

VirtualViewer®

VirtualViewer® HTML5 ICN Connector Installation Guide

Note:

An online version of this manual contains information on the latest updates to VirtualViewer. To find the most recent version of this manual, please visit the online version at www.virtualviewer.com or download the most recent version from our website at www.snowbound.com/support/manuals.html.



DOC-0190-01

Copyright Information

While Snowbound® Software believes the information included in this publication is correct as of the publication date, information in this document is subject to change without notice.

UNLESS EXPRESSLY SET FORTH IN A WRITTEN AGREEMENT SIGNED BY AN AUTHORIZED REPRESENTATIVE OF SNOWBOUND SOFTWARE CORPORATION MAKES NO WARRANTY OR REPRESENTATION OF ANY KIND WITH RESPECT TO THE INFORMATION CONTAINED HEREIN, INCLUDING WARRANTY OF MERCHANTABILITY AND FITNESS FOR A PURPOSE. Snowbound Software Corporation assumes no responsibility or obligation of any kind for any errors contained herein or in connection with the furnishing, performance, or use of this document.

Software described in Snowbound documents (a) is the property of Snowbound Software Corporation or the third party, (b) is

furnished only under license, and (c) may be copied or used only as expressly permitted under the terms of the license.

All contents of this manual are copyrighted by Snowbound Software Corporation. The information contained herein is the exclusive property of Snowbound Software Corporation and shall not be copied, transferred, photocopied, translated on paper, film, electronic media, or computer-readable form, or otherwise reproduced in any way, without the express written permission of Snowbound Software Corporation.

Microsoft, MS, MS-DOS, Windows, Windows NT, and SQL Server are either trademarks or registered trademarks of Microsoft Corporation in the United States and/or other countries.

Adobe, the Adobe logo, Acrobat, and the Acrobat logo are trademarks of Adobe Systems Incorporated.

Sun, Sun Microsystems, the Sun Logo, and Java are trademarks or registered trademarks of Sun Microsystems, Inc. in the United States and other countries.

Kakadu JPEG2000©, is copyrighted by Dr. David Taubman, and is proprietary to NewSouth Innovations, Pty. Ltd, Australia.

iText, the Initial Developers of the Original Code are Bruno Lowagie and Paolo Soares. Portions created by Bruno Lowagie are Copyright ©1999-2009 by Bruno Lowagie.

Aspose™, Aspose.Cells© (copyrighted 2003), Aspose.Words© (copyrighted 2003), and Aspose.Slides© (copyrighted 2004), are all proprietary to Aspose Software, Pty. Ltd, Australia.

I-NET JWebEngine© is proprietary to I-NET Software GmbH, and shall remain the sole and exclusive property of I-Net Software GmbH.

United States Government Restricted Rights

The Software is provided with RESTRICTED RIGHTS. Use, duplication or disclosure by the United States Government is subject to restrictions as set forth under subparagraph (c)(1)(ii) of The Rights in Technical Data and Computer Software clause of DFARS 252.227 –19 or subparagraphs (c)(i) and (2) of the Commercial Computer Software-Restricted Rights at 48 CFR 52.227 – 19 as applicable. The Manufacturer is Snowbound Software Corporation, 309 Waverley Oaks Rd., Suite 401, Waltham, MA 02452, USA.

All other trademarks and registered trademarks are the property of their respective holders.

Manual Title: *Snowbound Software VirtualViewer® ICN Connector Installation Guide*

Part Number: DOC-0190-01

Revision: 01

Printing Date: March 2017

Published by Snowbound Software Corporation.

309 Waverley Oaks Road, Suite 401

Waltham, MA 02452 USA

phone: 617-607-2000

fax: 617-607-2002

©1996 - 2017 by Snowbound Software Corporation. All rights reserved.

Table of Contents

Chapter 1 - Introduction	5
System Requirements	5
IBM Content Navigator Installation	5
VirtualViewer® HTML5	5
Servlet Container	5
Java Version	5
Chapter 2 - Overview	6
Repositories	6
FileNet P8	6
Document	6
Annotation	7
Content Manager	7
Design Implementation	8
ICN Plugin	8
Launch Web Page	9
Content Handler	9
Content Manager	10
Chapter 3 - Installing VirtualViewer	12
VirtualViewer Web Application Installation	12
Chapter 4 - Installing the VirtualViewer® HTML5 ICN Connector	18
Snowbound VirtualViewer IBM Content Navigator Plugin Installation	18
Chapter 5 - Configuring the VirtualViewer® HTML5 ICN Connector	21

Define a Viewer Map	21
Assign an Action Menu	23
Enable Pop-ups	26
Restart and Go!	26
Chapter 6 - Installing and Configuring the FileNet P8 Connector	27
\WorkPlaceDir	27
\WebXMLDir	27
The FileNet P8 Connector Installation and Configuration	28
Appendix A - Troubleshooting	30
Image Does Not Load in a Single Sign-On (SSO) Environment	30
Websphere Will Not Start	30
404 errors Occur After Updating the Configuration Settings in config.js	30
Windows 404 Server or File Not Found Error Occurs When Referencing Wrong web.xml	31

Chapter 1 - Introduction

This installation guide explains how to install VirtualViewer® HTML5 ICN Connector.

If you need any help installing, configuring or using VirtualViewer® HTML5 ICN Connector, please contact Snowbound Technical Support by entering a support ticket at <http://support.snowbound.com>.

System Requirements

This section describes the system requirements to install the VirtualViewer® HTML5 ICNConnector.

IBM Content Navigator Installation

The VirtualViewer® HTML5 ICN Connector requires an installation of IBM Content Navigator in order to function.

VirtualViewer® HTML5

The VirtualViewer® HTML5 ICN Connector also requires VirtualViewer® HTML5 This is included in this package.

Servlet Container

VirtualViewer® HTML5 requires a J2SE or J2EE servlet container to run. You may choose any compliant servlet container, although recommended servlet containers include Apache Tomcat 4.x and higher, IBM Websphere 5.1 and higher, and BEA Weblogic 8.1 and higher.

It also requires a JRE of 1.4 or higher.

Java Version

VirtualViewer requires the client machine to have a JRE of 1.4 or higher.

Chapter 2 - Overview

The Snowbound VirtualViewer product is integrated to an existing enterprise content management system environment. It provides a method to view, update, delete documents, notes, and annotations stored in the enterprise content management system.

The design is targeted to integrate and support the latest IBM Content Navigator front-end browser application software. IBM Content Navigator version 2.0.3 fix pack 3 is used during the design and delivery of this project.

VirtualViewer is integrated to IBM Content Navigator, supports access and manipulation of documents from three different IBM Content Navigator supported enterprise content manager repository types:

1. FileNet Content Manager (P8)
2. CMIS
3. Content Manager

The coupling of Snowbound VirtualViewer and IBM Content Navigator is achieved through three main components:

1. Snowbound VirtualViewer Plugin for IBM Content Navigator
2. launchViewer.jsp inside the Snowbound VirtualViewer web application (WAR) file
3. Pyramid ICN Connector which implements repository accesses

Repositories

Although the ICN mid-tier services provide much of the information required to get document, note, and annotation information from the repositories, it does not provide all the required interfaces to retrieve and manipulate the actual document, note, or annotations. This is achieved using individual repository application programming interface (API) calls created by the project.

