIMETypes - specifies the MIME types of files that will be processed by the Content Type. Each MIME type is specified in a separate Mime Type child node.

PropertyMap - stores the mappings of WebDAV properties to the CM System fields. The name attribute of the PSXPropertyFieldNameMapping element specifies the WebDAV property. The FieldName child of the PSXPropertyFieldNameMapping specifies the CM System Content Type field that stores the data for the specified WebDAV property.

PSXPropertyFieldNameMapping name="displayname" - Required for FastForward Content Types. Specifies the display title of the Content Item. WebDAV passes the filename as the display title.

PSXPropertyFieldNameMapping name="getcontenttype" – Required (unless mapping a hashed field). Specifies the MIME type of the content.

PSXPropertyFieldNameMapping name="getcontentlength"– Required(unless mapping a hashed field). Specifies the length, in bytes, of the content.

ExcludeFolderProperties – specifies which CM System custom properties of the parent WebDAV Folder should not be inherited by sub-folders created in WebDAV. Note that the Folder Name and Description are automatically not inherited (you must specify a new Folder Name and Description is always blank).

PropertyName – sys\_pubFileName is listed because it specifies the name of the folder to publish to. This name would most likely differ in the sub-folder.

NOTE: Fields specified to store data for a WebDAV configuration must be from a System, Shared, or Local definition. You cannot use child fields to store WebDAV data.

## Simple Example Implementation of WebDAV

To illustrate a typical implementation using WebDAV, let us configure two WebDAV-enabled CM System

Folders for two departments in a

company. We will configure:

- a WebDAV-enabled USMarketing Folder for the Default Community. The folder is accessible to the Marketing\_Author and Marketing\_Artist Roles as well as the Default Role.
- a WebDAV-enabled Editorial Folder for the Editorial Community. The folder is accessible only to Editor and Administrator Roles.

To set up our sample WebDAV configuration:

- 1 First, as an Administrator, we create the folders in Content Explorer that we want to WebDAV-enable. We create the USMarketing Folder and assign it to the Default Community.

We give the Role Admin Read, Write, and Admin permission, and we give the Roles Marketing\_Author, Marketing\_Artist, and Default Read and Write permissions:

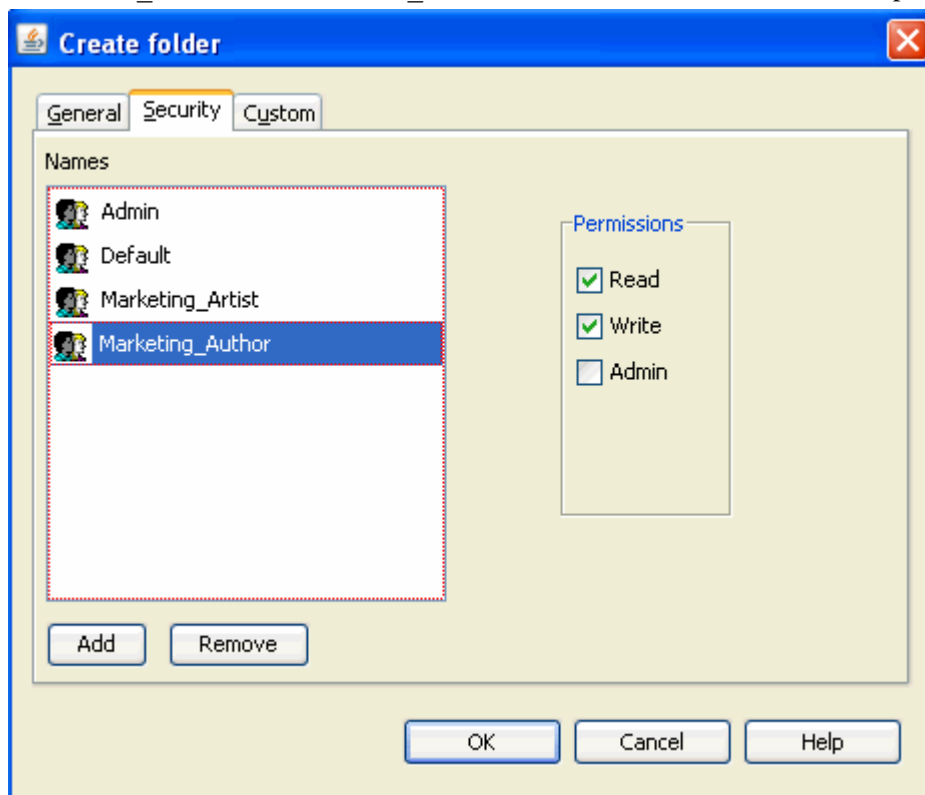


Figure 4: Create folder dialog, Security tab

Then we create the Editorial folder. We assign it to the Editorial Community and give the Admin Role Read, Write, and Admin permission and the Editor Role Read and Write permission.

