



Forms Diagnostic Techniques

An Oracle Technical White Paper

June 2000

CONTENTS

| | |
|--|-----------|
| 1. INTRODUCTION | 1 |
| 1.1 PURPOSE..... | 1 |
| 1.2 AUDIENCE | 1 |
| 1.3 TERMINOLOGY | 1 |
| 1.4 A DIAGNOSTIC APPROACH TO PROBLEM RESOLUTION..... | 1 |
| 1.4.1 Problem/Section Matrix | 3 |
| 2. FORMS SERVER CONFIGURATION | 4 |
| 2.1 OVERVIEW..... | 4 |
| 2.2 APPLETVIEWER CONFIGURATION..... | 6 |
| 2.2.1 Install Appletviewer on the client | 6 |
| 2.2.2 Create a test form | 7 |
| 2.2.3 Configure your Web Server | 7 |
| 2.2.4 Start the Forms Listener | 8 |
| 2.2.5 Create a HTML file..... | 9 |
| 2.2.6 Run the form over the Web. | 11 |
| 2.3 COMMON PROBLEMS USING APPLETVIEWER..... | 12 |
| 2.3.1 The Web Server is not reachable. | 12 |
| 2.3.2 Web Server Virtual Directory is not correct | 12 |
| 2.3.3 CODEBASE not correct | 12 |
| 2.3.4 Unable to contact the Forms Listener | 13 |
| 2.4 JINITIATOR CONFIGURATION | 13 |
| 2.4.1 Install Oracle JInitiator on the client | 13 |
| 2.4.2 Create a HTML file to load Oracle JInitiator | 14 |
| 2.4.3 JInitiator with Microsoft Internet Explorer | 14 |
| 2.4.4 JInitiator with Netscape Navigator..... | 16 |
| 2.4.5 JInitiator with both Netscape Navigator and Internet Explorer | 18 |
| 2.4.6 What version of Oracle JInitiator is being used by my browser? | 19 |
| 2.4.7 What version of JInitiator is installed in Netscape? | 19 |
| 2.5 HTML FILE PROBLEMS WITH ORACLE JINITIATOR AND INTERNET EXPLORER..... | 20 |
| 2.5.1 java.io.FileNotFoundException which refers to JAR file on startup | 20 |
| 2.5.2 Performance on startup slow, many class files loaded but no JAR files | 21 |
| 2.5.3 FRM-99999: Registry file http://ukp14998.uk.oracle.com/forms60codes/oracle/forms/registry/Registry.dat is missing. | 22 |
| 2.5.4 File not found error | 23 |
| 2.5.5 Browser hangs with message 'Loading Java Applet...' | 24 |
| 2.5.6 Applet does not appear, only an empty box with a red cross | 25 |
| 2.5.7 Do I need to exit and restart my browser? | 26 |
| 2.6 HTML FILE PROBLEMS WITH ORACLE JINITIATOR AND NETSCAPE NAVIGATOR..... | 26 |
| 2.6.1 NullPointerException in getHashKey loading applet | 26 |
| 2.6.2 Unable to contact http://.../xxx.jar and FileNotFoundException in getInputStream | 27 |
| 2.6.3 Blank page on Netscape when trying to run Forms with JInitiator | 28 |
| 2.6.4 Blank page in Netscape, many class files loaded | 28 |
| 2.6.5 Blank page in Netscape, Java console reports versions and user home directory | 28 |
| 3. FRM-99999 ERRORS | 30 |
| 3.1 WHAT CAUSES FRM-99999? | 30 |
| 3.2 RESOLVING FRM-99999 | 31 |
| 3.2.1 What does the applet show ? | 31 |
| 3.2.2 What does the Java console show ? | 33 |
| 3.2.3 Did the server recognise the incoming connection ? | 34 |
| 3.2.4 Starting a dedicated Forms process on the server..... | 36 |

| | |
|---|-----------|
| 3.2.5 Did the server crash ? | 37 |
| 3.3 KNOWN CAUSES OF FRM-99999..... | 37 |
| 3.3.1 Configuration Issues..... | 38 |
| 3.3.2 Known Bugs..... | 38 |
| 3.3.3 FRM-99999 Errors in Forms 6.0..... | 40 |
| 4. SERVER CRASH..... | 44 |
| 4.1 WHAT INFORMATION IS IN THE DUMP FILE ? | 44 |
| 4.2 HOW DOES THIS HELP ? | 46 |
| 4.3 TESTING IN MULTIPLE ENVIRONMENTS | 46 |
| 5. CLIENT CRASH..... | 47 |
| 5.1 EXAMPLE DUMP FILE | 47 |
| 5.2 WHAT DOES IT SHOW AND WHY IS IT USEFUL ? | 49 |
| 6. APPLICATION HANGING..... | 50 |
| 6.1 OBTAINING A CLIENT THREAD DUMP..... | 50 |
| 6.2 EXAMPLE STACK DUMP | 50 |
| 6.3 WHAT DOES IT SHOW AND HOW IS IT USEFUL ? | 53 |
| 6.4 COMMON CAUSES OF HANGING APPLICATIONS..... | 53 |
| 7. INTERMITTENT ERRORS | 54 |
| 7.1 DIAGNOSING NETWORK PROBLEMS..... | 54 |
| 7.2 USING PING TO CHECK THE NETWORK..... | 54 |
| 7.2.1 ping syntax..... | 54 |
| 7.2.2 Sample output | 55 |
| 7.2.3 Using traceroute / tracert | 56 |
| 7.2.4 Using netstat | 57 |
| 7.2.5 Tracing the problem | 58 |
| 7.3 NETWORK STATISTICS IN FORMS | 58 |
| 7.4 MEMORY PROBLEMS..... | 59 |
| 7.4.1 Setting JVM runtime options | 60 |
| 7.5 MEMORY LEAKS..... | 61 |
| 7.5.1 What is a memory leak ?..... | 61 |
| 7.5.2 Memory leaks in Java | 62 |
| 7.5.3 How to identify a memory leak | 62 |
| 8. PERFORMANCE PROBLEMS | 64 |
| 8.1 CACHING IN THE ORACLE JDK | 64 |
| 8.1.1 What is Caching ?..... | 64 |
| 8.1.2 Why is it important ? | 64 |
| 8.1.3 How do you check if caching is working? | 64 |
| 8.1.4 Performance Improvements in OJDK 1.1.7.15..... | 65 |
| 8.1.5 Configuring OJDK 1.1.7.15 | 67 |
| 8.2 THE FORMS LATENCY METER..... | 70 |
| 8.2.1 What is the Forms Latency Meter ? | 70 |
| 8.2.2 How is the Forms Latency Meter enabled ? | 71 |
| 8.2.3 Why use the Latency Meter ? | 73 |
| 8.2.4 Example Results..... | 73 |
| 8.3 EXPECTED PERFORMANCE EXAMPLES..... | 74 |
| APPENDIX A - FORMS SERVER CONNECTION ACTIVITY LOGGING..... | 75 |
| A 1 SWITCHING SERVER LOGGING ON | 75 |
| A 2 SAMPLE OUTPUT | 76 |
| APPENDIX B - FORMS RUNTIME DIAGNOSTICS (FRD)..... | 80 |
| B 1 INITIATING FRD..... | 80 |

| | |
|---|------------|
| B 1.1 INITIATINGFRD FROM THE COMMAND LINE..... | 80 |
| B 1.2 INITIATINGFRD FROM THE WEB | 81 |
| B 1.3 INITIATINGFRD FROM ORACLE ENTERPRISE MANAGER (OEM)..... | 83 |
| B 1.4 INITIATINGFRD FROM THE COMMAND LINE..... | 85 |
| B 2 FRD OUTPUT..... | 86 |
| B 2.1 INSTRUMENTED EVENTS..... | 86 |
| B 2.2 EXPECTED OUTPUT..... | 86 |
| <i>B 2.2.1 Navigation</i> | 87 |
| <i>B 2.2.2 Triggers</i> | 87 |
| <i>B 2.2.3 Built-ins</i> | 88 |
| <i>B 2.2.4 Messages</i> | 89 |
| <i>B 2.2.5 File Opens</i> | 89 |
| <i>B 2.2.6 Unhandled Exceptions</i> | 90 |
| B 2.3 DATA DUMPS..... | 90 |
| <i>B2.3.1 Environment Dump</i> | 90 |
| <i>B 2.3.2 State Dump</i> | 91 |
| <i>B 2.3.3 Event / Information Logged Matrix</i> | 92 |
| B 3 USINGFRD OUTPUT..... | 93 |
| B 4 FRD FEATURE / VERSION MATRIX..... | 94 |
| B 5 FRD CONSTANTS..... | 94 |
| APPENDIX C - GLOSSARY..... | 114 |

1. INTRODUCTION

1.1 PURPOSE

This document relates to resolving problems when running an application over the Web using Oracle Forms Server. The paper outlines the common causes of errors, how to verify your installation, and the techniques and tools available to diagnose the problem.

1.2 AUDIENCE

This document is intended for anyone involved in diagnosing and resolving problems with Oracle Forms, particularly Oracle Forms Server. After reading this paper you will have a good understanding of the diagnostic process, and will know the available techniques and tools to aid information gathering and ultimately problem resolution.

System administrators for Oracle Applications should note that some sections of this document will not apply to Oracle Applications. Oracle Applications has a tightly controlled environment, and you should check with Oracle Support Services that the steps you intend to take are supported for use with Oracle Applications before proceeding.

1.3 TERMINOLOGY

In this document, the term 'Forms Listener' refers to the process that listens for incoming connections; the term 'Forms Runtime' or 'Forms Runtime Engine' refers to a process started on the server to run the requested forms for a particular user, and the term 'Forms Server' refers to the Forms Listener plus its associated Forms Runtimes.

1.4 A DIAGNOSTIC APPROACH TO PROBLEM RESOLUTION

Everybody approaches a problem differently. A list of troubleshooting hints follows. This list is not a definitive guide to problem solving and will not guarantee resolution, but may help you approach a complex problem.

- Be methodical
 - Don't leap to the area you believe to be the cause based on a hunch, or a guess - make sure you eliminate the other possibilities first. An easy trap to fall into is that of spending long periods of time trying to find evidence to support your theory, rather than concentrating on what the evidence shows.

- Don't overlook the trivial or the obvious.
- Divide the problem into sections
 - Chop the problem into manageable sections - this helps eliminate whole areas from investigation. As you investigate an area and satisfy yourself that the problem does not lie there, you can proceed to the next section. An approach to diagnosing a problem that is often successful is to reduce it to its essential parts. This will be important if you need to discuss the problem with Oracle Support Services to obtain a solution.
 - Define what happens, when it happens, how often it happens. Of equal importance is understanding what does *not* happen, when it does *not* happen etc. For example, if a group of users in the same building all get the problem, and it always happens between 9 and 10am, it is just as important to know that it never reproduces in another building, or after 10pm. Perhaps the users only use a particular Form between 9 and 10, or the load on the system is highest between 9 and 10am.
- Read the error messages
 - It sounds obvious, but sometimes the information is right there in the error text. Unfortunately, interpreting the error message may not always be simple.

For example, when using JInitiator with Internet Explorer, if you omit the CODEBASE tag entirely, you will get an error message which begins :-

```
FRM-99999: Registry file
http://ukp14998.uk.oracle.com/oracle/forms/registry/Registry.dat
is missing.
```

Note that the missing file is reported as

```
http://ukp14998.uk.oracle.com/oracle/forms/registry/Registry.dat
```

This could lead to confusion as on machine ukp14998.uk.oracle.com, the Forms 6 installation has a directory structure which matches this and the file Registry.dat exists. However, the URL mentioned does not point to that file, and if you tried to access that file in a browser you would get the following error :-

```
HTTP/1.0 404 Object Not Found
```

- This document will help you understand the error messages, and help identify what action to take.
- Make sure you can reproduce the problem, if possible.
 - If you can reproduce the problem yourself, you may notice some behaviour that the end user never spotted - perhaps it had always happened, so they simply assumed it was meant to happen. If you cannot reproduce the problem then you have already started the first steps to resolving it. A laptop computer can be very useful in this scenario - if you cannot reproduce it at your desk, then take the laptop to the users desk, and see if it reproduces there. If it does, then you now know that its unlikely to be a software configuration problem, and may have more to do with the difference in networking at the two desks.
- Make sure you understand the tools you are trying to use

If you decide to use a diagnostic tool, make sure you know how to use it, and how to interpret the data it

produces. Time spent in investigating the usage of a tool *before* the problems happens is time well invested - typically you will be under time constraints when working on the problem, and will not have time to learn the tool as well.

1.4.1 Problem/Section Matrix.

This is a large document. Use the table below to help locate the sections relevant to the particular problem you are experiencing.

| | Forms Server Configuration | FRM-99999 Errors | Server Crash | Client Crash | Application Hanging | Intermittent Errors | Performance Problems | Connection Activity Logging | Forms Runtime Diagnostics |
|---------------------------------------|----------------------------|------------------|--------------|--------------|---------------------|---------------------|----------------------|-----------------------------|---------------------------|
| Problem\Chapter | 2 | 3 | 4 | 5 | 6 | 7 | 8 | A | B |
| No form will start | X | | | | | X | | | |
| Some forms will start, some will not. | X | | | | | | | X | X |
| FRM-99999 | X | X | X | | | X | | X | X |
| Java Errors on client | X | | | X | | | | | |
| Client Crash | X | | | X | | | | | |
| Server Crash | | X | X | | | | | X | X |
| Application Hangs | | | | | X | X | | X | X |
| Intermittent problems | | | | | | X | | X | X |
| Forms startup is very slow | X | | | | | X | X | | |
| Performance after startup is poor | X | | | | | X | X | | |

2. FORMS SERVER CONFIGURATION

This section will describe the minimum steps required to configure Forms Server using its simplest implementation methods, and how to verify the installation is correct.

For further information on configuring Forms Server and Forms Server architecture refer to the on-line help.

You can also find several white papers on the following Oracle Web sites

```
http://www.oracle.com
http://technet.oracle.com
http://metalink.oracle.com
```

2.1 OVERVIEW

If you are setting up the Forms Server for the first time, or are having problems installing on a new machine, it is strongly recommended that you first try the simplest type of installation possible to 'prove' the technology. Once this has been done, extra features can be added and tested until the final desired configuration is achieved. This will allow you to verify that the basic Forms Server is installed correctly, that your Web Server is configured correctly and works, and that the client machine can communicate with both the Forms Server and the Web Server.

Oracle Forms Server can be configured in a number of combinations

- Static HTML
 - This is the most straightforward implementation style. In this method, the HTML file contains all the information required for the applet to start the communication with the Forms Listener, and all the information the Forms Server needs to run the form.
- Cartridge
 - In this implementation method, some information is known by the cartridge regardless of what URL is passed to it. When the URL is passed to the cartridge then the cartridge uses the extra information passed in the URL to generate the HTML file required to run a form.
- CGI
 - New in 6i.
 - The forms CGI provides the same features as the Forms cartridge, but can be used with any Web Server that supports CGI. It dynamically creates a HTML file using the parameters passed in as part of the URL. It also allows load balancing between primary and secondary nodes.

There is also a choice of certified client browsers

- Appletviewer
 - Appletviewer is an application provided by Javasoft with the Java Development Kit. It allows the user

to view a Java Applet. It does not allow the user to view HTML although a HTML file with the relevant APPLET tag is needed to run an applet with Appletviewer.

- On the Forms distribution, we provide a copy of Appletviewer, based on the Oracle JDK (OJDK). The Oracle JDK is based on the Javasoft JDK, and only adds select bug fixes to improve stability and performance. It is not an Oracle-specific implementation of the JDK.
- Appletviewer is not certified for use with Oracle Applications 11*i*.
- JInitiator
 - JInitiator is a plug-in to a browser which provides a Java Runtime Environment based on the Oracle JDK. This plug-in allows the user to run an applet that may require a different Java version to the one their browser provides - the plug-in is responsible for running the applet. Using JInitiator allows the user to continue to use their preferred browser. The use of JInitiator is covered in more depth in section 2.4.
- Native Browser
 - Some browsers are supported with their native Java Runtime Environment. This list of browsers is under constant review as an increasing number of vendors implement up-to-date JREs. Please see <http://technet.us.oracle.com/products/developer/> for documentation on native browser support.

In addition, you have a choice of which communication protocol to use between the Forms Runtime and the Java applet :-

- Sockets
 - Many Internet based technologies use ‘Sockets’ for communication. A simple way to think of sockets is to imagine a numbering system for programs that communicate over the network. Typically these programs have a ‘client’ part and a ‘server’ part. The client and the server identify themselves by their sockets or port numbers (amongst other things). Communication between the client and the server parts of a program is done over what is called a ‘socket connection’.
 - Supported in Forms 4.5.10.x, 5.0.x, 6.0.x, 6*i*.
 - Uses the TCP/IP protocol to communicate with the server.
 - The default communication method in 6*i*.
- HTTP
 - This mode still uses a ‘socket connection’, but it is now a HTTP socket connection - the messages between the Forms Server and the Java client are now encapsulated in HTTP packets.
 - Beta in 6.0, production in 6*i*.
 - Allows communication through a firewall.
 - Uses HTTP 1.1
 - supports SSL to ensure secure communications over the Internet.

For more information on Sockets and HTTP connections, see the Oracle White Paper 'Deploying Internet Applications Using HTTP-Enabled Oracle Developer Server'

2.2 APPLETVIEWER CONFIGURATION

This list of requirements assumes that you already have a working Web Server and have installed Forms Server following the installation procedures for your platform. If you do not have a Web Server then you will need to install one. You should verify that your Web Server is capable of serving HTML pages before continuing.

2.2.1 Install Appletviewer on the client

Appletviewer is provided on the Oracle Forms Server distribution. You can also obtain Appletviewer from Oracle's Metalink Web site (<http://metalink.oracle.com>).

There are several versions of Appletviewer available, and the use of Appletviewer is not supported in all configurations. Please use

```
http://certify.us.oracle.com
```

to check certification issues. If in doubt, contact Oracle Support Services for clarification.

To check that Appletviewer has been installed correctly, open an MSDOS command prompt window. Change Directory to the place where you installed Appletviewer (for example `c:\appletviewer`)¹ and then navigate to `<install directory>\jdk\bin` directory and type

```
appletviewer
```

If Appletviewer is installed then a message detailing the usage of the command will be shown, otherwise a message stating the command was not recognised will be displayed.

¹ Note : Appletviewer can be installed in any directory - it does not need to reside in \$ORACLE_HOME. You may want to alter your PATH setting to include Appletviewer so that you can start it more easily.

```

Microsoft(R) Windows NT(TM)
(C) Copyright 1985-1996 Microsoft Corp.

D:\>cd jdk\1_1_7_20o\bin

D:\jdk\1_1_7_20o\bin>appletviewer
usage: appletviewer [-debug] [-J<runtime flag>] url|file ...

D:\jdk\1_1_7_20o\bin>cd ..

D:\jdk\1_1_7_20o>appletviewer
The name specified is not recognized as an
internal or external command, operable program or batch file.

D:\jdk\1_1_7_20o>

```

2.2.2 Create a test form

In order to test the Forms Server installation, you will need a simple form. Avoid choosing a 'simple' form from your application as it will rarely be as simple as it appears and may require a specific customised menu, attached libraries and a database connection.

Your simple form should require no database connection, and consist of a single canvas with some text and a button.

2.2.3 Configure your Web Server

Once you have installed the Forms Server software and have created a simple form with which to test the installation, you will need to complete some basic configuration of your Web Server.

In order to get the Forms Server working in its simplest configuration it is not required. Any Web Server which can service HTTP requests is sufficient. Using your Web Server of choice, you will need to create a number of virtual directories. A virtual directory is really an alias for a directory on the server machine. The exact details for creating a virtual directory will vary depending on the Web Server, so you may need to consult your Web Server documentation.

Using the method appropriate for your Web Server, create the following virtual directory

| Virtual Directory | NT Physical Directory | UNIX Physical Directory |
|-------------------|----------------------------|----------------------------|
| forms60code | %ORACLE_HOME%\forms60\java | \$ORACLE_HOME/forms60/java |

The actual name of the virtual directory is not important, but it must match exactly the name specified in the

HTML file.

You should replace %ORACLE_HOME% or \$ORACLE_HOME with the appropriate directory name.

This new virtual directory points to the Java class files for the Forms applet. When a Web Server sees a URL containing a reference to a file containing this virtual directory it will attempt to construct the real filename by replacing the virtual directory with the physical directory specified in the mapping.

For example:

| | |
|---|--|
| Virtual directory = mystuff | physical directory = /u01/website/misc |
| URL: http://test-pc.uk.oracle.com/mystuff/test.html | /u01/website/misc/test.html |

2.2.4 Start the Forms Listener

The exact syntax for starting the Forms Listener is dependent on the operating system and on the version of Forms.

If your Forms Server is installed on NT, then you will have a shortcut to start the listener on the Start menu, under Start->Programs->Developer6 ->Forms Server Listener. By default, the listener will listen for socket connections on port 9000.

If you want to start the listener by hand, the details are below.

| Operating System | Version | Command |
|------------------|-----------|------------------------------------|
| NT | FORMS 4.5 | f45srv32 port=<port number> |
| NT | FORMS 5.0 | f50srv32 port=<port number> |
| NT | FORMS 6.0 | if60srv -listen port=<port number> |
| UNIX | FORMS 4.5 | f45ctl start port=<port number> |
| UNIX | FORMS 5.0 | f50ctl start port=<port number> |
| UNIX | FORMS 6.0 | f60ctl start port=<port number> |

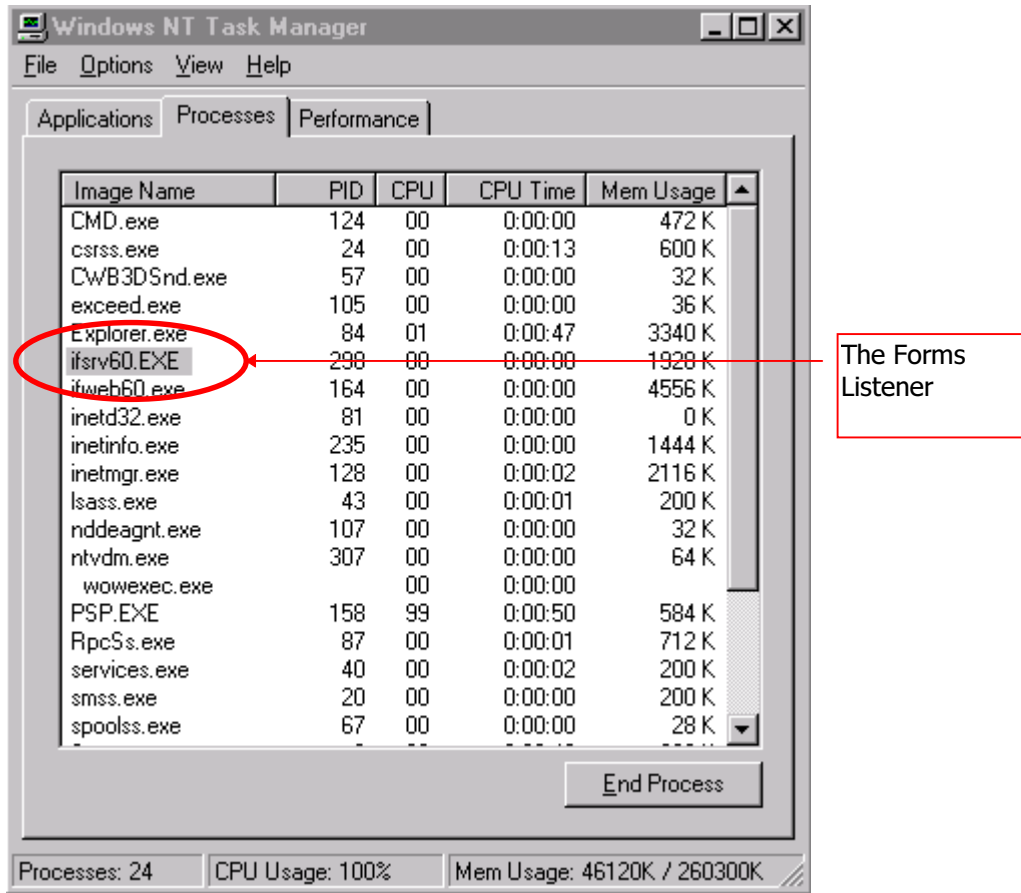
For more information see the on-line help, and the section on Forms Server Connection Activity Logging (Appendix A)

To check that the server has started correctly, you need to check the list of running processes. On UNIX, use the ps command. The process will be called f45srvm, f50srvm or f60srvm depending on your release of Forms.

E.g.

```
cxlewis-sun.us.oracle.com [7] > ps -ef | grep f60srvm
cxlewis 18723 18638 0 02:47:30 pts/4    0:00 grep f60srvm
cxlewis 18714 18638 0 02:47:08 pts/4    0:00 f60srvm port=9000
```

On NT, you will need to use Task Manager. You can start task manager by right-clicking on the NT taskbar.



If you are not sure what port the Forms listener is listening on, then you can restart the listener, specifying the port you want to use. Killing and restarting the Forms Listener will not impact any Forms Runtime processes that are running.

When you start the Forms Listener, you will often see another Forms process created. This extra process is a pre-spawned Forms Runtime, which the Forms Listener will use for the first incoming connection. Pre-spawning this process allows the initial connection time to be reduced. You can configure the number of pre-spawned processes created on Listener startup by using the pool parameter.

For example, the following statement will start the Forms Listener, and create ten Forms Runtime engines.

```
ifsrv60 -listen port=6000 pool=10
```

2.2.5 Create a HTML file

In order to test that the Forms Server is correctly installed and working, you must create a HTML file.

The HTML file tells the client which Java applet to run, and also passes other arguments required either by

the Java runtime or the Forms Server. The following is an example of a very simple static HTML file, based on the sample HTML file supplied with the demos. If you have installed the demos (on NT, for example) this file will be :-

C:\ORANT\TOOLS\DEVDEM60\WEB\STATIC.HTM

Example HTML File

```
<HTML>
<!-- FILE: static.html -->
<!-- Oracle Static (Non-Cartridge) HTML File Template (Windows NT) -->
<!-- Rename, and modify tags and parameter values as needed -->
<HEAD><TITLE>Oracle Forms Server</TITLE></HEAD>
<BODY><BR>Please wait while the Forms Client class files download and run.
      <BR>This will take a second or two...
<P>
<!-- applet definition (start) -->
<APPLET CODEBASE="/forms60code/"
        CODE="oracle.forms.engine.Main"
        ARCHIVE="/forms60code/f60web.jar"
        HEIGHT=20
        WIDTH=20>
<PARAM NAME="serverPort" VALUE="9000">

<PARAM NAME="serverArgs" VALUE="module=c:\temp\nolog">

<PARAM NAME="serverApp" VALUE="default">
</APPLET>
<!-- applet definition (end) -->
</BODY>
</HTML>
```

The highlighted items are the only parts which need to be changed for a basic installation².

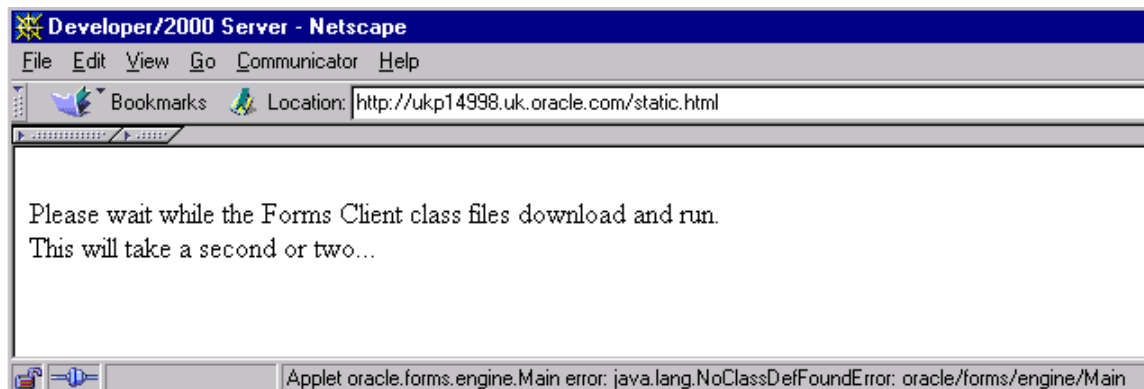
| HTML Tag | What is it ? | Comments |
|----------|--|--|
| CODEBASE | The location of the initial class (specified in CODE) to start running when starting the applet. | This is the virtual directory which you defined in step 2.2.3 |
| ARCHIVE | The location of JAR files. A JAR file is a collection of Java class files | This could be a separate virtual directory. However, in a standard Forms install, the jar files are in the same location as the class files. |

² In earlier versions of Forms, the CODE parameter is different to that shown. See the example HTML provided with the software for the correct value.

| | | |
|------------|---|---|
| serverPort | The port number used by the applet to communicate with the Forms Listener. | The default value is 9000 The value here must match the port number used by the Listener. |
| ServerArgs | The command line passed by the Forms Listener to the Forms Web Runtime process. | Note that restrictions on its syntax are the same as those when running if60run or f60runm on the server platform e.g. if FORMS60_PATH is not defined on the Forms Server, you must specify the full filename of the Forms module |

Once you have created this file and modified the relevant parameters, place it on the server in a location which will be visible to a Web browser on a client machine. You may choose to define another virtual directory for this purpose.

To verify that you can see this file from a client machine, use a Web browser :



If you get an error reported on the browser status line do not be alarmed. The purpose of this test is to verify that the URL can be resolved correctly. If you view the page source (in Netscape select View -> Page Source from the menu; in Internet Explorer, select View->Source) you should be able to confirm that the page loaded is the correct page. If this is not the case then you will need to check any virtual directories that you may have specified in the URL (not within the page itself - the page which references them does not even load, so that isn't a problem yet) and the configuration of the Web Server.

2.2.6 Run the form over the Web.

Once the Forms Listener is started and you have a test HTML file you will be able to run your test form using Appletviewer on your client. To do this, open an MSDOS window and navigate to the JDK\BIN directory as before. Enter the following command:

```
appletviewer <URL>
```

For example,

```
appletviewer http://test-pc.uk.oracle.com/myvirtualdir/testform.html
```

2.3 COMMON PROBLEMS USING APPLVIEWER

If you are unable to run the set up as described above, then the following section explains commonly encountered problems, how to identify them and what to do to resolve them.

