Oracle JInitiator

Technical FAQ
PM-CP-004
April 1999

ORACLE®
Enabling the Information Age™

Oracle JInitiator

PM-CP-004

CERTIFICATION AND AVAILABILITY

When will Oracle JInitiator be available?

Oracle JInitiator has been available since September 1998 for the deployment of custom Oracle Developer applications. Oracle Applications completed certification for Oracle JInitiator in February 1999.

How is Oracle JInitiator distributed?

Starting with release 6.0 of Oracle Developer, Oracle JInitiator will be shipped as part of the Oracle Developer distribution CD. Oracle JInitiator is also available for download from the Oracle Developer section of the Oracle website: http://www.oracle.com/tools/dev_server/dn.html. Updates for Oracle JInitiator may also be obtained through the Oracle Worldwide Support Organization.

Will Oracle JInitiator work on non Windows platforms?

Oracle has no current plans to port Oracle JInitiator to non Microsoft Windows platforms, however we are working very closely with Oracle Product Lines and a number of hardware vendors to provide support and certification for deploying Oracle Developer applications on non Microsoft Windows client platforms. For more information refer to PM-CP-001, *Oracle Developer Server: Client Platform Support Statement of Direction*. This is available from the Oracle Technology Network http://technet.oracle.com.

What versions of Netscape Navigator and Internet Explorer is Oracle JInitiator certified with?

Oracle JInitiator will be certified with the latest production releases of the respective browsers at the time that that each Oracle JInitiator release undergoes final QA testing. Oracle will also be providing support for earlier releases of the browsers. The exact browser versions that have been certified will be contained in the accompanying documentation for an Oracle JInitiator release.

What is the difference between the JavaSoft Java Plug-in and Oracle JInitiator?

The primary difference is that Oracle JInitiator includes the Oracle certified JRE whereas the JavaSoft Java Plug-In is shipped with a JavaSoft JDK reference implementation which has not been certified with Oracle Developer Server applications.

The JavaSoft Plug-in is simply a delivery mechanism for a JavaSoft JRE which can be launched from within a browser. Likewise, Oracle JInitiator is simply providing a delivery mechanism for an Oracle certified JRE which enables Oracle Developer applications to be run from within a browser in a stable and supported manner.

Why is Oracle certifying and delivering a specific JRE rather than using the JRE provided by JavaSoft?

Oracle Corporation is committed to a single, industry standard Java implementation, however the need to ensure Oracle Applications can respond to competitive pressures means the Oracle Developer Server team may need to add features or address critical issues in the platform at any time. This level of control is required to deliver a fully supported, certified deployment environment that meets the release criteria, time-frames, sustained engineering requirements and support requirements for Oracle Applications.

In addition to addressing critical bug fixes, Oracle JInitiator provides a number of additional features to enhance application performance. These features such as JAR file caching, incremental JAR file loading and applet caching enable enterprises to make best use of their available hardware and network assets. The goal is always to make changes in conjunction with JavaSoft agreement, but there will always be the potential for latency in feature/bug fix uptake.

Can the JavaSoft Java Plug-In be used to run Oracle Developer Server applications?

Recent testing efforts on platforms with a 1.1.7B-based JVM have been extremely positive. However, with two Win32 platforms already certified (specifically Silver Certification with Oracle JInitiator and Bronze Certification with native Microsoft Internet Explorer 5.0) Certification resources are focussed on expanding the number of non-Windows client platforms that are certified. As additional platforms become certified, the Java Plug-in will become a candidate for certification testing. Until this time the JavaSoft Java Plug-in is not certified for use with Oracle Developer Server applications.

Can native browsers be used to run Oracle Developer Server applications?

Recent advancements made in the Java platform, specifically with the 1.1.7B-based JVM and the tiered certification model introduced with Oracle Developer Server 6.0 have enabled Bronze Certification for native Microsoft Internet Explorer 5.0 support on Win32 platforms.

The Plug-in model adopted by Netscape is in line with the Java Plug-in strategy adopted across the industry and with Oracle's Oracle JInitiator.

SUPPORT

Who will provide support for Oracle JInitiator?

Oracle Corporation provides full support for Oracle JInitiator through the Oracle Worldwide Support Organization.

Which versions of Oracle Developer Server does Oracle JInitiator support?

Oracle JInitiator supports Oracle Developer Server Release 1.6.1 and later.