Each repository type has its own data model and access configurations. Important aspects of the data model and access configurations are noted below. All repositories are providing media-format file content to the Snowbound VirtualViewer application.

FileNet P8

FileNet content manager (aka P8) uses object-oriented class constructs to represent content.

Document

A Document object represents a single version of a document stored in a Content Engine object store. Fundamentally, a document can be thought of as two things:

- The document that the user creates. This document--a word processing file, a graphic, a spreadsheet, an html file--is stored either in the object store's content store or stored externally. For example, a Web page that is referenced by its URL address. In either case, it is known as the Document object's content element.
- The object that describes the user's document. This object, which the Content Engine uses to identify and find that particular version of the user's document, is called a Document object.

A document can have zero or more associated content elements. If a document has content, the content is either stored in a Content Engine content store (referred to as a content transfer element) or stored externally (a content reference element). These elements are represented by **ContentTransfer** and **ContentReference** objects, respectively. For content-carrying documents, the Content Engine APIs support creating a document with one or more content transfer elements, one or more content reference elements, or a mix of content transfer and content reference elements.

A document can be versioned. The Content Engine provides support for three levels of versioning: no versioning enabled, single-level versioning, and two-level versioning. When you version a document by checking it into a Content Engine object store, a VersionSeries object is associated with it that contains all versions of that document.

Annotation

An Annotation object represents incidental information that can be attached to an object (its parent) for the purpose of annotating or footnoting that object. An Annotation object allows you to link additional information to a containable object (custom object, document, or folder). You can modify and delete an annotation independently of its parent object. However, you cannot create versions of an annotation separately from its parent object. By design, an annotation is deleted whenever its parent object is deleted.

A document annotation is uniquely associated with a single document version. It is not versioned or carried forward when its associated document version is updated and a new version created.

Content Manager

Content Manager stores its data in DB2.

The following is a quick summary to the Content Manager repository structure:

- An item type is defined
- The item type has document parts or resource parts
- Item types are created with base parts, note parts, or annotation parts
- Item types have properties associated to the media content

Design Implementation

An IBM Content Navigator plugin retrieves the identification for the document from the ICN repository connection. The plugin calls the VirtualViewer web application. The receiver is `launchViewer.jsp` which is added to the existing VirtualViewer web application. The application receives the security configuration and document access properties necessary to render and manage the document through the implement java server page (jsp).

The VirtualViewer application implements a content handler application program interface (API). The content handler API is developed and configured to use the information provided by the ICN plugin. The content handler API is further developed to implement certain native repository accesses. These accesses supplement the existing, and some new, virtual viewer application functionality.

VirtualViewer is created in a new browser application tab within the same browser session.

ICN Plugin

The plugin implements specific methods to expose feature characteristics for a viewer integration, action characteristics for a context menu definition, and plugin service configuration characteristic to store site-specific install configuration.

The action and viewer implementation uses the service configuration characteristics to contact the VirtualViewer web application. The URL required to contact the `launchViewer.jsp` is configured to provide an endpoint for the plugin to call into the web application. This populates the web session. The URL required to contact the Snowbound Ajax servlet is configured to provide an endpoint to communicate the content for rendering. This populates and provides information to the server to communicate back from the Snowbound web application to the ICN context session. [SnowboundVirtualViewerPlugin, SnowboundVirtualViewerPluginConfigurationService, SnowboundVirtualViewerAction]

The plugin is also responsible for security interpretation. When the user invokes the action or viewer to invoke the Snowbound viewer the security for the documents selected is requested by the plugin code. This information instruments the security parameters passed to the Snowbound VirtualViewer. [SnowboundVirtualViewerPluginSecurity]

The registered action and viewer calls the service for the plugin. This service collects the security and document information based on the repository type to call the Snowbound web application. The service invokes a new tab in the same session to respect the authentication for the current user. The new tab is populated by the Snowbound VirtualViewer application. [SnowboundVirtualViewerPluginService]

The plugin defines a viewer definition to IBM Content Navigator. This viewer definition specifies the supported repository types and the MIME types that this viewer supports. These MIME types should be synchronized with Snowbound VirtualViewer supported MIME types. [SnowboundVirtualViewerPluginViewerDef]

Additional software artifacts that support the IBM Content Navigator plugin framework are developed to aid the deployment and installation of the plugin. These artifacts are used by the

browser to render configuration pages and to supply JavaScript, client-centric (versus server) functionality. [SnowboundVirtualviewerPlugin.js, ConfigurationPane.js, etc]

Launch Web Page

The launchViewer.jsp file is a Java Server Page implementation. This component accepts the information populated by the Snowbound plugin to call the Snowbound Ajax servlet. This component is added to the existing Snowbound web application.

Additionally the newly developed content handler implementation to access the repositories is added to the Snowbound web application. This requires that the installer obtains the target repository runtime files to support the Snowbound content handler implementation back to the IBM Content Navigator supported repositories. The content handler is responsible for managing the content byte stream representing the media or document requested from the IBM Content Navigator-supported repository.

Content Handler

The main file for the Content handler is **NavigatorContentHandler.java**. It contains methods that implement the Snowbound content handler interface. The code to handle the P8 repository methods is contained in this file while the code for CM8 and CMOD is contained in **ManageDbConnection.java**.

For document viewing, a call is made from VirtualViewer to **getDocumentContent** in NavigatorContentHandler. From the getDocumentContent method, a **FilenetElements.java** object is created. This object contains all of the parameters passed into the launchViewer.jsp method. Next, a call is made to **getFilenetDocumentByteArray** passing in the FilenetElements object. Inside getFilenetDocumentByteArray, the code determines what type of repository the document came from and calls the appropriate method to retrieve a document's content as a byte array. For P8, native repository retrieval code is used, within the function, to retrieve the document from the repository. For CM8 and CMOD, calls are made to retrieveCM8DocumentByteArray and retrieveCMODDocumentByteArray respectively in ManageDbConnection. These functions use native CM8 and CMOD document retrieval functionality to retrieve the document's content.

The above methods retrieve the document's contents as a byte array and return it back to VirtualViewer along with the document's file name via a ContentHandlerResult object.

For P8 documents, the Daeja style annotations are also retrieved via the getAllAnnotationsForDocument function. The annotations are returned via a ContentHandlerResult object. Currently CM* and CMOD annotations are not supported.

Creation of a new document or updating an existing document is handled by the **saveDocumentComponents** function within NavigatorContentHandler. Permissions are first checked using the FilenetElements object and then the appropriate methods are called to create or update the documents. If permission is not allowed, the function will throw a FlexSnapSIAPException containing user facing error messages to be displayed to the user via VirtualViewer. For P8, the code inside the getAllAnnotationsForDocument calls native repository functions within saveNewP8Document and updateExistingP8DocumentContent respectively. For CM8, saveNewCM8Document and updateCM8Document are called in ManageDbConnection.