2.3.1 The Web Server is not reachable.

If the Web Server is down, or you made a mistake typing in the address of the server, you will see an error similar to the following:

```
C:\users>appletviewer "http://notarealserver/test.html"  
I/O exception while reading: notarealserver  
Is http://notarealserver/test.html the correct URL?
```

By default most Web Servers are configured to listen on port 80 for URL requests. If you attempt to connect to the wrong port you will encounter the following error:

```
C:\users>appletviewer "http://cxlewis-sun.us.oracle.com:99/test.html"  
I/O exception while reading: Connection refused  
Is http://cxlewis-sun.us.oracle.com:99/test.html the correct URL?
```

2.3.2 Web Server Virtual Directory is not correct

If the virtual directory you specified in the URL is not defined on the application server you will encounter the following.

```
C:\users>appletviewer "http://ukp14901.uk.oracle.com/webhtml/testx.html"  
Warning: No Applets were started. Make sure the input contains an <applet> tag.  
usage: appletviewer [-debug] [-J<javaflag>] [-encoding <character encoding type>  
] url|file ...
```

2.3.3 CODEBASE not correct

You will get the following error when the codebase virtual directory is not set up.

```
C:\users>appletviewer "http://ukp14901.uk.oracle.com/webhtml/nocodebase.html"  
JAR caching enabled.  
Cache directory: d:\jdk\1_1_7_21o\bin\..\jcache  
Maximum cache size: 20971520 bytes  
Unable to contact http://ukp14901.uk.oracle.com/forms60jarsxxx/f60all.jar  
java.io.FileNotFoundException: http://ukp14901.uk.oracle.com/forms60jarsxxx/f60all.jar  
at sun.net.www.protocol.http.HttpURLConnection.getInputStream(Compiled Code)
```



```
    at sun.net.www.protocol.http.HttpURLConnection.openConnectionCheckRedire
cts(Compiled Code)
    at sun.applet.JARCache.beginStoring(JARCache.java:224)
    at sun.applet.AppletResourceLoader.loadJar(AppletResourceLoader.java:184)
    at sun.applet.AppletPanel.loadJarFiles(Compiled Code)
    at sun.applet.AppletPanel.runLoader(AppletPanel.java:392)
    at sun.applet.AppletPanel.run(Compiled Code)
    at java.lang.Thread.run(Thread.java:466)
```

2.3.4 Unable to contact the Forms Listener

If the forms listener is not started on the application server, or if the port specified in the HTML file does not match the port the Forms Listener is connected to, then the applet will report an FRM-99999 error. See Section 3 on resolving FRM-99999 Errors.

2.4 JINITIATOR CONFIGURATION

This section assumes you have previously configured the Forms Server, and have successfully deployed a simple form using Appletviewer.

2.4.1 Install Oracle JInitiator on the client

Install Oracle JInitiator onto a client PC running Windows NT or 95 - check the installation guide for details of how to do this. Oracle JInitiator is supplied on the Oracle Developer Server distribution, or the most recent version may be obtained from Oracle's Metalink Web site (<http://metalink.oracle.com>). Once the jinit.exe installation executable has been installed from the distribution or downloaded, you must run it to install Oracle JInitiator. Although the basic configuration of Oracle JInitiator will be covered later, for full details of the options available in the version you are using see the following files :-

| File | Contents |
|----------------|---|
| jinit_tags.htm | Explanation of the use of HTML tags to use Oracle JInitiator with both Netscape and Internet Explorer |
| readme.htm | Details of version compatibility of Oracle JInitiator and browsers, and also the Mime types and classid supported |

The location of these files may vary and will depend on the installation directory chosen for JInitiator.

For complete details specifying which versions of Jinitiator are certified for us on your particular operating system see

<http://certify.us.oracle.com>

2.4.2 Create a HTML file to load Oracle JInitiator

In order for your browser to recognise that you wish to use Oracle JInitiator to run the Forms applet, you will need to create a suitable HTML file. An example of a suitable static HTML file is supplied with Oracle JInitiator (static_jinit.htm) and can be found in the \docs directory of the JInitiator installation. This HTML file contains information required to use Oracle JInitiator with both Netscape and Internet Explorer. As both these browsers use different syntax to load plug-ins such as JInitiator, some of the information in the HTML file will be duplicated. The information contained within the <EMBED> tag is recognised by Netscape, and the <OBJECT> tag is recognised by Internet Explorer.

2.4.3 JInitiator with Microsoft Internet Explorer

A simple example of a HTML file for use with Microsoft Internet Explorer (Explorer) is shown below. Note that the HTML file for Explorer does not refer directly to the plug-in Oracle JInitiator - it simply defines the classid for the object being loaded. When Oracle JInitiator is installed, it registers its classid so that Explorer

knows to use it.

Example HTML file for Internet Explorer

```
<HTML>
<HEAD>
  <TITLE>Oracle Developer Server and Oracle JInitiator</TITLE>
</HEAD>
<BODY>
<OBJECT classid="clsid:020f6116-407b-11d3-a3bb-00c04fa32518"
  width="600" height="600"
  codebase="http://www.acme.com/jinit11741.exe#Version=1,1,7,18">
  <PARAM NAME="CODE"      VALUE="oracle.forms.engine.Main">
  <PARAM NAME="CODEBASE"  VALUE="/forms60code/">
  <PARAM NAME="ARCHIVE"   VALUE="/forms60code/f60all.jar">
  <PARAM NAME="type"
    VALUE="application/x-jinit-applet;version=1.1.7.18">
  <PARAM NAME="serverPort" VALUE="9000">
  <PARAM NAME="serverArgs" VALUE="module=c:\temp\nolog">
  <PARAM NAME="serverApp"  VALUE="default">
</OBJECT>
</BODY>
</HTML>
```

The highlighted sections are the only things which need to be changed for a basic installation of Developer Forms 6³.

| Tag | What is it? | Comments |
|----------|--|--|
| classid | This is the classid of the application which is being loaded. Explorer will use this id to determine which plug-in is needed to support this application, and will then attempt to load it | The value will depend on the version of JInitiator installed. See the Oracle JInitiator readme.htm file for the correct value for the release of JInitiator you are using. |
| Codebase | If Explorer cannot find a plug-in which can handle this type of application, the user will be able to download a copy of the | If you have installed Oracle JInitiator manually, this entry may be ignored for now. Do not confuse this with the entry PARAM NAME="CODEBASE" |

³ In earlier versions of Forms, the CODE parameter is different to that shown. See the example HTML provided with the software for the correct value.

| | | |
|-------------------------|---|---|
| | plug-in (Oracle JInitiator) from this URL. | |
| PARAM NAME="CODEBASE" | This entry points to the root directory of the Java class files on the server. | This is the virtual directory which you defined in step 2.2.3 |
| PARAM NAME="ARCHIVE" | Tells the JVM where to find JAR files. | This could be a separate virtual directory. However, in a standard Forms install, the jar files are in the same location as the class files. |
| PARAM NAME="type" | This is the Mime type of the application which is being loaded. | |
| PARAM NAME="serverPort" | The port number used by the applet to communicate with the Forms Listener. | The default value is 9000 The value here must match the port number used by the Listener. |
| PARAM NAME="serverArgs" | The command line passed by the Forms Listener to the Forms Web Runtime process. | The restrictions on its syntax are the same as those when running if60run or f60runm on the server platform e.g. if FORMS60_PATH is not defined on the Forms Server, you must specify the full filename of the Forms module |

2.4.4 JInitiator with Netscape Navigator

A simple example of a HTML file for use with Netscape Navigator (Netscape) is shown below. The HTML file for Netscape does not refer directly to the plug-in Oracle JInitiator - it simply defines the Mime-type for the application being loaded. The plug-in itself defines what Mime types it can handle, and Netscape recognises which plug-in to use for each Mime type.

```

<HTML>
  <HEAD>
    <TITLE>Oracle Developer Server and Oracle Jinitiator with
    Netscape</TITLE>
  </HEAD>
  <BODY>
    <EMBED      type="application/x-jinit-applet;version=1.1.7.18"
              width="600" height="600"
              java_code="oracle.forms.engine.Main"
              java_codebase="/forms60code/"
              java_archive="f60all.jar"
              serverport="9000"
              serverargs="module=c:\temp\nolog"
              serverapp="default"
              PLUGINSOURCE="http://ukpl5069.uk.oracle.com/jinit_download.htm" >
    <NOEMBED>
  </BODY>
</HTML>

```

The highlighted items are the only things which need to be changed for a basic installation⁴.

| Tag | What is it? | Comments |
|---------------|--|---|
| Type | This is the Mime type of the application which is being loaded. Netscape will attempt to load a plug-in which can support this Mime type | The value will depend on the version of JInitiator installed. See readme.htm in the \doc directory for the correct value. e.g. type="application/x-jinit-applet" This mime type will work with any version of Oracle JInitiator. However, it will not always choose the latest version if you have multiple versions of JInitiator installed. For that reason, it is strongly recommended that you specify the required version of JInitiator. |
| Java_codebase | This entry points to the root directory of the Java class files on the server. | This is the virtual directory which you defined in step 2.2.3 |
| Java_archive | Tells the JVM where to find JAR files. | This could be a separate virtual directory. However, in a standard |

⁴ In earlier versions of Forms, the CODE parameter is different to that shown. See the example HTML provided with the software for the correct value.

| | | |
|--------------|---|---|
| | | Forms install, the jar files are in the same location as the class files. |
| serverport | The port number used by the applet to communicate with the Forms Listener. | The default value is 9000 The value here must match the port number used by the Listener. |
| serverargs | The command line passed by the Forms Listener to the Forms Web Runtime process. | Note that restrictions on its syntax are the same as those when running if60run or f60runm on the server platform e.g. if FORMS60_PATH is not defined on the Forms Server, you must specify the full filename of the Forms module |
| PLUGINSPPAGE | If Netscape cannot find a plug-in which can handle this Mime type, the user will be able to download a copy of the plug-in (Oracle JInitiator) from this URL. | If you have installed Oracle JInitiator manually, this entry may be ignored for now. |

2.4.5 JInitiator with both Netscape Navigator and Internet Explorer

The example file below contains tags for JInitiator for both Netscape and Internet Explorer.

```
<HTML>
<!-- FILE: static_jinit.html -->
<!-- Oracle Static (Non-Cartridge) HTML File Template (Windows NT) -->
<!-- Tags and parameters have been modified for Oracle JInitiator -->

<HEAD><TITLE>Developer Server and Oracle JInitiator</TITLE></HEAD>

<BODY>
<P>
<OBJECT classid="clsid:9F77a997-F0F3-11d1-9195-00C04FC990DC"
        WIDTH=500
        HEIGHT=550
codebase="http://ukpl4901.uk.oracle.com/download/jinitiator/jinit11715.exe#Version=1,1
,7,15">
<PARAM NAME="CODE"      VALUE="oracle.forms.engine.Main" >
<PARAM NAME="CODEBASE"  VALUE="/forms60code/" >
<PARAM NAME="ARCHIVE"   VALUE="/forms60code/f60all.jar" >
<PARAM NAME="type"      VALUE="application/x-jinit-applet">
<PARAM NAME="serverPort" VALUE="6000">
<PARAM NAME="serverArgs" VALUE="module=c:\users\web\forms\f60test">
<PARAM NAME="serverApp"  VALUE="default">
<COMMENT>
```

```

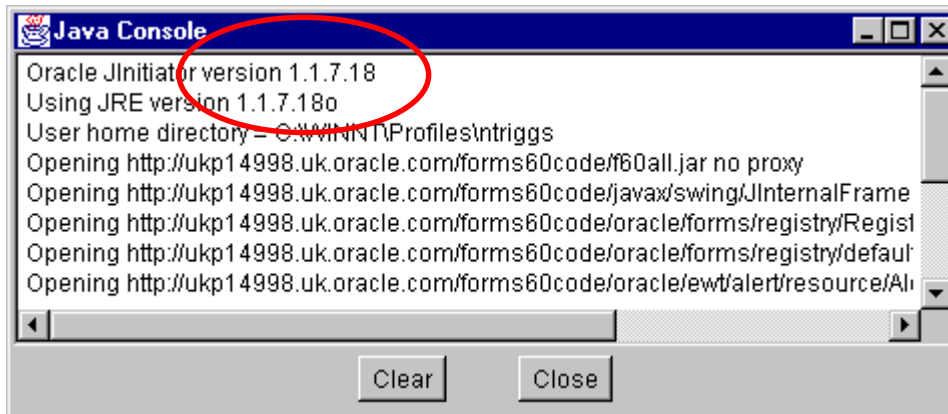
<EMBED type="application/x-jinit-applet"
  java_CODE="oracle.forms.engine.Main"
  java_CODEBASE="/forms60code/"
  java_ARCHIVE="/forms60code/f60all.jar"
  WIDTH=500
  HEIGHT=550
  serverPort="6000"
  serverArgs="module=c:\users\web\forms\f60test"
  serverApp="default"
pluginspage="http://ukp14901.uk.oracle.com/download/jinitiator/jinit_download.htm">
<NOEMBED>
</COMMENT>
</NOEMBED></EMBED>
</OBJECT>
</BODY>
</HTML>

```

2.4.6 What version of Oracle JInitiator is being used by my browser?

When you are running Forms using JInitiator, if you want to see which version of Oracle JInitiator is being used then you must enable the Java Console. See section 3.2.2 for details.

When the page which refers to the JInitiator EMBED or OBJECT tag is browsed, the console will be displayed and the version number will be shown :-



2.4.7 What version of JInitiator is installed in Netscape?

To check what versions of JInitiator have been installed, and which Mime types they support, use Help -> About Plug-Ins from the Netscape menu. You will see output displayed which will include information as follows :

Oracle JInitiator 1.1.7.11 for Netscape Navigator

File name: C:\Program Files\Netscape\Communicator\Program\plugins\NPJinit-11711.dll

Oracle JInitiator 1.1.7.11 for Netscape Navigator with OJDK/JRE 1.1.7.11

| Mime Type | Description | Suffixes | Enabled |
|---|-------------|----------|---------|
| application/x-jinit-applet | Java Applet | class | Yes |
| application/x-jinit-applet;version=1.1.5.3 | Java Applet | class | Yes |
| application/x-jinit-applet;version=1.1.5.21.1 | Java Applet | class | Yes |
| application/x-jinit-applet;version=1.1.7.9 | Java Applet | class | Yes |
| application/x-jinit-applet;version=1.1.7.10 | Java Applet | class | Yes |
| application/x-jinit-applet;version=1.1.7.11 | Java Applet | class | Yes |

Oracle JInitiator 1.1.7.18 for Netscape Navigator

File name: C:\Program Files\Netscape\Communicator\Program\plugins\NPJinit-11718.dll

Oracle JInitiator 1.1.7.18 for Netscape Navigator with OJDK/JRE 1.1.7.18

| Mime Type | Description | Suffixes | Enabled |
|---|-------------|----------|---------|
| application/x-jinit-applet | Java Applet | class | No |
| application/x-jinit-applet;version=1.1.7.18 | Java Applet | class | Yes |

As you can see from the above output, 2 versions of JInitiator are installed. However, if the generic application/x-jinit-applet mime type were specified, the older version of JInitiator would be used as that is the one marked as Enabled.

2.5 HTML FILE PROBLEMS WITH ORACLE JINITIATOR AND INTERNET EXPLORER

2.5.1 java.io.FileNotFoundException which refers to JAR file on startup

This error can occur if the VALUE is wrong in the ARCHIVE tag. Check that the ARCHIVE tag refers to the correct virtual directory, and that this virtual directory is configured correctly. Also check that the name of the JAR file is correct. If the ARCHIVE tag is as follows :-

```
<PARAM NAME="ARCHIVE" VALUE="/forms60code/f60all.jar">
```

Use your browser to attempt to browse to :-

```
http://yourserver/forms60code/f60all.jar
```

If this results in *HTTP/1.0 404 Object Not Found* then there is a problem with the configuration of your virtual directory.

Sample output when /forms60codes/ is used instead of /forms60code

```
Opening http://ukpl4998.uk.oracle.com/forms60codes/f60all.jar
proxy=emeacache.uk.oracle.com:80
Unable to contact http://ukpl4998.uk.oracle.com/forms60codes/f60all.jar
Opening http://ukpl4998.uk.oracle.com/forms60codes/f60all.jar
proxy=emeacache.uk.oracle.com:80
java.io.FileNotFoundException:
http://ukpl4998.uk.oracle.com/forms60codes/f60all.jar
    at sun.net.www.protocol.http.HttpURLConnection.getInputStream(Compiled
    Code)
    at
    sun.net.www.protocol.http.HttpURLConnection.openConnectionCheckRedirects(
    Compiled Code)
    at sun.applet.JARCache.beginStoring(JARCache.java:224)
    at sun.applet.AppletResourceLoader.loadJar(AppletResourceLoader.java:184)
    at sun.applet.JinitAppletPanel.loadJarFiles(Compiled Code)
    at sun.plugin.AppletViewer.loadJarFiles(Compiled Code)
    at sun.applet.JinitAppletPanel.runLoader(JinitAppletPanel.java:524)
    at sun.applet.JinitAppletPanel.run(Compiled Code)
    at java.lang.Thread.run(Thread.java:466)
```

2.5.2 Performance on startup slow, many class files loaded but no JAR files

If you omit the ARCHIVE tag or misspell the word ARCHIVE, performance will be poor. This could also occur if a previous tag were missing its trailing '>'.

Sample output with no ARCHIVE tag

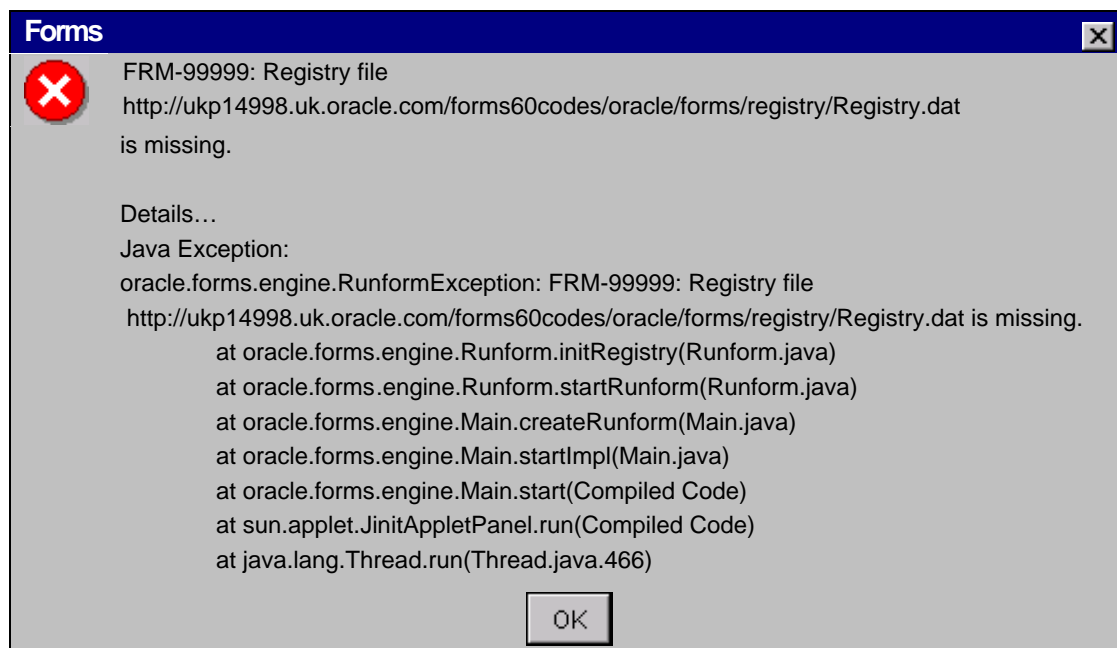
```
Oracle Jinitiator version 1.1.7.18
Using JRE version 1.1.7.180
User home directory = C:\WINNT\Profiles\nttriggs
JAR caching enabled.
    Cache directory: D:\JINIT1~1.18\jcache
    Maximum cache size: 20971520 bytes
Opening http://ukpl4998.uk.oracle.com/forms60code/oracle/forms/engine/Main.class
proxy=emeacache.uk.oracle.com:80
Opening
http://ukpl4998.uk.oracle.com/forms60code/oracle/ewt/lwAWT/BufferedApplet.class
proxy=emeacache.uk.oracle.com:80
Opening
http://ukpl4998.uk.oracle.com/forms60code/oracle/ewt/popup/PopupApplet.class
proxy=emeacache.uk.oracle.com:80
Opening
http://ukpl4998.uk.oracle.com/forms60code/oracle/ewt/popup/PopupOwner.class
proxy=emeacache.uk.oracle.com:80
Opening
```

```
http://ukp14998.uk.oracle.com/forms60code/oracle/ewt/event/tracking/MouseGrabPro
vider.class proxy=emeacache.uk.oracle.com:80
Opening
http://ukp14998.uk.oracle.com/forms60code/oracle/ewt/help/HelpCapable.class
proxy=emeacache.uk.oracle.com:80
Opening
http://ukp14998.uk.oracle.com/forms60code/oracle/ewt/lwAWT/ImmediatePainter.clas
s proxy=emeacache.uk.oracle.com:80
Opening
http://ukp14998.uk.oracle.com/forms60code/oracle/ewt/lwAWT/DoubleBufferer.class
proxy=emeacache.uk.oracle.com:80
Opening
http://ukp14998.uk.oracle.com/forms60code/oracle/ewt/lwAWT/VirtualComponent.clas
s proxy=emeacache.uk.oracle.com:80
...
...
```

2.5.3 FRM-99999: Registry file

http://ukp14998.uk.oracle.com/forms60codes/oracle/forms/registry/Registry.dat is missing.

If the CODEBASE value in the HTML file is incorrect, you will see the following error:



Check the value of the CODEBASE parameter. In the example above, /forms60codes/ is undefined. If you omit the CODEBASE tag entirely, you will get an error message which begins :-

```
FRM-99999: Registry file
http://ukp14998.uk.oracle.com/oracle/forms/registry/Registry.dat
is missing.
```

Note that the missing file is reported as

```
http://ukp14998.uk.oracle.com/oracle/forms/registry/Registry.dat
```

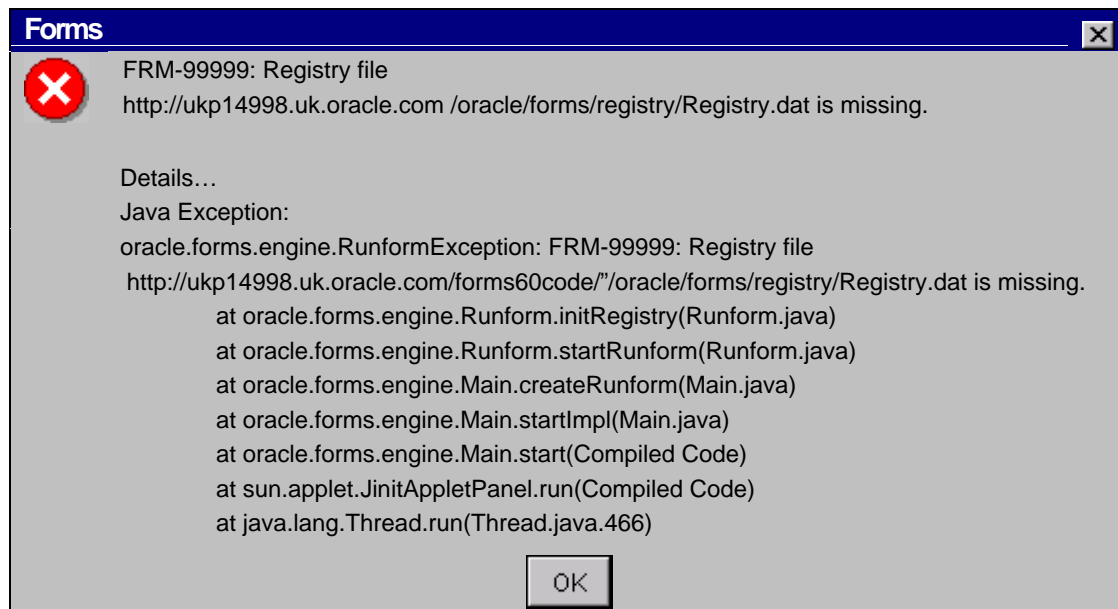
This could lead to confusion as on machine ukp14998.uk.oracle.com, the Developer 6 installation has a directory structure which matches this and the file exists. However, the *URL* mentioned does not point to that file, and if you tried to access that file in a browser you would get the following error :-

```
HTTP/1.0 404 Object Not Found
```

This could also be caused by a syntax error in the CODEBASE tag, such as a missing speech mark after VALUE= e.g.

```
<PARAM NAME="CODEBASE" VALUE="/forms60code/">
```

This will result in the following Forms error :-



2.5.4 File not found error

If there is a syntax error in the HTML file such as missing the trailing quote for the VALUE entry of the CODEBASE line e.g.

```
<PARAM NAME="CODEBASE" VALUE="/forms60code/">
```

the following errors will appear in the Java console :-

```
Opening http://ukpl4998.uk.oracle.com/forms60code/>
  <PARAM NAME=/oracle/forms/engine/Main.class
  proxy=emeacache.uk.oracle.com:80
File not found when looking for: oracle.forms.engine.Main
java.lang.NullPointerException
    at sun.applet.JinitAppletPanel.createApplet(JinitAppletPanel.java:603)
    at sun.applet.JinitAppletPanel.runLoader(JinitAppletPanel.java:525)
    at sun.applet.JinitAppletPanel.run(Compiled Code)
    at java.lang.Thread.run(Thread.java:466)
```

This error can also be caused by a wrong value for the CODE tag for Explorer e.g.

```
<PARAM NAME="CODE"          VALUE="Oracle.forms.engine.Main">
```

or the java_code tag for Netscape e.g.

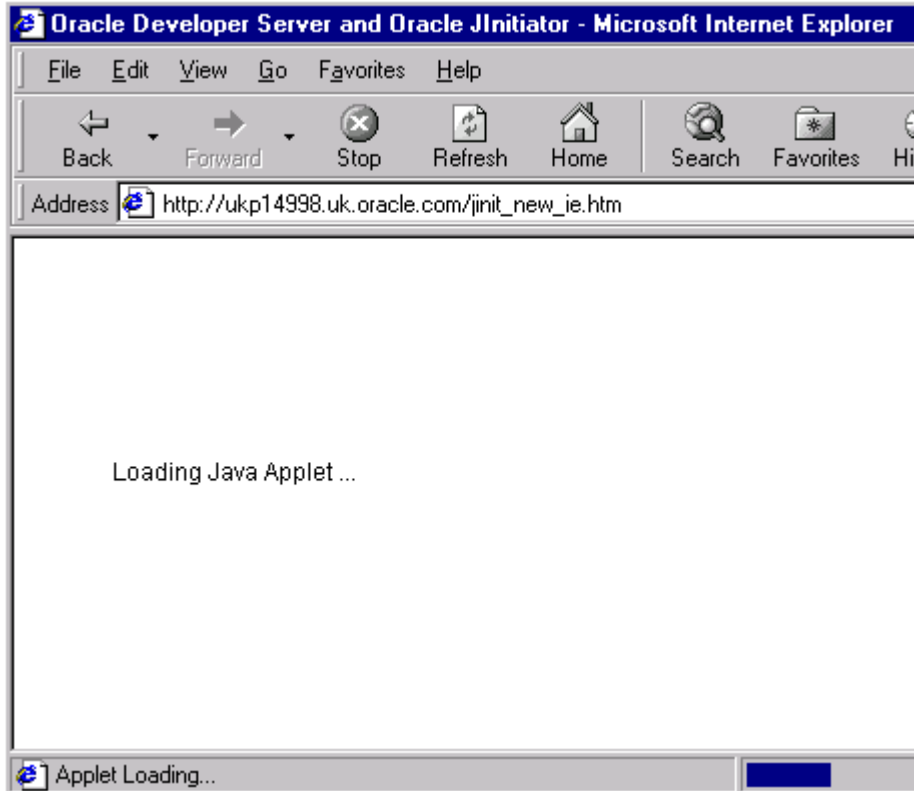
```
java_code="Oracle.forms.engine.Main"
```

Note that the value in the CODE and java_code tags is case sensitive (oracle should be lower case above) :-

```
Opening http://ukpl4998.uk.oracle.com/forms60code/Oracle/forms/engine.Main
proxy=emeacache.uk.oracle.com:80
File not found when looking for: Oracle.forms.engine
java.lang.NullPointerException
    at sun.applet.JinitAppletPanel.createApplet(JinitAppletPanel.java:603)
    at sun.applet.JinitAppletPanel.runLoader(JinitAppletPanel.java:525)
    at sun.applet.JinitAppletPanel.run(Compiled Code)
    at java.lang.Thread.run(Thread.java:466)
```

2.5.5 Browser hangs with message 'Loading Java Applet...'

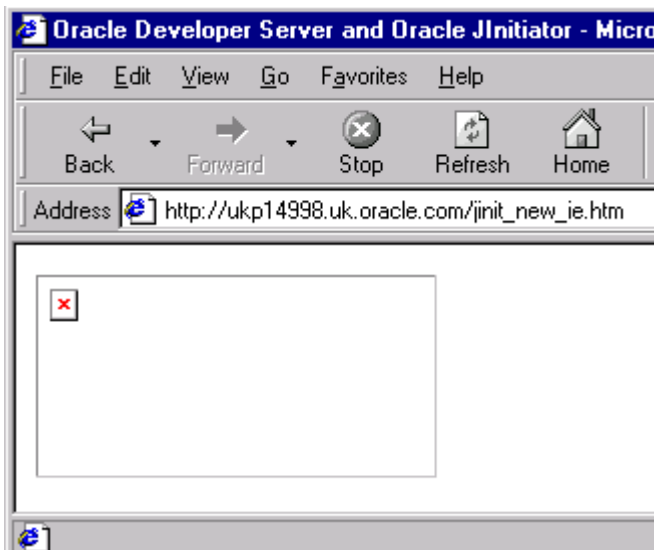
When trying to load the HTML file which runs the Forms applet, the browser appears to hang with the message 'Loading Java Applet ...' and the status line 'Applet Loading...' :-



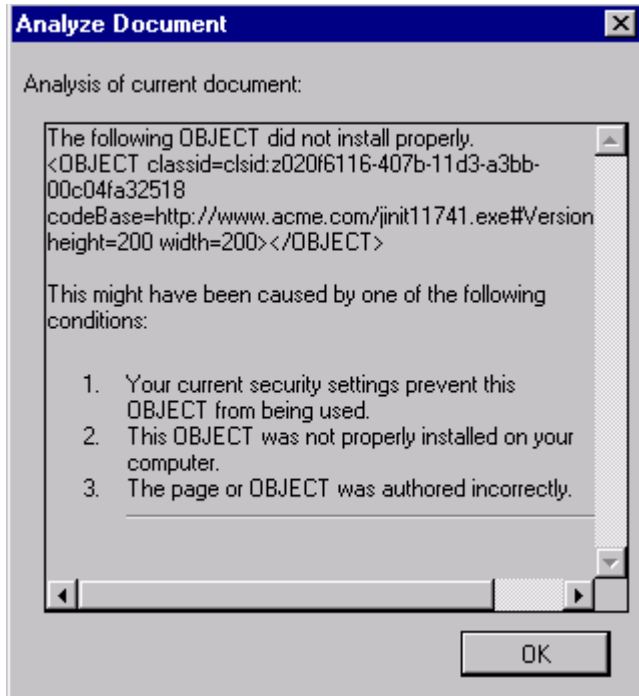
Check that the CODE parameter exists and is correctly defined.

2.5.6 Applet does not appear, only an empty box with a red cross

If the classid is incorrect, you will get the following red cross on the screen :-



Selecting File -> Properties from the menu, and then the Analyze button will show the following :-



2.5.7 Do I need to exit and restart my browser?

If you get a problem when using JInitiator which prevents the Forms applet from loading completely, you may need to exit your browser before retrying the operation.