Is Oracle JInitiator supported with Oracle Applications?

Yes. Oracle JInitiator is the only client platform on the market today that has met the stringent certification tests to pass Gold Certification that is required to support Oracle Applications. Oracle JInitiator has been certified with Oracle Applications within Netscape Navigator 4.06 and later and Microsoft Internet Explorer 4.0 and later.

INSTALLATION

What do I need to install on the client in order to run Oracle Developer Server applications in a Web browser?

By leveraging the standard browser extension mechanisms provided by Netscape Navigator and Microsoft Internet Explorer, Oracle JInitiator is able to automatically download itself to the client machine when the browser first encounters a HTML page that requires the use of it. The installation is then accomplished using the Plug-in/ActiveX model supported by the browser.

How large is Oracle JInitiator when it is downloaded to the client?

The compressed Oracle JInitiator distribution is approximately 8MB in size and expands to approximately 10MB in size once installed on the client.

Is it possible to install Oracle JInitiator without end user interaction?

Oracle JInitiator supports a silent installation mode which doesn't require end user interaction. To perform the silent installation the Oracle JInitiator distribution must be run with the following parameters: "-s -sm".

For example to perform a silent installation from the command line, the user would open a DOS shell and type:

```
C:\TEMP> jinit1179 -s -sm
```

To perform a silent installation using the Windows Run dialog, the user would click Start ➔ Run and then enter `jinit1179 -s -sm` in the Run dialog window that appears:

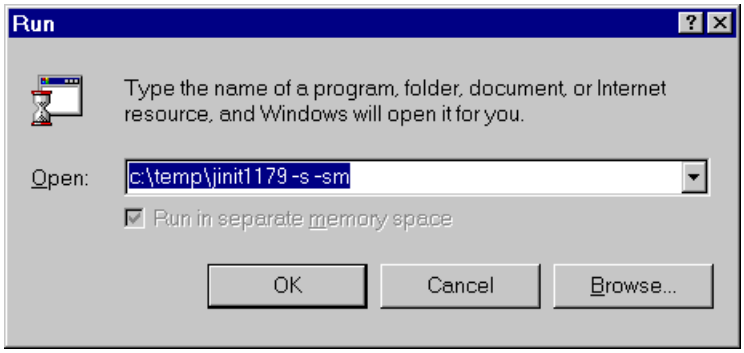

Figure 1: performing a silent installation via the Run dialog

Is it possible to perform the Oracle JInitiator installation from a central server such that user interaction is not required at all?

Using the facilities provided by the host operating systems, it is possible to install Oracle JInitiator on each client desktop. This in essence involves the System Administrator being able to access each client

machine and running the silent, non GUI installation option of Oracle JInitiator to perform the installation.

Can I force Oracle JInitiator to use the same configurations for Proxy Servers, etc. as the browser in which it is running?

The operation of Oracle JInitiator is controlled via the Oracle JInitiator Control Panel. The Oracle JInitiator Control Panel is installed at the same time Oracle JInitiator is installed and can be accessed from the Start ➔ Programs menu.

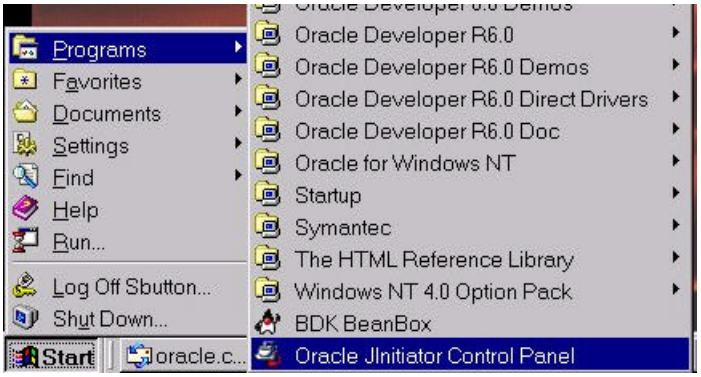

Figure 2: accessing the Oracle JInitiator Control Panel

Using the Oracle JInitiator Control Panel, Oracle JInitiator can be configured to use either its own specific Proxy settings or the default Proxy settings supplied by the browser from which it is invoked. This setting can be found in the Proxies tab of the Oracle JInitiator Control Panel.

