

Oracle 9iAS Forms Services – Using Sun's 1.4.1 Java Plug-in

*An Oracle White Paper
January 2003*

Oracle 9iAS Forms Services – Using Sun's 1.4.1 Java Plug-in

INTRODUCTION	3
Plug-in Version	3
Static version parameters and values:.....	3
Dynamic version parameters and values:.....	4
Applet Lifecycle	4
Dense Jar Format.....	4
Forms configuration files	4
<i>Example (from formsweb.cfg):</i>	5
<i>Example (from basejini.htm):</i>	6
<i>Example (Static HTML)</i>	6
MS Internet Explorer's Native JVM.....	7
Running with the 1.4.1x plug-in the first time.....	8
Additional Information	8

Oracle 9iAS Forms Services (Oracle Forms 6i) – Using Sun's 1.4.1 Java Plug-in

INTRODUCTION

This paper provides a guideline for deploying Oracle9iAS Forms Services (Oracle Forms 6i) using the 1.4.1_01 Java Plug-in from Sun. Sun's 1.4.1 Java Plug-in is a browser independent runtime environment including a Java virtual machine (JVM), class libraries, and other files that support the execution of programs written in the Java programming language.

Once you have completed installation and configuration and successfully tested the 1.4.1_01 Java Plug-in, you should be able to run Oracle9iAS Forms applications in the browser. Note that in order to use the Sun's Java Plug-in to run Oracle9iAS Forms applications, the HTML page that you use to launch the application must use the specific Sun's Java Plug-in tags that are recognized by the browser.

For additional details on the Sun's Java Plug-in refer to <http://www.javasoft.com>.

For more information regarding Client Platform Support and level of certification, please refer to the *Oracle Forms 6i Client Support, Statement of Direction* available from OTN (<http://otn.oracle.com/products/forms/content.html>).

PLUG-IN VERSION

The plug-in allows a static or dynamic versioning method to be used in determining the plug-in version that will run.

- **Static versioning:** If the user has a different version than indicated in the forms configuration file, the user will be prompted to download the newer version specified in the configuration file.
- **Dynamic versioning:** If the user has a lower version than specified, then the user will be directed to the plug-in page page. If the user has the same or a higher version, then the installed version will run.

Oracle recommends that the static versioning method be used so that the version certified by Oracle can be explicitly specified. See the Static HTML and Forms configuration file excerpts below for an example of how the static versioning is implemented.

Static version parameters and values:

There are two parameters to set in the forms configuration file:

```
jinit_classid=clsid: CAFEEFAC-0014-0001-0001-  
ABCDEFEDCBA
```

```
jinit_mimetype=application/x-java-applet;jpi-  
version=1.4.1_01
```

Dynamic version parameters and values:

Please visit the following link for more information on implementing the dynamic versioning method.

<http://java.sun.com/products/plugin/versions.html>

APPLET LIFECYCLE

In Oracle's JInitiator 1.1.8 we improved application performance within a browser session by applet instance caching. When a user navigates from the current page in the browser, the running Oracle9iAS Forms6i Services application is cached. When the user comes back to the page containing the applet, the applet that was running is automatically fully restored including all of the data entered in the application.

This functionality has now been introduced in Sun's 1.4.1 Java Plug-in, which provides parity with our current behavior in Oracle JInitiator 1.1.8. The difference is that an Applet parameter, **legacy_lifecycle**, must be added to the Form's configuration file in order to utilize this new enhancement in the 1.4.1 Java Plug-in. The value should be set to "true". Running in this mode allows a Forms end user to navigate to other web pages in the browser or reload the current page without immediately terminating the Forms Applet session. Thus the applet has a "legacy" which lasts beyond the current web page.

DENSE JAR FORMAT

With Forms we have found that we can improve compression and thus reduce download size of our JAR files by running them through the JAR utility an additional time. Therefore is Forms distributed with several jar files that contain jar files themselves, and Oracle JInitiator in turn allows the use of this dense jar format. The 1.4.1 plug-in from Sun also allows this format but requires that the jar file have an extension of '**jarjar**'. Any dense jar file listed in the archive parameter of the HTML or configuration file will need to have the new extension. The jar files provided with Forms can be copied and renamed with the new extension. For example, f60all_jinit.jar can be copied and renamed to **f60all_jinit.jarjar**.

Forms configuration files

The formsweb.cfg and basejini.htm files need to be modified in order for you to use the plug-in. By adding a new configuration section to the formsweb.cfg file you can specify the parameters that will be used with the 1.4.1 plug-in. In the example below we show two ways of using the 1.4.1 Java Plug-in. It can either be used as a standalone JVM in the same way as Oracle's JInitiator is used today, or it can be used to totally replace the native IE Java VM.

Add the **legacy_lifecycle** parameter to the new configuration section. See the Applet Lifecycle section above for more information on this parameter. Since this parameter is specific to the plug-in it will be ignored when using Oracle JInitiator.