For example, when attempting to run a Form without starting the Forms Server Listener first, the resulting error is :-

```
FRM-99999: Failed to connect to the Server
```

If this page is reloaded in the browser, the error is no longer reported. Also, if the Server is now started, the Form will still not load correctly until your browser is restarted. Note also this it is necessary to exit entirely from the browser - simply closing the window containing the Forms applet will not be adequate if another browser window is left open.

2.6 HTML FILE PROBLEMS WITH ORACLE JINITIATOR AND NETSCAPE NAVIGATOR

2.6.1 NullPointerException in getHashKey loading applet

If you omit the `java_code` tag (or misspell the tag name), you will get an error in the console window.

Missing `java_code` tag on Netscape

```
Oracle Jinitiator version 1.1.7.18
Using JRE version 1.1.7.18o
User home directory = C:\WINNT\Profiles\ntriggs
JAR caching enabled.
```

```
Cache directory: D:\JINIT1~1.18\jcache
Maximum cache size: 20971520 bytes
java.lang.NullPointerException
    at sun.applet.JinitAppletPanel.getHashKey(JinitAppletPanel.java:777)
    at
sun.applet.JinitAppletPanel.isAppletInCache(JinitAppletPanel.java:695)
    at sun.plugin.AppletViewer.init(AppletViewer.java:334)
    at
sun.plugin.navig.win32.AppletPluginPanel.startPanel(AppletPluginPanel.java:2
12)
    at
sun.plugin.navig.win32.PluginObject.startPlugin(PluginObject.java:132)
    at
sun.plugin.navig.win32.PluginObject.setDocumentURL(PluginObject.java:210)
```

2.6.2 Unable to contact http://.../xxx.jar and FileNotFoundException in getInputStream

On Netscape, an incorrect value for the java_archive entry will cause problems. For example, setting the following :-

```
java_archive="xf60all.jar"
```

The same symptoms will result from omitting the java_codebase tag, or setting it to an incorrect virtual directory.

Sample output

```
Oracle JInitiator version 1.1.7.18
Using JRE version 1.1.7.18o
User home directory = C:\WINNT\Profiles\nttriggs
JAR caching enabled.
  Cache directory: D:\JINIT1~1.18\jcache
  Maximum cache size: 20971520 bytes
Opening http://ukpl4998.uk.oracle.com/forms60code/xf60all.jar no proxy
Unable to contact http://ukpl4998.uk.oracle.com/forms60code/xf60all.jar
Opening http://ukpl4998.uk.oracle.com/forms60code/xf60all.jar no proxy
java.io.FileNotFoundException:
http://ukpl4998.uk.oracle.com/forms60code/xf60all.jar
    at sun.net.www.protocol.http.HttpURLConnection.getInputStream(Compiled
Code)
    at
sun.net.www.protocol.http.HttpURLConnection.openConnectionCheckRedirects(Com
piled Code)
    at sun.applet.JARCache.beginStoring(JARCache.java:224)
    at
sun.applet.AppletResourceLoader.loadJar(AppletResourceLoader.java:184)
    at sun.applet.JinitAppletPanel.loadJarFiles(Compiled Code)
```

```
at sun.plugin.AppletViewer.loadJarFiles(Compiled Code)
at sun.applet.JinitAppletPanel.runLoader(JinitAppletPanel.java:524)
at sun.applet.JinitAppletPanel.run(Compiled Code)
at java.lang.Thread.run(Thread.java:466)
```

2.6.3 Blank page on Netscape when trying to run Forms with JInitiator

If the 'type' tag is omitted from the EMBED tag, a blank page will be displayed in the browser.

2.6.4 Blank page in Netscape, many class files loaded

If the java_archive tag is omitted, JInitiator will begin to load lots of individual class files, but will then the browser will hang with a blank screen. (bug 919679)

```
Oracle JInitiator version 1.1.7.18
Using JRE version 1.1.7.18o
User home directory = C:\WINNT\Profiles\nttriggs
JAR caching enabled.
  Cache directory: D:\JINIT1~1.18\jcache
  Maximum cache size: 20971520 bytes
Opening
http://ukp14998.uk.oracle.com/forms60code/oracle/forms/engine/Main.class  no
proxy
Opening
http://ukp14998.uk.oracle.com/forms60code/oracle/ewt/lwAWT/BufferedApplet.cl
ass no proxy
Opening
http://ukp14998.uk.oracle.com/forms60code/oracle/ewt/popup/PopupApplet.class
no proxy
Opening
http://ukp14998.uk.oracle.com/forms60code/oracle/ewt/popup/PopupOwner.class
no proxy
.
.
.
Opening
http://ukp14998.uk.oracle.com/forms60code/oracle/ewt/graphics/ImageUtils.cla
ss no proxy
```

2.6.5 Blank page in Netscape, Java console reports versions and user home directory

If either of the 'width' or 'height' tags are omitted, Netscape will display a blank page and the console output will be as follows :-

```
Oracle JInitiator version 1.1.7.18
```

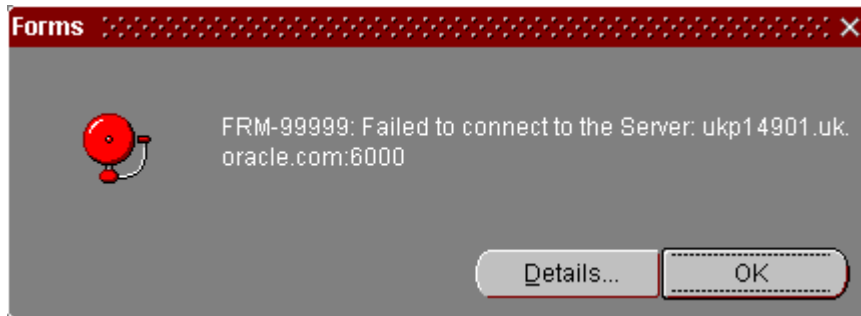


```
Using JRE version 1.1.7.18o  
User home directory = C:\WINNT\Profiles\ntriggs
```

3. FRM-99999 ERRORS

In Forms version 4.5.x, 5.x and 6.0.x, when running a Form over the Web you may encounter the error FRM-99999. FRM-99999 is a generic error message, and is used to indicate that the forms applet is no longer able to communicate with the forms server.

For example:



FRM-99999 can be raised by several different conditions. When you receive this error, you will need to obtain more information to resolve it. This section will introduce the diagnostic tools at our disposal and how to use them to resolve the problem.⁵

3.1 WHAT CAUSES FRM-99999?

The causes of FRM-99999 can be split into several categories:

- configuration problems
 - Some FRM-99999 errors are raised by configuration problems. For example, the forms server is not started, or is listening on a different port to that specified in the HTML file. Typically these errors will reproduce consistently.
- Forms server process has crashed
 - The majority of FRM-99999 errors that occur after a successful connection has been established and the form started, are due to the server crashing. Once the server process has died, then the client can no longer continue - the applet has no life of its own, and it cannot continue to run without being able to communicate with the server process.
 - These errors are often difficult to diagnose: the problem may not reproduce consistently, or the user may be unaware of the sequence of events that led to the crash.

⁵ In Developer 6*i*, many instances which previously raised the generic message FRM-99999 now raise more specific error messages. See Section 3.3.3 for details.

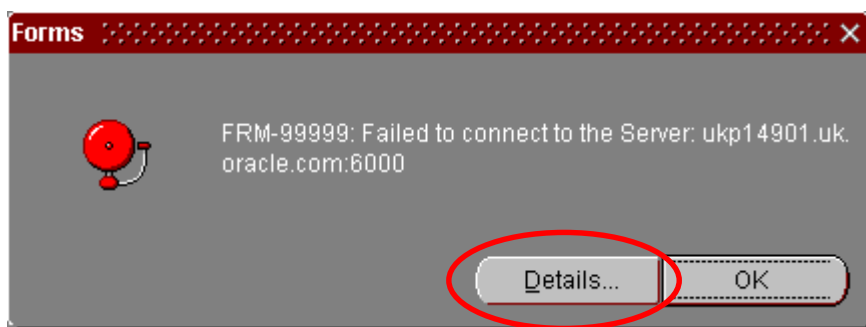
- The communication between the applet and the Forms Server process has experienced network problems, and the communication has been broken.
- Known bugs
 - A list of known causes of FRM-99999 is included in Section 3.3.

3.2 RESOLVING FRM-99999

3.2.1 What does the applet show ?

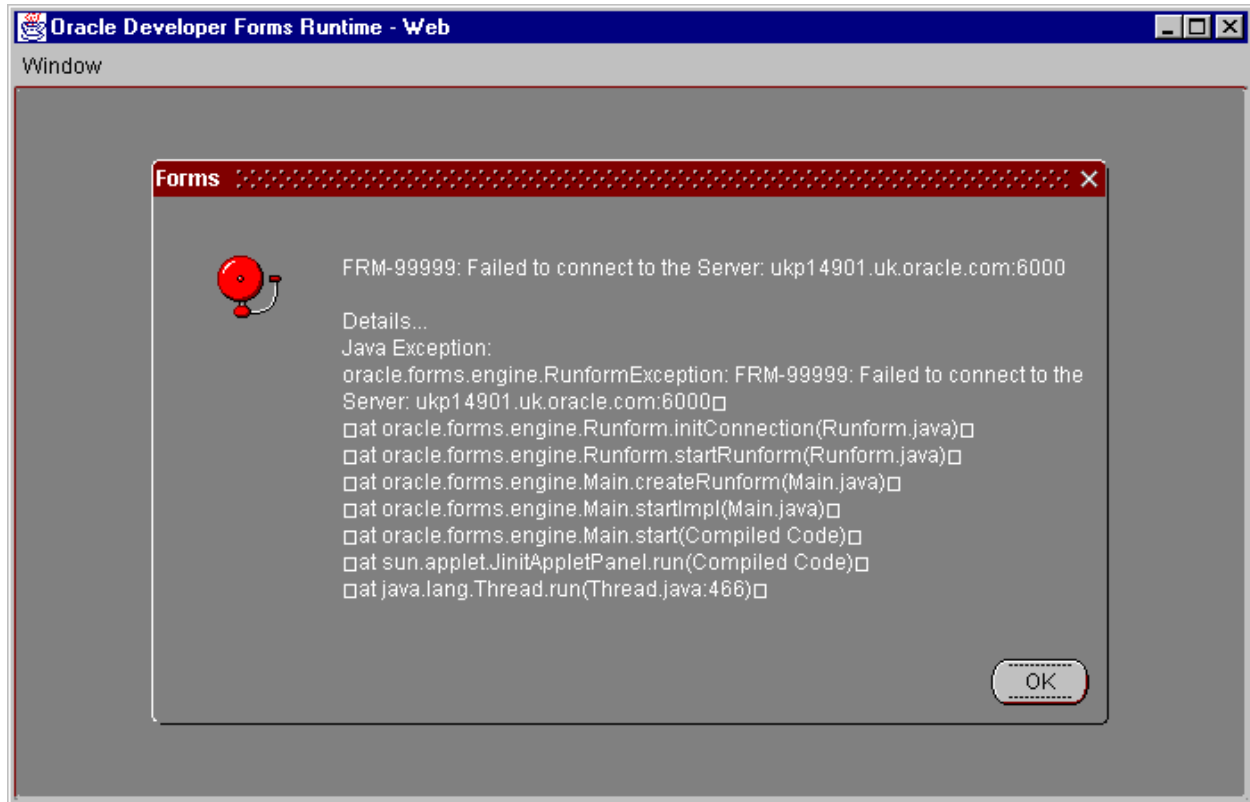
Often, everything required to identify the cause of the FRM-99999 is contained in the error reported by the Forms applet.

When a FRM-99999 is raised, the error dialog will have a 'Details' button.



Pressing the 'Details' button will show the current Java stack. The exact stack is dictated by the root cause *and* the release of Forms. This is due to the differing package structure used for the applet class files in the different releases.

Pressing this button will show the Java error stack.



For example, the following stack traces are both generated by the same root cause - the Forms Server is not started. However, they generate different stacks:

Forms 4.5

```

Java Exception:
oracle.forms.uiclient.v1_4.engine.RunformException:FRM-99999:failed to connect
to the server:cxlewis-pc:4500
at java.lang.Throwable.<init>(compiled code)
at java.lang.Exception.<init>(Compiled code)
at oracle.forms.uiclient.v1_4.engine.RunformException.<init>(compiled code)
at oracle.forms.uiclient.v1_4.engine.Runform.initConnection(compiled code)
at oracle.forms.uiclient.v1_4.engine.Runform.startRunform(Compiled code)
at oracle.forms.uiclient.v1_4.engine.Runform.run(Compiled code)
at java.lang.Thread.run(Compiled code)

```

Forms 6.0

```

Java Exception:
oracle.forms.engine.RunformException: FRM-99999: Failed to connect to the
Server: ukp14901.uk.oracle.com:6001
at oracle.forms.engine.Runform.initConnection(Runform.java)
at oracle.forms.engine.Runform.startRunform(Runform.java)
at oracle.forms.engine.Main.createRunform(Main.java)

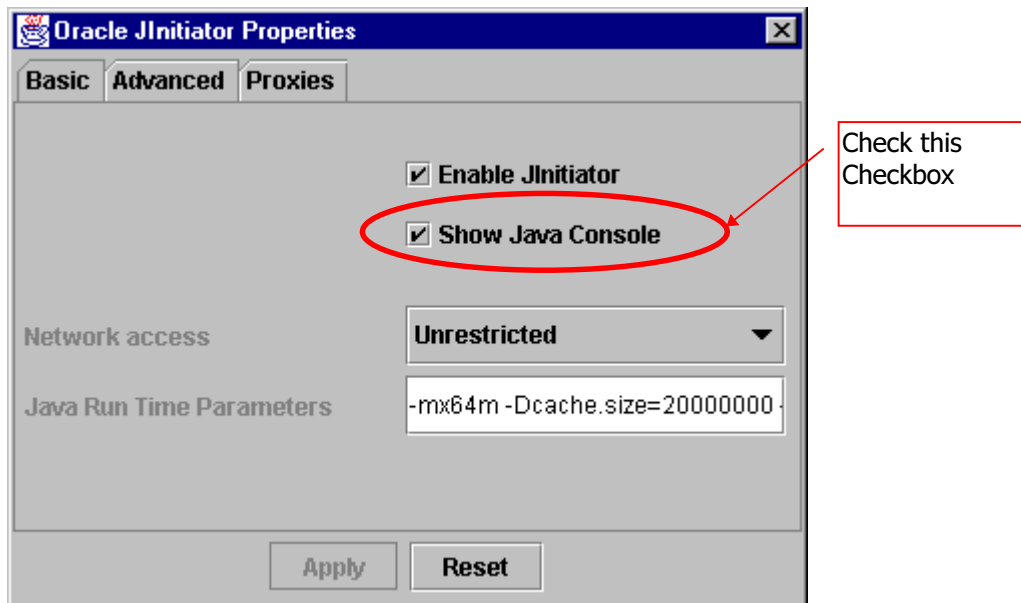
```

```
at oracle.forms.engine.Main.startImpl(Main.java)
at oracle.forms.engine.Main.start(Compiled Code)
at sun.applet.AppletPanel.run(Compiled Code)
at java.lang.Thread.run(Thread.java:466)
```

3.2.2 What does the Java console show ?

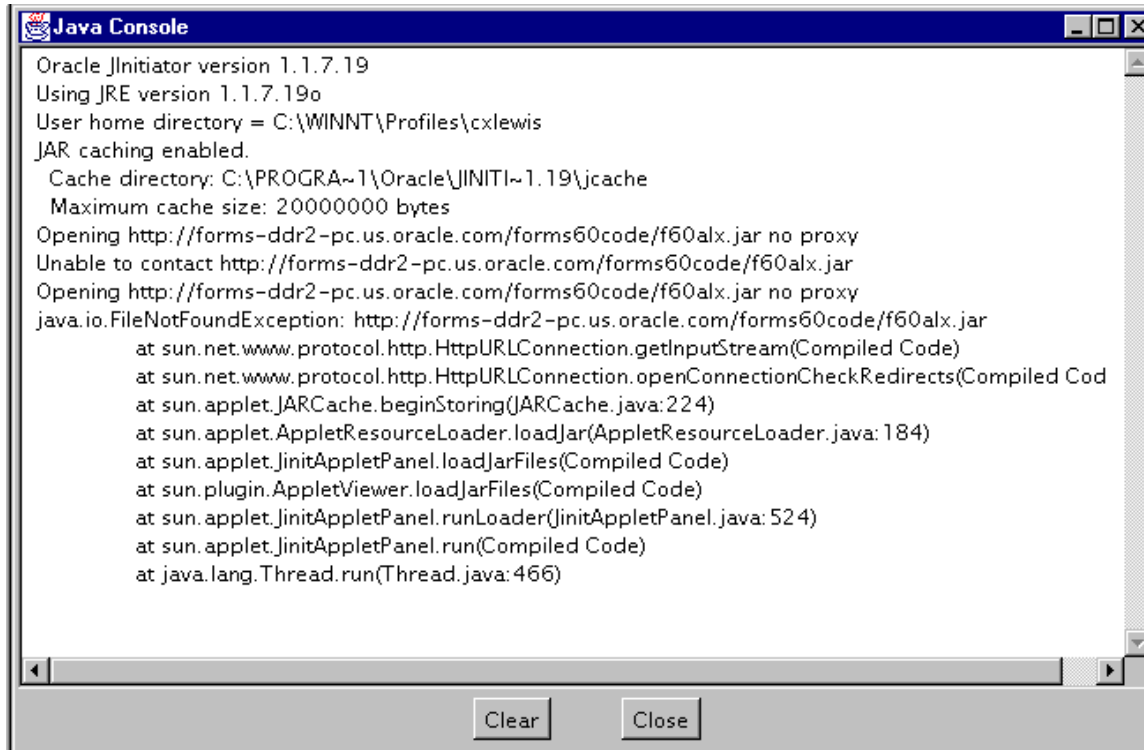
If you are running Appletviewer, then any Java errors will be reported directly to the console session you started Appletviewer from.

If you are using JInitiator and a Java error is encountered then the error will be written to the browser status line. However, this does not show the full error stack. To see the full error stack, you need to turn the Java Console on. This is done from the JInitiator Control Panel



The JInitiator Control Panel can be found on the 'Start->Programs' option in Windows

Note: You will need to re-start your browser for this to take effect.



3.2.3 Did the server recognise the incoming connection ?

This question is particularly important when the FRM-99999 error is raised on initial startup of the client. When the Forms applet starts, it attempts to communicate with the Forms Listener. The applet will determine how to attempt this communication based on parameters in the HTML file it was started from.

Depending on the communication method you have chosen as your deployment protocol, different parameters will be used to construct the port to talk to

| | |
|-------------|---|
| connectMode | <p>Tells the applet which communication method to use for communication with the Forms Server.</p> <p>Valid values are:</p> <p>HTTP - uses HTTP 1.1</p> <p>HTTPS - HTTPS with SSL for secure connections over the Internet</p> <p>socket - the default connection method.</p> |
| ServerHost | <p>Tells the applet which machine to send messages to.</p> <p>If this is not specified then it will default to the server specified in the URL</p> |

| | |
|------------|--|
| serverPort | Tells the applet which TCP/IP port to send messages to. It is essential that the PORT parameter in the HTML file matches the port the server is listening on. If this was not specified when the forms server was started then it will default to 9000. |
|------------|--|

For example, imagine a HTML file with the following parameters defined

```
PORT=4500  
SERVERHOST=cxlewis-sun.us.oracle.com
```

This would result in the forms applet attempting to contact a Forms server listening on port 4500 on the machine cxlewis-sun.us.oracle.com.

It is important to understand that the Forms Listener and the Web Server listener are two entirely different and unrelated processes. Typically, a Web Server will listen on port 80 or 81. If you try to start a Forms Listener on a port that is already in use, it will fail with the error “FRM-60009: Network bind error; probably the port is already in use”.



You can determine whether the forms listener is able to process incoming requests using the following command

```
telnet <machine name> <port number>  
eg.
```

```
telnet ukp14901.uk.oracle.com 6000  
Trying 138.3.65.126...  
Connected to ukp14901.uk.oracle.com.  
Escape character is '^]'.  
^C
```

If the port is open, you will be connected, and you can then use “^]” followed by quit to exit. This will not specifically check that the port is opened the Forms Listener, but it will show that something is keeping that port open. It could be a Web Server, for example.

If the port is not open, you will receive an error. This indicates that either the Forms Listener is not listening on that port number, or that the port has closed.

```
telnet ukp14901.uk.oracle.com 6001
```

```
Trying 138.3.65.126...
telnet: Unable to connect to remote host: Connection refused
```

Alternatively, stop the Forms Listener and restart it specifying the parameter

```
PORT=<port number to listen on>
```

The most straightforward way to establish whether the Forms Server saw the connection request is to switch on Forms Server Connection Activity Logging.

For details on Forms Server Connection Activity Logging, including how to switch it on, and what output to expect, see Appendix A.

If the server appears to recognise the incoming connection, but is unable to spawn the process to run the form on the server, then it is possible to bypass the Forms Listener altogether, and have the applet communicate directly with a pre-spawned Forms Runtime process.

3.2.4 Starting a dedicated Forms process on the server

To understand this procedure it is essential to understand what happens when a user points their browser at a HTML page to try to start a form. The first thing that happens is that the Web Server delivers the HTML file to the browser. The browser processes the file and discovers that there is a tag in it telling it either to start an applet or to start JInitiator. If the file contains JInitiator tags then JInitiator starts, and it tries to start the applet. When the applet starts it tries to initiate a conversation with the Forms Server. The information provided in the HTML file tells the client which machine to talk to, what protocol to use for the conversation, and what port to send its messages to. The Forms Listener process acts as a broker, listening on its designated port for incoming connections. When it receives a connection request, it spawns a Forms Runtime Engine process to run the form. Remember that only the user interface runs in the browser - the processing of the form is performed on the server. The spawned process must also open a port to communicate with the client, and this port number is assigned from the pool of available free ports. The applet is then 'redirected' to communicate with the new process using the newly opened port. In this way, the Forms Listener process does not become a bottleneck for passing messages between the various clients and their corresponding server processes.

By starting a dedicated Forms process on the server, we bypass the listener altogether. By taking away one more component, our understanding of where the problem lies may improve.

Note: This procedure does not provide a way of bypassing the Forms Listener for normal operation. It should be used only to help diagnose a problem.

To start the process, we run the Forms Web runtime process with the parameter

webfile=-<port number>

| Operating System | Version | Command |
|------------------|-----------|------------------------|
| Windows | FORMS 4.5 | f45web32 webfile=-4501 |
| Windows | FORMS 5.0 | f50web32 webfile=-5001 |
| Windows | FORMS 6.0 | ifweb60 webfile=-6001 |
| UNIX | FORMS 4.5 | f45webm webfile=-4501 |
| UNIX | FORMS 5.0 | f50webm webfile=-5001 |
| UNIX | FORMS 6.0 | f60webm webfile=-6001 |

The HTML file being used must be altered to use the port number specified when starting the dedicated process.

Note: The server process will terminate when the client disconnects.

3.2.5 Did the server crash ?

If you did not have logging switched on, then if the Forms Server process terminated unexpectedly, the process will have written a stack trace to a file in the directory the forms server was started in⁶. This dump file will be called:

<forms_runtime_process>_dump_<process id>

| Operating System | Version | Filename |
|------------------|-----------|---------------------|
| Windows | FORMS 4.5 | f45run32_dump_12345 |
| Windows | FORMS 5.0 | f50run32_dump_12345 |
| Windows | FORMS 6.0 | ifrun60_dump_12345 |
| UNIX | FORMS 4.5 | f45runm_dump_12345 |
| UNIX | FORMS 5.0 | f50runm_dump_12345 |
| UNIX | FORMS 6.0 | f60runm_dump_12345 |

See Section 4 for details.

3.3 KNOWN CAUSES OF FRM-99999

The following table lists known causes of FRM-99999, and what to do to resolve the problem.

⁶ Currently, this does not work. This has been logged as bug 1211961. If the Forms Server is running on UNIX, then setting the environment variable FORMSxx_CATCHTERM=0 (e.g. FORMS60_CATCHTERM) will force the Forms Runtime process to write a core file on the server. This core file can be interrogated using a debugging tool such as dbx, and a stack trace obtained.

3.3.1 Configuration Issues

| | |
|------------------|--|
| Problem | FRM-99999 Using F60LOV.JAR as Archive parameter |
| Solution | use F60ALL.JAR or F60SPLASH.JAR |
| Reference | Bug:548499 |
| Problem | WEBFORMS: APPLVIEWER WILL NOT RUN FORM IF IT CONTAINS A BOILERPLATE IMAGE |
| Solution | Set FORMS60_OUTPUT and FORMS60_MAPPING in the registry |
| Reference | Bug:548499 |
| Problem | Approximately 80 users are able to connect to the server at one time. At some point after the 80th user connects, subsequent users see FRM-99999 ... Failed to connect to server port 9000. On the server machine you see the error: "FAILED INITIALIZATION ON COMCTL32.DLL" |
| Solution | Create additional Web listener ports/ forms server ports and ensure that not all users are attempting to use the same combination of ports. |
| Problem | FRM-99999 No Applets were started. |
| Solution | JAR file is corrupt or empty |
| Reference | Solution 2089108.6 |

3.3.2 Known Bugs

| | |
|------------------|---|
| Problem | FRM-99999 CANNOT CONNECT TO FORMS SERVER <FORMS SERVER>:9000 WITH POOL > 0 |
| Solution | start with pool=0 |
| Reference | Fixed in 4.5.10.14.1 BUG:701797 |
| Problem | WEBFORMS: FIND IN LONG LOV GIVES ERROR FRM-99999: NETWORK ERROR |
| Solution | fixed: 4.5.10.1 |
| Reference | BUG:609004 |
| Problem | STARTING THE FORMS SERVER (F45CTL) WITHOUT DISPLAY SET WILL CAUSE FRM-99999 |
| Solution | Set the DISPLAY to a valid display. Do not use a Character mode terminal, and do not leave DISPLAY blank. Fixed: 4.5.10.9.1 |
| Reference | BUG:547459 |
| Problem | RUN_PRODUCT FAILS WITH FRM-99999 FRM-99999:A network error occurred, the client will not be able to continue. Details... Java Exception: java.io.EOFException |

| | |
|--------------------------------------|---|
| Solution Reference | Fixed: 4.5.9.0 BUG:662478 |
| Problem Solution Reference | WEBFORMS: FORMS60_TIMEOUT VALUE IS INVALID AND CAUSES FRM-99999 Fixed: 6.0.6.3 BUG:811390 |
| Problem Solution Reference | TAB CANVAS WITH POPUP MENU - WEB PREVIEWER CRASHES ON RIGHT MOUSE CLICK OFM0899 Fixed: 7.0.0 BUG:958270 |
| Problem Solution Reference | CRASH CLICKING MOUSE ON TAB CANVAS & WHEN-MOUSE-CLICK TRIGGER Fixed: 6.0.6.4 BUG:714903 |
| Problem Solution Reference | PATCH 8: SOLARIS SERVER NOT COMPATIBLE WITH NT CLIENT FRM-99999: A network error occurred, the client will not be able to continue. Details... Java Exception: java.lang.ClassCastException: java.lang.Object at oracle.forms.uiClient.v1_4.ui.ImageItem.OnUpdate(ImageItem.java) at oracle.forms.uiClient.v1_4.ui.UiCommon.onUpdate(Compiled Code) at oracle.forms.uiClient.v1_4.engine.Runform.onUpdateHandler Fixed: 4.5.10.14.1 BUG:968665 |
| Problem Solution Reference | WEBFORMS:DELETE_TREE_NODE AFTER POPULATING RECORD GROUP RESULTS IN FRM-99999 Fixed: 6.0.8.3.0 BUG:989310 |
| Problem Solution Reference | ASSERTION FAILED WHEN USING RUN_PRODUCT(GRAPHICS) Get a 'FRM-99999 Network Error' on the client and the error "Assertion failed: FALSE, file uisf.c, line 107" occurs on Forms Server when run RUN_PRODUCT(GRAPHICS, 'emp', SYNCHRONOUS, BATCH, FILESYSTEM, pl_id, 'DEPT.CHART_EMP') The applet hangs Fixed: 5.0.6.20.0 BUG:839418 |
| Problem Solution Reference | FRM-99999 WHEN CALLING FTREE.SET_TREE_SELECTION FOR NON VISIBLE NODE Fixed: 6.0.6.4 BUG:883818 |
| Problem Solution Reference | CLEAR_LIST GIVES FRM-99999 WITH TLIST IN FORMS 5.0 Fixed: 6.x BUG:868487 |
| Problem | WEBFORMS: FRM-99999 WHEN STARTING APPLICATION IN SEPARATE FRAME When starting the form in the Web using separateFrame=yes, you get the following error message: FRM-99999: A network error occurred, the client will not be able to continue. Details... |

| | |
|----------------|---|
| | <pre> Java Exception: java.lang.NullPointerException at oracle.forms.handler.UICCommon.moveAbove(UICCommon.java:2322) at oracle.forms.handler.UICCommon.onComponentUpdate(UICCommon.java:1409) at oracle.forms.handler.UICCommon.onUpdate(UICCommon.java:941) </pre> |
| Solution | Fixed: 6.0.5.0.3 |
| Reference | BUG:862193 |
| Problem | <pre> oracle.forms.engine.RunformException:FRM-99999: Internal Error: Failed to find Java Class. Full Details: FRM-99999: Internal Error: Failed to find name of class for handleClassId=0. Missing in registry? at oracle.forms.engine.Runform.onCreateHandler(Runform.java) at oracle.forms.engine.Runform.processMessage(Runform.java) at oracle.forms.engine.Runform.processSet(Runform.java) at oracle.forms.engine.Runform.onMessageReal(Runform.java) at oracle.forms.engine.Runform.onMessage(Runform.java) at oracle.forms.handler.DialogThread.doUserAlert(DialogThread.java:449) at oracle.forms.handler.DialogThread.run(DialogThread.java:369) at java.lang.Thread.run(Thread.java:474) </pre> |
| Solution | Fixed: 6.0.5.33 |
| Reference | BUG: 840668 |