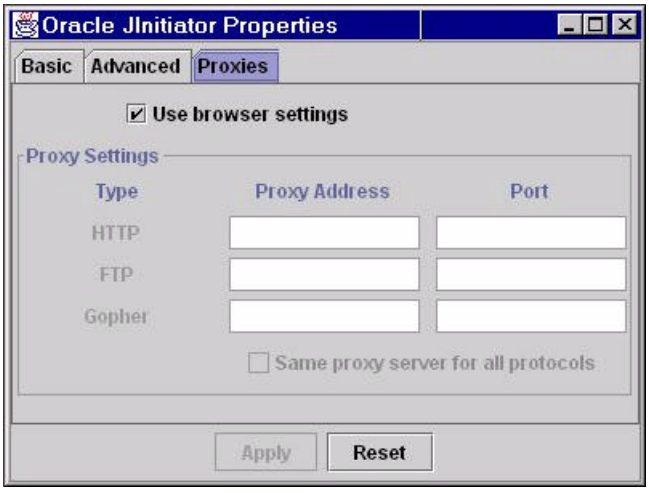

Figure 3: configuring Oracle JInitiator Proxy settings

How can I force my browser clients to download and install a new version of the Oracle JInitiator?

Oracle JInitiator functions as a Plug-in in Netscape Navigator or an ActiveX object in Microsoft Internet Explorer. The browser uses a MIME type to provide a mapping between an HTML page request and the required Plug-in/ActiveX object. Each Oracle JInitiator installation has a specific MIME type associated with it. When a browser loads a HTML page that contains a MIME type that it is not aware of, the browser informs the user that it does not have the required Plug-in/ActiveX object and will open a dialog to ask the user if they wish to retrieve it.

By changing the MIME type specified in the HTML page for your application to a later version, the browser will detect that it does not have a valid Plug-in/ActiveX object for that MIME type and will notify the user that they need to download a newer version of Oracle JInitiator.

For example:

An HTML page HR.HTML allows users to run the HR application. The HR.HTML page indicates to the required version of Oracle JInitiator (1.1.5.21.1) through the MIME type value:

```
<OBJECT classid="clsid:9F77a997-F0F3-11d1-9195-00C04FC990DC" WIDTH=180
HEIGHT=20 codebase="http://mymachine/jinit115211.exe#Version=1,1,5,21,1">
<PARAM NAME="CODE"     VALUE="oracle.forms.uiClient.v1_4.engine.Main" >
<PARAM NAME="CODEBASE" VALUE="/form50code/" >
<PARAM NAME="ARCHIVE"  VALUE="/form50code/f50all.jar" >
<PARAM NAME="type"     VALUE="application/x-jinit-applet;version=1.1.5.21.1">
<PARAM NAME="serverPort" VALUE="9000">
<PARAM NAME="serverArgs"  VALUE="module=fmx  userid=u/p@datasource">
<PARAM NAME="serverApp"  VALUE="default">

<EMBED type="application/x-jinit-applet;version=1.1.5.21.1"
       java_CODE="oracle.forms.uiClient.v1_4.engine.Main"
       java_CODEBASE="/form50code/"
       java_ARCHIVE="/form50code/f50all.jar"
       WIDTH=180
       HEIGHT=20
       serverPort="9000"
       serverArgs="module=fmx_name  userid=u/p@datasource"
       serverApp="default"
       pluginspage="http://mymachine/jinit_download.htm">
</EMBED>
</OBJECT>
```

Figure 4: HTML page using Oracle JInitiator 1.1.5.21.1

If a later release of Oracle JInitiator is obtained and placed on the server, the client browser can be forced to use the newer version by updating the version specific lines in the HR.HTML.

```
<OBJECT classid="clsid:9F77a997-F0F3-11d1-9195-00C04FC990DC" WIDTH=180
HEIGHT=20 codebase="http://mymachine/jinit1179.exe#Version=1,1,7,9">
<PARAM NAME="CODE"     VALUE="oracle.forms.uiClient.v1_4.engine.Main" >
<PARAM NAME="CODEBASE" VALUE="/form50code/" >
<PARAM NAME="ARCHIVE"  VALUE="/form50code/f50all.jar" >
<PARAM NAME="type"     VALUE="application/x-jinit-applet;version=1.1.7.9">
<PARAM NAME="serverPort" VALUE="9000">
<PARAM NAME="serverArgs"  VALUE="module=fmx  userid=u/p@datasource">
<PARAM NAME="serverApp"  VALUE="default">

<EMBED type="application/x-jinit-applet;version=1.1.7.9"
       java_CODE="oracle.forms.uiClient.v1_4.engine.Main"
       java_CODEBASE="/form50code/"
       java_ARCHIVE="/form50code/f50all.jar"
       WIDTH=180
```

```
       HEIGHT=20
       serverPort="9000"
       serverArgs="module=fmx_name  userid=u/p@datasource"
       serverApp="default"
       pluginspage="http://mymachine/jinit_download.htm">
</EMBED>
</OBJECT>
```