Our completed folder structure appears as:

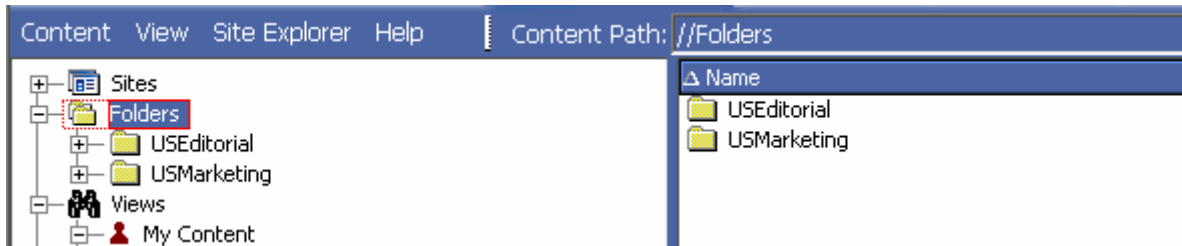


Figure 5: WebDAV-enabled Folders in CM System

## 2 We modify the WebDAV configuration file to define our two folders:

- We change the first `<bean id="name">` node to `<bean id="USMarketing">`;
- In the sub-element `<prop key="RxWebDAVConfig">` we change the name of the folder configuration file to `RxWebdavConfigMarketing.xml`.
- We change the second `<bean id="name">` node to `<bean id="USEditorial">`;
- In the sub-element `<prop key="RxWebDAVConfig">` we change the name of the folder configuration file to `RxWebdavConfigEditorial.xml`.

```
<!-- US -->
  <bean id="USMarketing"
    class="com.percussion.utils.spring.PSWebDavServletController">
    <property name="initParameters">
      <props>
        <prop key="debug">>true</prop>
        <prop
key="RxWebDAVConfig">RxWebDAVConfigMarketing.xml</prop>
      </props>
    </property>
  </bean>
  <bean id="USEditorial"
    class="com.percussion.utils.spring.PSWebDavServletController">
    <property name="initParameters">
      <props>
        <prop key="debug">>true</prop>
        <prop
key="RxWebDAVConfig">RxWebDAVConfigEditorial.xml</prop>
      </props>
    </property>
  </bean>
```

At the bottom of the file change the name in two of the `<prop>` elements in the `<bean id="urlMapping">` element to custom URL paths and the custom names.

```
<!-- add mappings here -->
<bean id="urlMapping"
  class="com.percussion.utils.spring.PSUrlHandlerMapping">
  <property name="mappings">
```

```
<props>
  <prop key="Folders/USMarketing/**">USMarketing</prop>
  <prop key="Folders/USEditorial/**">USEditorial</prop>
</props>
</property>
</bean>
```

We delete the beans and properties that we are not using to prevent WebDAV from failing.

- 3 Next, we copy existing folder configuration files to create `RxWebdavConfigMarketing.xml` and `RxWebdavConfigEditorial.xml`. Then we open each folder configuration file and specify the CM System Folder and Community.

In `RxWebdavConfigMarketing.xml`, we make the following changes:

- we change root to `//Folders/USMarketing`;
- to make the Folder `USMarketing` available to members of the Default Community, we change the value of `communityname` to `Default` and the value of `communityid` to `Default's Community ID`.
- we leave the default Content Type of `rffFile` and modify the `rffImage` Content Type node to define the `PDFfile` Content Type;
- in the `PDFfile` definition:
  - we specify `contentfield` as `pdfupload`, the name of the `PDFFile` Content Type field that stores WebDAV content;
  - we leave `ownerfield` as `webdavowner`, since the name of the `PDFFile` Content Type field that stores the WebDAV user who has a lock on the item is also `webdavowner`;
  - we set `default` to `false`, because the `File` Content Type is already the default.
- we change `<Mimetype>` to the appropriate value for `PDFfile`, `application/pdf`;
- in `<PropertyMap>`, we map the WebDAV field `getcontenttype` to the CM System Content Type field `pdfupload_type`. These are the fields in each application that store the MIME type. We also map the WebDAV field `getcontentlength` to the CM System Content Type field `pdfupload_size`. These are the fields in each application that store the Content Length.

```
<PSXWebdavConfigDef root="//Folders/USMarketing" communityname="Default"
communityid="10" locale="en-us" deleteas="remove" publicValidTokens="y"
QEValidTokens="i">>
```

```
<!--
  Defines one or more supported content type for WebDAV

  "id" - the id of the content type in Rhythmyx

  "name" - the name of the content type in Rhythmyx, this is only
used as a
          reference to the id above.

  "contentfield" - the field name for the content
```

"ownerfield" - the field name for the lock owner that is submitted by WebDAV clients

"default" - "true" or "false". There must be only one element with the value "true".

If it is "false", must define one or more mime types (hence must define the "Mimetypes" element);

If it is "true", the "Mimetypes" element is ignored.

```
-->
```

```
<PSXWebdavContentType id="307" name="PDFFile"
contentfield="pdfupload" ownerfield="webdavowner" default="false">
```

```
<!-- Defines the mime type(s) that be accepted by the current
Rhythmyx content type.-->
```

```
<MimeTypes>
  <MimeType>application/pdf</MimeType>
</MimeTypes>
```

```
<!--
```

```
  Defines the mapper between WebDAV properties and its
  corresponding field name of the current Rhythmyx content type
```

Required mappings:

"getcontenttype" - maps to a Rhythmyx Content Editor field for the mime-type.