3.3.3 FRM-99999 Errors in Forms 6.0

From Forms6 source:

| | | |
|---|-------------|---|
| 1 | Error | "FRM-99999: Internal Error: Failed to access Java Class.\nFull Details: {0}" |
| | Description | This is an internal error that occurs when the Java language throws an IllegalAccessException whilst we are trying to load some classfiles. It usually indicates that the system is misconfigured in some way. The [generated] text in Parameter 0 often helps to work out why it occurred. |
| | Parameters | Parameter 0: The [Java generated] reason for the problem. |
| 2 | Error | "FRM-99999: serverArgs parameter is either not set or null" |
| | Description | This is a Client mis-configuration error that occurs when the Applet parameter "serverArgs" is either not present or has a null value. |
| 3 | Error | "FRM-99999: Invalid URL {0} sent to browser with target {1}\nFull Details: {2}" |
| | Description | Indicates that either the URL, or the Browser Target, |

| | | |
|----|-------------|--|
| | Parameters | requested was rejected in some way by the browser. Parameter 0: the requested URL Parameter 1: the requested Browser Target Parameter 2: the [Java generated] reason for the problem. |
| 4 | Error | "FRM-99999: Internal Error: Failed to find name of class for handleClassId={0}. Missing in registry?" |
| | Description | A Client mis-configuration error, due to a missing Java class file and |
| | Parameters | Parameter 0: the numeric handlerClassId that wasn't registered. |
| 5 | Error | "FRM-99999: Internal Error: Failed to find Java Class.\nFull Details: {0}" |
| | Description | A Server mis-configuration error, due to a missing Java class file. This error occurs when the Client requests a Java class that couldn't be located on the Server. |
| | Parameters | Parameter 0: the [Java generated] reason for the problem. |
| 6 | Error | "FRM-99999: Failed to connect to the Server: {0}:{1,number,#####}" |
| | Description | The Client was unable to establish a connection to the Server machine {host} on the designated socket {port}. |
| | Parameters | Parameter 0: the Server host that was contacted. Parameter 1: the Port, or Socket, on the server that was used. |
| 7 | Error | "FRM-99999: Failed to connect to the Server.\nBad machine specification: {0}:{1,number,#####}" |
| | Description | The Client was unable to establish a connection to the Server because the format of the host |
| | Parameters | Parameter 0: the Server host that was contacted. Parameter 1: the Port, or Socket, on the server that was used. |
| 8 | Error | "FRM-99999: Internal Error: Failed to create Java Object.\nFull Details: {0}" |
| | Description | The Client was unable to create a new Object for some reason. The value of Parameter0 may give some indication as to why the error occurred. |
| | Parameters | Parameter 0: the [Java generated] reason for the problem. |
| 9 | Error | "FRM-99999: Failed to execute command.\n Command = {0} {1}\nFull Details: {2}" |
| | Description | Executing an Operating System command, in an attempt to start an external Browser module, caused some problem. |
| | Parameters | Parameter 0: the Executable name requested for the Browser. Parameter 1: the URL Target given to the Browser. Parameter 2: the [Java generated] reason for the problem. |
| 10 | Error | "FRM-99999: An unexpected error occurred." |
| | Description | An Unexpected error occurred. |
| 11 | Error | "FRM-99999: A network error occurred, the client will not be able to continue." |

| | | |
|----|-------------|---|
| | Description | An Unexpected Network error occurred. |
| 12 | Error | "FRM-99999: New passwords do not match. Please make them identical." |
| | Description | In the Change Password dialog, the new password and the retyped passwords do not match. They have to be identical. |
| 13 | Error | "FRM-99999: Registry file {0} is missing." |
| | Description | A Server configuration error that indicates that an important file (the Registry) could not be located by the Client. |
| | Parameters | Parameter 0: the fully specified URL for the Registry file that is missing. |
| 14 | Error | "FRM-99999: Registry is either missing, or contains invalid, entry for Dispatcher.\nFull Details: {0}" |
| | Description | A Server configuration error that indicates that an important file (the Registry) isn't setup. |
| | Parameters | Parameter 0: the [Java generated] reason for the error. |
| 15 | Error | "FRM-99999: Error {0,number,#####} occurred. See the release notes file (relnotes) for information about this error." |
| | Description | See Release notes for details. |
| 16 | Error | "FRM-99999: WEB Client Version too new." |
| | Description | The version of the Client is newer than the version of the Server. |
| 17 | Error | "FRM-99999: WEB Client Version too old." |
| | Description | The version of the Client is older than the version of the Server. |

Table of Forms 6i messages codes for existing errors.

| | | |
|---|-------|---|
| 1 | Error | "FRM-92000: Internal Error: Failed to access Java Class.\nFull Details: {0}" |
| 2 | Error | "FRM-92010: serverArgs parameter is either not set or null" |
| 3 | Error | "FRM-92020: Invalid URL {0} sent to browser with target {1}\nFull Details: {2}" |
| 4 | Error | "FRM-92030: Internal Error: Failed to find name of class for handleClassId={0}. Missing in registry?" |
| 5 | Error | "FRM-92040: Internal Error: Failed to find Java Class.\nFull Details: {0}" |
| 6 | Error | "FRM-92050: Failed to connect to the Server: {0}:{1,number,#####}" |
| 7 | Error | "FRM-92060: Failed to connect to the Server.\nBad machine specification: {0}:{1,number,#####}" |
| 8 | Error | "FRM-92070: Internal Error: Failed to create Java Object.\nFull Details: {0}" |
| 9 | Error | "FRM-92080: Failed to execute command.\n Command = {0} {1}\nFull Details: {2}" |

| | | |
|----|-------|---|
| 10 | Error | "FRM-92090: An unexpected error occurred." |
| 11 | Error | "FRM-92100: Your connection to the Server was interrupted.\nThis may be the result of a network error, or a failure on the Server.\nYou will need to reestablish your session."}, |
| 12 | Error | "FRM-92110: New passwords do not match. Please make them identical." |
| 13 | Error | "FRM-92120: Registry file {0} is missing." |
| 14 | Error | "FRM-92130: Registry is either missing, or contains invalid, entry for Dispatcher.\nFull Details: {0}" |
| 15 | Error | "FRM-92140: Error {0,number,#####} occurred. See the release notes file (relnotes) for information about this error." |
| 16 | Error | "FRM-92150: WEB Client Version too new." |
| 17 | Error | "FRM-92160: WEB Client Version too old." |

New Error Messages in 6i

| | | |
|-----|-------------|---|
| new | Error | FRM-92170: Firewall or proxy server not responding |
| | Description | The connection was dropped because of the firewall |
| new | Error | FRM-92180: We have detected network errors that will adversely affect performance.\nYour firewall or proxy server may not be HTTP 1.1 compliant |
| | Description | The firewall or proxy is the source of the performance hit. |

4. SERVER CRASH

If the Forms Runtime Engine terminated unexpectedly, then the runtime process on the server will have written a stack trace to one of the following places

- Server Connection Activity Log
 - If the forms server was started with connection activity log switched on then the stack trace will have been written to the log file.
 - See Appendix A for details
- File in the working directory
 - This file will have the format

<forms_runtime_process>_dump_<process id>

4.1 WHAT INFORMATION IS IN THE DUMP FILE ?

The dump file contains a stack trace of the running process, and shows the last successful operation performed by Forms.

Example Dump File

```
[Tue Dec 7 15:13:11 1999 PDT]::Client Status [ConnId=0, PID=2220]
  >> ERROR: Abnormal termination of connection, Error Code: 11

FORM/BLOCK/FIELD: X:BLOCK3.STATUS
Last Trigger: WHEN-BUTTON-PRESSED - (In Progress)
Msg: <NULL>
Last Builtin: REPORT_OBJECT_STATUS - (In Progress)
----- Call Stack Trace [ConnId = 0, ProcId = 2220] -----

calling          call      entry          argument values in hex
location         type     point          (? means dubious value)
-----
siehjmpterm()+412  CALL     siehdst()+0    BA1958 ? A8CDB8 ? A8CC4C ?
                  A8CDA4 ? A8CD88 ? EFFF3BA4 ?
sigacthandler()+40  PTR_CALL
                  B ? B6A95C ? B6AB44 ?
                  AE7DEC ? 0 ? E ?
zrcctcx_CheckAuthRe  CALL     zrcctcc_ClientCommo  B ? 0 ? EFFF4758 ? 0 ? 50 ?
                  DB2112 ?
zrcctco_Connect     CALL     zrcctcx_CheckAuthRe  DAE5B8 ? 0 ? 0 ? 0 ?
                  FFFFFFFC0 ? FFFFFFFF0 ?
ibfnjurJobUpdate    CALL     zrcctco_Connect()+0  DAE5B8 ? EFFF4B40 ?
                  EFFF4B38 ? 0 ? 76 ?
```


| | | | |
|----------------|----------|------------------|---|
| | | | FFFF4B40 ? |
| ibfrun()+5940 | CALL | ibfnjurJobUpdate | BE8198 ? BE7BB0 ? BDF328 ? FFFF56A4 ? AE7DEC ? FFFF4B40 ? |
| pfrrun()+23596 | PTR_CALL | | CC4E40 ? AE7DEC ? 0 ? 0 ? BE8198 ? CBC6A8 ? |
| peicnt()+248 | CALL | pfrrun()+0 | CC5720 ? CC4E40 ? 0 ? 1 ? 25 ? BF1858 ? |
| | CALL | | BF1858 ? CC4E40 ? CC4E40 ? BF1858 ? EFFF7B64 ? CBD0C0 ? |
| ipkxcr()+32 | CALL | | 191F7000 ? D396F8 ? 0 ? 0 ? 0 ? FFBF ? |
| ipfrun()+916 | CALL | ipkxcr()+0 | BE8198 ? CBC6A8 ? BE8198 ? 0 ? 0 ? BEEE20 ? |
| ifzgmt()+3260 | CALL | ipfrun()+0 | BE8198 ? 0 ? D396F8 ? D291C8 ? EE7427EC ? 0 ? |
| ifzmkv()+64 | CALL | ifzgmt()+0 | BE7BB0 ? EE7432A0 ? AE7DEC ? BED6D8 ? EE7427EC ? 2000000 ? |
| ifzevl()+80 | CALL | ifzmkv()+0 | BE7BB0 ? BE7BB0 ? A8395C ? AE7DEC ? 7FFFFFFF ? EE7427EC ? |
| ifzevf()+176 | CALL | ifzevl()+0 | BE7BB0 ? BE7BB0 ? AE7DEC ? 80000000 ? B3B460 ? BDC8C8 ? |
| ifzevf()+900 | CALL | ifzevf()+0 | BE7BB0 ? BE7BB0 ? AE7DEC ? 1 ? D29250 ? EE743184 ? |
| ifzerc()+2244 | CALL | ifzevf()+0 | 4000 ? FFFFBF00 ? BE7BB0 ? 4000 ? EE743184 ? EE742B28 ? |
| ifzebk()+1976 | CALL | ifzerc()+0 | 0 ? AE7DEC ? BE7BB0 ? BDC8C8 ? BED6D8 ? EE7427EC ? |
| ifzefm()+316 | CALL | ifzebk()+0 | 0 ? AE7DEC ? BE7BB0 ? 4000 ? FFFEFFFF ? 10000 ? |
| ifzeif()+844 | CALL | ifzefm()+0 | BE7BB0 ? BE9F08 ? BE7BB0 ? AE7DEC ? 0 ? BDC8C8 ? |
| ifzexf()+1188 | CALL | ifzeif()+0 | BE7BB0 ? 0 ? 0 ? 1 ? BED6D8 ? 2000000 ? |
| ifzexe()+164 | CALL | ifzexf()+0 | BE7BB0 ? 1 ? 0 ? BE8198 ? BDC8C8 ? 0 ? |
| ifzman()+5276 | CALL | ifzexe()+0 | 0 ? BEAD68 ? 2 ? BEAB48 ? BE7BB0 ? AE7DEC ? |
| if4mmo()+40 | CALL | ifzman()+0 | 10000000 ? 0 ? 1 ? 0 ? BB2C30 ? 0 ? |
| | PTR_CALL | | 0 ? 0 ? 0 ? 3 ? BB2250 ? CC57C ? |
| | PTR_CALL | | BB12CC ? EFFFFA42C ? 0 ? 2 ? EFFF99D0 ? 0 ? |
| main()+168 | CALL | | 0 ? EFFF9918 ? 3 ? EFFF99CC ? EFFF986C ? EFFF992C ? |
| _start()+220 | CALL | main()+0 | 3 ? EFFF99CC ? EFFF99CC ? |

```
B12400 ? 0 ? 0 ?
```

```
----- End of Call Stack Trace -----
```

In this stack trace, the top two functions `siehjmpterm()` and `sigacthandler()` are the signal handling code - these functions will often be present in the stack trace. To see the function the program was in when the error occurred you need to read further down the stack.

4.2 HOW DOES THIS HELP ?

The stack trace is useful on two different levels

- the information in the stack can be used to identify a known issue. It is not 100% reliable, but an identical stack trace is a good indicator of a matching problem. Even if it is not the same, there may be a workaround or patch for an existing bug that can be tested.
- If the problem is not a known bug, then the stack may provide valuable information to assist development efforts to pinpoint the cause.

4.3 TESTING IN MULTIPLE ENVIRONMENTS

Sometimes a problem may reproduce in one version of Forms, but behave differently in another. Similarly, a problem may not reproduce on a different operating system, or implementation method.

Testing in a different version can prove that a bug is fixed in a later version, or it may provide a different error message and different diagnostic information. A bug is usually logged against a specific version of Forms - you may find that the stack trace in one version does not match any known bugs, but that the stack trace from another version finds an exact match. This could provide you with additional workarounds or information to allow you to avoid the problem altogether.

A particular area to be aware of is running a form from the 'Run Forms Web' button in the Forms Builder. Some problems do not reproduce when running the form over the Web using the normal deployment methods. Although still a genuine issue, it may lessen the perceived impact of the problem if it only affects the development environment. Also, running the form via a HTML file will provide a Java stack if there is a Java error - running from within the builder will not.

Apart from providing solutions or workarounds, the information gathered by running in different environments and versions is valuable information if you need to log a bug.

5. CLIENT CRASH

If the Forms applet disappears unexpectedly, accompanied by a dialog indicating a fatal error, then the Forms applet has crashed. On Windows, a crash will result in the operating system raising an 'illegal operation' dialog, or may cause Dr. Watson to report the error.

To verify the crash, check for a stack trace file on the client. If the client has crashed then a file with the .rpt extension will be created in the same directory as the executable. The root of the filename will be the name of the executable.

E.g.

Using Appletviewer, started from the directory c:\jdk\1_1_7_20\bin, the client stack trace file will be c:\jdk\1_1_7_20\bin\appletviewer.rpt

Using JInitiator, then the executable is considered to be the browser.

| Browser | Filename |
|-------------------|------------------|
| Appletviewer | appletviewer.rpt |
| Netscape | netscape.rpt |
| Internet Explorer | ie4.rpt |

Sometimes the applet may appear to have crashed, but no corresponding .rpt file can be found. In this case it is likely that the Forms Server process has unexpectedly disconnected from the client. The applet will still be running, but it has shutdown all the Forms windows, giving the appearance of a client crash.

5.1 EXAMPLE DUMP FILE

```
=====  
Exception code: C0000005 ACCESS_VIOLATION  
Fault address: 004A5892 01:000A4892  
Module: C:\Program Files\Netscape\Communicator\Program\netscape.exe  
  
System Information:  
Operating System: Windows NT Version 4.0 Build 1381 Service Pack 3  
Date and Time: 4/14/1999, 10:02  
Command line: "C:\Program Files\Netscape\Communicator\Program\netscape.exe"  
  
Registers:  
EAX:00000000  
EBX:006A3FCA  
ECX:00000000  
EDX:FFFFFFFF
```

ESI:0019F61E
EDI:009BFBE0
CS:EIP:001B:004A5892
SS:ESP:0023:0012FE84 EBP:00862F5C
DS:0023 ES:0023 FS:0038 GS:0000
Flags:00010246

Could not find Module32First

Call stack:

| Address | Frame | Logical addr | Module |
|----------|----------|---------------|---|
| 004A5892 | 00862F5C | 0001:000A4892 | C:\Program Files\Netscape\Communicator\Program\netscape.exe |

=====
Exception code: C0000005 ACCESS_VIOLATION

Fault address: 0064D943 01:0024C943

Module: C:\Program Files\Netscape\Communicator\Program\netscape.exe

System Information:

Operating System: Windows NT Version 4.0 Build 1381 Service Pack 3

Date and Time: 12/14/1999, 10:30

Command line: "C:\Program Files\Netscape\Communicator\Program\netscape.exe"

Registers:

EAX:736966A8
EBX:00000001
ECX:009BF8A0
EDX:00000000
ESI:00000000
EDI:009BF8A0
CS:EIP:001B:0064D943
SS:ESP:0023:0012FBA0 EBP:0012FBBC
DS:0023 ES:0023 FS:0038 GS:0000
Flags:00010216

Could not find Module32First

Call stack:

| Address | Frame | Logical addr | Module |
|----------|----------|---------------|---|
| 0064D943 | 0012FBBC | 0001:0024C943 | C:\Program Files\Netscape\Communicator\Program\netscape.exe |
| 0066ECF8 | 0012FBD4 | 0001:0026DCF8 | C:\Program Files\Netscape\Communicator\Program\netscape.exe |
| 5F4039DB | 0012FC04 | 0001:000029DB | C:\WINNT\System32\MFC42.DLL |
| 5F403C5A | 0012FC20 | 0001:00002C5A | C:\WINNT\System32\MFC42.DLL |
| 0066D908 | 0012FC50 | 0001:0026C908 | C:\Program Files\Netscape\Communicator\Program\netscape.exe |
| 5F411E08 | 0012FCA0 | 0001:00010E08 | C:\WINNT\System32\MFC42.DLL |

```
5F411FED 0012FD6C 0001:00010FED C:\WINNT\System32\MFC42.DLL
5F40230B 0012FD8C 0001:0000130B C:\WINNT\System32\MFC42.DLL
5F402294 0012FDEC 0001:00001294 C:\WINNT\System32\MFC42.DLL
5F40221F 0012FE08 0001:0000121F C:\WINNT\System32\MFC42.DLL
5F4021D6 0012FE34 0001:000011D6 C:\WINNT\System32\MFC42.DLL
77E7288D 0012FE50 0001:0000188D C:\WINNT\system32\USER32.dll
77E72918 0012FE70 0001:00001918 C:\WINNT\system32\USER32.dll
006A36E9 0012FE98 0001:002A26E9 C:\Program Files\Netscape\Communicator
\Program\netscape.exe
77E71250 00862F64 0001:00000250 C:\WINNT\system32\USER32.dll
```

5.2 WHAT DOES IT SHOW AND WHY IS IT USEFUL ?

The information contained in the dump file is extremely useful to Oracle development, and should be included in any bug filed to report the problem.

6. APPLICATION HANGING

If the client appears to hang then it is important to verify that the server process is still alive.

If the server process has not crashed, but the client no longer appears to respond to user interaction then the application is said to be hanging.

In such cases a thread dump can point to the deadlock.

6.1 OBTAINING A CLIENT THREAD DUMP

A Java thread dump can only be obtained from Appletviewer.

To obtain the stack dump press CTRL+BREAK in the command prompt (or DOS session) that you started Appletviewer from.

There is currently no way to obtain a Java thread dump from JInitiator.

6.2 EXAMPLE STACK DUMP

```
Full thread dump:
  "AWT-Finalizer" (TID:0xf91c50, sys_thread_t:0x907b40, Win32ID:0x127,
state:W) prio=4
    java.lang.Object.wait(Compiled Code)
    sun.awt.AWTFinalizer.run(Compiled Code)
  "Thread-3" (TID:0xf527e0, sys_thread_t:0x8fce50, Win32ID:0x145, state:CW)
prio=4
    oracle.forms.engine.Heartbeat.run(Compiled Code)
    java.lang.Thread.run(Thread.java:466)
  "Thread-2" (TID:0xf52a80, sys_thread_t:0x8fc630, Win32ID:0xec, state:CW)
prio=4
    java.lang.Object.wait(Compiled Code)
    oracle.forms.engine.FlushQueue.run(Compiled Code)
    java.lang.Thread.run(Thread.java:466)
  "TaskScheduler timer" (TID:0xf8c0d8, sys_thread_t:0x8fae50, Win32ID:0x87,
state:CW) prio=4
    java.lang.Object.wait(Compiled Code)
    oracle.ewt.timer.Timer._waitTilScheduled(Compiled Code)
    oracle.ewt.timer.Timer.run(Compiled Code)
    java.lang.Thread.run(Thread.java:466)
  "Busy indicator" (TID:0xf55d08, sys_thread_t:0x8f7280, Win32ID:0x146,
state:CW) prio=2
    oracle.ewt.timer.Timer.accurateSleep(Compiled Code)
    oracle.ewt.timer.Timer.run(Compiled Code)
    java.lang.Thread.run(Thread.java:466)
```

```

"Forms-StreamMessageWriter" (TID:0xf7f0a0, sys_thread_t:0x8d2770,
Win32ID:0dc, state:CW) prio=4
    java.lang.Object.wait(Compiled Code)
    oracle.forms.net.SingleAccessorQueue.get(Compiled Code)
    oracle.forms.net.StreamMessageWriter.run(Compiled Code)
"Forms-StreamMessageReader" (TID:0xf7fa38, sys_thread_t:0x8d2690,
Win32ID:0111, state:R) prio=4
    java.net.SocketInputStream.read(Compiled Code)
    oracle.forms.net.EncryptedInputStream.fill(Compiled Code)
    oracle.forms.net.EncryptedInputStream.read(Compiled Code)
    java.io.DataInputStream.readUnsignedByte(Compiled Code)
    oracle.forms.engine.Message.readDetails(Compiled Code)
    oracle.forms.engine.Message.readDetails(Compiled Code)
    oracle.forms.net.StreamMessageReader.run(Compiled Code)
"Screen Updater" (TID:0xf8d798, sys_thread_t:0x8c5970, Win32ID:0xef,
state:W) prio=4
    java.lang.Object.wait(Compiled Code)
    sun.awt.ScreenUpdater.nextEntry(Compiled Code)
    sun.awt.ScreenUpdater.run(Compiled Code)
"thread applet-oracle.forms.engine.Main" (TID:0xf7dbd8, sys_thread_t:0x8b85
0, Win32ID:0x136, state:CW) prio=4
    java.lang.Object.wait(Compiled Code)
    sun.applet.AppletPanel.getNextEvent(Compiled Code)
    sun.applet.AppletPanel.run(Compiled Code)
    java.lang.Thread.run(Thread.java:466)
"AWT-Windows" (TID:0xf7a0b0, sys_thread_t:0x8a6ec0, Win32ID:0x13b, state:R)
prio=5
    sun.awt.windows.WToolkit.run(WToolkit.java:98)
    java.lang.Thread.run(Thread.java:466)
"AWT-EventQueue-0" (TID:0xf79e90, sys_thread_t:0x8a6e60, Win32ID:0xfc,
state:CW) prio=5
    java.lang.Object.wait(Compiled Code)
    oracle.forms.net.SingleAccessorQueue.get(Compiled Code)
    oracle.forms.net.StreamMessageReader.get(Compiled Code)
    oracle.forms.net.SocketConnection.get(Compiled Code)
    oracle.forms.engine.Runform.processSet(Compiled Code)
    oracle.forms.engine.Runform.onMessageReal(Compiled Code)
    oracle.forms.engine.Runform.onMessage(Compiled Code)
    oracle.forms.engine.Runform.processEventEnd(Compiled Code)
    oracle.ewt.lwAWT.LWComponent.redispatchEvent(Compiled Code)
    oracle.ewt.lwAWT.LWComponent.processEvent(Compiled Code)
    java.awt.Component.dispatchEventImpl(Compiled Code)
    java.awt.Container.dispatchEventImpl(Compiled Code)
    java.awt.Component.dispatchEvent(Compiled Code)
    java.awt.LightweightDispatcher.retargetMouseEvent(Compiled Code)
    java.awt.LightweightDispatcher.processMouseEvent(Compiled Code)
    java.awt.LightweightDispatcher.dispatchEvent(Compiled Code)
    java.awt.Container.dispatchEventImpl(Compiled Code)

```

```

    java.awt.Component.dispatchEvent(Compiled Code)
    java.awt.EventDispatchThread.run(Compiled Code)
    "Finalizer thread" (TID:0xf60088, sys_thread_t:0x879fa0, Win32ID:0xee,
state:CW) prio=2
    "main" (TID:0xf600b0, sys_thread_t:0x87b5d0, Win32ID:0xfd, state:CW) prio=5
Monitor Cache Dump:
    <unknown key> (0x8f7280): <unowned>
        Waiters: 1
    oracle.ewt.timer.Timer@F8BD60/145A940: <unowned>
        Waiters: 1
    java.lang.Object@F7F038/1394130: <unowned>
        Waiters: 1
    sun.awt.ScreenUpdater@F8D798/136F748: <unowned>
        Waiters: 1
    sun.awt.AWTFinalizer@F91C50/135AE00: <unowned>
        Waiters: 1
    oracle.forms.engine.FlushQueue@F8B440/14D1100: <unowned>
        Waiters: 1
    java.lang.Object@F7F130/1394350: <unowned>
        Waiters: 1
    sun.applet.AppletViewerPanel@F79508/101DF78: <unowned>
        Waiters: 1
    oracle.forms.net.EncryptedInputStream@F80490/14E30D0: owner "Forms-StreamMe
sageReader" (0x8d2690, 1 entry)
        oracle.forms.engine.Heartbeat@F52748/152E8C0: <unowned>
            Waiters: 1
Registered Monitor Dump:
    SvmcJIT Method Monitor: <unowned>
    Verifier lock: <unowned>
    SvmcJIT Method Monitor: <unowned>
    SvmcJIT Method List Monitor: <unowned>
    SvmcJIT Fixups Allocation: <unowned>
    SvmcJIT Code Allocation: <unowned>
    SvmcJIT Data Allocation: <unowned>
    Thread queue lock: <unowned>
        Waiters: 1
    Name and type hash table lock: <unowned>
    String intern lock: <unowned>
    JNI pinning lock: <unowned>
    JNI global reference lock: <unowned>
    BinClass lock: <unowned>
    Class loading lock: <unowned>
    Java stack lock: <unowned>
    Code rewrite lock: <unowned>
    Heap lock: <unowned>
    Has finalization queue lock: <unowned>
    Finalize me queue lock: <unowned>
        Waiters: 1

```


6.3 WHAT DOES IT SHOW AND HOW IS IT USEFUL ?

The information contained in the thread dump can help Oracle development identify the problem in the code. The thread dump should be included in any bug filed to report the problem.

6.4 COMMON CAUSES OF HANGING APPLICATIONS

- there could be a mismatch between the Java class files and the forms server version.
 - Communication between the applet and the forms server process is based on message id. If these message id's are out of step, then the applet may not understand an instruction from the server, and vice versa. If you are using jar files, then try with the ARCHIVE tag removed. If the problem persists then pull the correct class files off the installation/patch CD by hand
- the Forms Runtime may have died
 - check that the Forms Runtime process on the server is still alive.
 - You should check that the FORMSxx_TIMEOUT parameter is set. The timeout facility acts like a heartbeat and forces the forms server to 'ping' the client on a regular basis, only cleaning up the forms server process when there has been no reply from the client for some time. Although this is primarily intended to prevent orphaned server processes, it can also prevent the unwanted premature cleanup of server processes.

7. INTERMITTENT ERRORS

Usually intermittent or apparently random errors are the most difficult to solve. The lack of predictability means that it is difficult to know what to debug, or when to switch debugging on.

Many intermittent errors are caused by one of the following

- problems with the underlying network stack
- memory related issues

7.1 DIAGNOSING NETWORK PROBLEMS

The most common error reported when the network has terminated the connection unexpectedly is

```
FRM-99999 A network error occurred, the client will not be able to continue.
```

In this scenario, the 'Details' screen in the applet will show

```
java.net.socketexception. connection reset by peer.
```

7.2 USING PING TO CHECK THE NETWORK.

Most developers and system administrators are familiar with the ping command. We often use it to check that a machine is contactable, or that the machine name can be resolved by the operating system.

However, ping is much more than a simple utility to check whether a machine on the network is alive. It can be used to identify a problem on the network, check the performance of the network or to check the robustness of the network.

When diagnosing FRM-99999 we are only concerned with packet loss. Poor network performance will affect startup time for Forms, and response times, but it will not cause Forms to crash. A lost packet will cause Forms to assume that the network connection is no longer open, and will report FRM-99999. If the packets are not reaching their intended destination then Forms has no way to differentiate that from the server process no longer being alive.

7.2.1 ping syntax.

On Windows

```
PING -l 1472 -n 90 <ipaddress or hostname>  
where :  
-l defines number of bytes to transfer.
```

-n option defines the number of pings to perform.

| Operating System | Command to get Help |
|------------------|---------------------|
| UNIX | man ping |
| Windows | ping -? |

On UNIX the exact syntax can vary from vendor to vendor.

For example, on Solaris

```
ping -s <hostname> 1472 90
```

7.2.2 Sample output

The PING command should return something similar to the following:

```
# ping -s ukpl4901.uk.oracle.com 1472 10
PING ukpl4901.uk.oracle.com: 1472 data bytes
1480 bytes from ukpl4901.uk.oracle.com (138.3.65.126): icmp_seq=0. time=264. Ms
1480 bytes from ukpl4901.uk.oracle.com (138.3.65.126): icmp_seq=1. time=211. Ms
1480 bytes from ukpl4901.uk.oracle.com (138.3.65.126): icmp_seq=2. time=227. Ms
1480 bytes from ukpl4901.uk.oracle.com (138.3.65.126): icmp_seq=3. time=212. Ms
1480 bytes from ukpl4901.uk.oracle.com (138.3.65.126): icmp_seq=4. time=210. Ms
1480 bytes from ukpl4901.uk.oracle.com (138.3.65.126): icmp_seq=5. time=212. ms
1480 bytes from ukpl4901.uk.oracle.com (138.3.65.126): icmp_seq=6. time=225. ms
1480 bytes from ukpl4901.uk.oracle.com (138.3.65.126): icmp_seq=7. time=212. ms
1480 bytes from ukpl4901.uk.oracle.com (138.3.65.126): icmp_seq=8. time=232. ms
1480 bytes from ukpl4901.uk.oracle.com (138.3.65.126): icmp_seq=9. time=255. ms

----ukpl4901.uk.oracle.com PING Statistics----
10 packets transmitted, 10 packets received, 0% packet loss
round-trip (ms)  min/avg/max = 210/226/264
```

If the packet loss is high, the response time is very slow, or the packets arrive out of order then there could be a network problem. On a LAN, the time to perform a round trip should be very low - typically less than 10 milliseconds.

In the above example, the round trip time is quite high. This is because the machine being pinged is on a WAN. The number of hops required to reach the machine is high. See Section 7.2.3 on `tracert` for details.

The following output from `ping` indicates a problem somewhere on the network:

```
C:\>ping jcarlin-sun -l 1472 -n 90

Pinging jcarlin-sun.us.oracle.com [144.25.80.48] with 1472 bytes of data:
```

```
Reply from 144.25.80.48: bytes=1472 time=15ms TTL=253
Reply from 144.25.80.48: bytes=1472 time<10ms TTL=253
Reply from 144.25.80.48: bytes=1472 time<10ms TTL=253
Reply from 144.25.80.48: bytes=1472 time<10ms TTL=253
Request timed out.
Request timed out.
Request timed out.
Request timed out.
Request timed out.
Request timed out.
Request timed out.
Request timed out.
Request timed out.
Request timed out.
Request timed out.
Reply from 144.25.80.48: bytes=1472 time<10ms TTL=253
Reply from 144.25.80.48: bytes=1472 time=16ms TTL=253
...
```

This shows a number of request time outs and packet losses which could be causing the connection reset by peer.

Note: Windows NT does display the number of packets transmitted and received or the percentage of packet loss. In these cases, you must manually calculate the percentage loss using the following equation

$$\% \text{packet loss} = (\text{number of timeouts} / \text{number of pings}) * 100$$

eg. In the example above, jcarlin-sun was pinged 90 times, and failed 23 times

Therefore,

$$\% \text{packet loss} = (23 / 90) * 100 = 26\%$$

7.2.3 Using traceroute / tracert

On UNIX systems, traceroute is a utility to trace the route taken by a TCP/IP packet from one machine to another.