Figure 5: upgraded HTML page using Oracle JInitiator 1.1.7.9

When the browser next requests the page, the user will be prompted with the a dialog box, indicating that a new Oracle JInitiator release can be installed. To install the newer release, the user performs the same steps required to install the initial release. Following is an example of this dialog box from Netscape Navigator.

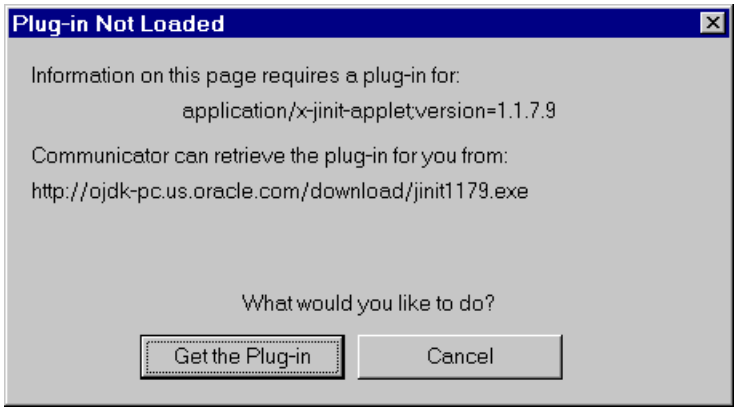

Figure 6: browser Plug-in download dialog

I pressed the cancel button on the Netscape 'Plug-in Not Loaded' dialog and now I never get prompted to install Oracle JInitiator. How do I install the Plug-in?

Netscape uses the Windows registry to store information about the Plug-ins it has installed. As soon as the 'Plug-in Not Loaded' dialog appears, Netscape writes the details for the Plug-in into the registry, irrespective of whether the Plug-in is actually installed or not. When a page is encountered that calls for the use of that specific Plug-in, Netscape will find that is had the Plug-in installed via the registry information. This results in the 'Plug-in Not Loaded' dialog box not being shown again.

To overcome this, you can force Netscape to load a Plug-in by clicking on the Plug-in missing icon as shown in Figure 7. This will result in Netscape redisplaying the Plug-in download dialog.

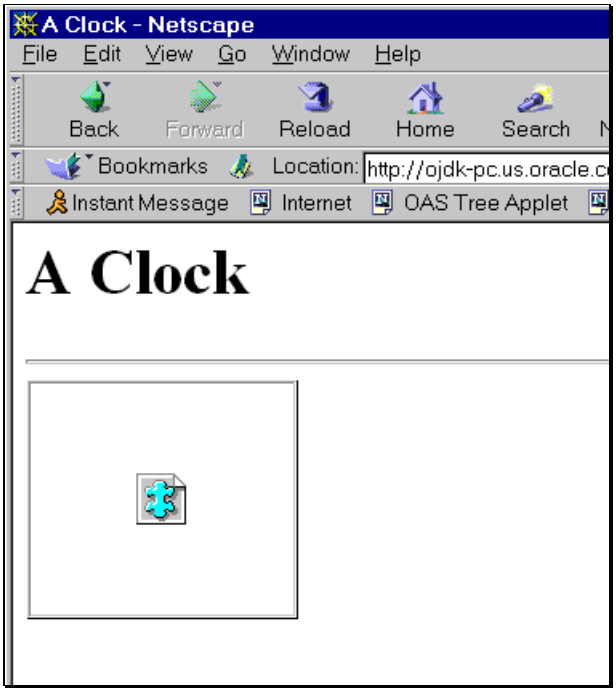

Figure 7: Plug-in not found Icon

I have a lot of HTML pages that have different MIME types in them. Will the latest Oracle JInitiator release still run with these earlier MIME types?