"getcontentlength" - maps to a Rhythmyx Content Editor field for the content length.

Rhythmyx WebDAV automatically maps the following:

"displayname" - always maps to Rhythmyx sys\_title field

"creationdate" - maps to Rhythmyx sys\_contentcreatedate field

"getlastmodified" - maps to Rhythmyx sys\_contentlastmodifieddate field

"getcontentlanguage" - always maps to Rhythmyx sys\_lang field

```
-->
```

```
<PropertyMap>
  <PSXPropertyFieldNameMapping name="displayname">
    <FieldName>displaytitle</FieldName>
  </PSXPropertyFieldNameMapping>
  <PSXPropertyFieldNameMapping name="getcontenttype">
    <FieldName>pdfupload_type</FieldName>
  </PSXPropertyFieldNameMapping>
```

```
        <PSXPropertyFieldNameMapping name="getcontentlength">
            <FieldName>pdfupload_size</FieldName>
        </PSXPropertyFieldNameMapping>
    </PropertyMap>
</PSXWebdavContentType>

<PSXWebdavContentType id="309" name="rffFile"
contentfield="item_file_attachment" ownerfield="webdavowner"
default="true">

    <PropertyMap>
        <PSXPropertyFieldNameMapping name="displayname">
            <FieldName>displaytitle</FieldName>
        </PSXPropertyFieldNameMapping>
        <PSXPropertyFieldNameMapping name="getcontenttype">
            <FieldName>item_file_attachment_type</FieldName>
        </PSXPropertyFieldNameMapping>
        <PSXPropertyFieldNameMapping name="getcontentlength">
            <FieldName>item_file_attachment_size</FieldName>
        </PSXPropertyFieldNameMapping>
    </PropertyMap>
</PSXWebdavContentType>

<!--
    A list of excluded folder properties when creating a folder in
WebDAV.
    A created folder will inherit all properties of its parent folder
except
    the list of excluded properties and the folder name.
-->
<ExcludeFolderProperties>
    <PropertyName>sys_pubFileName</PropertyName>
</ExcludeFolderProperties>
```

```
</PSXWebdavConfigDef>
```

In `RxWebdavConfigEditorial.xml`, we change root to `//Folders/USEditorial`, and we make the Folder Editorial available to members of the Editorial Community. We make the `rffImage` Content Type the default, and add a node for the Logo Content Type. The changes we make to the `<PSXWebdavContentType>` node for Logo are similar to those we made for the `PDFFile` Content Type, above.

```
<PSXWebdavConfigDef root="//Folders/USEditorial"
communityname="Editorial" communityid="1007" locale="en-us"
deleteas="remove" publicValidTokens="y" QEValidTokens="i">>
```

```
<!--
    Defines one or more supported content type for WebDAV

    "id" - the id of the content type in Rhythmyx

    "name" - the name of the content type in Rhythmyx, this is only
used as a
        reference to the id above.
```

"contentfield" - the field name for the content

"ownerfield" - the field name for the lock owner that is submitted by WebDAV clients

"default" - "true" or "false". There must be only one element with the value "true".

If it is "false", must define one or more mime types (hence must define the "Mimetypes" element);

If it is "true", the "Mimetypes" element is ignored.

-->

```
<PSXWebdavContentType id="307" name="rffImage" contentfield="img1"
ownerfield="webdavowner" default="true">
```

<!-- Defines the mime type(s) that be accepted by the current Rhythmyx content type.-->

```
<MimeTypes>
  <MimeType>image/gif</MimeType>
  <MimeType>image/bmp</MimeType>
  <MimeType>image/jpeg</MimeType>
  <MimeType>image/png</MimeType>
</MimeTypes>
```

<!--

Defines the mapper between WebDAV properties and its corresponding field name of the current Rhythmyx content type

Required mappings:

"getcontenttype" - maps to a Rhythmyx Content Editor field for the mime-type.

"getcontentlength" - maps to a Rhythmyx Content Editor field for the content length.

Rhythmyx WebDAV automatically maps the following:

"displayname" - always maps to Rhythmyx sys\_title field

"creationdate" - maps to Rhythmyx sys\_contentcreatedate field

"getlastmodified" - maps to Rhythmyx sys\_contentlastmodifieddate field

"getcontentlanguage" - always maps to Rhythmyx sys\_lang field

-->

```
<PropertyMap>
  <PSXPropertyFieldNameMapping name="displayname">
```

```
        <FieldName>displaytitle</FieldName>
    </PSXPropertyFieldNameMapping>
    <PSXPropertyFieldNameMapping name="getcontenttype">
        <FieldName>img1_type</FieldName>
    </PSXPropertyFieldNameMapping>
    <PSXPropertyFieldNameMapping name="getcontentlength">
        <FieldName>img1_size</FieldName>
    </PSXPropertyFieldNameMapping>
    </PropertyMap>
</PSXWebdavContentType>

<PSXWebdavContentType id="339" name="Logo" contentfield="logobody"
ownerfield="webdavowner" default="false">

    <PropertyMap>
        <PSXPropertyFieldNameMapping name="displayname">
            <FieldName>displaytitle</FieldName>
        </PSXPropertyFieldNameMapping>
        <PSXPropertyFieldNameMapping name="getcontenttype">
            <FieldName>logobody_type</FieldName>
        </PSXPropertyFieldNameMapping>
        <PSXPropertyFieldNameMapping name="getcontentlength">
            <FieldName>logobody_size</FieldName>
        </PSXPropertyFieldNameMapping>
    </PropertyMap>
</PSXWebdavContentType>

<!--
    A list of excluded folder properties when creating a folder in
WebDAV.
    A created folder will inherit all properties of its parent folder
except
    the list of excluded properties and the folder name.
-->
<ExcludeFolderProperties>
    <PropertyName>sys_pubFileName</PropertyName>
</ExcludeFolderProperties>

</PSXWebdavConfigDef>
```



After configuration is complete, we restart the Rhythmyx Server. In our Web browser, we enter:

```
http://hostname:9992/Rhythmyx/rxwebdav/Folders/USMarketing?sys_command=getConfig
```

The browser may prompt you to enter a username and password. Enter a username/password associated with the same Community as the WebDAV folder. If the WebDAV servlet has deployed correctly the browser displays the contents of the WebDAV configuration file.

```

- <PropertyMap>
- <PSXPropertyFieldNameMapping name="displayname">
  <FieldName>displaytitle</FieldName>
</PSXPropertyFieldNameMapping>
- <PSXPropertyFieldNameMapping name="getcontenttype">
  <FieldName>pdfupload_type</FieldName>
</PSXPropertyFieldNameMapping>
- <PSXPropertyFieldNameMapping name="getcontentlength">
  <FieldName>pdfupload_size</FieldName>
</PSXPropertyFieldNameMapping>
</PropertyMap>
</PSXWebdavContentType>
- <PSXWebdavContentType id="309" name="rffFile">
  contentfield="item_file_attachment" ownerfield="webdavowner" default="true">
- <PropertyMap>
- <PSXPropertyFieldNameMapping name="displayname">
  <FieldName>displaytitle</FieldName>
</PSXPropertyFieldNameMapping>
- <PSXPropertyFieldNameMapping name="getcontenttype">
  <FieldName>item_file_attachment_type</FieldName>
</PSXPropertyFieldNameMapping>
- <PSXPropertyFieldNameMapping name="getcontentlength">
  <FieldName>item_file_attachment_size</FieldName>
</PSXPropertyFieldNameMapping>
</PropertyMap>
</PSXWebdavContentType>
- <!--
      A list of excluded folder properties when creating a folder
in WebDAV.

```

Figure 6: Folder Config file

We repeat using

```
http://hostname:9992/Rhythmyx/rxwebdav/Folders/USEditorial?sys_command=getConfig
```

- Then, we enter:

```
http://hostname:9992/Rhythmyx/rxwebdav/Folders/USMarketing?sys_command=validateConfig
```

If the WebDAV servlet is configured correctly, the screen displays a confirmation message.

We repeat using

```
http://hostname:9992/Rhythmyx/rxwebdav/Folders/USEditorial?sys_command=validateConfig.
```

Note that for this command, you must log in as a user in the Editorial Community.

- Now, we can create Network Places in Windows Explorer that make these CM System folders available through Windows Explorer. In the Add Network Place Wizard, we enter the locations for our WebDAV folders in the format `http://localhost:<Rhythmyx Server port>/Rhythmyx/rxwebdav/<URL pattern>`. The URL pattern is specified in the `root` parameter of the `<PSXWebDAVConfigDef>` element of the folder configuration file.