On Windows, the equivalent command is TRACERT.

The following outputs compare the tracert results for a machine on the LAN, to a machine on the WAN.

Lan

```
D:\>tracert jcarlin-pc.us.oracle.com

Tracing route to jcarlin-pc.us.oracle.com [130.35.96.107]
```

```
over a maximum of 30 hops:
```

```
1 <10 ms <10 ms <10 ms jcarlin-pc.us.oracle.com [130.35.96.107]
```

```
Trace complete.
```

Wan

```
D:\>tracert ukp14901.uk.oracle.com
```

```
Tracing route to ukp14901.uk.oracle.com [138.3.65.126]
```

```
over a maximum of 30 hops:
```

```
1 <10 ms <10 ms <10 ms whq4op3-rtr-744-f2.us.oracle.com [130.35.96.1]
2 <10 ms <10 ms 15 ms whq4op3-rtr-771-f0.us.oracle.com [144.25.252.71]

3 <10 ms 16 ms <10 ms usrtr11-f0-0.us.oracle.com [137.254.20.11]
4 156 ms 172 ms 156 ms ukrtr8-atm5-0-0-1.us.oracle.com [137.254.22.2]
5 156 ms 172 ms 172 ms ukrtr1-f0.us.oracle.com [137.254.1.1]
6 203 ms 297 ms 172 ms bracknell-rtr-2-atm-301.uk.oracle.com
[138.3.0.38]
7 172 ms 172 ms 172 ms edinburgh-rtr-1-s0.uk.oracle.com [138.3.1.54]
8 172 ms 188 ms 187 ms ukp14901.uk.oracle.com [138.3.65.126]
```

```
Trace complete.
```

From these results, it would come as no great surprise to discover that the return-time reported by ping is significantly higher pinging the machine on the WAN. If your network connections are sometimes timing out, then using tracert will allow you to see where the problem lies, and allow you to focus your investigation.

7.2.4 Using netstat

If your initial testing indicates that the connection is unreliable, then netstat can yield useful information.

Unix

```
# netstat -I
Name Mtu Net/Dest Address Ipkts Ierrs Opkts Oerrs Collis Queue
lo0 8232 loopback localhost 965676 0 965676 0 0 0
fddi0 4352 haggis haggis 652193 43 193048 0 0 0
```

We can ignore the line for loopback, and concentrate on the line for haggis.

The figure for 'Queue' should be zero - there should be no network packets queued that cannot be transmitted.

The high ratio of collisions ('Collis') might indicate that the network is saturated. Anything above 5% should be investigated.

A high number of 'Terrs' or 'Oerrs' often indicate a physical network problem. The values should be close to zero. A value of 100 or more should be investigated (regardless of the amount of data transmitted)

Windows also has a netstat command. However, the syntax is a little different, and there is no way to get a short summary.

On use the following command to obtain information on the network status

```
C:\>netstat -s
```

7.2.5 Tracing the problem

Run the form and see if the FRM-99999 error coincides with a large number of time-outs on the network. If it does, then it is probable that the network time-outs are the root cause of the problem. Any dropped packet can cause Forms a problem - the form may not crash immediately, but may become confused when it subsequently receives an instruction that it considers to be out of context.

By using a combination of ping, traceroute and ipconfig, you should be able to identify where to start your investigations. Problems involving cabling may require an electronic cable tester to resolve. You may find that the cabling is loose or the connections are bad. Often hubs and switches will have built-in diagnostic software which can be checked. If the problem is difficult to locate, you may need to use a Protocol Analyzer to track it down.

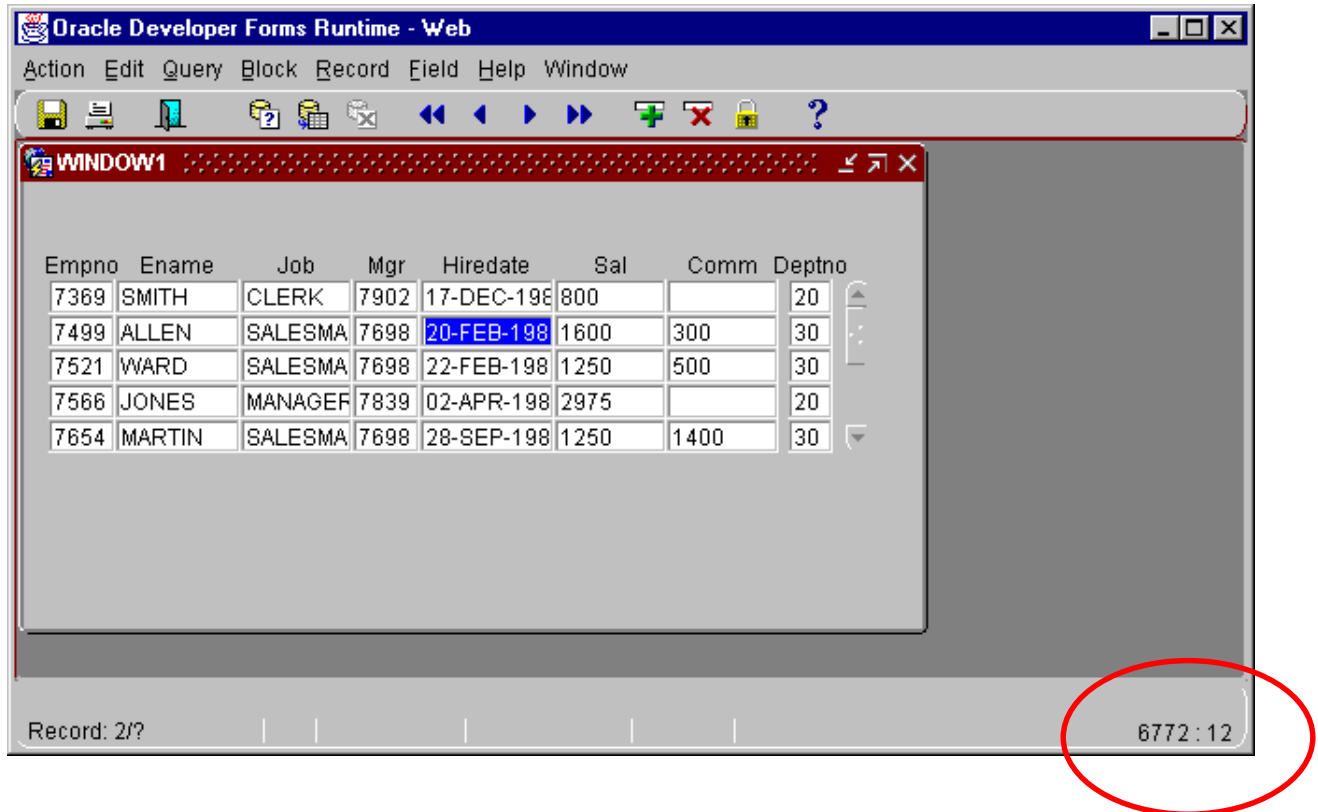
7.3 NETWORK STATISTICS IN FORMS

By altering the HTML file, Forms can be made to report the number of round-trips and the total number of bytes transmitted by the Forms communication layer whilst the form is running.

To see the network statistics, add the following parameter to your HTML file:

```
<param name=" networkStats " value="true">
```

Once the form has started, the network statistics will be shown in the 'busybar' component of the status line.



The network statistics reported by this parameter are not the same as those detected by a network ‘sniffer’. By using networkStats, Forms reports the number of bytes sent by the internal Forms communication method. This does not include any protocol specific sections of the network packets, such as a header. This extra information will increase the total number of bytes passing along the network.

networkStats can be very useful in tracking down the function which takes most time in a Web environment. For example, you may discover that a particular operation generates considerably more round trips and network traffic than you would have expected, and this can help focus efforts to improve performance.

Note: networkStats is not a supported parameter. Faulty behaviour caused by enabling network statistics that cannot be reproduced when networkStats is not specified cannot be logged as a bug, and will not be treated as such by Oracle.

7.4 MEMORY PROBLEMS

Like all software programs, a Java applet uses memory. For Java, the language specification requires a ‘garbage collector’, which is in an internal memory manager for the Java Virtual Machine (JVM). When a Java program needs memory, it requests this memory from the JVM. If there is no memory left, then the JVM will attempt to free some memory by using the garbage collector. The garbage collector will try to release memory that is no longer required to run the program back to the JVM. If there is still insufficient memory to perform the required task then the JVM will attempt to get more memory from the operating

system. If that memory allocation fails, then the Java program will be unable to continue.

For example, the following output shows a typical 'out of memory' error:

```
D:\>appletviewer -J-mx24M "http://testpc.us/webforms/mem.html"
Default cache directory d:\forms6i\jdk\bin\..\jcache not found.  JAR caching
disabled.
Forms Applet version is : 4
java.lang.OutOfMemoryError
    at sun.awt.image.GifImageDecoder.<init>(Compiled Code)
    at sun.awt.image.InputStreamImageSource.getDecoder(Compiled Code)
    at sun.awt.image.URLImageSource.getDecoder(URLImageSource.java:119)
    at sun.awt.image.InputStreamImageSource.doFetch(Compiled Code)
    at sun.awt.image.ImageFetcher.fetchloop(Compiled Code)
    at sun.awt.image.ImageFetcher.run(Compiled Code)
```

7.4.1 Setting JVM runtime options

The initial memory allocation for the JVM can be configured, as can the maximum memory allocation. A full list of parameters can be found at <http://java.sun.com/products/jdk/1.1/docs/tooldocs/solaris/java.html>

Some, but not all of the parameters are shown below:

```
D:\testjava>appletviewer -J-help x
usage: java [-options] class
```

where options include:

```
-help                print out this message
-version             print out the build version
-v -verbose          turn on verbose mode
-debug               enable remote JAVA debugging
-noasyncgc           don't allow asynchronous garbage collection
-verbosegc           print a message when garbage collection occurs
-noclassgc           disable class garbage collection
-ss<number>         set the maximum native stack size for any thread
-oss<number>         set the maximum Java stack size for any thread
-ms<number>          set the initial Java heap size
-mx<number>          set the maximum Java heap size
-classpath <directories separated by semicolons>
                    list directories in which to look for classes
-prof[:<file>]      output profiling data to .\java.prof or .\<file>
-verify              verify all classes when read in
-verifyremote        verify classes read in over the network [default]
-noverify            do not verify any class
-nojit               disable JIT compiler
```


When using Appletviewer, you need to use the syntax

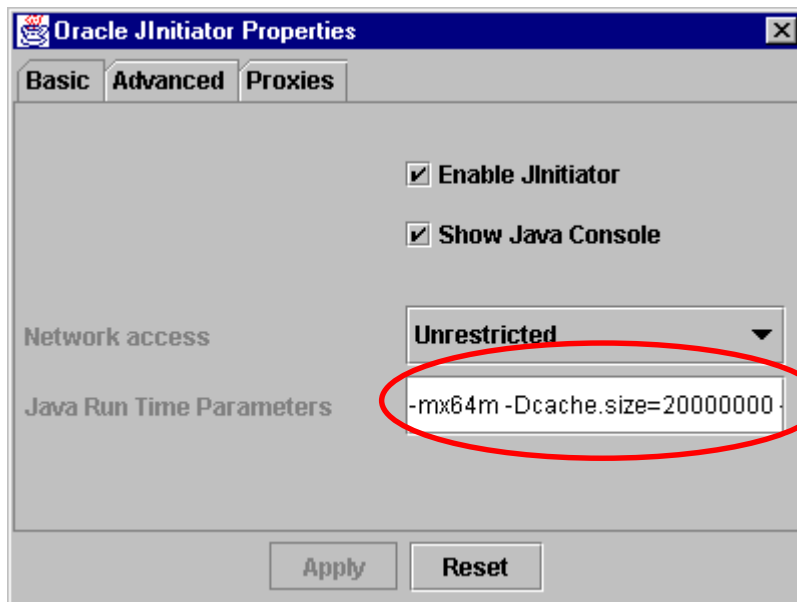
```
appletviewer -J<-options> <filename>
```

For example, the following command will set the initial 'Java Heap' (the memory used by the JVM) to 20Mb, and the maximum memory to be used by the JVM to 32Mb

```
Appletviewer -J-ms20m -J-mx32Mb "http://test-pc.us/webforms/maxmem.html"
```

The default initial and maximum sizes for Appletviewer are 16Mb and 20Mb respectively.

Using JInitiator, you will need to set the runtime options in the JInitiator control panel.



Note: the JVM will only use the memory it is told it is allowed to use. Even if you have memory available to the Operating System, the JVM will not use it.

7.5 MEMORY LEAKS

7.5.1 What is a memory leak ?

A memory leak is an error in a program's dynamic-store allocation logic that causes it to fail to reclaim discarded memory, leading to eventual collapse due to memory exhaustion.

What does this mean ? Well, when a program runs it may need to allocate some memory to perform a particular task. If the program has finished with that memory and no longer has any use for it, but fails to make that memory available to other programs running on the machine, then it is said to have leaked the memory.

A typical method used to spot memory leaks is to repeat a series of steps, and observe the memory in use by the application - if the memory usage continues to rise with each iteration, then the assumption is often that the program has a memory leak.

However, it is not quite as simple as that: some complex software may choose to retain control of memory it has previously allocated so that it can reuse it at a later point - memory allocation can be an expensive operation, and if the program expects that it will need more memory later it may be more efficient to keep the unused memory available for reuse.

7.5.2 Memory leaks in Java

The Java language specification demands that the JVM has a Garbage Collector (GC). In Java, the programmer allocates memory by creating a new object. There is no way to de-allocate that memory. Periodically the Garbage Collector sweeps through the memory allocated to the program, and determines which objects it can safely destroy, therefore releasing the memory.

To determine which objects it can safely destroy, the garbage collector uses a 'mark and sweep' algorithm. The garbage collector scans the dynamically allocated memory for objects, marking those which still have active references to them. After all possible paths to objects have been investigated, unmarked objects are known to be no longer needed and can be garbage collected.

A common myth with Java programming is that the presence of a garbage collector means that there can be no memory leaks. This is not true. The garbage collector simply marks those objects which have active references, and destroys those that do not. It is possible to have an active reference to an object that is no longer needed. This is a memory leak in Java. The solution to the leak is to destroy the references to the object once it is no longer needed so that the garbage collector can identify it as safe to destroy. If a memory leak exists in a Java program, then calling the garbage collector more frequently will not help.

To complicate matters further, the JVM may choose not to release unused memory back to the operating system. In the real world this seldom matters, as most programs will typically require more memory at some point in the near future and can reuse the free memory in the JVM. However, it is worth bearing in mind that not all the memory allocated to the JVM will be in use by the program running in the JVM.

7.5.3 How to identify a memory leak

Typically, if a growth in memory usage is observed each time a particular series of operations is performed, then it is a memory leak. The ideal proof is to

1. get the form into an initial base state, and record the memory usage
2. perform a series of steps to illustrate the problem

3. return to the initial base state, and record the memory usage

By repeating steps 2 & 3, it is possible to determine whether there is a steady memory leak or not. If the growth in memory is small over a large number of iterations, then it may not be a leak at all - it could be the JVM retaining unused memory, or the garbage collector not activating as frequently as expected.

8. PERFORMANCE PROBLEMS

8.1 CACHING IN THE ORACLE JDK

When any Java program runs, the Java virtual machine needs to load class files. When running over the Internet, the time taken to download a class file each time the program runs can lead to performance problems. In order to solve this download problem, the JDK supports Java Archive (JAR) files. A jar file is simply a collection of class files bundled into one compressed file. Typically, the size of the jar file will be much smaller than the combined size of the class files it contains. In addition to reducing the amount of data to be transferred, jar files also allow JInitiator and Oracle's JDK to use a process referred to as caching.

Starting with OJDK 1.1.7.15, several performance improvements were made to the caching process. These are discussed in section 8.1.4

8.1.1 What is Caching ?

When the JVM first references a class, it checks the local machine to see if any of the previously cached jar files contain this class. If the class does exist in one of the pre-cached jar files, then the JVM checks to see if there is a newer version of this jar file on the application server. If there is a newer jar file available then the new copy of the jar file is downloaded to the client cache. If the cached jar file is up to date, then the class file is loaded from the cached jar file rather than from over the network.

8.1.2 Why is it important ?

If the application jar files do not change, then after the application has run once, and all the jar files required have been cached on the client, then subsequent invocations of the application will always load the classes from the local cached copies. This can lead to significant performance improvements in the startup time for the application. If new classes are needed to run a specific part of the application, these will be downloaded as required.

8.1.3 How do you check if caching is working?

If the version of the OJDK you are using is older than 1.1.7.15.1, then to check caching is enabled, you will need to check for files in the cache directory. If you are using OJDK 1.1.7.15.1 or above, then you can configure the caching mechanism to write logging output. (See section 8.1.5 for Details)

If you are not sure which version of the OJDK you are using,

- check the readme files under the JInitiator or Appletviewer directory.

- JInitiator will write the OJDK version to the JInitiator console.
- Appletviewer has an option to show the version:

```
Appletviewer -J-version test
```

8.1.4 Performance Improvements in OJDK 1.1.7.15

Whilst caching meant that the jar file would not be downloaded from the server every time the application is invoked, there were still some issues affecting performance

- The JAR files contain compressed data. The time to decompress this data from the cached jar file outweighs the time saved downloading less data from the network.
- JAR files can be digitally signed and undergo authentication to ensure they have not been modified in transit over the network. This procedure involves expensive mathematical calculations.

The new caching mechanism introduced in 1.1.7.15 addresses these issues in the following way:

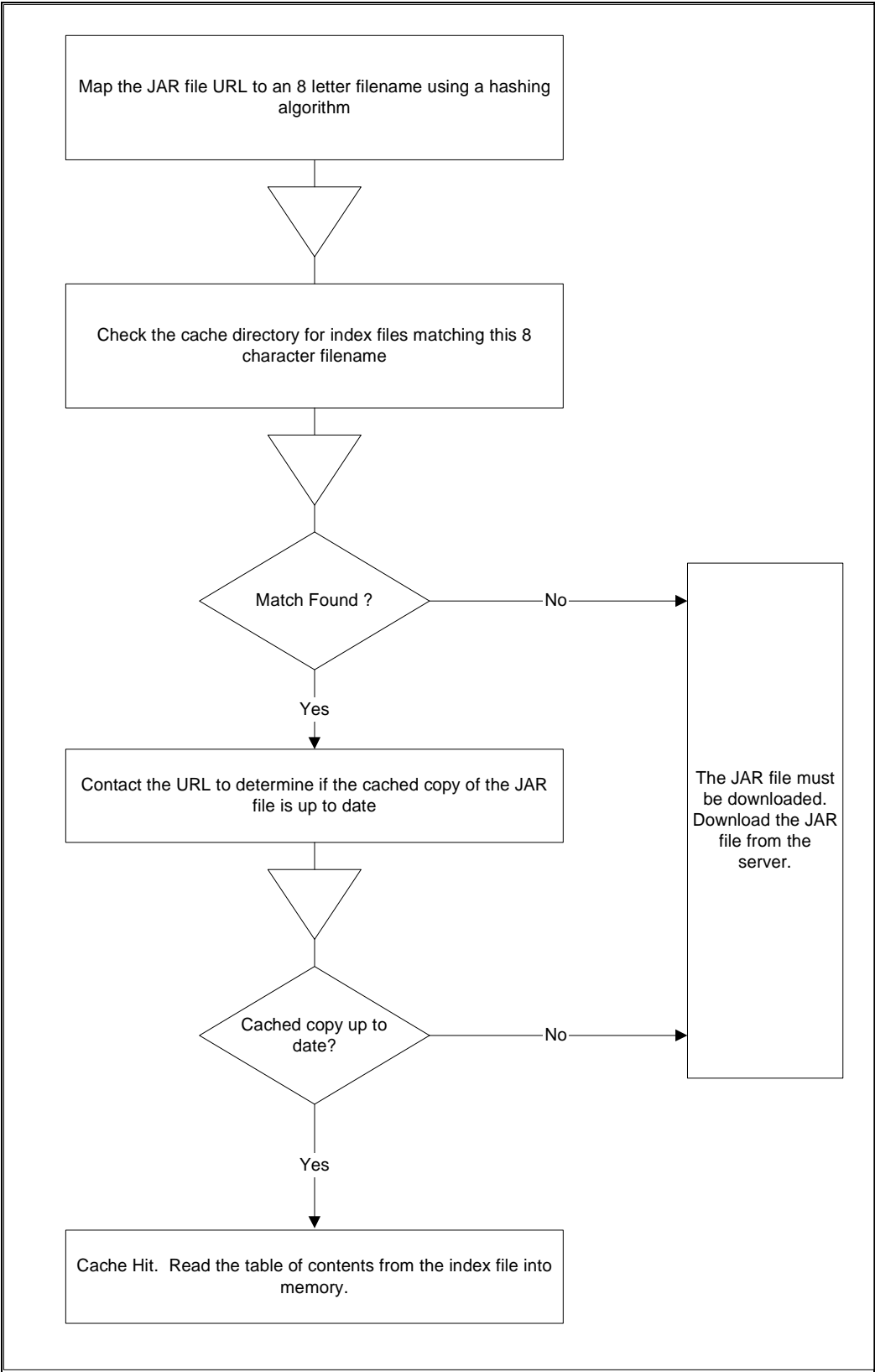
When the JAR file is downloaded for the first time, two files are created

- A data file which contains all of the unzipped data from the JAR file.
 - Data files have the extension .dxx, where xx is a number between 00 and 99. For example 10f756b8.d00
- An index file which contains information about the JAR file, including the URL it was loaded from, the date it was last modified on the server, and a table of contents. The table of contents lists all of the entries in the JAR file, their offsets in the data file, and the authentication status of each entry.
 - Index files have the extension .ixx, where xx is a number between 00 and 99. For example 10f756b8.i00

The information in these files is stored in a binary format. There is no easy way to read them by eye, and there is little value in doing so.

The first eight characters of all cache files represent the URL where the JAR file was downloaded from. This allows the caching mechanism to quickly find a URL in the cache by mapping the URL to its corresponding eight character representation and looking for files with that name.

When a JAR file is needed to run an applet, the following logic is used:



When a class file is required by the application, the OJDK uses the information in the table of contents to locate the class file in the data cache. If the data was digitally signed, then the list of authenticated signers is

read from the data file.

8.1.5 Configuring OJDK 1.1.7.15

In the Oracle JDK from version 1.1.7.15 onwards, the caching mechanism can output logging information to allow for the viewing of caching operations as they are performed. This logging information can be configured to operate in 5 modes: off (default), verbose, cache hit, cache miss and log to file. Using verbose mode all cache operations are logged when they are performed. This information will contain details such as the when a cache miss or hit occurs, what files are being added to the cache and what files are being deleted from the cache Using cache hit mode, only caching operations that load JAR files from the local cache will be logged. Using cache miss mode, only caching operations that load JAR files from the network are logged. Using the logfile mode enables the caching messages to be written to a file for later use.

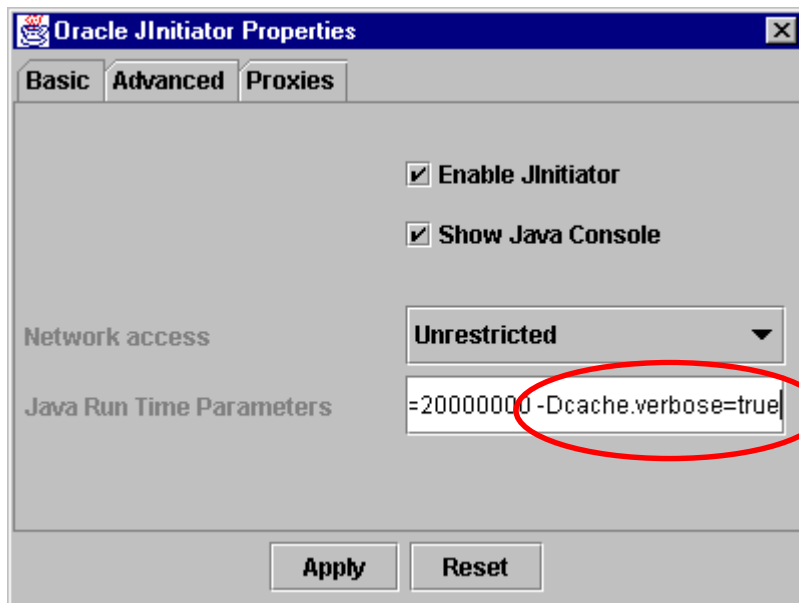
| Property Name | Property Value | Action |
|--------------------|----------------|--|
| None Specified | | Indicates whether caching is enabled, and outputs the name of the cache directory and its size. |
| Cache.verbose | true | Output all cache operations to the Oracle JInitiator console window. Includes information such as cache hits and cache misses, cache directory, cache size, file additions to the cache and file deletions from the cache. |
| Cache.verbose.hit | true | Output message when a file is retrieved from the cache. |
| Cache.verbose.miss | true | Output message when a required file is not able to be retrieved from the cache. |
| Cache.logfile | logfile name | When this property is specified, the verbose output generated by the caching mechanism will be stored in a file. The verbose output will be appended to the specified file each time Oracle Jinitiator is run. Each run will be separated from previous information with a header containing the date and time of the run. By specifying an output log file, verbose information will always be directed to the file. If cache.verbose, cache.hit or |

| | | |
|--|--|---|
| | | cache.miss is specified in addition to the output file property, the output messages will also be sent to the console window. |
|--|--|---|

To configure caching, specify the relevant Java property for the cache logging option to be configured with the form of:

`-D<parameter>=<value>.`

I.e. In JInitiator, specify the parameter in the JInitiator Control Panel



When a simple form is run using JInitiator with

`-Dcache.verbose=true`

the following output is produced in the Java Console:

```
Oracle JInitiator version 1.1.7.18
Using JRE version 1.1.7.18o
User home directory = C:\WINNT\Profiles\cxlewis.000
Cache: JAR caching enabled.
  Cache directory: C:\PROGRA~1\Oracle\JINITI~1.18\jcache
  Maximum cache size: 20000000 bytes
Opening http://ukpl4901.uk.oracle.com/forms60code/f60a11.jar
proxy=emeacache.uk.oracle.com:80
Cache: Cache hit for http://ukpl4901.uk.oracle.com/forms60code/f60a11.jar
Opening http://ukpl4901.uk.oracle.com/forms60code/javax/swing/JinternalFrame.class
proxy=emeacache.uk.oracle.com:80
```



```
Opening
http://ukp14901.uk.oracle.com/forms60code/oracle/forms/engine/RunformBundle_en_GB.clas
s proxy=emeacache.uk.oracle.com:80
Opening
http://ukp14901.uk.oracle.com/forms60code/oracle/forms/engine/RunformBundle_en_GB.prop
erties proxy=emeacache.uk.oracle.com:80
Opening http://ukp14901.uk.oracle.com/forms60code/oracle/forms/registry/Registry.dat
proxy=emeacache.uk.oracle.com:80
Opening http://ukp14901.uk.oracle.com/forms60code/oracle/forms/registry/default.dat
proxy=emeacache.uk.oracle.com:80
Forms Applet version is : 4
```

When a value for logfile is provided, then the output will be directed to the specified file:

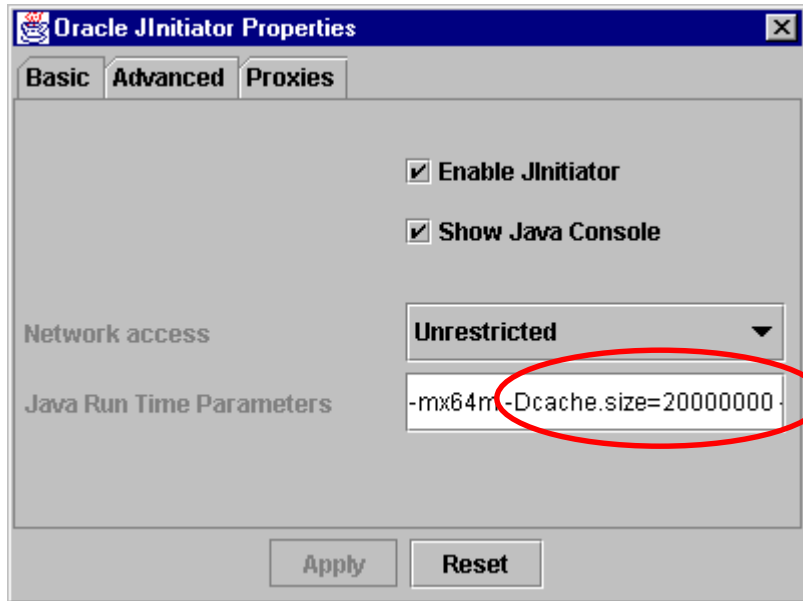
```
=====
Mon Jan 10 11:27:00 GMT 2000
JAR caching enabled.
  Cache directory: C:\PROGRA~1\Oracle\JINITI~1.18\jcache
  Maximum cache size: 20000000 bytes
Cache hit for http://ukp14901.uk.oracle.com/forms60code/f60all.jar
```

To override the caching directory, or to change the size of the cache, then use the parameters

```
-Dcache.directory=<directory name>
-Dcache.size=<cache size>
```

NOTE: All the parameters discussed in this section can be specified in JInitiator and Appletviewer

```
appletviewer -J-Dcache.directory="c:\mycache" -J-Dcache.size=32000000
"http://ukp14901.uk.oracle.com/webhtml/test.html"
```



For further information on improving performance, see the Oracle White Paper “Oracle Developer Server: How to Tune for the Deployment of Internet Applications.”

8.2 THE FORMS LATENCY METER

The Forms Latency Meter is new in Forms 6i. It is not a supported part of the product. Faulty behaviour in the latency meter, or behaviour caused by enabling the latency meter that cannot be reproduced when the latency meter is not used cannot be logged as a bug, and will not be treated as such by Oracle.

8.2.1 What is the Forms Latency Meter ?

The Forms Network Latency Meter tests the network in a mode that is similar to the way Forms Server will use the network by simulating typical messages between the applet and the Forms Server. The latency meter measures the response time of the network, and reports the average response time.

The default test examines the performance of the network for 1.4K chunks and 6K chunks of data by transmitting packets of these sizes three times. The average of this response time is displayed in the format X:Y, where X represents the average response time for a 1.4K message, and Y the average response time for a 6K message.

The unit of measurement is a millisecond.

1.4K byte chunks represent the regular stream generated on the network during the regular interaction with the Form such as entering data and navigating across items in a Canvas.

The 6K chunks represent the data stream usually generated in more complex operations such as opening and closing a form.