Currently the Netscape browser has limit of 256 characters that may be used to store the recognized MIME types for a particular Plug-in. Microsoft IE does not have this restriction with their extensible browser Objects architecture. Working within this limit, Oracle JInitiator will provide backwards support for as many earlier MIME types as is possible.

The accompanying documentation and release notes for an Oracle JInitiator release will provide an accurate description of what MIME types are supported for a specific release.

Is it possible to make Oracle Developer applications run in any version of Oracle JInitiator?

A generic MIME type is provided that will allow any installed version of Oracle JInitiator to run the Oracle Developer application. This MIME type `application/x-jinit-applet` is recognized by every version of Oracle JInitiator. Note that always using this MIME type will not enforce the upgrading of later Oracle JInitiator versions since this MIME type is supported in every release.

OPERATION OF ORACLE JINITIATOR

Can the Forms Applet window be run within the same browser window from which it was launched?

Yes. Using Oracle Developer 6.0 the Forms applet may be run either within the same browser window or in a separate window. This is a configurable option. It is set as a parameter in the HTML file. Please see the Oracle Developer 6.0 release notes for more information.

What happens to the running Oracle Developer application if the user navigates off of the current browser page?

Oracle JInitiator allows a running Java application to be cached. This feature allows a user to run a Forms application, navigate to a different page and return to the Forms application page. The Forms application will appear as exactly as it was when the user left it.

Can I use the Oracle JInitiator to run my custom developed Java applications?

Yes but this is not supported by Oracle. Oracle JInitiator uses a standard JavaSoft JVM. It differs only in that certain bugs identified by the Oracle development team have been fixed. Therefore it should be capable of running customer Java applications. However at this time, Oracle only provides support for Oracle JInitiator when running Oracle Java based applications such as Oracle Developer, Oracle Enterprise Manager and Oracle Discoverer. The use of Oracle JInitiator to run custom Java applications is not supported by Oracle.

Can Oracle JInitiator and the JavaSoft Java Plug-In coexist on the same machine?

Yes. Oracle JInitiator and the JavaSoft Java Plug-in can be installed and coexist in the same browser installation since they each use different MIME types to launch the plug-in.

Will Oracle JInitiator coexist and operate correctly when used at the same time as the JavaSoft Plug-in, in the same browser instance?

Due to the way that dynamically loadable libraries are loaded and the way that the JVM dynamically loadable libraries are named, the Oracle JRE and the JavaSoft JRE can not be run simultaneously from within the same browser instance. This means that a browser user can not switch from using the JavaSoft Java Plug-in to Oracle JInitiator in the same browser instance. The browser must be stopped and restarted when switching between the different applications that use Oracle JInitiator and Java Plug-in from JavaSoft.

With the JavaSoft Java Plug In and Oracle JInitiator there is an option to use a different JRE. Can I use the JavaSoft Java Plug In when it is configured to use the Oracle certified JRE to run Oracle Developer applications?

No. The only certified and supported combination will be using the Oracle JInitiator with the supplied Oracle JRE. The Oracle JRE, while conforming to the JavaSoft standard, contains bug fixes to the JavaSoft JRE that allows Oracle Developer applications to run correctly. Oracle works closely with JavaSoft to ensure that these bug fixes are communicated to JavaSoft and applied to the standard JRE, but is unable to wait for the fixed JavaSoft JRE to be released.

The figure below shows the Oracle JInitiator Control Panel and the correct settings for the Java Run Time Environment value.

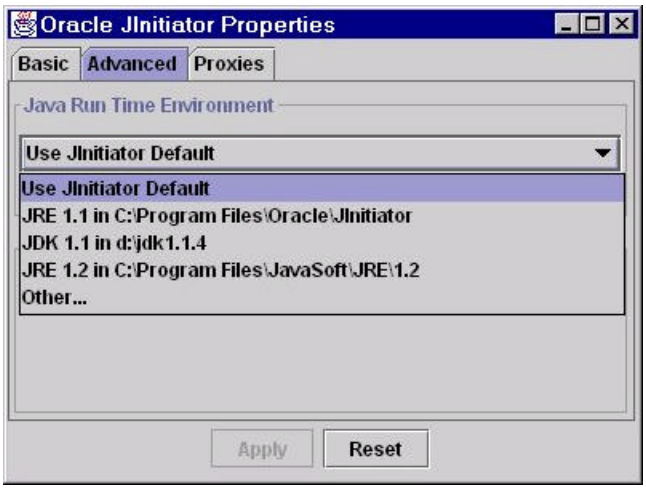

Figure 8: Setting the Oracle JInitiator JRE

CACHING

Does Oracle JInitiator cache the Java class files downloaded when an application is run? If so, does this feature therefore mean that the Java class files would have to be downloaded only once and not each time the application is started?