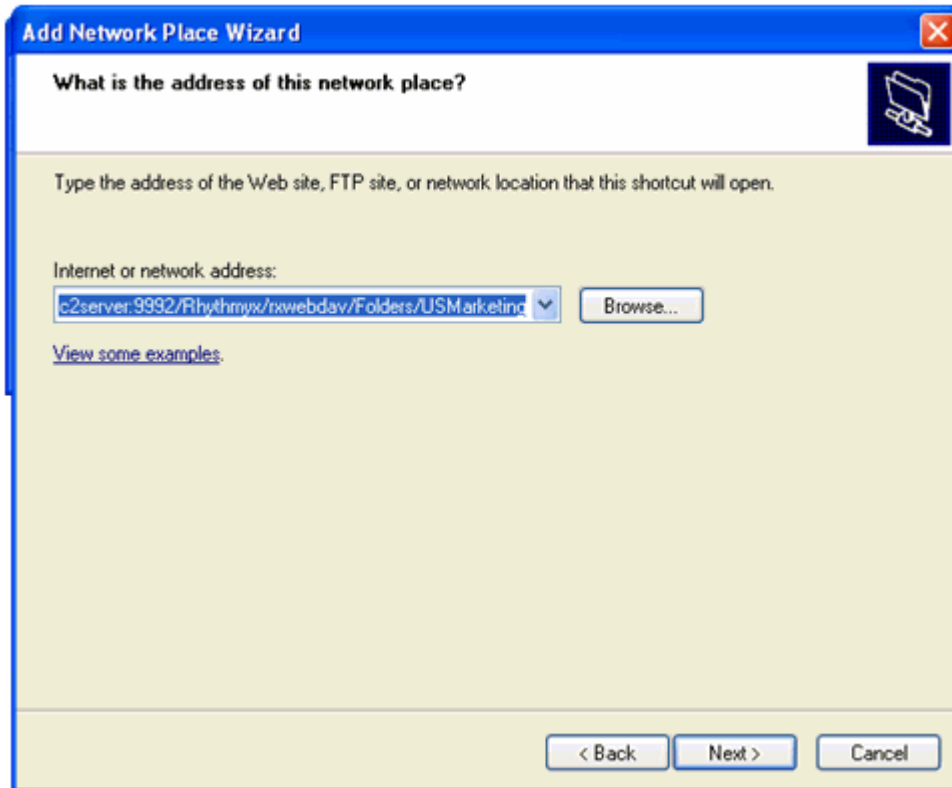


Figure 7: Add Network Place Wizard

Windows Explorer will ask us to log in when it connects to the WebDAV URL. The username must be associated with the same Community as the WebDAV folder.

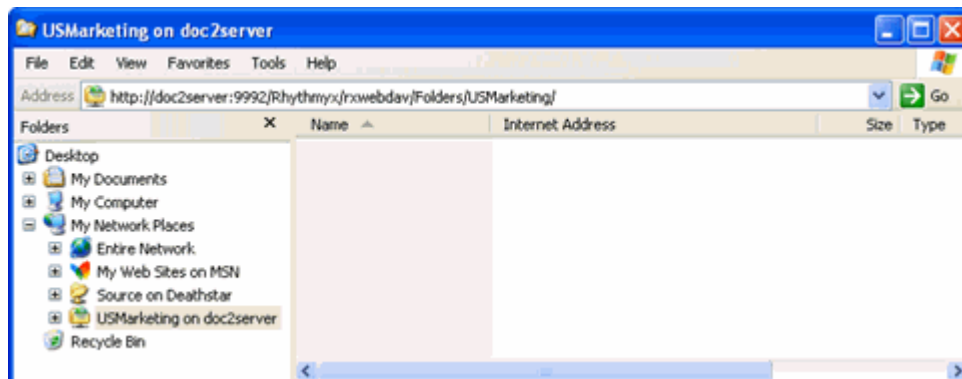


Figure 8: Network Places

- We perform any steps that are required to connect a third-party application to the WebDAV enabled folders.

## Webdav for Content Types with hash fields

Content Types with hash fields can be imported into the CMS using WebDav.

Webdav has been configured to accept Content Types with hash fields as well as out of box binary fields. For Content Types whose binary fields have been replaced with hash fields, the Webdav configuration

does not need to be changed. By default, Webdav first checks for binary {fieldname}. If it is no longer assigned to the content type, it checks instead for {fieldname}\_hash. So, for example, for the Image Content Type, Webdav will first look for img1 field and if it is not found will look for img1\_hash instead. If desired, the Webdav configuration can be updated to check directly for {fieldname}\_hash.

When using hash fields, the size and type field mappings are no longer required in the Webdav configuration or on the content type. To be able to remove these fields, the Webdav configuration should be updated to directly specify the {fieldname}\_hash. To continue to use these fields on the Content Type, they will be populated as before if the mappings exist.

For Content Types with both, binary as well as hash fields, the file will be uploaded to both. However, it is recommended that as soon as the de-duplication process has been run, the old, redundant binary fields be removed from the Content Types.

For hashed fields the getcontenttype and getcontentlength are not required. The \_size and \_type fields will be automatically populated if they are provided on the type, but are not themselves required.

---

## Turning off validation for Items uploading through WebDav

As of 7.3 a `sys_client` html parameter is provided to the editor when uploading files through webdav and will have a value of 'WebDav'. This can be used in field validation prerequisites to turn off a field that would normally validate. This will allow an item to be initially uploaded but a user would have to provide the field details for the required field when editing the item in Content Explorer. If field validation is used the user may still be able to transition the item without first opening it up. Item validation through an extension is required to prevent transition.

## Troubleshooting WebDAV

If your WebDAV implementation is not working, use the following HTTP commands to check your configuration:

- `getConfig`: retrieves the configuration file;
- `help`: lists help for other commands;
- `validateconfig`: checks the validity of the configuration and returns a message indicating whether the configuration passed or failed the validation. If it fails the validation, the returned message includes a list of errors and warnings.

Issue these commands as values of the `sys_command` parameter:

```
http://localhost:9992/Rhythmyx/rxwebdav/Sites/EnterpriseInvestments/Files?sys_command=getconfig
http://localhost:9992/Rhythmyx/rxwebdav/Sites/EnterpriseInvestments/Files?sys_command=help
http://localhost:9992/Rhythmyx/rxwebdav/Sites/EnterpriseInvestments/Files?sys_command=validateconfig
```

## CHAPTER 3

# Working with WebDAV

This chapter outlines specific techniques for working with WebDAV, including

- How to access WebDAV-enabled Folders in Windows Explorer and Internet Explorer
- How to work with CM System Content in WebDAV-enabled Folders
- How WebDAV works with CM System Workflow

---

## Opening Web Folders in Windows Explorer

In Windows, you can access the WebDAV server from Web Folders.