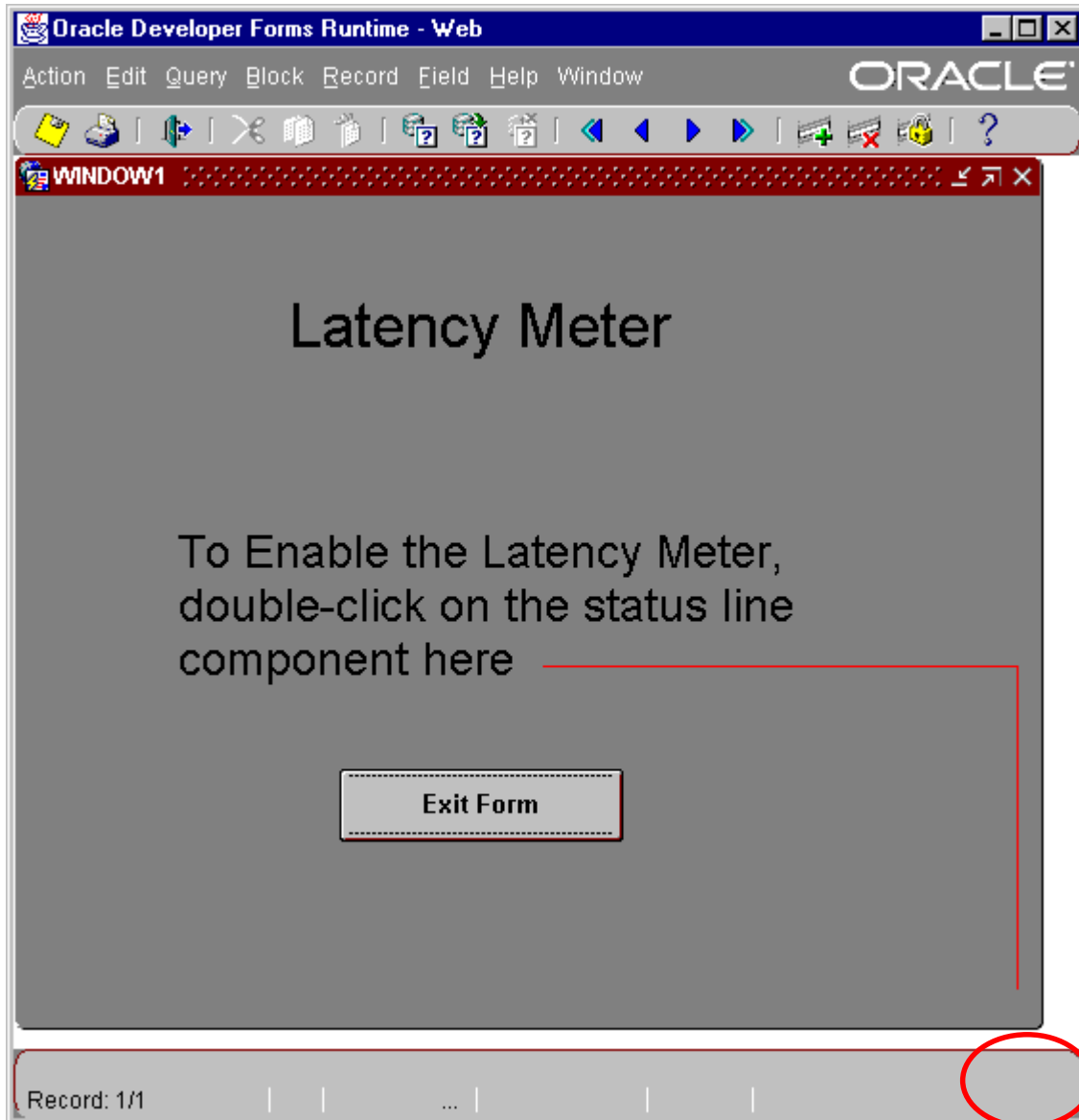
8.2.2 How is the Forms Latency Meter enabled ?

To enable the Forms Latency Meter, the HTML file used to launch the form must be modified to include the "latencyCheck" applet parameter.

Eg.

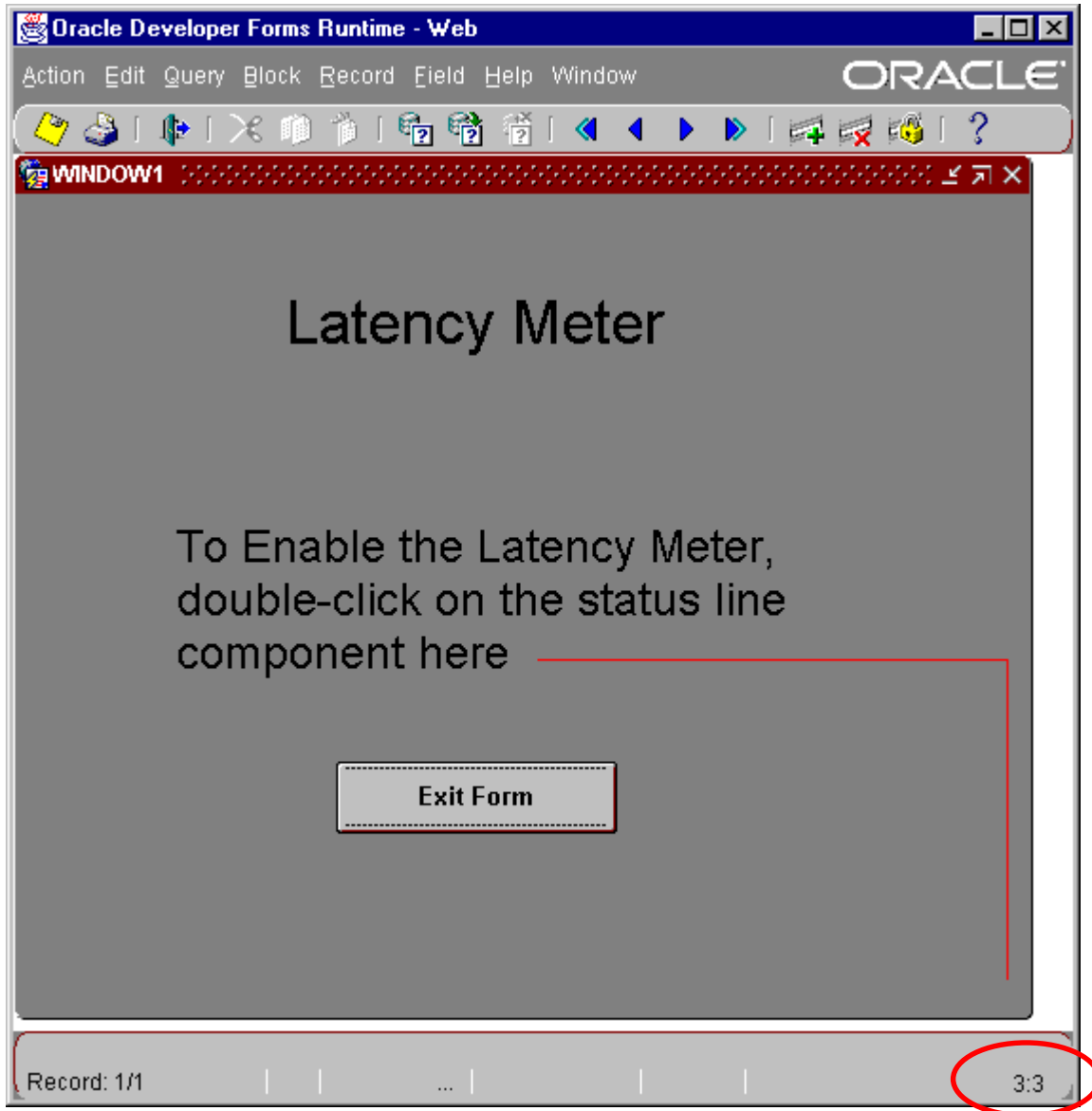
```
...  
<param name="serverPort" value="9000">  
<param name="latencyCheck" value="true">  
...
```

Once the form has started, then double-clicking on the 'busybar' component of the status line starts the latency test.



The text 'Latency' followed by a counter will appear in this section of the status line.

The counter will stop at 5, and the latency figures will be reported.



8.2.3 Why use the Latency Meter ?

You may be asking yourself how the Latency Meter can help you. After all, ping can report network latency, so why do you need to know about the Forms Latency Meter ?

Unlike ping, the latency meter can report on network latency over firewalls, using the http protocol. For example, when running Forms Server 6i over http, you could have firewalls at both the client and server. The latency meter allows you to determine if the network latency is affecting performance in this more complicated environment.

8.2.4 Example Results

| Test | X (1.4K) | Y (6K) |
|---|----------|--------|
| Local Forms Server (i.e. client and server on same machine) | 0 | 3 |
| Forms Server on same LAN (4 hops from client to server) | 33 | 105 |
| Forms Server on WAN (8 hops from client to server) | 185 | 440 |

8.3 EXPECTED PERFORMANCE EXAMPLES.

In the following tables, the environment is as follows:

- The form being used is a simple form consisting of a master-detail form based on dept and emp.
- The Forms Server and the Database are on the LAN. I.e. no significant delay is introduced by Sql*Net traffic between the forms runtime engine and the database.
- The client PC is a Pentium P3, 450 MHz.
- The Forms version is 6i.

NOTE: These figures are intended to illustrate the effects of both caching and network latency on Forms performance. They do not guarantee a performance level.

| Network Speed (1.4K/6K) | Time in seconds to start simple form (nothing cached) | Time in seconds to start simple form (f60all.jar cached) |
|----------------------------|---|--|
| <10 / <10 | 27 | 7 |
| 40 / 130 | 30 | 8 |

APPENDIX A - FORMS SERVER CONNECTION ACTIVITY LOGGING

Forms Server connection activity logging provides a log file of all Forms Server activity, monitoring connection and disconnection requests, and can prove helpful when diagnosing a variety of problems.

Switching this feature on incurs minimal overhead, and is simple to do. Therefore, it is recommended that this feature is enabled at all times.⁷

All messages produced by the Forms Server are written to a specified log file.

This includes information about the start-up of the Forms Server, connection requests, connection establishment (can the request be satisfied), connection and disconnection, and abnormal terminations. The log file also records IP addresses, port numbers, and process ID information to allow server administrators to determine which processes belong to which users.

In addition, if the forms process dies unexpectedly, a stack trace will be recorded in the log file.

A 1 SWITCHING SERVER LOGGING ON

To turn forms server connection activity logging on, specify the location of the log file when starting the forms server

Windows NT

| Version | Command |
|-----------|--|
| FORMS 4.5 | f45srv32 log=d:\logs\f45srv.log |
| FORMS 5.0 | f50srv32 log=d:\logs\f50srv32.log |
| FORMS 6.0 | ifsrv60 -listen log=d:\logs\f60srv.log |

Unix

| Version | Command |
|-----------|---------------------------------------|
| FORMS 4.5 | f45ctl start log=/u01/logs/f45srv.log |
| FORMS 5.0 | f50ctl start log=/u01/logs/f50srv.log |
| FORMS 6.0 | f60ctl start log=/u01/logs/f60srv.log |

⁷ On NT, with Forms versions prior to 6i, there is no way to have a NT service set to startup automatically with Forms Server Connection Activity Logging switched on. Starting with Forms 6i, you can configure options when installing the server as a service, or subsequently by editing the appropriate registry variables.

If no log file is specified in the command line, then logging is not enabled, and no messages will be written.

A 2 SAMPLE OUTPUT

The log file produced will look similar to the one below:

```
FORMS CONNECTION ACTIVITY LOG FILE
Developer:Forms/LogRecord

[12/02/99 19:49:27 Pacific Standard Time]::Server Start-up Data:
    Server Log Filename: d:\users\web\logs\f60srv.txt
    Server Hostname: forms-ddr2-pc
    Server Port: 6000
    Server Pool: 1
    Server Process Id: 271

[12/02/99 19:50:05 Pacific Standard Time]::LISTN: Connection Request [ConnId=0,
Addr=130.35.99.68:37640]
[12/02/99 19:50:05 Pacific Standard Time]::RUNFORM Client Connected [ConnId=0,
PID=294]
[12/02/99 19:50:13 Pacific Standard Time]::RUNFORM Client Disconnected
[ConnId=0, PID=294]
[12/02/99 19:50:41 Pacific Standard Time]::LISTN: Connection Request [ConnId=1,
Addr=130.35.99.68:39432]
[12/02/99 19:50:41 Pacific Standard Time]::RUNFORM Client Connected [ConnId=1,
PID=290]
[12/02/99 19:50:44 Pacific Standard Time]::RUNFORM Client Disconnected
[ConnId=1, PID=290]
```

If a forms runtime process crashes, then it will attempt to write a stack trace. If you have logging switched on, then the stack trace will be redirected to the log file:

```
FORMS CONNECTION ACTIVITY LOG FILE
Developer:Forms/LogRecord

[Wed Feb 23 07:39:02 2000 PST]::Server Start-up Data:
    Server Log Filename: log.txt
    Server Hostname: cxlewis-sun.us.oracle.com
    Server Port: 6992
    Server Pool: 1
    Server Process Id: 16838

[Wed Feb 23 07:39:13 2000 PST]::LISTN: Connection Request [ConnId=0,
Addr=138.3.65.126:2485]
[Wed Feb 23 07:39:13 2000 PST]::RUNFORM Client Connected [ConnId=0, PID=16839]
```


[Wed Feb 23 07:39:19 2000 PST]::Client Status [ConnId=0, PID=16839]

>> ERROR: Abnormal termination of connection, Error Code: 11

FORM/BLOCK/FIELD: DUMP45:<NULL>.<NULL>

Last Trigger: ON-LOGON - (Successfully Completed)

Msg: <NULL>

Last Builtin: - (No error number)

----- Call Stack Trace [ConnId = 0, ProcId = 16839] -----

| calling location | call type | entry point | argument values in hex (? means dubious value) |
|---------------------|-----------|---------------------|--|
| siehjmptrm()+408 | CALL | siehdst()+0 | C3F5D8 ? B67274 ? B67260 ? B67250 ? B67244 ? B67108 ? |
| sigacthandler()+40 | PTR_CALL | | B ? 0 ? C08614 ? A ? C3F5D8 ? 0 ? |
| lmmstcalloc()+44 | CALL | lmmstmalloc()+0 | B ? 0 ? EFFF7EE0 ? 0 ? EFFF8174 ? BC3484 ? |
| iwpbxo()+32 | CALL | | EFFF815C ? A ? EFFF815C ? 2 ? EE863100 ? 1066 ? |
| iwpbeh()+556 | CALL | iwpbxo()+0 | CA6448 ? C9B168 ? 1 ? 0 ? 0 ? CB2428 ? |
| ixobjhm_handlemessa | PTR_CALL | | C9A4F0 ? 800000 ? C9B168 ? EEED209C ? 1 ? 1 ? |
| ixwsru()+184 | CALL | ixobjhm_handlemessa | C9A4F0 ? 0 ? C9B6C8 ? D80B48 ? D82330 ? D82330 ? |
| ifzget()+964 | CALL | iftogi()+0 | D7DB38 ? C9A4F0 ? C9B6C8 ? C9B6C8 ? BC3484 ? C74428 ? |
| ifzeky()+84 | CALL | ifzget()+0 | CA6448 ? 1 ? 0 ? EEED1BC8 ? 1 ? CB2428 ? |
| ifzgmt()+1152 | CALL | ifzeky()+0 | CA6448 ? CA6448 ? FFFFFFFD ? CB2428 ? EEED1BC8 ? CB2428 ? |
| ifzmky()+48 | CALL | ifzgmt()+0 | CA6448 ? 0 ? 0 ? CB2428 ? EEED1BC8 ? CB2428 ? |
| ifzevl()+56 | CALL | ifzmky()+0 | 0 ? CA6448 ? B5E7B0 ? 7FFFFFFF ? 8000000 ? EEED1BC8 ? |
| ifzefi()+140 | CALL | ifzevl()+0 | CA6448 ? CA6448 ? 80000000 ? A3E79C ? C9A4F0 ? 1 ? |
| ifzevf()+868 | CALL | ifzefi()+0 | CA6448 ? CA6448 ? 1 ? CB1330 ? EEED209C ? CB2428 ? |
| ifzerc()+2172 | CALL | ifzevf()+0 | 4000 ? FFFFBFFF ? CA6448 ? 8000000 ? 4000 ? EEED209C ? |
| ifzebk()+1940 | CALL | ifzerc()+0 | CB2428 ? D71D38 ? CA6448 ? C9A4F0 ? CB2428 ? EEED1BC8 ? |
| ifzefm()+232 | CALL | ifzebk()+0 | 0 ? 4000 ? CA6448 ? 1 ? FFFFFFF ? 10000 ? |
| ifzeif()+816 | CALL | ifzefm()+0 | CA6448 ? CB3860 ? CA6448 ? |

```

ifzexf()+1212      CALL      ifzeif()+0      2000000 ? 0 ? C9A4F0 ?
CA6448 ? 0 ? 0 ? 1 ? CB2428 ?
2000000 ?
ifzexe()+148      CALL      ifzexf()+0      CA6448 ? 1 ? 0 ? CA6A30 ? 0 ?
0 ?
ifzman()+5564     CALL      ifzexe()+0      0 ? CB0488 ? 6 ? CB0188 ?
CA6448 ? CB1288 ?
if4mmo()+40      CALL      ifzman()+0      CBEA50 ? 8 ? 0 ? 0 ? 0 ? 0 ?
PTR_CALL        0 ? 0 ? 0 ? 2 ? C71410 ?
112830 ?
PTR_CALL        C71014 ? 0 ? EEA0925C ?
C712CC ? C712E4 ? C712C0 ?
CALL            0 ? EFFF9730 ? 2 ? EFFF97E4 ?
6E ? 0 ?
CALL            8000 ? A0BC7C ? EFFF97E4 ?
64 ? 7 ? A0BF9C ?

----- End of Call Stack Trace -----
[Wed Feb 23 07:39:30 2000 PST]::RUNFORM Client Disconnected [ConnId=0, PID=16839]

```

If a server process crashes, then the termination message will contain an operating system specific error code.

```
>> ERROR: Abnormal termination of connection, Error Code: 11
```

On UNIX these are defined in signal.h, commonly located in /usr/include/sys/signal.h

On NT, the error codes are also defined in signal.h, typically in C:\Program Files\Microsoft Visual Studio\VC98\Include.

The following table shows some of the more common signals in NT.

| Name | Value | Description |
|---------|-------|--|
| SIGINT | 2 | CTRL+C interrupt. The default action issues INT 23H. |
| SIGILL | 4 | Illegal instruction. The default action terminates the calling program. |
| SIGFPE | 8 | Floating-point error, such as overflow, division by zero, or invalid operation. The default action terminates the calling program. |
| SIGSEGV | 11 | Illegal storage access. The default action terminates the calling program. |
| SIGTERM | 15 | Termination request sent to the program. The default action terminates the calling program. |
| SIGABRT | 22 | Abnormal termination. The default action terminates the calling program with exit code 3. |

Knowing the signal raised during the crash can help Oracle development investigate the bug.

APPENDIX B - FORMS RUNTIME DIAGNOSTICS (FRD)

Forms Runtime Diagnostics is a mechanism for collecting diagnostic information from a running form. It can be used by a developer to assist in debugging their application, and it can be used by a system administrator or Oracle support staff to collect information which may help identify the root cause of a problem they are investigating .

FRD is enabled at an individual process level. That is, enabling FRD for one user will not enable it for any others. It can be switched on at the command line for forms, or can be switched on for a form that is already running using an operating system command or utility.

When a form is run with FRD enabled, a chronological representation of external user events and internal processing events are written to the log file. This log file can be examined to determine what the form was doing at any particular time, or what internal events are triggered by a particular user event.

FRD generates extensive output, and it is therefore recommended that it is used only when specific issues are encountered. This can be achieved by either switching FRD on dynamically (see sections B1.3 and B1.4) or by having an alternate HTML file which will switch FRD on. Users experiencing problems can then be directed towards this alternate HTML file while investigation is carried out.

The following sections describe the user interface of FRD, details the events that are logged, and shows examples of the output which one can expect to find in the log file itself.

B 1 INITIATING FRD

B 1.1 INITIATING FRD FROM THE COMMAND LINE

To initiate data collection, provide the string "collect" to the runtime argument "record" as follows:

Windows

| Version | Command |
|-----------|--|
| FORMS 4.5 | f45run32 module=myform userid=scott/tiger@hockey record=collect |
| FORMS 5.0 | f50run32 module=myform userid=scott/tiger@hockey record=collect |
| FORMS 6.0 | ifrun60 module=myform userid=scott/tiger@hockey record=collect |

UNIX

| Version | Command |
|-----------|---|
| FORMS 4.5 | f45runm module=myform userid=scott/tiger@hockey record=collect |
| FORMS 5.0 | f50runm module=myform userid=scott/tiger@hockey record=collect |
| FORMS 6.0 | f60runm module=myform userid=scott/tiger@hockey record=collect |

This will create a file in the current working directory with name:

collect_<process ID>.

In order to choose the name of the file yourself, provide the string as the value of the "log=" argument:

Windows

| Version | Command |
|-----------|--|
| FORMS 4.5 | f45run32 module=myform userid=scott/tiger@hockey record=collect log=c:\logs\frd45.log |
| FORMS 5.0 | f50run32 module=myform userid=scott/tiger@hockey record=collect log=c:\logs\frd50.log |
| FORMS 6.0 | ifrun60 module=myform userid=scott/tiger@hockey record=collect log=c:\logs\frd60.log |

Unix

| Version | Command |
|-----------|---|
| FORMS 4.5 | f45runm module=myform userid=scott/tiger@hockey record=collect log=/u01/logs/frd45.log |
| FORMS 5.0 | f50runm module=myform userid=scott/tiger@hockey record=collect log=/u01/logs/frd50.log |
| FORMS 6.0 | f60runm module=myform userid=scott/tiger@hockey record=collect log=/u01/logs/frd60.log |

B 1.2 INITIATING FRD FROM THE WEB

To invoke FRD on the Web, the launching HTML file must specify the command line parameter in the same way as above.

As FRD generates a large amount of data, it is recommended that FRD is not switched on by default. The

system administrator should provide an alternate HTML file with FRD switched on, and redirect those users experiencing problems to the alternate FRD-enabled file.

Jinitiator

```
...
<OBJECT classid="clsid:9F77a997-F0F3-11d1-9195-00C04FC990DC"
        WIDTH=400
        HEIGHT=400
        codebase="http://mymachine/jinit.exe">
  <PARAM NAME="CODE"          VALUE="oracle.forms.engine.Main" >
  <PARAM NAME="CODEBASE"     VALUE="/form60code/" >
  <PARAM NAME="ARCHIVE"      VALUE="/form60code/f60all.jar" >
  <PARAM NAME="type"         VALUE="application/x-jinit-applet">
  <PARAM NAME="serverPort"   VALUE="6000">
  <PARAM NAME="serverArgs"   VALUE="module=test userid=scott/tiger@hockey
record=collect log=frd.txt">
  <PARAM NAME="serverApp"    VALUE="default">
  <COMMENT>
  <EMBED type="application/x-jinit-applet"
        java_CODE="oracle.forms.engine.Main"
        java_CODEBASE="/form60code/"
        java_ARCHIVE="/form60code/f60all.jar"
        WIDTH=400
        HEIGHT=400
        serverPort="6000"
        serverArgs="module=test userid=scott/tiger@hockey
record=collect log=frd.txt"
        serverApp="default"
        pluginspage="http://mymachine/jinit.html">
    <NOEMBED>
  </COMMENT>
  ...
  ...
```

Appletviewer

```
...
<APPLET
  CODEBASE="/forms60code/"
  CODE="oracle.forms.engine.Main"
  ARCHIVE="f60all.jar"
  HEIGHT=500
  WIDTH=500
>
```

```
<PARAM NAME="serverPort" value="9000">
<PARAM NAME="serverApp" value="default">
<PARAM NAME="serverArgs"
  VALUE="module=test_form record=collect log=c:\temp\frdlog.txt">

</APPLET>
...
```

B 1.3 INITIATING FRD FROM ORACLE ENTERPRISE MANAGER (OEM)

To install and configure Oracle Enterprise Manager, see the on-line documentation.

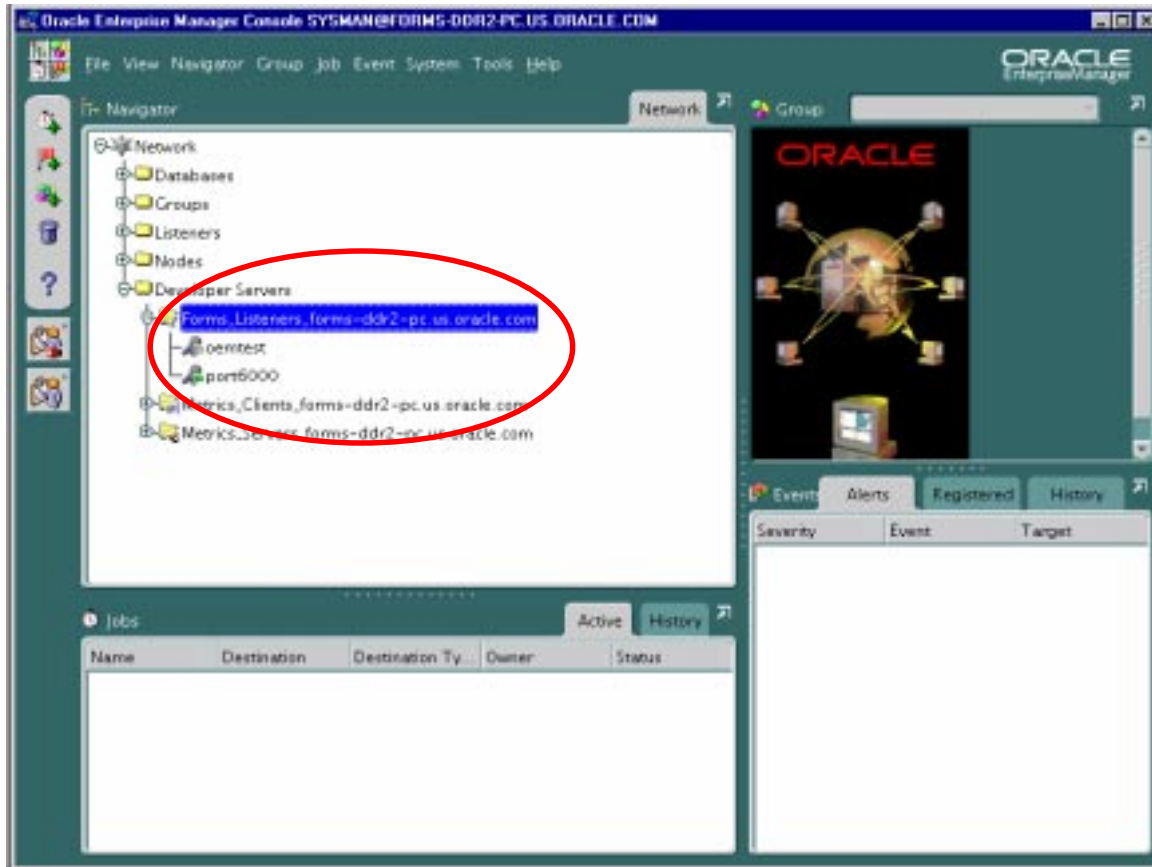
To install Forms 6i support in OEM Repository, see the install guide, and refer to the following Technical White Paper:

Title: Deploying Forms Applications to the Web with Oracle Developer Server Release 6I

=>Chapter 13, Oracle Enterprise Manager Forms Support.

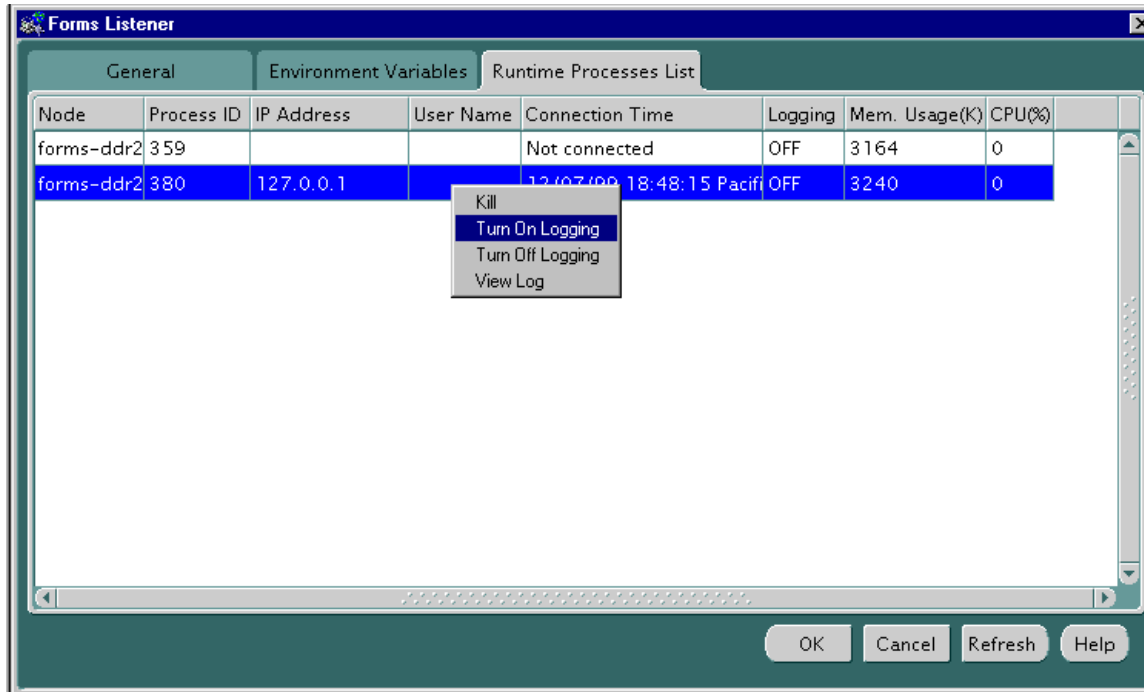
Location: Oracle Developer 6I Documentation.

Once you have successfully configured Forms 6i support then you will have a OEM screen similar to the following image:



To see the running processes for a forms listener, right-click on the listener, and then pick 'properties'. The right tab of the properties screen shows the running process.

Once you have selected the Forms server process you wish to trace, then right-click on the row in the processes table, and choose 'Turn On Logging'.



Once logging is enabled, you can view the FRD log from within OEM.

B 1.4 INITIATING FRD FROM THE COMMAND LINE

On NT, this is done by sending a message to the thread ID of the runtime process. Once you know the process ID of the runtime, you'll have to look up the corresponding thread ID in a temporary file written by the runtime process. This file is located in `$orahome/forms60/em`. The name of the file is `em_PID.rti`. The thread ID of the runtime process is one of the entries in this text file. Once the thread ID is found, just execute this command in the cmd shell:

To turn ON dynamic logging:

```
$orahome/bin/ifctrl160 -s thread_ID 1
```

To turn OFF dynamic logging:

```
$orahome/bin/ifctrl160 -s thread_ID 0
```

The log file will be written to `$orahome/forms60/log` directory.

On Solaris, turning on/off dynamic logging is done by sending a signal to the runtime process. Once you know the PID of the runtime, just issue the following commands:

To turn ON dynamic logging:

```
kill -16 PID
```

To turn OFF dynamic logging:

```
kill -1 PID
```

B 2 FRD OUTPUT

B 2.1 INSTRUMENTED EVENTS

The following events are traced by FRD:

- Navigation
- Trigger firing: includes name and hierarchical location.
- Built-in Execution: includes name and IN and OUT parameter types and values.
- Messages: includes message numbers and text
- FMX/MMX file opens.
- Unhandled exceptions: includes error message if available, otherwise only error numbers.