Oracle JInitiator provides a persistent caching mechanism for JAR files that it downloads when running Java applications. A JAR file is a standard Java archive that contains a set of Java class files that are used by a Java application. By putting all the required class files into a single JAR file, a single download of the JAR file is performed rather than multiple downloads for each of the individual class files required.

By caching the JAR files on the client, Oracle JInitiator alleviates the need to download JAR files each time they are required for an application. The first time a JAR file is required it is downloaded from the Web server and then saved to the local client machine. The next time it is required, Oracle JInitiator will load the JAR file directly from the cache directory on the client machine.

This saves a lot of end user time and network traffic. For example, if an application requires a 2MB JAR file there were a fast Ethernet connection which was capable of downloading a 2MB file in 5 seconds then each user would save 5 seconds at application startup. However if a dial-up network is used which took 10 minutes to download a 2MB file, then each user would save 10 minutes at application startup.

How does Oracle JInitiator caching technology work?

Oracle JInitiator provides browser session independent caching of JAR files. Oracle JInitiator stores the downloaded JAR files on the local client machine so that it does not need to download them when next they are required.

When a JAR file is requested, Oracle JInitiator will check the cache directory to determine if the file has been previously requested, downloaded and subsequently stored. If the JAR file is not present, Oracle JInitiator will download the JAR file from the Web server and then store it for future use in the

cache. Some additional information is stored in the cache file. This information enables Oracle JInitiator to uniquely identify the JAR file and to determine the Last-Modified date of the requested file as reported by the Web server.

If a required file is present in the cache, then Web server is queried to determine if the stored JAR file is current. Oracle JInitiator takes the Last-Modified date contained in the cached JAR file and asks the Web server (using standard HTTP interactions) if the file on the server has subsequently been modified. The Web server compares the Last-Modified date and the timestamp of the file stored on the server. If the server file is newer, it is sent to Oracle JInitiator with a status code of 200. Otherwise it returns a status code of 304 which indicates that the file in the cache is current.

If the cached JAR file is not current, a new one is downloaded and stored for future use in the cache directory. If the file is current, Oracle JInitiator loads it from the cache directory and updates the timestamp on the cached file to indicate the last time it was used.

The Oracle JInitiator cache loading process is depicted below.