CMOD is an archive based repository and does not allow creation of or updating new documents.

Content Manager

The DB2 Content Manager Version 8 access control model is applied at the level of the controlled entity. A controlled entity is a unit of protected user data. In different DB2 Content Manager Installations, the controlled entity can be an individual item, item-type, or the entire library. Operations on the controlled entities are regulated by one or more control rules. The access control list (ACL) is the container for these control rules. The `DKAccessControlListICM` class represents a DB2 Content Manager Version 8 ACL. Every controlled entity in a DB2 Content Manager system must be bound to an ACL. Default settings can be configured by the system administrator, if appropriate. An ACL is associated with an item type to enforce access control at the item type level. Similarly, an item level access control is established by binding an ACL to the desired item when the item is created. When a user initiates an operation on the item, the system will check the user's privilege and the ACL associated with this item to determine if this user has the right to perform such an operation on this item. The right to access an item also requires the right to access the item type based on which this item type is defined. An ACL is created by calling one of its multiple constructors. If the data store with which this ACL shall be associated is known at the time of object creation, then one of the constructors that take a data store parameter is used. Otherwise, the default constructor is used and the `setDatastore` method is called on the created ACL instance once the data store instance has also been created.

A privilege is an ability to use the DB2 Content Manager system. DB2 Content Manager Privileges are used to grant to individual users and to define access control lists (ACLs). DB2 Content Manager Privileges must be grouped into privilege sets before they can be used. The DB2 Content Manager system provides a number of un-modifiable pre-defined privileges, called System-defined Privileges. Each of these privileges authorizes a certain operation(s). System-defined Privileges are enforced by the DB2 Content Manager Library Server stored procedures. The DB2 Content Manager system also allows users to define their application-specific privileges, called User-defined Privileges. Each Privilege has a system-generated unique code, called the Privilege Definition Code. Privilege Definition Code 0 to 999 are reserved to store DB2 Content Manager System-defined Privileges. Code 1000 and up are open for user-defined privileges. User-defined privileges contains the rights to access and modify the DB2 Content Manager controlled entities. Access to the controlled entities, in addition to ACL checking, is controlled by this category of privileges.

Some examples of user-defined privileges include:

- The privilege to query item types and attributes
- The privilege to query items
- The privilege to create items
- The privilege to move items between different item types

- The privilege to delete items
- The DKPrivilegeICM class is used to represent privileges in DB2 Content Manager data stores.

Chapter 3 - Installing VirtualViewer

Follow the steps below to install and configure VirtualViewer:

The Snowbound web application is delivered as a ZIP archive. The zip archive must be reviewed and modified by the target environment user to manage and manipulate the configuration of the web application before and during deployment to the target server.

To compliment the P8 login application security stanza in WebSphere for JBoss, add the following xml to jboss' **conf/login-config.xml** file:

```
<application-policy name="FileNetP8WSI">
<authentication>
<login-module code="com.filenet.api.util.WSILoginModule" flag-
g="required"/>
</authentication>
</application-policy>
```

VirtualViewer Web Application Installation

The Snowbound VirtualViewer WAR file needs to be installed in the IBM application server as the first step. The WAR file must be modified with specific content from the customer's application server before it can be installed. In this section, you will learn how to configure and then deploy the WAR file. A list of the installation steps are provided below.

Follow the steps below to install VirtualViewer:

1. Retrieve the Virtual Viewer, **VirtualViewerJavaHTML5.war file**.
2. Unzip the **VirtualViewerJavaHTML5.war** file by right-clicking on the zip file and select extract all.
3. Right-click on WAR file, use 7-zip (Windows zip will not work correctly) and extract the contents of WAR file.
In the archive is a web.xml file that will need to be updated with system specific properties.
4. Edit **PyraSol.properties**.
5. In the **PyraSol.properties** file, edit the LIBRARYSERVER parameter for CMOD.
For example: LIBRARYSERVER=w12r2ecm.
6. Update the following connection information values:

ObjectStore=<target P8 Object store>

PEConnectionPoint=<target P8 connection point>

userid=<administrative user for P8>

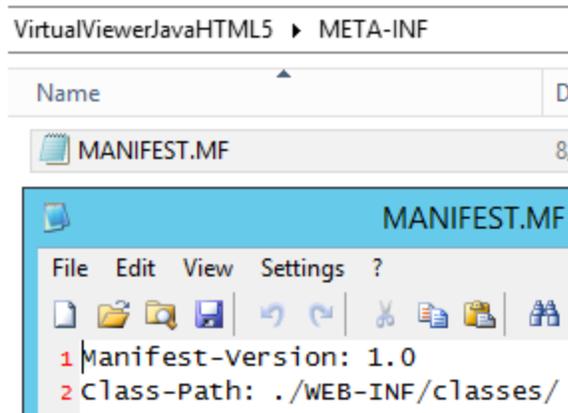
password=<administrative user for P8 password>

CEUri=<http://localhost:9080/wsi/FNCEWS40MTOM or equivalent P8 WebService endpoint>

7. Add or update **MANIFEST.MF**.

1. If VirtualViewerJavaHTML5/META-INF folder does not exist, create it.
2. Create or update the document MANIFEST.MF (not MANIFEST.MF.txt).
3. Add, insert, or append the following directive used to locate PyraSol.properties file:

Class-Path: ./WEB-INF/classes



8. Verify or add the IBM Content Navigator handler library.

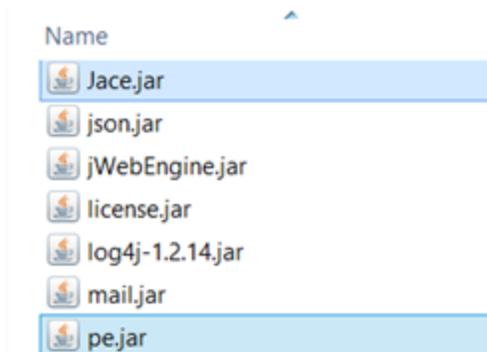
1. In the **VirtualViewerJavaHTML5/WEB-INF/lib** folder check for PyramidICNHandler.jar.
2. If PyramidICNHandler.jar is present, continue to the next step.
3. If PyramidICNHandler.jar is not present, request or contact Snowbound Support at <http://support.snowbound.com> to obtain the JAR file.
4. Add the JAR file to the **VirtualViewerJavaHTML5/WEB-INF/lib** folder.

9. Add the required repository runtime support files.

1. FileNet Content Manager (P8)
2. Locate **pe.jar** and **Jace.jar** from the target installation in one of the following loc-

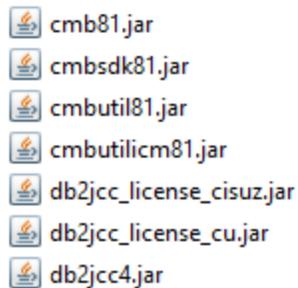
ations: **IBM\FileNet\CEClient\lib** or **IBM\FileNet\ContentEngine\lib**.