You can create Web Folders that correspond to your WebDAV-enabled CM System Folders in Windows 2000 and later versions. No special configuration is required to access WebDAV folders through Windows Explorer.

To set up a Web Folder through Windows Explorer:

- 1 In the Windows Explorer navigation pane, click on the My Network Places node, then double-click the Add Network Place icon.

Depending on your version of Windows, you may have to follow a different set of instructions to access the Add Network Place wizard. See your Windows Help for the correct instructions.

- 2 Windows Explorer displays the Add Network Place wizard.

- 3 Enter the URL of the CM System WebDAV server. Use the format:

`http://hostname:9992/Rhythmyx/rxwebdav/<URL pattern>`.

- 4

The URLs for the default CM System WebDAV files are:

`http://hostname:9992/Rhythmyx/rxwebdav/Sites/EnterpriseInvestments/Files`

`http://hostname:9992/Rhythmyx/rxwebdav/Sites/EnterpriseInvestments/Images`

`http://hostname:9992/Rhythmyx/rxwebdav/Sites/CorporateInvestments/Files`

`http://hostname:9992/Rhythmyx/rxwebdav/Sites/CorporateInvestments/Images`.

NOTE: The URL pattern is specified in the *root* parameter of the <PSXWebDAVConfigDef> element of the folder configuration file.

- 5 When you click [**Next**], Windows Explorer displays the Enter Network Password dialog. Enter a CM System username that has access to the same Community as the WebDAV folder and its password.
- 6 When you click [**Next**], Windows Explorer displays the Completing the Add Network Place dialog. Click [**Finish**].
- 7 Windows Explorer adds the Folder as a Network Place. Now it appears when you click the My Network Places node. You can add files to this location, and open, modify, and save files in the location. All of your changes will appear in the Folder of the same name in CM System, and all of the changes that you make to the Folder in CM System will appear in this Folder.

When you attempt to access a file in this location, Windows Explorer may display the Enter Network Password dialog. Enter a CM System username/password that has access to the same Community as the folder to complete the action.

---

If your version of Windows Explorer does not allow you to add a Network Place, open your WebDAV Folders in Internet Explorer instead.

---

---

# Opening Web Folders in Internet Explorer

In Windows, you can access WebDAV folders from Internet Explorer.

You can create Web Folders that correspond to your WebDAV-enabled CM System Folders in Internet Explorer Version 5.0 and above; however, the Web Folders feature must be installed in Internet Explorer. If the Web Folders feature is not installed in your Internet Explorer installation, go to [support.microsoft.com](http://support.microsoft.com) (<http://support.microsoft.com>) or refer to Internet Explorer's help for installation instructions.

To set up a Web Folder through Internet Explorer:

- 1 In the menu bar, choose *File > Open*.

Internet Explorer displays the Open dialog.

- 2 Enter the URL of the WebDAV folder you want to access from Internet Explorer. Use the format:

`http://<WebDAV host>:<WebDAV port>/Rhythmyx/rxwebdav/<URL`

`pattern>`. The URLs for the default Web folders configured in Rhythmyx are:

`http://hostname:9992/Rhythmyx/rxwebdav/Sites/EnterpriseInvestments/Files`

`http://hostname:9992/Rhythmyx/rxwebdav/Sites/EnterpriseInvestments/Images`

`http://hostname:9992/Rhythmyx/rxwebdav/Sites/CorporateInvestments/Files`

`http://hostname:9992/Rhythmyx/rxwebdav/Sites/CorporateInvestments/Images`

NOTE: The servlet URL pattern is specified in the Root parameter of the folder configuration file.

- 3 Check **Open as Web Folder**.

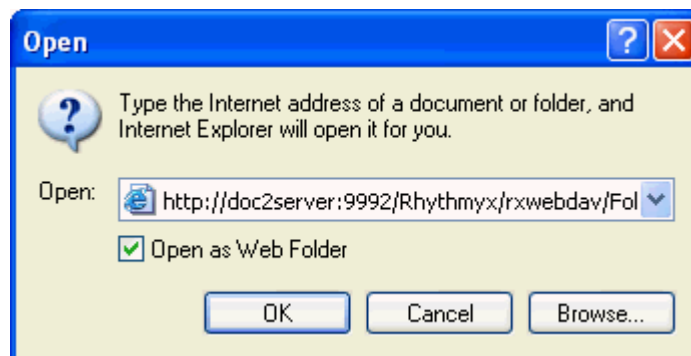


Figure 9: Opening a WebDAV Folder in Internet Explorer

- 4 Click **[OK]**.

- 5 If Internet Explorer prompts you for a username and password, enter a CM System username that has access to the same Community as the folder and its password.
- 6 Internet Explorer adds the Folder. You can add files to this location, and open, modify, and save files in the location. All of your changes will appear in the Folder of the same name in CM System, and all of the changes that you make to the Folder in CM System will appear in this Folder.