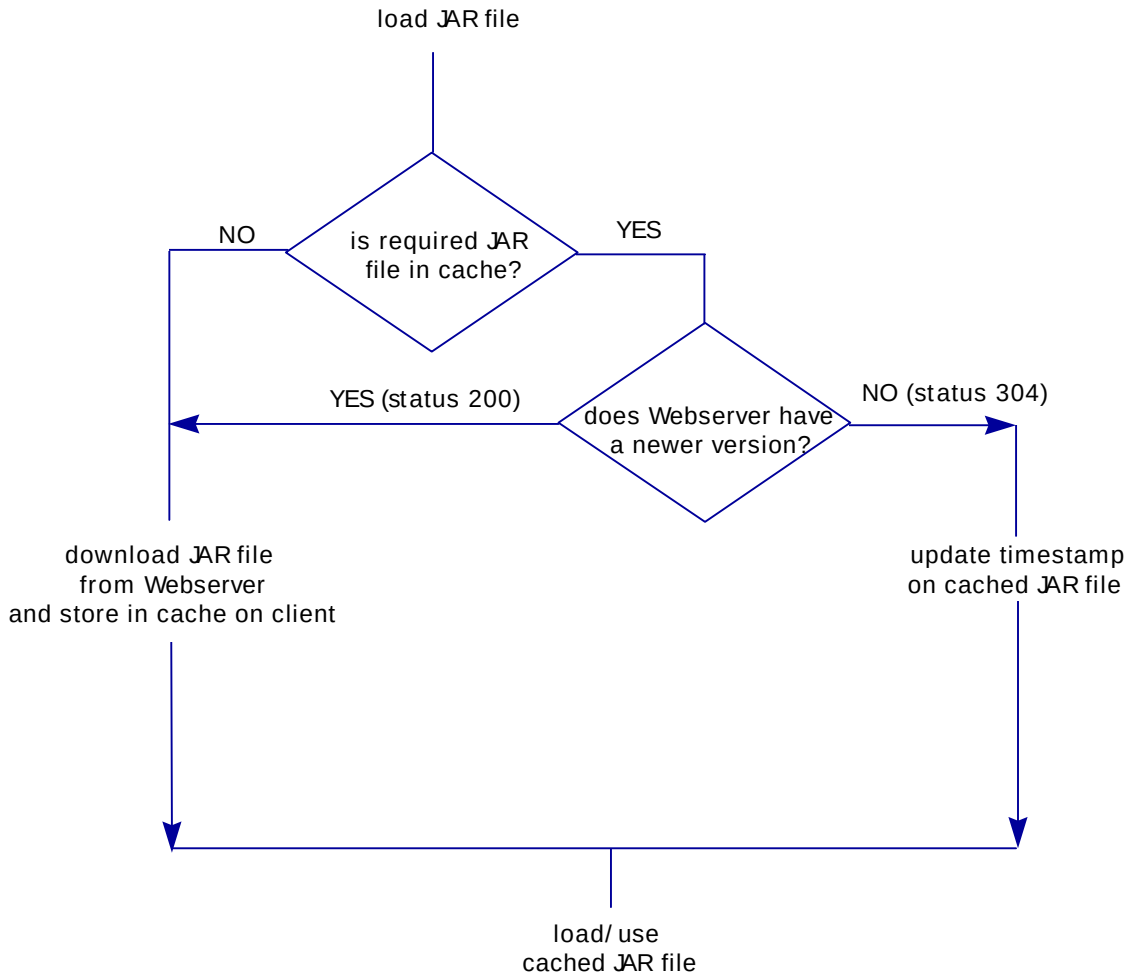

Figure 9: cache loading operation

Where are the cached JAR files stored?

By default, Oracle JInitiator stores the downloaded JAR files in the `jcache` subdirectory which under the Oracle JInitiator installation directory.

Why does the jcache directory contain strange names for the cached JAR files?

Each JAR on a Web server is identified by a URL (URL = codebase + JAR filename). The Oracle JInitiator caching mechanism uses this to uniquely identify the JAR file. On Windows operating systems, since the full URL is not a valid filename for a file, Oracle JInitiator transforms it via a hashing algorithm into an acceptable filename and then uses this as the stored JAR filename. When a request is made for a JAR file, Oracle JInitiator performs same transformation on the complete URL to determine if the JAR file is already in the cache.

How does JAR file caching work with server load balancing?

As outlined in the previous question, JAR files are identified in the cache based on the URL from which they were retrieved. This means that the same JAR file from different servers will be store separately. This is done deliberately to ensure security and application integrity. If this were not done then a malicious application could replace the JAR file from another application. In this case when the original application was re-run it would use the wrong files. Since JAR files are not guaranteed to have unique names, there exists the ability to have JAR file naming collisions. Storing the JAR file using the explicit URL from where it came prevents this.

It appears that the timestamp on the cached JAR files is updated every time I run an Oracle Developer application. Is this normal? Does it mean that the file is being downloaded every time?

Oracle JInitiator supports a configurable cache maximum size. Every time a cached JAR file is used, Oracle JInitiator updates the timestamp to indicate the date and time that the cached file was last used.

If the cache grows to the point where a file must be removed in order to maintain below the maximum cache size, Oracle JInitiator removes the least recently used file according to this timestamp.

How can I tell that my cache is functioning correctly and that the JAR files are not being downloaded everytime?

When Oracle JInitiator needs to download a required file, it does so via the Web server that has been configured to run Oracle Developer applications. Most Web servers support the use of log files that record what files have been downloaded, by whom and when. The Web server log file uses a standard format to describe the transactions that have occurred. This log format includes the name of the requested item and the result of the request. The result of the request is indicated using a set of standard HTTP status codes.

HTTP Status Code	Meaning
200	OK
304	NOT MODIFIED

If a JAR file was downloaded to the client, the log file will contain the name of the requested JAR file and the HTTP status code 200.

If a JAR file was not downloaded because the timestamp on it was earlier than the cached file timestamp, then the log file will contain the name of the requested JAR file and the HTTP status code 304.