VirtualViewerJavaHTML5 > WEB-INF > lib

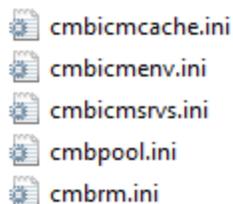


10. Copy **pe.jar** and **Jace.jar** to VirtualViewerJavaHTML5/WEB-INF/lib.

A number of CMB jars as shown below are required. These can be found on the installation host for Content Manager in the lib directory. Example: D:\IBM\db2cmv8\lib

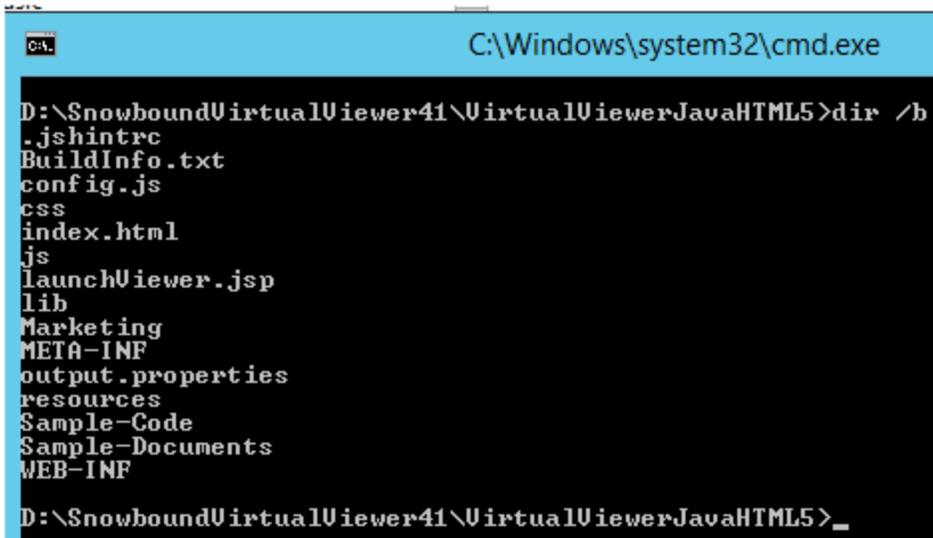


A number of INI files as shown below are required. These can be found on the installation host for Content manager in the cmgmt\connectors directory. Example: D:\IBM\db2cmv8\cmgmt\connectors



1. Copy the above files to VirtualViewerJavaHTML5/WEB-INF/lib.
2. Verify that **launchViewer.jsp** is present in web application folder.

3. Update or assemble the SnowboundVirtualViewer.war file.
4. Change the directory to the root where WEB-INF is:

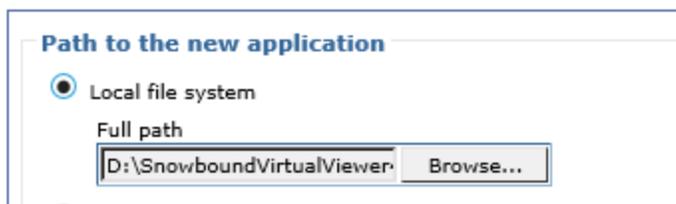


```
C:\Windows\system32\cmd.exe
D:\SnowboundVirtualViewer41\VirtualViewerJavaHTML5>dir /b
.jshintrc
BuildInfo.txt
config.js
css
index.html
js
launchViewer.jsp
lib
Marketing
META-INF
output.properties
resources
Sample-Code
Sample-Documents
WEB-INF
D:\SnowboundVirtualViewer41\VirtualViewerJavaHTML5>_
```

jar -cvMf VirtualViewerJavaHTML5.war *

5. Install or deploy SnowboundVirtualViewer.war file.
6. Login to WebSphere Admin Console: <https://<your-websphere-server>:9043/ibm/console/logon.jsp>.
7. Select **Applications > Application Types > application**.
8. Select the **Install** button.
9. Browse to the location of VirtualViewerJavaHTML5.war and select it:

Specify the EAR, WAR, JAR, or SAR module to upload and install.



Path to the new application

Local file system

Full path

D:\SnowboundVirtualViewer Browse...

10. Select **Next**.
11. Select Fast Path, select **Next**.

Preparing for the application installation

How do you want to install the application?

Fast Path - Prompt only when additional information is required.

Detailed - Show all installation options and parameters.

Choose to generate default bindings and mappings

Previous Next Cancel

Step 1. No changes, press **Next**.

Step 2. Select the target Application Server and optionally webserver (CPE02, webserverIHS).

Check box next to module and press **Apply**.

Press **Next**.

Install New Application

Specify options for installing enterprise applications and modules.

Step 1 Select installation options

→ **Step 2: Map modules to servers**

✦ **Step 3** Map virtual hosts for Web modules

✦ **Step 4** Map context roots for Web modules

✦ **Step 5** Metadata for modules

Step 6 Summary

Map modules to servers

Specify targets such as application servers or clusters of application servers where you want to install modules. Modules can be installed on the same application server or dispersed among several application servers as routers for requests to this application. The plug-in configuration file (plugin-cfg.xml) for each server is routed through.

Display clusters and servers in the following security domain:
All security domains

Clusters and servers:

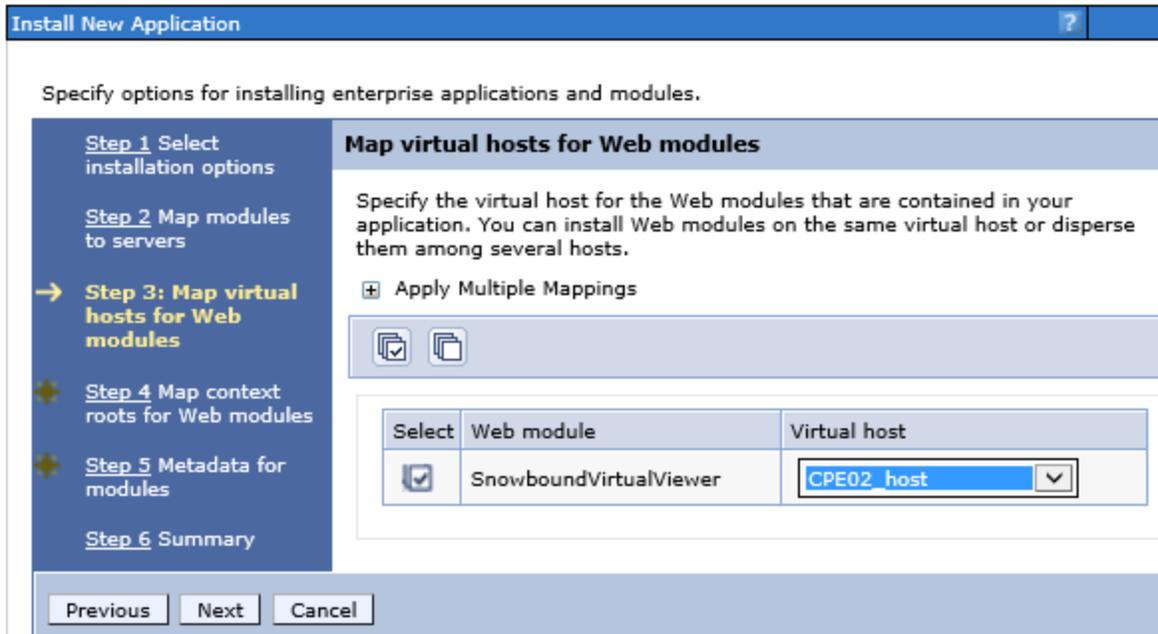
WebSphere:cell=w12r2ecmCell,node=w12r2ecmNode01,server=IFXADMIN
WebSphere:cell=w12r2ecmCell,node=w12r2ecmNode01,server=TranslatorServer
WebSphere:cell=w12r2ecmCell,node=w12r2ecmNode01,server=webserverIHS
WebSphere:cell=w12r2ecmCell,node=w12r2ecmNodeCPE01,server=CPE01
WebSphere:cell=w12r2ecmCell,node=w12r2ecmNodeCPE02,server=CPE02

