



CISCO CONFIDENTIAL

CHAPTER 22

Using the Java Plug-in

Java Plug-in (JPI) technology is a Sun Microsystems product that allows Java2 applets to run in web browsers.

All applets have been removed beginning with this version of CWCS. CiscoWorks Common Services does not require or depend on a browser VM or JPI technology to provide its services. However, the JPI is included in the shipping version of CWCS so that applications who do depend on the JPI can deploy their applets. CWCS includes the extractable EXE file for Windows, and a compressed TAR file and customized installation script (pam.sh) for Solaris. The software packages are CSCOPlug on Solaris and plug on Windows; at runtime, the JPI installables are located in *NMSROOT*/htdocs/plugin.

Applications can evaluate whether to use the Java Plug-in depending on their needs. The CWCS JPI team recommends that developers:

1. Determine if your application's features really require the JPI.
2. If your application does need it, use only the JPI version supplied with CWCS, and ensure your application works with that version of the JPI.
3. Do not hard-code the Plug-in version in your JSPs. Instead, pick the JPI version dynamically, as explained in the [“Using the Java Plug-in API”](#) section on page 22-2.

The following topics describe the Java Plug-in support in CWCS and how to use it with your application:

- [About the Java Plug-in Requirements](#)
- [Using the Java Plug-in API](#)
- [Accessing the JPI Configuration from CCR](#)
- [Using Tags Java Plug-in](#)
- [Using Client Local Resources](#)
- [JPI Technology References](#)

About the Java Plug-in Requirements

The Java Plug-in requires specific operating systems, browsers, and patches to run successfully. Both Solaris and Windows client machines need to have approximately 12 MB disk space for downloading and installing the Plug-in. On Windows, the C: drive needs to have approximately 10 MB.



Caution

It is recommended to uninstall any other versions of the Java Plug-in from the client machine before installing the CWCS- supported version.

CISCO CONFIDENTIAL**Table 22-1 Java Plug-in Requirements**

Client OS/Browsers	JPI Version	System Requirements
Solaris 2.8/Netscape 7.0, Mozilla 1.7.13	1.4.2_10	To obtain patches, see the SunSolve support website . You will find patch clusters for Solaris operating system platforms J2SE Solaris 9 and J2SE Solaris 8.
Solaris 2.9/Netscape 7.0, Mozilla 1.7.13		
Windows 2000/Netscape 7.1/7.2, Microsoft Internet Explorer 6.0 SP2, Mozilla 1.7.13	1.4.2_10	No special requirements

Using the Java Plug-in API

The Java plug-in API is available to all applications. This API dynamically substitutes the values of the plug-in versions used by CWCS clients for Windows and Solaris platforms.

You can use the API to retrieve the Plug-in values by sending the operating system name, as follows:

- Windows: WIN-IE, WIN-NS
- Solaris: SOL

For Windows alone, WIN-IE returns the class ID for CWCS Internet Explorer clients and WIN-NS returns the version to be used by CWCS Netscape clients.

For example:

1. In a JSP file, include `com.cisco.nm.cmf.ssl.GetPluginVersion`.
2. `%=GetPluginVersion.getPluginVersion("WIN-IE")%` can be used in places where the class ID for the Java Plug-in is hard-coded.
3. `%=GetPluginVersion.getPluginVersion("WIN-NS")%` can be used in places where the type of EMBED tag for Java Plug-in is hard-coded.
4. `%=GetPluginVersion.getPluginVersion("SOL")%` can be used in places where the type of EMBED tag for Java Plug-in is hard-coded.
5. `%=GetPluginVersion.getPluginVersion("IE_PLUGINS PAGE")%` points to the URL from which the Plug-in can be downloaded.

You can view these values in the file `NMSROOT/lib/classpath/javaplugin.properties`.

Accessing the JPI Configuration from CCR

You can retrieve the values of the JPI configuration parameters from the Core Client Registry. CCR will contain a separate entry for each configuration parameter, as follows:

```
MDC (Application) Name: Core
Resource Type: Custom
Resource Name: WIN_IE_VERSION

MDC (Application) Name: Core
Resource Type: Custom
Resource Name: WIN_NS_VERSION
```

CISCO CONFIDENTIAL

```
MDC (Application) Name: Core
Resource Type: Custom
Resource Name: SOL_VERSION
```

```
MDC (Application) Name: Core
Resource Type: Custom
Resource Name: JavaPluginExe
```

The CCR resources WIN_IE_VERSION, WIN_NS_VERSION, SOL_VERSION, and JavapluginExe correspond to the WIN_IE, WIN_NS,SOL, and IE_PLUGINSPAGE parameters discussed in the “Using the Java Plug-in API” section on page 22-2.

To retrieve a JPI configuration parameter from CCR, use code like the following:

```
CCREntry ccrentry = new CCREntry("Core", "Custom", "Custom", "", "", "WIN_IE_VERSION");
CCRResponse ccrresponse = ccrinterface.retrieveEntry(ccrentry);
ccrentry = (CCREntry)ccrresponse.getReturnedValues().elementAt(0);
String resourceData = ccrentry.getResourceData();
```

You can query WIN_NS_VERSION, SOL_VERSION, and JavaPluginExe in the same way.

Using Tags Java Plug-in

To use the Java Plug-in, you must change your HTML pages to use Sun's JRE via the Java Plug-in software. Sun provides an [HTML converter tool](#) with the Sun JPI SDK. This tool automatically makes the necessary changes to the HTML web pages. For more information about the HTML converter tool, see http://java.sun.com/j2se/1.4.2/docs/guide/plugin/developer_guide/html_converter_more.html.

The tagging structure that the JPI requires, including the correct OBJECT and EMBED tags, is explained in the Sun Java Developer Guide chapter “Using OBJECT, EMBED and APPLET Tags in Java Plug-in” (see http://java.sun.com/j2se/1.4.2/docs/guide/plugin/developer_guide/using_tags.html).

Developers must ensure that the correct HTML tags have been used for each browser (EMBED for Netscape and OBJECT tag for Microsoft Internet Explorer). Otherwise, the Java Plug-in is not used and the applet is started using the browser's default Java Virtual Machine. After downloading and installing the Java Plug-in software from the CWCS Server, the applets would continue to execute under the Java Plug-in for any pages that are converted to use the Java Plug-in.

Using Client Local Resources

To allow each client to use its local resources (such as printing and cut/copy/paste), you must use the clientservices13.jar file, as follows:

- The clientservices13.jar file is located in *NMSROOT*/www/classpath. Applications need to download this version of the clientservices13.jar with the applets that need access to local resources.
- Applications may need to sign the application JAR file. Contact the Release Engineering team for details on doing this.

CISCO CONFIDENTIAL

JPI Technology References

The following external references may assist your understanding of Sun's implementation of the Java Plug-in.

- Sun main plug-in page: <http://java.sun.com/products/plugin/>
- Technical documentation: <http://java.sun.com/j2se/1.4.2/docs/guide/plugin/index.html>
- Applet caching:
http://java.sun.com/j2se/1.4.2/docs/guide/plugin/developer_guide/applet_caching.html
- Standard extension versioning: <http://java.sun.com/products/plugin/versions.html>
- Related jar indexing:
http://java.sun.com/j2se/1.4.2/docs/guide/plugin/developer_guide/jar_indexing.html