The example below shows an entry made in a log file using standard NCSA log formatting in which the JAR file in the cache was not current had to be downloaded from the Web server.

```
ferret.us.oracle.com - - [19/Feb/1999:17:40:12 -0800] "GET
/forms_java/f60all.jar  HTTP/1.0" 200 -
```

The example below shows an entry made in a log file using standard NCSA log formatting when the JAR file in the cache was current and was not downloaded from the Web server.

```
ferret.us.oracle.com - - [19/Feb/1999:17:42:29 -0800] "GET
/forms_java/f60all.jar  HTTP/1.0" 304 -
```

It seems that when the jar file is downloaded, a .JCX file is created in the jcache directory. What is this file?

As a JAR file is being downloaded a temporary copy of it is written to the filesystem. This temporary copy is identified by the .jcx file extension. Once the download has successfully completed, the .jcx extension is channged to a .jc extension. If the download is interrupted at any point or the connection is dropped the temporary .jcx file will remain. Oracle JInitiator will not load a file with a .jcx extension.

I have verified that the caching is working correctly, but my application is still taking longer than I would like to start. Any Advice?

The JAR file caching provided by Oracle JInitiator does not perform any magic to increase the execution speed of Java. What it does is save the amount of time it needed to download the required JAR file to run the application via a caching mechanism. The operation of unzipping a JAR file, loading the contained classes into memory and then authenticating them takes a significant amount of the startup time. In fact, on a fast network the amount of time taken to download the JAR file will be smaller than the amount of time required to load the Java classes into memory and perform the authentication. In this case, caching saves very little in terms of overall application startup time. On a slower network, the time required to download the JAR file will become proportionately greater in the overall startup time. In this case JAR file caching becomes important.

The Oracle JInitiator development team is working on ways to address the generic Java issues such as JAR file unzipping, class loading and authentication in the effort to make application startup time as rapid as possible.

SECURITY

Does Oracle JInitiator use the same level of encryption as the AppletViewer, that is 40 bit RSA RC4 encryption?

Yes. Oracle JInitiator and the Oracle Appletviewer supplied with the OJDK use the same encryption technology (RSA RC4 40-bit).

Is there a way to run Oracle JInitiator with 128 bit encryption?

Oracle JInitiator uses the RSA RC4 40-bit algorithm for the encryption of Forms messages sent between the Applet client and the Oracle Developer Server. The US government forbids the exporting of software containing encryption software higher than 40 bits (Arms Export Control Act, 22 U.S.C. § 2778).

Can Oracle JInitiator be used with SSL ? When an application is connected to using SSL, the error message "Unknown protocol : https" is displayed.

Any time a running Oracle Developer application needs to load a resource such as an image or open a HTML document, it does so via Oracle JInitiator and not the Web browser itself. Two examples of this occur when are when the WEB.SHOW_DOCUMENT PL/SQL built-in used or when an images is downloading for use within the application.

Oracle JInitiator uses the standard Java class libraries as supplied by JavaSoft. These standard class libraries do not provide support for the HTTPS protocol. Any attempt to connect to an SSL enabled Web server using the HTTPS protocol using Oracle JInitiator will not succeed. The output from such an attempt is the error message shown above.

ADDITIONAL READING

The following additional papers on related topics can be downloaded from the Oracle Technology Network http://technet.oracle.com.

PM-CP-001 Oracle Developer Server: Client Platform Support, Statement of Direction

PM-CP-002 Oracle Developer Server: Solaris Client Support

PM-CP-003 Oracle Developer Server: Microsoft Internet Explorer 5.0 Support

PM-CP-005 Oracle JInitiator, Whitepaper

Oracle Corporation
World Headquarters
500 Oracle Parkway
Redwood Shores, CA 94065
U.S.A.

Worldwide Inquiries:
+1.650.506.7000
Fax +1.650.506.7200
http://www.oracle.com

Oracle JInitiator
Technical FAQ

Copyright © Oracle Corporation 1999
All Rights Reserved

This document is provided for informational purposes only, and the information herein is subject to change without notice. Please report any errors herein to Oracle Corporation. Oracle Corporation does not provide any warranties covering and specifically disclaims any liability in connection with this document.

Oracle is a registered trademark.

All other company and product names mentioned are used for identification purposes only and may be trademarks of their respective owners.