Apply

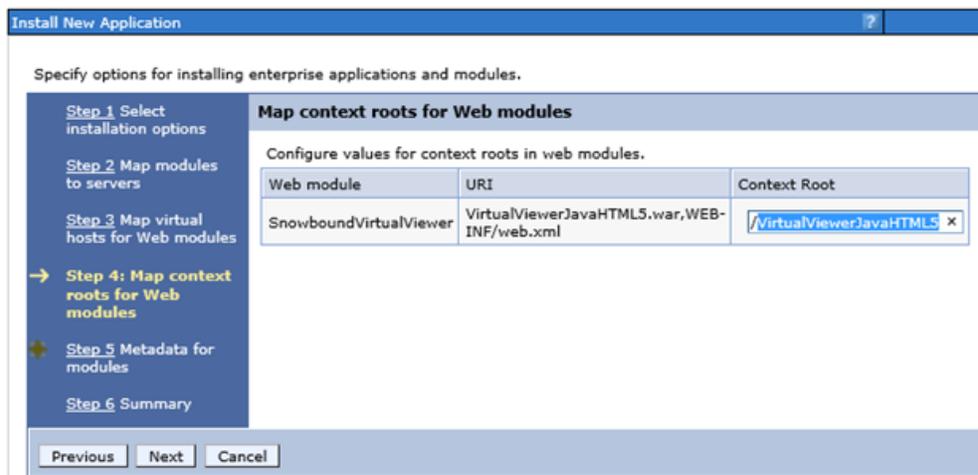
Select	Module	URI	Server
<input checked="" type="checkbox"/>	SnowboundVirtualViewer	VirtualViewerJavaHTML5.war,WEB-INF/web.xml	WebSphere:cell=w12r2ecmCell,node=w12r2ecmNodeCPE02,server=CPE02 WebSphere:cell=w12r2ecmCell,node=w12r2ecmNodeCPE01,server=CPE01

Previous Next Cancel

Step 3. Select correct web server host association.



Step 4. Select context: VirtualViewerJavaHTML5.



Step 5. Metadata for modules. No changes, press **Next**.

Step 6. Summary, No changes, press **Finish**.

Wait for deployment completion.

Save.

Select application checkbox, select **Start** button.

Chapter 4 - Installing the VirtualViewer® HTML5 ICN Connector

Follow the steps below to install and configure VirtualViewer® HTML5 ICN Connector:

Snowbound VirtualViewer IBM Content Navigator Plugin Installation

Follow the steps below to setup VirtualViewer plugin into Content Navigator:

1. Obtain the IBM Content Navigator (ICN) plugin file -- **SnowboundVirtualViewerPlugin.jar**.
2. We recommend that you back up the contents of the IBM Content Navigator directory before installing VirtualViewer® HTML5.
3. Copy the **pe.jar**, **jace.jar** and your **SnowboundLicense.jar** to the **VirtualViewerJavaHTML5/WEB-INF/lib** folder in the war file.

If running on a separate machine, you may have to copy the **commons.codec.1.x.jar** to the WEB-INF/lib folder. An example of the location for commons.codec.1.x.jar file could be in the navigator.ear in \navigator.ear\navigator.war\WEB-INF\lib\commons-codec-1.8.jar.

4. Login to **ICN Admin**.
`http://<your IBM Content Navigator Server>:<port>/navigator/?desktop=admin.`
5. Deploy the war file.
6. Start **VirtualViewerJavaHTML5.jar**.
7. Run
`http://<server>:8080/VirtualViewerJavaHTML5/encryptor.jsp.`
8. Encrypt the administrator **userid**.

Encrypt the **password**.

Edit the **VirtualViewerJavaHTML5.war\WEB-INF\classes\PyraSol.properties** file in the war file with the **userid** and **password**.

Make sure that the CEUri and PEConnectionPoint are correct.

For Example, edit the following text in the PyraSol.properties file:

```
#CE configsettings

#P8 settings
userid=<enter your userid>==
password=<enter your password>
JaasStanza=FileNetP8WSI
CEUri=http://localhost:9080/wsi/FNCEWS40MTOM/
PEConnectionPoint=cpeOS1
policyNoDocumentUpdate=false

#CM8 settings
CM8User=icmadmin
CM8Password=password
JDBCURL=jdbc:db2://192.168.1.14:50000/icmnlbdb

#CMOD settings
CMODAFP2PDFInstallDirectory=D:\\IBM\\OnDemand\\V9.5\\www
CMODLogDirectory=D:\\temp
CMODPort=1445
CMODTraceLevel=1
LIBRARYSERVER=w12r2ecm
```

9. Re-deploy the war file.
10. Login to **ICN Admin**.
11. Navigate to **Plugins > New Plug-in**.
12. Assign JAR file path. For Example:
D:\IBM\ECMClient\plugins\SnowboundVirtualViewerPlugin.jar
13. Press **Load**.
14. Scroll down.

Plug-in: Snowbound VirtualViewer Plugin

A plug-in can be either a JAR file or a compiled class file.

Important: The IBM Content Navigator web application server must be able to access the plug-in file on the local file

JAR file path:

Class file path:

Class name:

Name:	Snowbound VirtualViewer Plugin
Version:	2.0.3.113
Repository types:	None
Actions:	Open with VirtualViewer
Open Actions:	None
Viewers:	Snowbound VirtualViewer
Features:	None
Layouts:	None

* VirtualViewer URL:

* VirtualViewer servlet path:

15. Assign parameters:
VirtualViewer URL: /VirtualViewerJavaHTML5
VirtualViewer servlet path:/VirtualViewerJavaHTML5/AjaxServlet
16. Press **Save** and **Close**.
Note: If errors occur try delete the plugin and re-add it.
17. Logout of ICN Admin.