When you attempt to access a file in this location, Internet Explorer may display the Enter Network Password dialog. Enter the CM System username that has access to the same Community as the folder and its password.



---

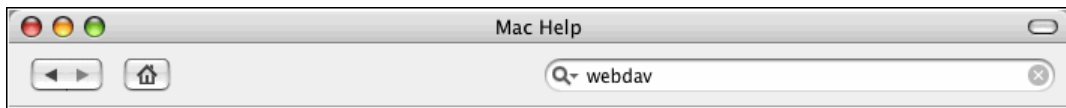
## Opening Web Folders in Mac OS X

In a Mac OS X, you can access the WebDAV server from Web Folders.

You can use the Connect to Server command to access a WebDAV server.

To set up a Web Folder through a Mac OS:

- 1 In the Finder, select *Go > Connect to Server*.  
Enter the address of the server in the Server Address field.  
The server address should be in a form similar to this: `http://servername.com/path`
- 2 Click the Connect button.



*Figure 10: Connecting to WebDAV on the Mac*

If you change your network settings (by opening Network preferences or by choosing a new location from the Apple menu) while you are connected to the WebDAV server, you should disconnect from the server, then reconnect to it.

To disconnect, select the server and choose Eject from the File menu.

---

## Working with CM System Content in Web Folders

Some third-party applications require you to set up WebDAV-enabled folders to communicate with a WebDAV server; others require more specialized configuration. Consult your application's documentation to determine whether it supports WebDAV and for instructions about configuring WebDAV support.

Once you have configured WebDAV in CM System, you can:

- add new Content Items to CM System directly from your third party applications by saving them to the WebDAV Folder (you will be prompted for your rxwebdav login);
- add new Content Items to CM System by dragging them into Web Folders in Windows Explorer;
- access existing Content Items directly from your third-party application or Windows Explorer.

In all cases, you will be prompted for your rxwebdav login when you initially access the WebDAV folder. New Files are added to CM System in the Content Type specified for their MIME Type, or to the default Content Type if you have not specified a MIME Type.

In CM System, if you copy or create a Content Item in a WebDAV-enabled Folder, the Content Item will appear in the Folder from third-party applications only if the Folder's configuration file handles the MIME type associated with the Content Item.

---

## WebDAV and CM System Workflow

Content Items in WebDAV folders are subject to standard CM System Workflow restrictions. Two issues in particular affect WebDAV.

First, when a user attempts to open a file in a WebDAV-enabled directory, CM System attempts to check out the associated Content Item to that user. If the Content Item is not in a Public State, the checkout occurs normally. If the Content Item is in a Public State, the WebDAV client issues a second request to Transition the Content Item to Quick Edit, then checkout the Content Item. The following processing occurs:

1. The WebDAV client requests the Content Item.
2. CM System returns a message that the Content Item is not available. The message includes the Content Item metadata, part of which is the value of the CONTENTVALID Flag.
3. The WebDAV client compares the value of CONTENTVALID flag to the values in the publicValidTokens parameter in the WebDAV servlet configuration.
4. If any of these values match, WebDAV assumes the Content Item is in a Public state and issues another request to Transition the Content Item to the first State in the Workflow where CONTENTVALID matches the value of the QEValidTokens in the WebDAV configuration.
5. Once this Transition is completed successfully, CM System checks the Content Item out to the user. CM System also records a Transition comment indicating that the Transition was performed by WebDAV.

Note that the default values of these attributes:

publicValidToken=y

QEValidToken=i

These values match the standard CONTENTVALID values used for Public (y) and Quick Edit (i) States in CM System Workflows. If you use different values for these States, you will need to change the values of these attributes in the WebDAV configuration to match.

Next, the State of a Content Item affects the processing of Purge requests from WebDAV. When you attempt to delete a Content Item from a WebDAV folder WebDAV issues a PURGE request. The value of the `deleteas` attribute of the WebDAV configuration determines the processing of this request. If the value of the `deleteas` attribute is *purge*, then CM System purges the Content Item from the Repository. The Content Item cannot be recovered. If the value of the `deleteas` attribute is *remove*, CM System only removes the Content Item from the Folder; it still exists in the Repository and can be recovered. In either case, however, if the Content Item is in a Public State, no processing occurs.