B 2.2 EXPECTED OUTPUT

FRD identifies each event with a sequence number. This sequence number is unique within the runform session. Several pieces of related information may be logged against the same event id.

Event ID

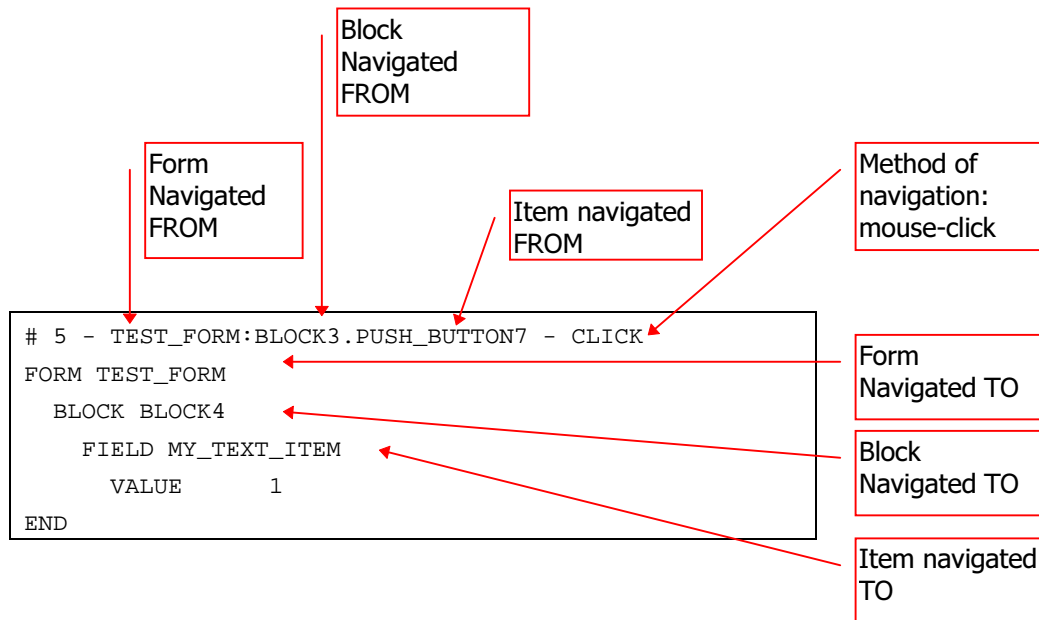
```
# 4 - EMP:EMP.EMPNO - LOV
FORM EMP
  STATUS      CHANGED
BLOCK EMP
  STATUS      CHANGED
  RECSTATUS   CHANGED
  FIELD DEPTNO
    VALUE      10
END

# 4 - EMP:EMP.EMPNO.1225090206
LOV 10 ACCOUNTING
Out Argument 0 - Type: Boolean  Value: TRUE
```

The expected output for each of the FRD events is detailed in the following sections.

B 2.2.1 Navigation

- Form Level - Record form names navigated from and to.
- Block Level - Record block names navigated from and to.
- Item Level - Record item names navigated from and to.



B 2.2.2 Triggers

When a trigger fires, the following information is recorded

```
<trigger_name> Trigger Fired:
Form: <form_name>
Block: <block_name>
Item: <item_name>
```

The level of the trigger corresponds to the last item in the list. A trigger on Form level, therefore, would have only the form name listed below the trigger name. A trigger fired at item level would have the form, block and item specified.

A "State Delta" is written to the log following a trigger fire. A State Delta is a dump of all internal state information that has changed since the occurrence of the previous state delta. The first state delta will dump the entire forms state to the log. The delta mechanism has been used to decrease the size of the output file.

```
# 5 - TEST_FORM:BLOCK3.TEXT_ITEM4.3962681314971500
CLICK TEST_FORM BLOCK3 PUSH_BUTTON7 1 MOUSE
```

```

WHEN-BUTTON-PRESSED Trigger Fired:
Form: TEST_FORM
Block: BLOCK3
Item: PUSH_BUTTON7

State Delta:
FORM TEST_FORM
CURFIELD PUSH_BUTTON7

```

In this example, focus at the start of the event was in BLOCK3.TEXT_ITEM4. The user clicked on BLOCK3.PUSH_BUTTON7, causing the WHEN-BUTTON-PRESSED trigger to fire. At the end of the event, the state delta shows that the focus is now on PUSH_BUTTON7

B 2.2.3 Built-ins

When a PL/SQL Forms built-in is executed, the following output will be logged:

```

Executing <built-in name> Built-In:
In Argument 0 - Type: <type> Value: <value>
In Argument 1 - Type: <type> Value: <value>
Out Argument 0 - Type: <type> Value: <value>

```

Out argument 0 generally represents the function return value.

```

# 20 - EMP:EMP.JOB.1223921846
KEY Next_item

WHEN-VALIDATE-ITEM Trigger Fired:
Form: EMP
Block: EMP
Item: JOB

State Delta:

Executing FIND_ITEM Built-in:
In Argument 0 - Type: String Value: comm
Out Argument 0 - Type: Integer Value: 65544

Executing SET_ITEM_PROPERTY/SET_FIELD Built-in:
In Argument 0 - Type: Integer Value: 65544
In Argument 1 - Type: Number Value: 79
In Argument 2 - Type: String Value: NULL
In Argument 3 - Type: Oracle Number Value: 5
In Argument 4 - Type: Oracle Number Value: 0

```

Note: in the example above, the pl/sql code was

```
set_item_property('comm',enabled,property_false);
```

The call to FIND_ITEM was made implicitly by Forms.

B 2.2.4 Messages

Any messages produced by Forms will appear in the log as it was displayed on screen:

```
Error Message: <message>
```

```
# 33 - EMP:EMP.DEPTNO.1223954763
MENU DEFAULT Action Save
Error Message: FRM-40508: ORACLE error: unable to INSERT record.
```

Note: FRD will record the error message even if the message does not appear on the screen. A common example where this is useful is when :SYSTEM.MESSAGE_LEVEL has been set to a value that would normally suppress the message

B 2.2.5 File Opens

FRD logs attempts to open the following file types⁸:

- libraries (pll/plx)
- icons (ico)
- form binaries (fmx)
- menu binaries (mmx)

Files opened are written to the log in the following format:

```
Opened file: <file_name>
```

```
Opened file: test_form.fmx
```

If a file fails to be opened, the failure is logged as follows:

```
Failed to open file: <file_name>
```

```
Failed to open file: text_form2.fmx
```

⁸ Currently this does not work for all file types. This has been logged as bug 1216294

B 2.2.6 Unhandled Exceptions

When an exception is raised that is not caught, it is logged to the file in the following format:

```
Unhandled Exception: <exception_name>
```

If the name is unavailable, FRD will record the exception number for reference:

```
Unhandled Exception ORA- <exception_number>
```

Unhandled exceptions should be followed in the log by a state delta (see above under Triggers for a description of state delta).

```
# 2 - FRD:BLOCK3.CNT.1227978389
CLICK FRD BLOCK3 PUSH_BUTTON8 1 MOUSE

WHEN-BUTTON-PRESSED Trigger Fired:
Form: FRD
Block: BLOCK3
Item: PUSH_BUTTON8

State Delta:
FORM FRD
  CURFIELD  PUSH_BUTTON8

Unhandled Exception ORA-03114
State Delta:

Error Message: FRM-40735: WHEN-BUTTON-PRESSED trigger raised unhandled exception
ORA-03114.
```

B 2.3 DATA DUMPS

Most events cause a dump of relevant data to be written to the log file.

B2.3.1 Environment Dump

An environment dump is the first entry in the log. It is written only at startup of FRD. It includes the following information:

- File Name
- Process ID of associated runform instance
- User IP address (taken from connection activity log)
- Version information - Forms and all Forms subcomponents

- Values of all environment variables and applicable Oracle registry entries

```
Forms Runtime Diagnostic Collection Log
File Name: /tmp/cxl.log
Process ID: 4774
Forms 6.0 (Forms Runtime) Version 6.0.8.4.0 (Production)
PL/SQL Version 8.0.6.0.0 (Production)
Oracle Virtual Graphics System Version 6.0.5.31.0 (Production)
Oracle Multimedia Version 6.0.5.33.0 (Production)
Oracle Tools Integration Version 6.0.5.32.0 (Production)
Oracle Tools Common Area Version 6.0.5.31.0
Oracle CORE Version 4.0.6.0.0 - Production
```

B 2.3.2 State Dump

State dumps provide a snapshot of the internal context at a given moment in time. The output of a state dump to the log can be triggered by a number of events including the following: on startup, post-trigger fire, on exception or failure.

```
# 3 - EMP:EMP.EMPNO - CLICK
FORM EMP
  STATUS      QUERY
BLOCK EMP
  STATUS      QUERY
  RECSTATUS   QUERY
  FIELD ROWID
    VALUE      AAAAosAABAAAEpmAAA
  FIELD EMPNO
    VALUE      7369
  FIELD ENAME
    VALUE      SMITH
  FIELD JOB
    VALUE      CLERK
  FIELD MGR
    VALUE      7902
  FIELD HIREDATE
    VALUE      17-DEC-1980
  FIELD SAL
    VALUE      800
  FIELD DEPTNO
    VALUE      20
END
```

B 2.3.3 Event / Information Logged Matrix

| Event | Description | Associated Items |
|----------------------|--------------------------------------|---|
| Startup | A new instance of the runtime | environment_dump state_dump |
| Form Navigate | User navigates between forms. | to_form / from_form |
| Block Navigate | User navigates between blocks. | to_form / from_form to_block / from_block |
| Item Navigate | User navigates between items. | to_form / from_form to_block / from_block to_item / from_item |
| Trigger Fire | A trigger is executed. | trigger_name trigger_location state_dump |
| Built-in Execution | Forms Built-ins executed at runtime. | builtin_name arguments return_value |
| File open | Any open call to native OS | file_name file_path |
| Message Sent | A message is sent by forms. | Message_code message_text |
| Unhandled Exceptions | An unhandled exception is raised. | error_code error_text state_dump |

| Item Name | Description |
|-----------|-------------|
|-----------|-------------|

| | |
|------------------|--|
| environment_dump | File name, process ID, user IP, version info, system variables |
| state_dump | Hierarchical and value information. |
| to_form | The name of the form which was navigated to. |
| from_form | The name of the form which was navigated from. |
| to_block | The name of the block which was navigated to. |
| from_block | The name of the block which was navigated from. |
| to_item | The name of the item which was navigated to. |
| from_item | The name of the item which was navigated from. |
| trigger_location | The fully qualified location of the trigger in the module hierarchy. |
| object_name | The name of the object containing the trigger. |
| builtin_name | The name of the built-in which was executed. |
| Arguments | Values of arguments passed to the built-in |
| return_value | The value returned from the built-in |
| file_name | The name of the file opened |
| file_path | The full path of the file opened |
| message_code | The message code (i.e. FRM-12345) |
| message_text | The text of the message. |
| Error_code | The code of the error. |
| Error_text | The text of the error. |

B 3 USING FRD OUTPUT

Once you have all this information how do use it ?

If the FRD log file has a stack trace in it, then this information can be used to search for relevant issues in Metalink (<http://metalink.oracle.com>). The information can also be passed to Oracle Support Services.

By examining the FRD log, you can identify what the last event in the form was, what triggers fired, and what internal navigation took place. This information will help narrow the problem down. Look for patterns in problem reproduction

For Example,

- Does the user experiencing the problem always perform the actions in a specific order ?
- Is that sequence of events the same as for a user who does not experience the same problem ?
- If the problem always happens in a certain trigger, what built-in is forms executing at the point of failure ?
- Are any error messages being reported to the FRD log that are not visible on the screen ?

Such questions help isolate the problem. They may not help resolve it directly, but they help understanding, and this may lead to identification of a workaround. Providing a concise and accurate problem definition will help both Support Services and Forms Development identify a solution to the problem.

B 4 FRD FEATURE / VERSION MATRIX

| Feature | 1.6.1 | 2.1 | 6.0 | 6i |
|--|-------|-----|-----|----|
| Switch FRD on from runform command line/HTML | ✓ | ✓ | ✓ | ✓ |
| Switch FRD on from OEM | ✓ | | | ✓ |
| Switch FRD on from command line utility | ✓ | | | ✓ |
| NT Stack Trace ⁹ | ✓ | ✓ | ✓ | ✓ |
| UNIX Stack Trace ¹⁰ | ✓ | ✓ | ✓ | ✓ |

B 5 FRD CONSTANTS

FRD output frequently includes constant values defined in Forms, for example, when showing the arguments to a Forms builtin. The following section lists the forms constants and their meaning.

```

/*
** Record number constants.
*/
#define IP_CURRENT_RECORD ((ip_plsql_int)0)
/* also a block property */

/*

```

⁹ This does not work correctly. This has been logged as bug 1211969

¹⁰ This does not work correctly when running character-mode forms. This has been logged as bug 1208557.

```

** Constants defined in package TOOLS.
*/
#define IP_TEXT_PARAMETER      ((ip_plsql_int)1)
#define IP_DATA_PARAMETER     ((ip_plsql_int)6)
#define IP_FILESYSTEM         ((ip_plsql_int)11)
#define IP_DB                  ((ip_plsql_int)12)
#define IP_FORMS               ((ip_plsql_int)1)
#define IP_REPORTWRITER       ((ip_plsql_int)2)
#define IP_GRAPHICS            ((ip_plsql_int)3)
#define IP_BOOKVIEWER         ((ip_plsql_int)4)
#define IP_BOOKVIEWER2        ((ip_plsql_int)5)
#define IP_SYNCHRONOUS         ((ip_plsql_int)1)
#define IP_ASYNCHRONOUS       ((ip_plsql_int)2)
#define IP_BATCH               ((ip_plsql_int)21)
#define IP_RUNTIME             ((ip_plsql_int)22)

/*
** Constants defined in ibp40ws
**
** These match the values of synchronous and asynchronous.
*/
#define IP_RESTRICTED          ((ip_plsql_int)1)
#define IP_UNRESTRICTED        ((ip_plsql_int)2)

/*
** Internal constants
*/
#define IP_INTERNAL_NULL_PARAMETERLISTID ((ip_plsql_int)-1)

#define IP_NON_KWD              ((ip_plsql_int)0)
#define IP_ALL_RECORDS          ((ip_plsql_int)-1) /* Execute_, Enter_Query */
#define IP_FOR_UPDATE           ((ip_plsql_int)2) /* Execute_, Enter_Query */
#define IP_NO_SCREEN            ((ip_plsql_int)3) /* Host */
#define IP_ATTR_ON               ((ip_plsql_int)4) /* Set_Field */
#define IP_ATTR_OFF              ((ip_plsql_int)5) /* Set_Field */
#define IP_PROPERTY_ON           ((ip_plsql_int)4) /* Set_XXX_Property */
#define IP_PROPERTY_OFF          ((ip_plsql_int)5) /* Set_XXX_Property */
#define IP_NO_PROMPT            ((ip_plsql_int)60) /* Host */

/*
** Call, Call_Query
*/
#define IP_HIDE                  ((ip_plsql_int)6)
#define IP_NO_HIDE               ((ip_plsql_int)7)

/*

```

```

** Clear_Forms, Clear_Block
*/
#define IP_ASK_COMMIT      ((ip_plsql_int)8)
#define IP_NO_COMMIT      ((ip_plsql_int)9)
#define IP_DO_COMMIT      ((ip_plsql_int)10)
#define IP_NO_VALIDATE    ((ip_plsql_int)11)

/*
** Set_Field, Field_Characteristic, Block_Characteristic attributes
*/
#define IP_AUTO_HELP      ((ip_plsql_int)12)
#define IP_AUTO_SKIP      ((ip_plsql_int)13)
#define IP_BASE_TABLE     ((ip_plsql_int)14)
#define IP_DATATYPE       ((ip_plsql_int)15)
#define IP_DISPLAYED      ((ip_plsql_int)16)
#define IP_DISPLAY_LENGTH ((ip_plsql_int)17)
#define IP_ECHO            ((ip_plsql_int)18)
#define IP_ENTERABLE      ((ip_plsql_int)19)
#define IP_FIXED_LENGTH   ((ip_plsql_int)21)
#define IP_FIELD_LENGTH   ((ip_plsql_int)22)
#define IP_LIST           ((ip_plsql_int)23)
#define IP_PAGE           ((ip_plsql_int)24)
#define IP_PRIMARY_KEY    ((ip_plsql_int)25)
#define IP_QUERY_LENGTH   ((ip_plsql_int)26)
#define IP_QUERYABLE      ((ip_plsql_int)27)
#define IP_REQUIRED       ((ip_plsql_int)28)
#define IP_UPDATEABLE     ((ip_plsql_int)29)
#define IP_UPDATE_NULL    ((ip_plsql_int)30)
#define IP_UPPER_CASE     ((ip_plsql_int)31)
#define IP_X_POS          ((ip_plsql_int)32)
#define IP_Y_POS          ((ip_plsql_int)33)
#define IP_FIRST_FIELD    ((ip_plsql_int)34)
#define IP_LAST_FIELD     ((ip_plsql_int)35)
#define IP_FIRST_ITEM     ((ip_plsql_int)34)
#define IP_LAST_ITEM      ((ip_plsql_int)35)
#define IP_LAST_QUERY     ((ip_plsql_int)510)

/*
** Create_List
*/
#define IP_REMOVE_LIST    ((ip_plsql_int)36)
#define IP_NO_REMOVE_LIST ((ip_plsql_int)37)

/*
** Call, Call_Query
*/
#define IP_DO_REPLACE     ((ip_plsql_int)38)
#define IP_NO_REPLACE     ((ip_plsql_int)39)

```

```

/*
** Block_Characteristic attributes
*/
#define IP_NEXT_BLOCK      ((ip_plsql_int)40)
#define IP_NEXTBLOCK      ((ip_plsql_int)40)
#define IP_PREVIOUS_BLOCK ((ip_plsql_int)41)
#define IP_TOP_RECORD      ((ip_plsql_int)42)

/*
** Field_Characteristic
*/
#define IP_NEXT_FIELD      ((ip_plsql_int)43)
#define IP_PREVIOUS_FIELD ((ip_plsql_int)44)

/*
** Forms_Characteristic
*/
#define IP_FIRST_BLOCK     ((ip_plsql_int)45)
#define IP_LAST_BLOCK      ((ip_plsql_int)46)

/*
** Application_Characteristic
*/
#define IP_CURRENT_FORM    ((ip_plsql_int)47)
#define IP_CALLING_FORM    ((ip_plsql_int)48)

/*
** List_Values
*/
#define IP_DO_RESTRICT     ((ip_plsql_int)49)
#define IP_NO_RESTRICT     ((ip_plsql_int)50)

/*
** Replace_Menu
*/
#define IP_PULL_DOWN       ((ip_plsql_int)51)
#define IP_BAR              ((ip_plsql_int)52)
#define IP_FULL_SCREEN     ((ip_plsql_int)53)

/*
** Block_Characteristic
*/
#define IP_RECORDS_DISPLAYED ((ip_plsql_int)54)

/*
** Execute_Query, Enter_Query
*/

```

```

#define IP_NO_WAIT          ((ip_plsql_int)55)

/*
** Clear_Form, Exit_Form, New_Form
*/
#define IP_TO_SAVEPOINT    ((ip_plsql_int)56)
#define IP_FULL_ROLLBACK   ((ip_plsql_int)57)

/*
** Set_Input_Focus
*/
#define IP_MENU            ((ip_plsql_int)58)

/*
** Exit_Form, New_Form
*/
#define IP_NO_ROLLBACK     ((ip_plsql_int)59)

/*
** Set_Block
*/
#define IP_AUTO            ((ip_plsql_int)62)
#define IP_KEY_MODE        ((ip_plsql_int)63)
#define IP_LOCKING_MODE    ((ip_plsql_int)64)
#define IP_UNIQUE_KEY      ((ip_plsql_int)65)
#define IP_IMMEDIATE       ((ip_plsql_int)66)
#define IP_DELAYED         ((ip_plsql_int)67)

/*
** Set_Form
*/
#define IP_CURSOR_MODE     ((ip_plsql_int)68)
#define IP_SAVEPOINT_MODE  ((ip_plsql_int)69)
#define IP_OPEN_AT_COMMIT  ((ip_plsql_int)70)
#define IP_CLOSE_AT_COMMIT ((ip_plsql_int)71)

/*
** Application characteristics
*/
#define IP_SAVEPOINT_NAME   ((ip_plsql_int)72)
#define IP_USERNAME         ((ip_plsql_int)73)
#define IP_PASSWORD         ((ip_plsql_int)74)
#define IP_CONNECT_STRING   ((ip_plsql_int)75)

/*
** Field_Characteristic
*/

```

```

#define IP_DATABASE_VALUE      ((ip_plsql_int)76)

/*
** Set_Block
*/
#define IP_UPDATEABLE_PRIMARY_KEY      ((ip_plsql_int)77)
#define IP_NON_UPDATEABLE_PRIMARY_KEY  ((ip_plsql_int)78)

/*
** Item characteristics
*/
#define IP_ENABLED                ((ip_plsql_int)79)
#define IP_NAVIGABLE              ((ip_plsql_int)80)

/*
** Create_Timer
*/
#define IP_NO_CHANGE              ((ip_plsql_int)-1)
#define IP_REPEAT                 ((ip_plsql_int)82)
#define IP_NO_REPEAT             ((ip_plsql_int)83)

/*
** Application Characteristic
*/
#define IP_TIMER_NAME            ((ip_plsql_int)84)

/*
** Set/Get block characteristics
*/
#define IP_QUERY_HITS            ((ip_plsql_int)85)
#define IP_QUERY_OPTIONS        ((ip_plsql_int)86)
#define IP_RECORDS_TO_FETCH     ((ip_plsql_int)87)

/* really UPDATE_CHANGED_COLUMNS */
#define IP_UPDATE_CHANGED_ONLY  ((ip_plsql_int)88)

/*
** Show_Alert return values
*/
#define IP_ALERT_BUTTON1        ((ip_plsql_int)88)
#define IP_ALERT_BUTTON2        ((ip_plsql_int)89)
#define IP_ALERT_BUTTON3        ((ip_plsql_int)90)

#define IP_BOOLEAN_PARAMETER    ((ip_plsql_int)91)      /* Add_Parameter */
#define IP_DATE_PARAMETER       ((ip_plsql_int)92)      /* Add_Parameter */

#define IP_BLOCK_NAME           ((ip_plsql_int)93)      /* Get_Item */

```

```

/*
** Status constants (100-120)
*/
#define IP_STATUS                ((ip_plsql_int)100)
#define IP_INSERT_STATUS        ((ip_plsql_int)101)
#define IP_CHANGED_STATUS       ((ip_plsql_int)102)
#define IP_QUERY_STATUS         ((ip_plsql_int)103)
#define IP_NEW_STATUS           ((ip_plsql_int)104)

/*
** Message line constants (121-140)
*/
#define IP_ACKNOWLEDGE          ((ip_plsql_int)121)
#define IP_NO_ACKNOWLEDGE       ((ip_plsql_int)122)
#define IP_ACKNOWLEDGE_PREVIOUS ((ip_plsql_int)123)
#define IP_NO_ACKNOWLEDGE_PREVIOUS ((ip_plsql_int)124)

/*
** Misc properties
*/
#define IP_INITIAL_VALUE        ((ip_plsql_int)141) /* Get_Field_Property */
#define IP_UNUSED_142           ((ip_plsql_int)142)
#define IP_GROUP_NAME           ((ip_plsql_int)143) /* Get_LOV_Property */
#define IP_LOV_SIZE             ((ip_plsql_int)144) /* Set_LOV_Property */
#define IP_AUTO_REFRESH         ((ip_plsql_int)145) /* Get/Set_LOV_Property */
#define IP_AUTO_DISPLAY         ((ip_plsql_int)146) /* Get/Set_LOV_Property */
#define IP_AUTO_CONFIRM         ((ip_plsql_int)147) /* Get/Set_LOV_Property */
#define IP_LONGLIST             ((ip_plsql_int)148) /* Get/Set_LOV_Property */

#define IP_RETURNITEM           ((ip_plsql_int)149) /* Get/Set_LOV_Column_Property */

/*
** Relation constants (151-180)
*/
#define IP_MASTER_NAME          ((ip_plsql_int)151)
#define IP_DETAIL_NAME          ((ip_plsql_int)152)
#define IP_NEXT_RELATION        ((ip_plsql_int)153)
#define IP_NEXT_MASTER_RELATION IP_NEXT_RELATION
#define IP_MASTER_DELETES       ((ip_plsql_int)154)
#define IP_DEFERRED_COORDINATION ((ip_plsql_int)155)
#define IP_AUTOQUERY            ((ip_plsql_int)156)
#define IP_ISOLATED             ((ip_plsql_int)157)
#define IP_NON_ISOLATED         ((ip_plsql_int)158)
#define IP_CASCADING            ((ip_plsql_int)159)
#define IP_FIRST_RELATION        ((ip_plsql_int)160)
#define IP_COORDINATED          ((ip_plsql_int)161)
#define IP_COORDINATION_OPERATION ((ip_plsql_int)162)
#define IP_NON_COORDINATED      ((ip_plsql_int)163)

```



```

#define IP_COORDINATION_STATUS          ((ip_plsql_int)164)
#define IP_NEXT_DETAIL_RELATION        ((ip_plsql_int)165)
#define IP_PREVENT_MASTERLESS_OPERATION ((ip_plsql_int)166)
#define IP_DEFER_UNTIL_VISIBLE         ((ip_plsql_int)167)

/*
** Block Properties (181-250)
*/
#define IP_ORDER_BY                    ((ip_plsql_int)181)
#define IP_DELETE_ALLOWED              ((ip_plsql_int)182)
#define IP_UPDATE_ALLOWED              ((ip_plsql_int)183)
#define IP_INSERT_ALLOWED              ((ip_plsql_int)184)
#define IP_COLUMN_SECURITY             ((ip_plsql_int)185)
#define IP_OPTIMIZER_HINT              ((ip_plsql_int)186)
#define IP_FIRST_MASTER_RELATION       IP_FIRST_RELATION
#define IP_FIRST_DETAIL_RELATION       ((ip_plsql_int)187)
#define IP_NEXT_NAVIGATION_BLOCK       ((ip_plsql_int)188)
#define IP_PREVIOUS_NAVIGATION_BLOCK   ((ip_plsql_int)189)
#define IP_DEFAULT_WHERE               ((ip_plsql_int)190)
#define IP_NAVIGATION_STYLE            ((ip_plsql_int)191)
#define IP_SAME_RECORD                 ((ip_plsql_int)192)
#define IP_CHANGE_RECORD               ((ip_plsql_int)193)
#define IP_CHANGE_BLOCK                ((ip_plsql_int)194)
#define IP_QUERY_ALLOWED               ((ip_plsql_int)195)
#define IP_QUERY_SOURCE                ((ip_plsql_int)196)
#define IP_QUERY_SOURCE_TYPE           ((ip_plsql_int)197)
#define IP_INSERT_SOURCE               ((ip_plsql_int)198)
#define IP_INSERT_SOURCE_TYPE          ((ip_plsql_int)199)
#define IP_UPDATE_SOURCE               ((ip_plsql_int)200)
#define IP_UPDATE_SOURCE_TYPE          ((ip_plsql_int)201)
#define IP_DELETE_SOURCE               ((ip_plsql_int)202)
#define IP_DELETE_SOURCE_TYPE          ((ip_plsql_int)203)
#define IP_TABLE_SOURCE                ((ip_plsql_int)204)
#define IP_PROCEDURE_SOURCE            ((ip_plsql_int)205)
#define IP_ALIAS                       ((ip_plsql_int)206)
#define IP_BLOCKSCROLLBAR_POSITION    ((ip_plsql_int)207)
#define IP_BLOCKSCROLLBAR_X_POS       ((ip_plsql_int)208)
#define IP_BLOCKSCROLLBAR_Y_POS       ((ip_plsql_int)209)
#define IP_FIRST_BUTTON                ((ip_plsql_int)210)
#define IP_NEXT_BUTTON                 ((ip_plsql_int)211)
#define IP_SELECTED_RADIO_BUTTON       ((ip_plsql_int)212)
#define IP_ICON_IN_HTOOLBAR           ((ip_plsql_int)213)
#define IP_ICON_IN_VTOOLBAR           ((ip_plsql_int)214)
#define IP_ICON_IN_MENU                ((ip_plsql_int)215)

/*
** Application properties (251-300)
*/

```

```

#define IP_OPERATING_SYSTEM ((ip_plsql_int)251)
#define IP_TOOLKIT ((ip_plsql_int)252)
#define IP_DATASOURCE ((ip_plsql_int)253)
#define IP_VERSION ((ip_plsql_int)254)
#define IP_DISPLAY_WD ((ip_plsql_int)255)
#define IP_DISPLAY_HT ((ip_plsql_int)256)
#define IP_CURRENT_FORM_NAME ((ip_plsql_int)257)
#define IP_CALLING_FORM_NAME ((ip_plsql_int)258)
#define IP_USER-NLS_CHARACTER_SET ((ip_plsql_int)259)
#define IP_USER-NLS_LANG ((ip_plsql_int)260)
#define IP_USER-NLS_LANGUAGE ((ip_plsql_int)261)
#define IP_USER-NLS_TERRITORY ((ip_plsql_int)262)
#define IP_CURSOR_STYLE ((ip_plsql_int)263)
#define IP_BUILTIN_DATE_FORMAT ((ip_plsql_int)264)
#define IP_BUILTIN_DECIMAL_CHAR ((ip_plsql_int)265)
#define IP_MENU_BUFFERING ((ip_plsql_int)266)
#define IP_DB_DESIGN_PROPERTIES ((ip_plsql_int)267)
#define IP_PLSQL_DATE_FORMAT ((ip_plsql_int)268)
#define IP_USER-NLS_DATE_FORMAT ((ip_plsql_int)269)
#define IP_DATE_FORMAT_COMPATIBILITY_MODE ((ip_plsql_int)270)
#define IP_USER_DATE_FORMAT ((ip_plsql_int)271)
#define IP_USER_DATETIME_FORMAT ((ip_plsql_int)272)
#define IP_OUTPUT_DATE_FORMAT ((ip_plsql_int)273)
#define IP_OUTPUT_DATETIME_FORMAT ((ip_plsql_int)274)
#define IP_PLSQL_NUMERIC_CHARACTERS ((ip_plsql_int)275)
#define IP_FLAG_USER_VALUE_TOO_LONG ((ip_plsql_int)276)

/*
** Form Properties (301-400)
*/
#define IP_DEFAULT ((ip_plsql_int)301)
#define IP_FORM ((ip_plsql_int)302)
#define IP_BLOCK ((ip_plsql_int)303)
#define IP_RECORD ((ip_plsql_int)304)
#define IP_ITEM ((ip_plsql_int)305)
#define IP_VALIDATION_UNIT ((ip_plsql_int)306)
#define IP_FIRST_NAVIGATION_BLOCK ((ip_plsql_int)307)
#define IP_FORM_NAME ((ip_plsql_int)308)
#define IP_FILE_NAME ((ip_plsql_int)309)
#define IP_GLOBAL ((ip_plsql_int)310)
#define IP_MODULE-NLS_CHARACTER_SET ((ip_plsql_int)311)
#define IP_MODULE-NLS_LANG ((ip_plsql_int)312)
#define IP_MODULE-NLS_LANGUAGE ((ip_plsql_int)313)
#define IP_MODULE-NLS_TERRITORY ((ip_plsql_int)314)
#define IP_ERROR_NOTIFICATION ((ip_plsql_int)315)
#define IP_GENERAL_EXCEPTION ((ip_plsql_int)316)
#define IP_NO_EXCEPTION ((ip_plsql_int)317)
#define IP_SPECIFIC_EXCEPTION ((ip_plsql_int)318)