Chapter 5 - Configuring the VirtualViewer® HTML5 ICN Connector

Follow the steps below to configure VirtualViewer® HTML5 ICN Connector.

Once the application and plugins are installed, the System Administrator can configure VirtualViewer in the Content Navigator interface. First, VirtualViewer will need to be added as a viewer map. This tells Content Navigator which mime types the viewer will support. At that point, the System Administrator can decide which mime types will open in VirtualViewer and the relative order of viewer selection when multiple viewer options are available for the same mime type. For example, VirtualViewer can be made the default viewer for all mime types.

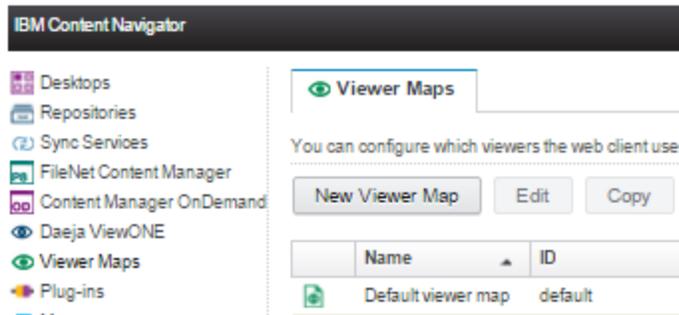
The second step is optional. The System Administrator can configure a specific “Open VirtualViewer” action when a user right-clicks a document.

Both of these configurations are described below.

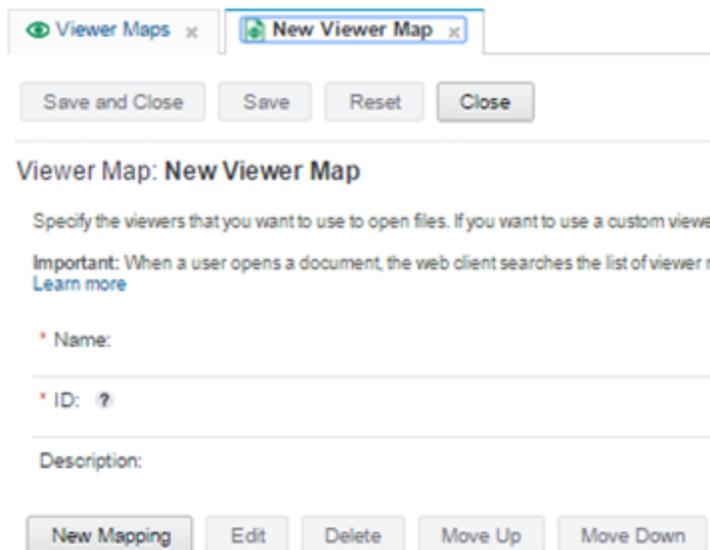
Define a Viewer Map

Follow the steps below to define a viewer map:

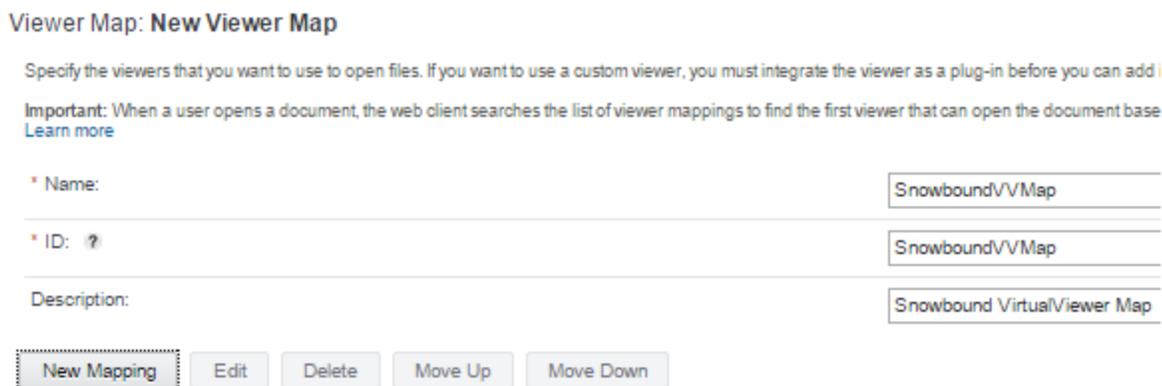
1. Login to **ICN Admin**.
<http://<your IBM Content Navigator Server>:<port>/navigator/?desktop=admin>
2. Navigate to **Viewer Maps**.



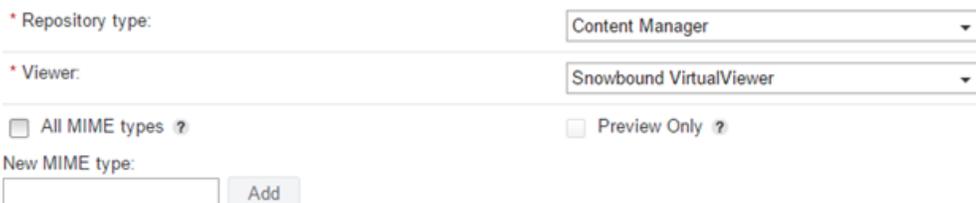
3. Select **New Viewer Map**.



4. Create new map.
 Name: SnowboundVVMMap
 ID: SnowboundVVMMap
 Desc: Snowbound VirtualViewer Map



5. Click **New Mapping**.



6. Select **Repository Type**.
 One of: .
 Content Manager
 Content Manager On-Demand

FileNet Content Manager
Content Management Interoperability Services (CMIS)

7. Select **Viewer**: VirtualViewer (Plugin registered viewer name).
Note: Plugin viewer definition will declare which MIME types are available/supported for the next step below.
8. Select **All MIME types**.
OR
Optionally add individual MIME types to the Select MIME Types list
9. Press **Okay**.
10. Scroll down the viewer map list.
11. Press the Move Up to make this viewer the first in the list.
12. Press **Save** and **Close**.
Note: The Viewer map will search top to bottom the above referenced list for the first supported MIME type for a given viewer definition.
13. Navigate to Desktops in the IBM Content Navigator Administration.
14. Select the target Desktop definition to assign the new viewer map.

▼ Desktop Configuration

* Viewer map: ?

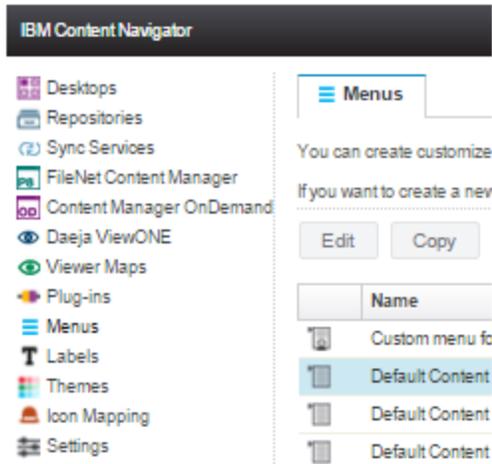
SnowboundVVMa|

15. In Desktop Configuration, choose the new Viewer map.
16. Press Save and Close.

Assign an Action Menu

Follow the steps below to assign an action menu:

1. Login to **ICN Admin**.
`http://<your IBM Content Navigator Server>:<port>/navigator/?desktop=admin`
2. Navigate to **Menus**.