# Appendices

---

## APPENDIX I

### Default Folder Configuration File

```
<?xml version="1.0" encoding="UTF-8"?>
<!-- The WebDAV configuration information

"root" - the virtual root path of the Rhythmyx for all resource
         specified in the request. It must start from "Site" or "Folder"
         For example, root="//Site/mysite", root="//Folder/myfolder".
         It must be an existing folder.

"communityid" - the name of the community, which is used to communicate
               with Rhythmyx Server.

"communityname" - the name of the community, this is only a reference
                 to the communityid, but is not used by the WebDAV.

"locale" - the locale that is used for the created items

"deleteas" - an optional attribute, indicating the behavior of the
             DELETE operation (in DELETE, COPY and MOVE method).
             The possible values are:

             "purge": indicate the DELETE operation will purge the
                     target items and/or folders. The purged items
                     will not be able to recovered afterwards.
             "remove": indicate the DELETE operation will remove the
                     folder relationships with the target items. The
                     removed items can be recovered afterwards. This
                     is the default behavior if this attribute is not
                     specified.

"publicValidTokens" - comma separated list of chars for public state
                    default "y"

"QEValidTokens" - comma separated list of chars for quick-edit state
                 default "i"
-->
<PSXWebdavConfigDef root="//Sites/EnterpriseInvestments/Images"
communityname="Enterprise_Investments" communityid="1002" locale="en-us"
deleteas="remove"
publicValidTokens="y" QEValidTokens="i">
```

```
<!--  
  Defines one or more supported content type for WebDAV  
  
  "id" - the id of the content type in Rhythmyx  
  
  "name" - the name of the content type in Rhythmyx, this is only  
used as a      reference to the id above.
```

"contentfield" - the field name for the content

"ownerfield" - the field name for the lock owner that is submitted by WebDAV clients

"default" - "true" or "false". There must be only one element with the value "true".

If it is "false", must define one or more mime types (hence must define the "Mimetypes" element);

If it is "true", the "Mimetypes" element is ignored.

-->

```
<PSXWebdavContentType id="307" name="rffImage" contentfield="img1"
ownerfield="webdavowner" default="false">
```

<!-- Defines the mime type(s) that be accepted by the current Rhythmyx content type.-->

```
<MimeTypes>
  <MimeType>image/gif</MimeType>
  <MimeType>image/bmp</MimeType>
  <MimeType>image/jpeg</MimeType>
  <MimeType>image/png</MimeType>
</MimeTypes>
```

<!--

Defines the mapper between WebDAV properties and its corresponding field name of the current Rhythmyx content type

Required mappings:

"getcontenttype" - maps to a Rhythmyx Content Editor field for the mime-type.

"getcontentlength" - maps to a Rhythmyx Content Editor field for the content length.

Rhythmyx WebDAV automatically maps the following:

"displayname" - always maps to Rhythmyx sys\_title field

"creationdate" - maps to Rhythmyx sys\_contentcreatedate field

"getlastmodified" - maps to Rhythmyx sys\_contentlastmodifieddate field

"getcontentlanguage" - always maps to Rhythmyx sys\_lang field

-->

```
<PropertyMap>
  <PSXPropertyFieldNameMapping name="displayname">
    <FieldName>displaytitle</FieldName>
```





```

        </PSXPropertyFieldNameMapping>
        <PSXPropertyFieldNameMapping name="getcontenttype">
            <FieldName>img1_type</FieldName>
        </PSXPropertyFieldNameMapping>
        <PSXPropertyFieldNameMapping name="getcontentlength">
            <FieldName>img1_size</FieldName>
        </PSXPropertyFieldNameMapping>
    </PropertyMap>
</PSXWebdavContentType>

<PSXWebdavContentType id="309" name="rffFile"
contentfield="item_file_attachment" ownerfield="webdavowner" default="true">

    <PropertyMap>
        <PSXPropertyFieldNameMapping name="displayname">
            <FieldName>displaytitle</FieldName>
        </PSXPropertyFieldNameMapping>
        <PSXPropertyFieldNameMapping name="getcontenttype">
            <FieldName>item_file_attachment_type</FieldName>
        </PSXPropertyFieldNameMapping>
        <PSXPropertyFieldNameMapping name="getcontentlength">
            <FieldName>item_file_attachment_size</FieldName>
        </PSXPropertyFieldNameMapping>
    </PropertyMap>
</PSXWebdavContentType>

<!--
    A list of excluded folder properties when creating a folder in
WebDAV.
    A created folder will inherit all properties of its parent folder
except
    the list of excluded properties and the folder name.
-->
<ExcludeFolderProperties>
    <PropertyName>sys_pubFileName</PropertyName>
</ExcludeFolderProperties>

</PSXWebdavConfigDef>

```

## APPENDIX II

# WebDAV Methods

Supported WebDAV Methods:

WebDAV Method	CM System Processing
OPTIONS	Requests the allowed WebDAV methods for the resource identified by the Request-URI.

---

GET	Retrieves the content of the resource (or item) identified by the Request-URI. If the resource is a folder, then retrieves all properties of its immediate children.
HEAD	Retrieves the header information as in the GET method, but not the content.
PROPFIND	Retrieves properties defined on the resource identified by the Request-URI.
PROPPATCH	Processes instructions specified in the request body to set properties defined on the resource identified by the Request-URI. Removing properties is not supported.
MKCOL	Creates a new collection or folder.
PUT	Creates a new item or updates the content of an existing item. A PUT cannot be performed on a collection (or folder) according to the WebDAV specification.
DELETE	Deletes an item or a folder identified by the Request-URI. Deletes a folder will also recursively delete all its child items and sub-folders if exists.
COPY	Copies items and folders from one location to another. Supporting “0” and “infinity” depth for copying folders.
MOVE	Moves items and folders from. Move for properties is not supported.
LOCK	Checks out the resource identified by the Request-URI.
UNLOCK	Checks in the resource identified by the Request-URI.