```

```

#define IP_COORDINATE_SYSTEM          ((ip_plsql_int)319)
#define IP_CHARACTER_CELL_WIDTH      ((ip_plsql_int)320)
#define IP_CHARACTER_CELL_HEIGHT    ((ip_plsql_int)321)
#define IP_BLOCKING                  ((ip_plsql_int)322)
#define IP_NON_BLOCKING              ((ip_plsql_int)323)
#define IP_PROPERTY_4_5              ((ip_plsql_int)324)

/*
** Item Properties (401-500)
*/
#define IP_CASE_INSENSITIVE_QUERY    ((ip_plsql_int)401)
#define IP_LOCK_RECORD_ON_CHANGE     ((ip_plsql_int)402)
#define IP_DERIVED_COLUMN            ((ip_plsql_int)403)
#define IP_HINT_TEXT                 ((ip_plsql_int)404)
#define IP_LOV_NAME                   ((ip_plsql_int)405)
#define IP_ALTERABLE                  ((ip_plsql_int)406)
#define IP_FORMAT_MASK                ((ip_plsql_int)407)
#define IP_FORMATTED_VALUE           ((ip_plsql_int)408)
#define IP_UNFORMATTED_VALUE         ((ip_plsql_int)409)
#define IP_UPDATE_PERMISSION         ((ip_plsql_int)410)
#define IP_ITEM_IS_VALID              ((ip_plsql_int)411)
#define IP_VALUE_CHANGED              ((ip_plsql_int)413) /* UPDATE_COLUMN */
#define IP_NEXT_NAVIGABLE_ITEM       ((ip_plsql_int)414)
#define IP_PREVIOUS_NAVIGABLE_ITEM   ((ip_plsql_int)415)
#define IP_OLE_IUNKNOWN_PTR          ((ip_plsql_int)416)
#define IP_CMP_QUALITY                ((ip_plsql_int)417)
#define IP_IMAGE_DEPTH               ((ip_plsql_int)418)
#define IP_IMAGE_FORMAT               ((ip_plsql_int)419)

/*
** Misc
*/
#define IP_QUERY_ONLY                ((ip_plsql_int)501)
#define IP_NO_QUERY_ONLY              ((ip_plsql_int)502)
#define IP_DEVELOPER_NAMES            ((ip_plsql_int)503)
#define IP_ENABLE_VALIDATION          ((ip_plsql_int)504)
#define IP_DISABLE_VALIDATION         ((ip_plsql_int)505)
#define IP_VALIDATION                 ((ip_plsql_int)506)
#define IP_DEFER_REQUIRED_ENFORCEMENT ((ip_plsql_int)507)
#define IP_CURRENT_ROW_VA             ((ip_plsql_int)508)

/*
** Open Form (peer form)
*/
#define IP_NO_ACTIVATE                ((ip_plsql_int)511)
#define IP_ACTIVATE                   ((ip_plsql_int)512)
#define IP_SESSION                    ((ip_plsql_int)513)
#define IP_NO_SESSION                 ((ip_plsql_int)514)

```

```

#define IP_SHAREDATA ((ip_plsql_int)515)
#define IP_NO_SHAREDATA ((ip_plsql_int)516)

/*
** Forms_Characteristic (continued)
*/
#define IP_INTERACTION_MODE ((ip_plsql_int)517)
#define IP_ISOLATION_MODE ((ip_plsql_int)518)
#define IP_MAX_QUERY_TIME ((ip_plsql_int)519)
#define IP_MAX_RECORDS_FETCHED ((ip_plsql_int)520)

/*
** Misc (continued)
*/
#define IP_DIRECTION_DEFAULT ((ip_plsql_int)521)
#define IP_ALTERABLE_PLUS ((ip_plsql_int)522)
#define IP_ENTERABLE_PLUS ((ip_plsql_int)523)
#define IP_DISPLAYED_PLUS ((ip_plsql_int)524)
#define IP_BALLOON_TEXT ((ip_plsql_int)525)
#define IP_BALLOON_ATTRIBUTE ((ip_plsql_int)526)
#define IP_HORIZONTAL ((ip_plsql_int)527)
#define IP_VERTICAL ((ip_plsql_int)528)
#define IP_TOPMOST_TAB_PAGE ((ip_plsql_int)529)
#define IP_ITEM_TAB_PAGE ((ip_plsql_int)530)

#define IP_RECORDS_TO_INSERT ((ip_plsql_int)531)
#define IP_RECORDS_TO_UPDATE ((ip_plsql_int)532)
#define IP_RECORDS_TO_DELETE ((ip_plsql_int)533)

#define IP_TAB_X_OFFSET ((ip_plsql_int)535)
#define IP_TAB_Y_OFFSET ((ip_plsql_int)536)

#define IP_HELP_TOPIC ((ip_plsql_int)541)
#define IP_HELPBOOK_TITLE ((ip_plsql_int)542)

#define IP_UNUSED_551 ((ip_plsql_int)551)
#define IP_UNUSED_552 ((ip_plsql_int)552)
#define IP_UNUSED_553 ((ip_plsql_int)553)
#define IP_UNUSED_554 ((ip_plsql_int)554)
#define IP_UNUSED_555 ((ip_plsql_int)555)
#define IP_UNUSED_556 ((ip_plsql_int)556)
#define IP_UNUSED_557 ((ip_plsql_int)557)
#define IP_UNUSED_558 ((ip_plsql_int)558)
#define IP_UNUSED_559 ((ip_plsql_int)559)
#define IP_UNUSED_560 ((ip_plsql_int)560)
#define IP_UNUSED_561 ((ip_plsql_int)561)
#define IP_UNUSED_562 ((ip_plsql_int)562)
#define IP_UNUSED_563 ((ip_plsql_int)563)

```

```

#define IP_UNUSED_564                ((ip_plsql_int)564)
#define IP_UNUSED_565                ((ip_plsql_int)565)
#define IP_UNUSED_566                ((ip_plsql_int)566)
#define IP_UNUSED_567                ((ip_plsql_int)567)
#define IP_UNUSED_568                ((ip_plsql_int)568)
#define IP_UNUSED_569                ((ip_plsql_int)569)
#define IP_UNUSED_570                ((ip_plsql_int)570)

#define IP_ALIGNMENT_LEFT            ((ip_plsql_int)571)
#define IP_ALIGNMENT_RIGHT          ((ip_plsql_int)572)
#define IP_ALIGNMENT_CENTER          ((ip_plsql_int)573)
#define IP_ALIGNMENT_START           ((ip_plsql_int)574)
#define IP_ALIGNMENT_END             ((ip_plsql_int)575)
#define IP_LEFT_TO_RIGHT             ((ip_plsql_int)576)
#define IP_RIGHT_TO_LEFT            ((ip_plsql_int)577)
#define IP_DIRECTION                 ((ip_plsql_int)578)

#define IP_UNUSED_581                ((ip_plsql_int)581)
#define IP_UNUSED_582                ((ip_plsql_int)582)
#define IP_UNUSED_583                ((ip_plsql_int)583)
#define IP_UNUSED_584                ((ip_plsql_int)584)
#define IP_UNUSED_585                ((ip_plsql_int)585)
#define IP_UNUSED_586                ((ip_plsql_int)586)
#define IP_UNUSED_587                ((ip_plsql_int)587)
#define IP_UNUSED_588                ((ip_plsql_int)588)
#define IP_UNUSED_589                ((ip_plsql_int)589)
#define IP_UNUSED_590                ((ip_plsql_int)590)

/*
** Block_Characteristic attributes (continued)
*/
#define IP_QUERY_DATA_SOURCE_NAME    ((ip_plsql_int)591)
#define IP_QUERY_DATA_SOURCE_TYPE    ((ip_plsql_int)592)
#define IP_DML_DATA_TARGET_NAME      ((ip_plsql_int)593)
#define IP_DML_DATA_TARGET_TYPE      ((ip_plsql_int)594)

/*
** Misc (continued)
*/
#define IP_RAISED                     ((ip_plsql_int)595)
#define IP_LOWERED                    ((ip_plsql_int)596)
#define IP_PLAIN                      ((ip_plsql_int)537)

/*
** Group constants
*/
#define IP_END_OF_GROUP                ((ip_plsql_int)-2) /* used by Add_Group_Row */

```

```

#define IP_CHAR_COLUMN      ((ip_plsql_int)1300) /* CHAR col selector */
#define IP_DATE_COLUMN     ((ip_plsql_int)1301) /* DATE col selector */
#define IP_NUMBER_COLUMN   ((ip_plsql_int)1302) /* NUMBER col selector */
#define IP_LONG_COLUMN     ((ip_plsql_int)1303) /* LONG col selector */

/*
** Canvas && Window constants (1400-1499)
*/
#define IP_DISPLAY          ((ip_plsql_int)1400)
#define IP_VISIBLE         ((ip_plsql_int)1401)
#define IP_POSITION        ((ip_plsql_int)1402)
#define IP_WINDOW_SIZE     ((ip_plsql_int)1403)
#define IP_VIEW_SIZE       ((ip_plsql_int)1404)
#define IP_POSITION_ON_CANVAS ((ip_plsql_int)1405)
#define IP_X_POS_ON_CANVAS ((ip_plsql_int)1406)
#define IP_Y_POS_ON_CANVAS ((ip_plsql_int)1407)
#define IP_DISPLAY_POSITION ((ip_plsql_int)1408)
#define IP_DISPLAY_X_POS    ((ip_plsql_int)1409)
#define IP_DISPLAY_Y_POS    ((ip_plsql_int)1410)
#define IP_CANVAS_SIZE      ((ip_plsql_int)1411)
#define IP_TITLE            ((ip_plsql_int)1412)
#define IP_WINDOW_NAME     ((ip_plsql_int)1413)
#define IP_STACKED         ((ip_plsql_int)1414)

#define IP_WINDOW_X_POS    ((ip_plsql_int)1416)
#define IP_WINDOW_Y_POS    ((ip_plsql_int)1417)
#define IP_WINDOW_WIDTH    ((ip_plsql_int)1418)
#define IP_WINDOW_HEIGHT   ((ip_plsql_int)1419)
#define IP_WINDOW_POSITION ((ip_plsql_int)1420)

#define IP_PROPERTY_MAXIMIZE ((ip_plsql_int)1421)
#define IP_PROPERTY_MINIMIZE ((ip_plsql_int)1422)
#define IP_PROPERTY_NORMAL   ((ip_plsql_int)1423)
#define IP_WINDOW_STATE      ((ip_plsql_int)1424)
#define IP_REMOVE_ON_EXIT    ((ip_plsql_int)1425)

/*
** Get/Set_Item_Property (1500-1599)
*/
#define IP_NEXT_ITEM        ((ip_plsql_int)1501)
#define IP_PREVIOUS_ITEM    ((ip_plsql_int)1502)
#define IP_ITEM_SIZE        ((ip_plsql_int)1503)
#define IP_ITEM_TYPE        ((ip_plsql_int)1504)
#define IP_LABEL            ((ip_plsql_int)1505)
#define IP_ICONIC_BUTTON    ((ip_plsql_int)1506)
#define IP_MAX_LENGTH       ((ip_plsql_int)1507)
#define IP_WIDTH            ((ip_plsql_int)1508)

```

```

#define IP_HEIGHT ((ip_plsql_int)1509)
#define IP_ITEM_NAME ((ip_plsql_int)1510)
#define IP_SECURE ((ip_plsql_int)1511)
#define IP_ITEM_CANVAS ((ip_plsql_int)1512)
#define IP_WRAP_STYLE ((ip_plsql_int)1513)
#define IP_KEEP_POSITION ((ip_plsql_int)1514)
#define IP_SCROLLBAR ((ip_plsql_int)1515)
#define IP_AUTO_NAVIGATE ((ip_plsql_int)1516)
#define IP_ALIGNMENT ((ip_plsql_int)1517)
#define IP_CASE_RESTRICTION ((ip_plsql_int)1518)
#define IP_BORDER_BEVEL ((ip_plsql_int)1519)
#define IP_MULTI_LINE ((ip_plsql_int)1520)
#define IP_AUTO_HINT ((ip_plsql_int)1521)
#define IP_UPPERCASE ((ip_plsql_int)1522)
#define IP_LOWERCASE ((ip_plsql_int)1523)
#define IP_NONE ((ip_plsql_int)1524)
#define IP_VISUAL_ATTRIBUTE ((ip_plsql_int)1525)
#define IP_ITEM_DEFAULT_VALUE ((ip_plsql_int)1526)
#define IP_ICON_NAME ((ip_plsql_int)1527)
#define IP_RANGE_LOW ((ip_plsql_int)1528)
#define IP_RANGE_HIGH ((ip_plsql_int)1529)
#define IP_ENFORCE_KEY ((ip_plsql_int)1530)
#define IP_EDITOR_NAME ((ip_plsql_int)1531)
#define IP_EDITOR_XPOS ((ip_plsql_int)1532)
#define IP_EDITOR_YPOS ((ip_plsql_int)1533)
#define IP_LOV_XPOS ((ip_plsql_int)1534)
#define IP_LOV_YPOS ((ip_plsql_int)1535)
#define IP_WINDOW_HANDLE ((ip_plsql_int)1536)
#define IP_APPLICATION_INSTANCE ((ip_plsql_int)1537)
#define IP_MNEMONIC ((ip_plsql_int)1538)
#define IP_SHOW_POPUPMENU ((ip_plsql_int)1539)
#define IP_POPUPMENU_CUT_ITEM ((ip_plsql_int)1540)
#define IP_POPUPMENU_COPY_ITEM ((ip_plsql_int)1541)
#define IP_POPUPMENU_PASTE_ITEM ((ip_plsql_int)1542)
#define IP_POPUPMENU_PASTESPEC_ITEM ((ip_plsql_int)1543)
#define IP_POPUPMENU_INSOBJ_ITEM ((ip_plsql_int)1544)
#define IP_POPUPMENU_DELOBJ_ITEM ((ip_plsql_int)1545)
#define IP_POPUPMENU_LINKS_ITEM ((ip_plsql_int)1546)
#define IP_POPUPMENU_OBJECT_ITEM ((ip_plsql_int)1547)
#define IP_LOV_VALIDATION ((ip_plsql_int)1548)
#define IP_SHOW_PALETTE ((ip_plsql_int)1549)
#define IP_PROMPT_TEXT ((ip_plsql_int)1550)
#define IP_PROMPT_EDGE ((ip_plsql_int)1551)
#define IP_PROMPT_EDGE_OFFSET ((ip_plsql_int)1552)
#define IP_PROMPT_EDGE_ALIGNMENT ((ip_plsql_int)1553)
#define IP_PROMPT_ALIGNMENT_OFFSET ((ip_plsql_int)1554)
#define IP_PROMPT_TEXT_ALIGNMENT ((ip_plsql_int)1555)
#define IP_PROMPT_DISPLAY_STYLE ((ip_plsql_int)1556)

```

```

#define IP_PROMPT_VISUAL_ATTRIBUTE      ((ip_plsql_int)1557)
#define IP_SHOW_PLAY_BUTTON            ((ip_plsql_int)1558)
#define IP_SHOW_REWIND_BUTTON         ((ip_plsql_int)1559)
#define IP_SHOW_FAST_FORWARD_BUTTON   ((ip_plsql_int)1560)
#define IP_SHOW_RECORD_BUTTON         ((ip_plsql_int)1561)
#define IP_SHOW_VOLUME_CONTROL        ((ip_plsql_int)1562)
#define IP_SHOW_TIME_INDICATOR        ((ip_plsql_int)1563)
#define IP_SHOW_SLIDER                 ((ip_plsql_int)1564)
#define IP_MERGE_VISUAL_ATTRIBUTE     ((ip_plsql_int)1565)
#define IP_MERGE_PROMPT_VISUAL_ATTRIBUTE ((ip_plsql_int)1566)
#define IP_MERGE_CURRENT_ROW_VA       ((ip_plsql_int)1567)
#define IP_MERGE_BALLOON_ATTRIBUTE    ((ip_plsql_int)1568)

/*
** IMAGE Zoom constants
*/
#define IP_ZOOM_IN                     ((ip_plsql_int)-1)
#define IP_ZOOM_OUT                    ((ip_plsql_int)-2)
#define IP_SELECTION_RECTANGLE        ((ip_plsql_int)-3)
#define IP_ADJUST_TO_FIT               ((ip_plsql_int)-4)
#define IP_ZOOM_PERCENT                 ((ip_plsql_int)-5)

/*
**      OLE Object Set_Item_Property constants
*/
#define IP_POPUP_MENUITEM_HIDDEN      ((ip_plsql_int)-3)
#define IP_POPUP_MENUITEM_DISABLED   ((ip_plsql_int)-2)
#define IP_FILENAME                   ((ip_plsql_int)100)
#define IP_CLASSNAME                   ((ip_plsql_int)101)

/*
** Get/Set_Menu_Item_Property (1600-1699)
*/
#define IP_CHECKED                     ((ip_plsql_int)1600)
#define IP_COLUMN_NAME                 ((ip_plsql_int)1601)

/*
** Set_Alert_Property (1700-1799)
*/
#define IP_ALERT_MESSAGE_TEXT          ((ip_plsql_int)1700)
#define IP_ALERT_DEFAULT               ((ip_plsql_int)1701)

/*
** Get/Set_Report_Object_Property (1800-1899)
*/
#define IP_REPORT_FILENAME              ((ip_plsql_int)1800)
#define IP_REPORT_EXECUTION_MODE       ((ip_plsql_int)1801)
#define IP_REPORT_COMM_MODE            ((ip_plsql_int)1802)

```



```

#define IP_REPORT_SOURCE_BLOCK ((ip_plsql_int)1803)
#define IP_REPORT_QUERY_NAME ((ip_plsql_int)1804)
#define IP_REPORT_DESNAME ((ip_plsql_int)1805)
#define IP_REPORT_DESTYPE ((ip_plsql_int)1806)
#define IP_REPORT_DESFORMAT ((ip_plsql_int)1807)
#define IP_REPORT_SERVER ((ip_plsql_int)1808)
#define IP_REPORT_OTHER ((ip_plsql_int)1809)
#define IP_PREVIEW ((ip_plsql_int)1810)
#define IP_FILE ((ip_plsql_int)1811)
#define IP_PRINTER ((ip_plsql_int)1812)
#define IP_MAIL ((ip_plsql_int)1813)
#define IP_CACHE ((ip_plsql_int)1814)
#define IP_SCREEN ((ip_plsql_int)1815)

/*
** Get/Set_xx_Property - visual properties (1900-1999)
*/
#define IP_FONT_NAME ((ip_plsql_int)1900)
#define IP_FONT_SIZE ((ip_plsql_int)1901)
#define IP_FONT_WEIGHT ((ip_plsql_int)1902)
#define IP_FONT_STYLE ((ip_plsql_int)1903)
#define IP_FONT_SPACING ((ip_plsql_int)1904)
#define IP_FOREGROUND_COLOR ((ip_plsql_int)1905)
#define IP_BACKGROUND_COLOR ((ip_plsql_int)1906)
#define IP_FILL_PATTERN ((ip_plsql_int)1907)
#define IP_WHITE_ON_BLACK ((ip_plsql_int)1908)
#define IP_PROMPT_FONT_NAME ((ip_plsql_int)1909)
#define IP_PROMPT_FONT_SIZE ((ip_plsql_int)1910)
#define IP_PROMPT_FONT_WEIGHT ((ip_plsql_int)1911)
#define IP_PROMPT_FONT_STYLE ((ip_plsql_int)1912)
#define IP_PROMPT_FONT_SPACING ((ip_plsql_int)1913)
#define IP_PROMPT_FG_COLOR ((ip_plsql_int)1914)
#define IP_PROMPT_BG_COLOR ((ip_plsql_int)1915)
#define IP_PROMPT_FILL_PATTERN ((ip_plsql_int)1916)
#define IP_PROMPT_WHT_ON_BLK ((ip_plsql_int)1917)
#define IP_TOOLTIP_FONT_NAME ((ip_plsql_int)1918)
#define IP_TOOLTIP_FONT_SIZE ((ip_plsql_int)1919)
#define IP_TOOLTIP_FONT_WEIGHT ((ip_plsql_int)1920)
#define IP_TOOLTIP_FONT_STYLE ((ip_plsql_int)1921)
#define IP_TOOLTIP_FONT_SPACING ((ip_plsql_int)1922)
#define IP_TOOLTIP_FG_COLOR ((ip_plsql_int)1923)
#define IP_TOOLTIP_BG_COLOR ((ip_plsql_int)1924)
#define IP_TOOLTIP_FILL_PATTERN ((ip_plsql_int)1925)
#define IP_TOOLTIP_WHT_ON_BLK ((ip_plsql_int)1926)
#define IP_CURRENT_ROW_FONT_NAME ((ip_plsql_int)1927)
#define IP_CURRENT_ROW_FONT_SIZE ((ip_plsql_int)1928)
#define IP_CURRENT_ROW_FONT_WEIGHT ((ip_plsql_int)1929)
#define IP_CURRENT_ROW_FONT_STYLE ((ip_plsql_int)1930)

```

```

#define IP_CURRENT_ROW_FONT_SPACING ((ip_plsql_int)1931)
#define IP_CURRENT_ROW_FG_COLOR ((ip_plsql_int)1932)
#define IP_CURRENT_ROW_BG_COLOR ((ip_plsql_int)1933)
#define IP_CURRENT_ROW_FILL_PATTERN ((ip_plsql_int)1934)
#define IP_CURRENT_ROW_WHT_ON_BLK ((ip_plsql_int)1935)

/*
** Get/Set_Graphics_Property (2000-2099)
*/
#define IP_GRAPHICS_TEXT ((ip_plsql_int)2000)
#define IP_GRAPHICS_TYPE ((ip_plsql_int)2001)

/*
** Hierarchical Tree properties (2100-2199)
*/
#define IP_RECORD_GROUP ((ip_plsql_int)2100)
#define IP_QUERY_TEXT ((ip_plsql_int)2101)
#define IP_ALLOW_EMPTY_BRANCHES ((ip_plsql_int)2102)
#define IP_ALLOW_MULTI_SELECT ((ip_plsql_int)2103)
#define IP_NODE_STATE ((ip_plsql_int)2104)
#define IP_NODE_DEPTH ((ip_plsql_int)2105)
#define IP_NODE_LABEL ((ip_plsql_int)2106)
#define IP_NODE_ICON ((ip_plsql_int)2107)
#define IP_NODE_VALUE ((ip_plsql_int)2108)
#define IP_NODE_COUNT ((ip_plsql_int)2109)
#define IP_SELECTION_COUNT ((ip_plsql_int)2110)
#define IP_DATA_SOURCE ((ip_plsql_int)2111)

/*
** Tree Node offsets.
*/
#define IP_PARENT_OFFSET ((ip_plsql_int)1)
#define IP_SIBLING_OFFSET ((ip_plsql_int)2)
#define IP_LAST_CHILD ((ip_plsql_int)0)
#define IP_PREVIOUS_NODE ((ip_plsql_int)-1)
#define IP_NEXT_NODE ((ip_plsql_int)-2)
#define IP_FIRST_CHILD ((ip_plsql_int)-3)

/*
** Tree search types.
*/
#define IP_FIND_NEXT ((ip_plsql_int)1)
#define IP_FIND_NEXT_CHILD ((ip_plsql_int)2)

/*
** Tree Node states.
*/
#define IP_EXPANDED_NODE ((ip_plsql_int)1)

```

```

#define IP_COLLAPSED_NODE ((ip_plsql_int)-1)
#define IP_LEAF_NODE ((ip_plsql_int)0)

/*
** Tree Node selection types.
*/
#define IP_SELECT_ON ((ip_plsql_int)0)
#define IP_SELECT_OFF ((ip_plsql_int)1)
#define IP_SELECT_TOGGLE ((ip_plsql_int)2)

/*
** Tree Node deletion types.
*/
#define IP_NODE_AND_CHILDREN ((ip_plsql_int)1)
#define IP_CHILDREN_ONLY ((ip_plsql_int)2)

/*
** Get_File_Name constants
*/
#define IP_OPEN_FILE ((ip_plsql_int)1)
#define IP_SAVE_FILE ((ip_plsql_int)2)

/*
** Write_Image_File constants
*/
#define IP_ORIGINAL_DEPTH ((ip_plsql_int)0)
#define IP_MONOCHROME ((ip_plsql_int)1)
#define IP_GRAYSCALE ((ip_plsql_int)2)
#define IP_LUT ((ip_plsql_int)3)
#define IP_RGB ((ip_plsql_int)4)
#define IP_NO_COMPRESSION ((ip_plsql_int)0)
#define IP_MINIMIZE_COMPRESSION ((ip_plsql_int)1)
#define IP_LOW_COMPRESSION ((ip_plsql_int)2)
#define IP_MEDIUM_COMPRESSION ((ip_plsql_int)3)
#define IP_HIGH_COMPRESSION ((ip_plsql_int)4)
#define IP_MAXIMIZE_COMPRESSION ((ip_plsql_int)5)

/*
** Write_Sound_File constants
*/
#define IP_ORIGINAL_SETTING ((ip_plsql_int)0)
#define IP_ORIGINAL_QUALITY ((ip_plsql_int)0)

#define IP_MONOPHONIC ((ip_plsql_int)1)
#define IP_STEREOPHONIC ((ip_plsql_int)2)

#define IP_COMPRESSION_OFF ((ip_plsql_int)1)

```

```

#define IP_COMPRESSION_ON                ((ip_plsql_int)2)

#define IP_HIGHEST_SOUND_QUALITY         ((ip_plsql_int)1)
#define IP_HIGH_SOUND_QUALITY            ((ip_plsql_int)2)
#define IP_MEDIUM_SOUND_QUALITY          ((ip_plsql_int)3)
#define IP_LOW_SOUND_QUALITY              ((ip_plsql_int)4)
#define IP_LOWEST_SOUND_QUALITY          ((ip_plsql_int)5)

/*
** item Prompt constants
*/
#define IP_TOP_EDGE                       ((ip_plsql_int)0)
#define IP_BOTTOM_EDGE                    ((ip_plsql_int)1)
#define IP_START_EDGE                     ((ip_plsql_int)2)
#define IP_END_EDGE                       ((ip_plsql_int)3)
#define IP_HIDDEN                         ((ip_plsql_int)-3)
#define IP_PROMPT_FIRST_RECORD            ((ip_plsql_int)1)

/*
** VA constant.
*/
#define IP_UNSPECIFIED                    ((ip_plsql_int)-1)

/*
** Font weight constants.
*/
#define IP_FONT_MEDIUM                    ((ip_plsql_int)0)
#define IP_FONT_ULTRALIGHT                ((ip_plsql_int)1)
#define IP_FONT_EXTRALIGHT                ((ip_plsql_int)2)
#define IP_FONT_LIGHT                     ((ip_plsql_int)3)
#define IP_FONT_DEMILIGHT                 ((ip_plsql_int)4)
#define IP_FONT_DEMIBOLD                  ((ip_plsql_int)5)
#define IP_FONT_BOLD                      ((ip_plsql_int)6)
#define IP_FONT_EXTRABOLD                 ((ip_plsql_int)7)
#define IP_FONT_ULTRABOLD                 ((ip_plsql_int)8)

/*
** Font style constants.
*/
#define IP_FONT_PLAIN                     ((ip_plsql_int)0)
#define IP_FONT_ITALIC                    ((ip_plsql_int)1)
#define IP_FONT_OBLIQUE                    ((ip_plsql_int)2)
#define IP_FONT_UNDERLINE                  ((ip_plsql_int)3)
#define IP_FONT_OUTLINE                    ((ip_plsql_int)4)
#define IP_FONT_SHADOW                     ((ip_plsql_int)5)
#define IP_FONT_INVERTED                  ((ip_plsql_int)6)
#define IP_FONT_OVERSTRIKE                 ((ip_plsql_int)7)
#define IP_FONT_BLINK                      ((ip_plsql_int)8)

```

```
/*
** Font spacing constants.
*/
#define IP_FONT_NORMAL ((ip_plsql_int)0)
#define IP_FONT_ULTRADENSE ((ip_plsql_int)1)
#define IP_FONT_EXTRADENSE ((ip_plsql_int)2)
#define IP_FONT_DENSE ((ip_plsql_int)3)
#define IP_FONT_SEMIDENSE ((ip_plsql_int)4)
#define IP_FONT_SEMIEXPAND ((ip_plsql_int)5)
#define IP_FONT_EXPAND ((ip_plsql_int)6)
#define IP_FONT_EXTRAEXPAND ((ip_plsql_int)7)
#define IP_FONT_ULTRAEXPAND ((ip_plsql_int)8)
```

APPENDIX C - GLOSSARY

| | |
|----------------------|---|
| JDK | <p>Java Development Kit</p> <p>Produced by Javasoft, the JDK allows a developer to build Java programs and run them.</p> |
| OJDK | <p>Oracle Java Development Kit</p> <p>A version of the JDK which includes specific bug fixes made by Oracle</p> |
| JVM | <p>Java Virtual Machine.</p> <p>The Java Virtual Machine interprets and runs the code generated by the Java compiler.</p> |
| JInitiator | <p>Oracle's Java plug-in. JInitiator allows browser independence, as it uses its own JVM to execute the Java code.</p> |
| Appletviewer | <p>A simple application which can run any Java Applet. Appletviewer is part of the JDK</p> |
| Java | <p>Java is both a programming language and a platform.</p> <p>The Java language is an object-oriented, interpreted, high-level language. For a complete discussion of the Java language see the white paper 'The Java Language Environment' at http://java.sun.com</p> <p>The Java platform is a software only platform that sits on top of a hardware platform. The platform consists of two components - the JVM and the Java Application Programming Interface (API)</p> |
| Forms Listener | <p>The Forms Listener is the process that monitors a specified port number for incoming connection requests.</p> |
| Forms Server | <p>The Forms Server is the collective term for the Forms Listener and its associated Forms Runtime Engines.</p> |
| Forms Runtime Engine | <p>The Forms Runtime Engine is the Forms process which runs the requested Form. It is spawned by a successful connection request to the Forms Listener</p> |
| Web Server | <p>A Web Server services http requests, delivering a file to the requester.</p> |



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/>

Title: Oracle Forms Diagnostic Techniques
Version: 1.0.1
Date: June 2000
Authors: Chris Lewis, Nick Triggs
Editors:
Contributors: Ros Rason, Matt Hawkins

Copyright © Oracle Corporation 1999,2000
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, and Oracle8i, Oracle8, PL/SQL, and Oracle Expert are trademarks of Oracle Corporation. All other company and product names mentioned are used for identification purposes only and may be trademarks of their respective owners.