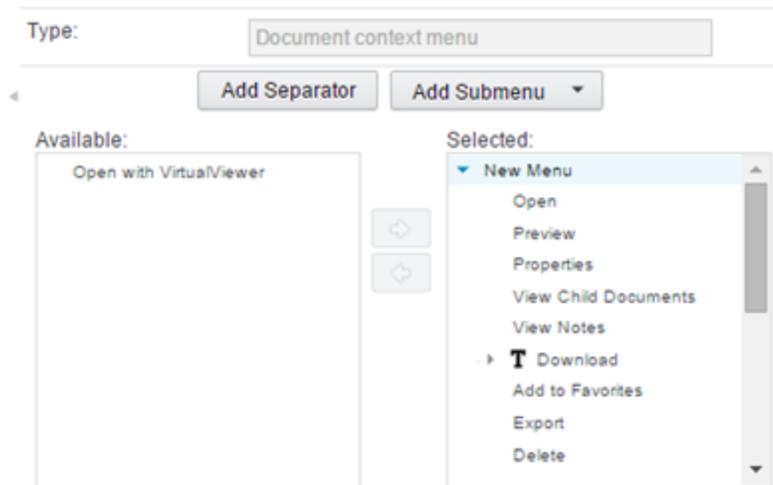


3. Select the Default document context menu.
4. Click **Copy**.
5. Rename the copy: Snowbound Document Context menu.

Context Menu: **New Menu**

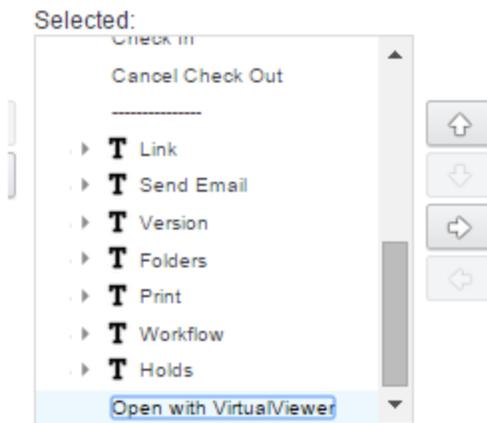
You can restrict the actions that are available for this context menu or you can add custom actions that are defined in a plug-in. You can also organize the actions by adding separators and submenus to the context menu.

* Name:	Snowbound Document Context Menu
* ID: ?	SnowboundDocumentContextMenu
Description:	Displayed when the user right-clicks a document
Type:	Document context menu

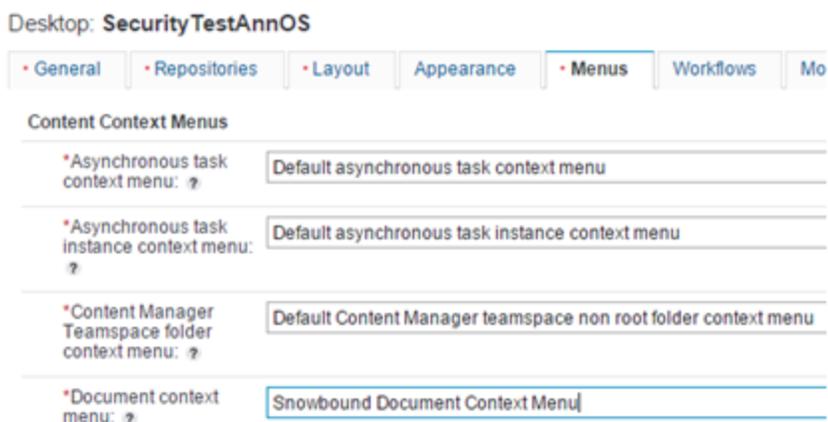


6. Add action –
“Open with VirtualViewer”.

7. Select newly added action and move up to desired location.



8. Save and Close.
9. Assign to desktop.



10. Save and Close.

Enable Pop-ups

If the browser does not show the new VirtualViewer you may need to enable pop-ups:



This icon means that the pop-up is blocked:

Click on the icon and select “Always allow pop-ups from site.”

Restart and Go!

Once all configuration steps have been completed, we highly recommend restarting the server running your web server to assure VirtualViewer® HTML5 functions correctly with IBM Content Navigator.

Once your server has been restarted, simply load IBM Content Navigator into a browser and double click a document to load VirtualViewer® HTML5.

Chapter 6 - Installing and Configuring the FileNet P8 Connector

The VirtualViewer Workplace and WorkplaceXT Connector contains the following important components:

VirtualViewerWorkPlace.zip (The zip file name may vary with version) - The zip file containing the files needed to install the product.

WebXMLDir - Contains web.xml tags that will need to be added to the Workplace or WorkplaceXT web.xml for VirtualViewer to work correctly with Workplace and WorkplaceXT.

WorkPlaceDir - Contains VirtualViewer and Workplace and WorkplaceXT connector files.

sample content_redir.properties - A sample version of Workplace's `content_redir.properties` modified to show how to launch certain document MIME types to be viewed in VirtualViewer.

WorkPlaceDir

The **WorkPlaceDir** directory contains the following for WorkplaceXT Connector:

- The **VirtualViewerHTML5** directory - Contains the files for VirtualViewer.
- The **WEB-INF** directory - Contains the **lib** and **resources** directories.

The **lib** directory contains **VVAJAXJavaServer.jar**, the standard VirtualViewer Content Server servlet code, and other various .jar files.

The **resources** directory contains images.

- **SnowboundHTML5Viewer.jsp** - Enables VirtualViewer client to be run with WorkplaceXT.
- **SnowboundHTML5Viewer.jsp.Workplace** - Enables VirtualViewer client to be run with Workplace.

WebXMLDir

The **WorkPlaceDir** directory contains the following files :

- **snowbound-servlet-mapping-web.xml.tags** - Contains the `<servlet-mapping>` tags that must be added to the **web.xml** in order to have VirtualViewer work with either WorkplaceXT or Workplace.
- **snowbound-servlet-web.xml.WorkplaceXT.tags** - Contains the `<servlet>` tags that must be added to the **web.xml** in order to have VirtualViewer work with

WorkplaceXT.

- **snowbound-servlet-web-xml.Workplace.tags** - Contains the `<servlet>` tags that must be added to the **web.xml** in order to have VirtualViewer work with Workplace.
- **web.xml** is a sample only and should only be used as a reference.

The FileNet P8 Connector Installation and Configuration

Your Workplace and WorkplaceXT install should have an `<Application Server Root>\<Workplace Application Root>` application directory. It is recommended that you make a backup copy of this directory structure before adding VirtualViewer.