Example (from formsweb.cfg):

```
; *****  
; SPECIFIC CONFIGURATIONS  
; *****  
  
[jdk141]  
  
; Used the same way as Oracle JInitiator  
  
archive_jini=f60all_jinit.jarjar  
  
jinit_classid=clsid: CAFEEFAC-0014-0001-0001-  
ABCDEFEDCBA  
  
jinit_mimetype=application/x-java-applet;jpi-  
version=1.4.1_01  
  
legacy_lifecycle=true  
  
[nativeVM]  
  
; Used as the native Plug-in VM in the browser.  
  
legacy_lifecycle=true  
  
IE50=native
```

Example (from basejini.htm):

```
<!-- Forms applet definition (start) -->
<OBJECT classid="%jinit_classid%"
...
<PARAM NAME="serverApp" VALUE="%serverApp%">
<PARAM NAME="legacy_lifecycle" VALUE="%legacy_lifecycle%">
<COMMENT>
<EMBED SRC="" PLUGINSOURCE="%jinit_download_page%"
...
serverApp="%serverApp%"
legacy_lifecycle="%legacy_lifecycle%"
...
</NOEMBED></EMBED>
</OBJECT>
<!-- Forms applet definition (end) -->
```

This is a new parameter!

This is a new parameter!

Example (Static HTML)

The following is an example of a static html file that uses the 1.4.1 plug-in with static versioning.

```
<HTML>
<BODY>
<P>
<OBJECT classid="clsid:CAFEEFAC-0014-0001-0001-ABCDEFEDCBA"
...
<PARAM NAME="CODEBASE" VALUE="/forms60code/" >
<PARAM NAME="ARCHIVE" VALUE="/forms60code/f60all_jinit.jarjar"
>
<PARAM NAME="type" VALUE="application/x-java-applet;jpi-
version=1.4.1">
...
<EMBED type="application/x-java-applet;jpi-version=1.4.1_01"
java_CODE="oracle.forms.engine.Main"
java_CODEBASE="/forms60code/"
```

```
java_ARCHIVE="/forms60code/f60all_jinit.jarjar"  
...  
serverApp="default"  
legacy_lifecycle="true">  
...
```

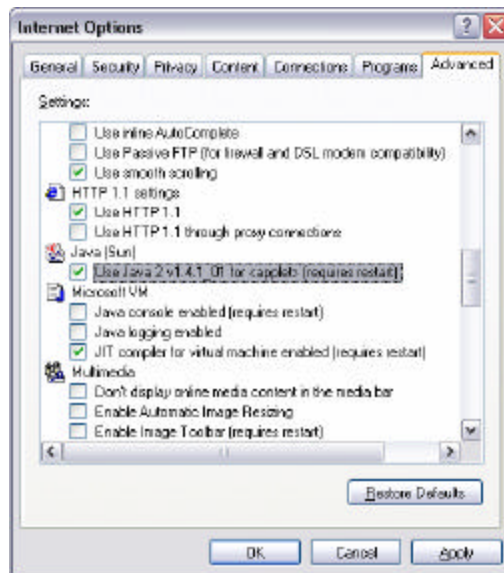
MS INTERNET EXPLORER'S NATIVE JVM

There is two ways of exposing the new Java plug-in in Internet Explorer, you can either use it the same way as Oracle's JInitiator as mentioned above, or as the native VM in Internet Explorer replacing the IE native JVM . If you still want to use the IE native JVM there is two ways of disabling the 1.4.1 plug-in after installing the plug-in.

The first option is to open the control panel for the Java Plug-in and change the browser settings accordingly. See picture:



The second option is to disable the 1.4.1 plug-in in Internet Explorers Java settings. This will allow you to use the Native JVM or the plug-in based on the parameters provided in the html files. See picture:



RUNNING WITH THE 1.4.1X PLUG-IN THE FIRST TIME

The first time an end user accesses a Forms application registered to use the 1.4.1_01 plug-in a security warning is shown, asking the user to grant permission to the code signed by Developer. The user should select either Yes or Always if the user don't want the message again for the Forms applet.



Additional Information

Oracle Forms 6i Client Platform Support, Statement of Direction:
http://otn.oracle.com/products/forms/htdocs/clientsod_forms6i.html

Download:
<http://java.sun.com/j2se/1.4.1/download.html>

Developer Guide:
http://java.sun.com/j2se/1.4.1/docs/guide/plugin/developer_guide/

Please see the Java Plug-in 1.4.1 Developer Guide for details about the CLSID and MIME type.

http://java.sun.com/j2se/1.4.1/docs/guide/plugin/developer_guide/version.html



Oracle 9iAS Forms Services – Using Sun's 1.4.1 Java Plug-in
November 2002
Author: Todd Newlin, Jonas Jacobi

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

Worldwide Inquiries:
Phone: +1.650.506.7000
Fax: +1.650.506.7200
www.oracle.com

Oracle is a registered trademark of Oracle Corporation. Various product and service names referenced herein may be trademarks of Oracle Corporation. All other product and service names mentioned may be trademarks of their respective owners.

Copyright © 2001 Oracle Corporation
All rights reserved.