1. Unzip the **VirtualViewerWorkPlace.zip** (The zip file name may vary with version) and place the contents of **WorkPlaceDir** into your **WorkplaceXT (or Workplace)** root directory. If you are using Workplace, you will have to switch the **config.js** file with **config.js.Workplace** and switch **SnowboundHTML5Viewer.jsp** with **SnowboundHTML5Viewer.jsp**. To make a switch, simply remove the current configuration files and then remove the Workplace tags from the desired configuration files.
2. From the zip file, take the contents of **WebXMLDir** and place them into your **working web.xml directory**.

Note:

Please note that the correct location of the **web.xml** is typically not the same directory as your root Workplace or WorkplaceXT directory. The following is an example of a common **web.xml** directory:

```
C:\Program Files\IBM\WebSphere\AppServer\profiles\AppSrv01\config\cells\filenet-p8Node01Cell\applications\Workplace.ear\deployments\Workplace\app_engine.war\WEB-INF
```

Once the contents of **WebXMLDir** have been placed into the correct **web.xml** directory, the existing **web.xml** will need to be modified as follows:

- a. Insert the contents of the **snowbound-servlet-web-xml.tags** into your existing **web.xml** immediately following the last `</servlet>` tag.
- b. Insert the contents of the **snowbound-servlet-mapping-web-xml.tags** into your existing **web.xml** immediately before the first `<servlet-mapping>` tag, to allow all of the Snowbound modifications to appear consecutively.
- c. Verify that the `servletURL` and `codebaseURL` parameters are using the appropriate server name (or localhost) and port number.
- d. Save **web.xml** and exit your editor.

3. Open the **content_redir.properties** file located at <FileNet Installation Folder>\Config\AE\content_redir.properties. This location is not within the web application itself.
4. You will need to edit the MIME types you wish to open in VirtualViewer. For example:

```
image/pjpeg=/SnowboundViewer.jsp?{JSP_QUERY_STRING}
image/jpg=/SnowboundViewer.jsp?{JSP_QUERY_STRING}
image/jpeg=/SnowboundViewer.jsp?{JSP_QUERY_STRING}
image/bmp=/SnowboundViewer.jsp?{JSP_QUERY_STRING}
image/tiff=/SnowboundViewer.jsp?{JSP_QUERY_STRING}
image/gif=/SnowboundViewer.jsp?{JSP_QUERY_STRING}
image/png=/SnowboundViewer.jsp?{JSP_QUERY_STRING}

application/msword=/SnowboundHTML5Viewer.jsp?{JSP_QUERY_STRING}
application/postscript=/ SnowboundHTML5Viewer.jsp?{JSP_QUERY_STRING}
application/pdf=/ SnowboundHTML5Viewer.jsp?{JSP_QUERY_STRING}
application/vnd.ms-excel=/SnowboundHTML5Viewer.jsp?{JSP_QUERY_STRING}
```

It is possible that you will have other modifications to the default file that need to be preserved. It is also possible that you will not want to have each MIME type defined above to be viewed in VirtualViewer. Appropriate edits will have to be made in this case. It is recommended that any installation procedure includes a backup of the original file.

See **sample content_redir.properties** included in the package for a more detailed example.

5. Restart your web server.
6. To launch the VirtualViewer client, log into Workplace and choose to view a file with a MIME type that you have configured to use VirtualViewer. The VirtualViewer client should appear with the chosen file.

Appendix A - Troubleshooting

This section explains solutions to issues that may occur when installing the VirtualViewer Java AJAX FileNet P8 Connector.

Image Does Not Load in a Single Sign-On (SSO) Environment

An issue may occur where an image will not load in the viewer. If this occurs, check the SystemErr.log file. If you see the following error message `Anonymous users aren't allowed access to the Content Engine` and you have a Single Sign-On (SSO) environment, then add the following URLs to the `<security-constraint>` section of the web.xml file.

```
<url-pattern>/RequestServer/*</url-pattern>
<url-pattern>/ResponseServer/*</url-pattern>
<url-pattern>/AjaxServlet/*</url-pattern>
<url-pattern>/snowboundAjaxRedirect.jsp</url-pattern>
<url-pattern>/ajaxClient.jsp</url-pattern>
<url-pattern>/SnowboundViewer.jsp</url-pattern>
```

Websphere Will Not Start

If Websphere will not start, check your web.xml file for syntax errors. Make sure that all parameters and the XML syntax are valid. This issue may occur in all environments and not just in a Single Sign-on (SSO) environment.

404 errors Occur After Updating the Configuration Settings in config.js

If you see 404 errors when trying to access VirtualViewer AJAX through FileNet Connector, especially after updating the configuration settings in config.js, make sure an older version of the configuration settings in config.js is not in your browser's cache on the client machine.

Enter the following URL to access the config.js file directly and ensure you are seeing the most up to date version, substituting in your **server** and **port number** where the text is bold:

```
http://server:port/WorkplaceXT/config.js
```

If you see the contents of the old config.js, press refresh in the browser (ctrl+F5) to load and use the most recent config.js.

Windows 404 Server or File Not Found Error Occurs When Referencing Wrong web.xml

This issue occurs when you receive a Windows 404 - server or file not found error and a “Please Wait While Your File Loads” message and the viewer loads with no image when integrating with FileNet on a Windows server.

To solve this issue, ensure that the correct web.xml is referenced. Ensure that the proper name of the directory is referenced in the config.js, the web.xml, and the JSP file. For example if WorkplaceXT is the proper name of the directory, make sure that **WorkplaceXT** instead of **Workplace** is referenced in those mentioned configuration files.

Please note that there may be multiple web.xml files present on the system. It is important to choose the correct one. The web.xml being referenced may be located in another directory as shown in the following example:

```
C:\Program Files\IBM\WebSphere\  
AppServer\profiles\AppSrv01\config\  
cells\filenet-p8Node01Cell\applications\  
Workplace.ear\deployments\  
Workplace\app_engine.war\  
WEB-INF
```

Index

4

404 error 30-31

A

annotation 7

C

CMB jars 14

CMIS 6

config.js 30

configuration settings 30

Content Manager 6

D

DKAccessControlListICM 10

E

environment

 Single Sign-on 30

F

file not found error 31

FileNet Content Manager (P8) 6, 13

FilenetElements.java 9

G

getDocumentContent 9

getFilenetDocumentByteArray 9

I

ICN installation 5

image

 does not load 30

INI files 14

installing

 VirtualViewer 12

J

Jace.jar 13

M

ManageDbConnection.java 9

MANIFEST.M 13

N

NavigatorContentHandler.java 9

P

pe.jar 13

PyramidICNHandler.jar 13

PyraSol 12

R

repository 6

Websphere

does not start 30

S

saveDocumentComponents 9

server error 31

servlet container 5

Single Sign-on environment 30

Snowbound-servlet-mapping-web-
xml.tags 27

snowbound-servlet-web-xml.tags 27-
28

SSO 30

system requirements 5

 ICN installation 5

 servlet container 5

T

t IBM Content Navigator 6

V

VirtualViewerJavaHTML5-4.1.x.zip 12

W

web.xml 30

web.xml file

 referencing 31