Oracle9*i*AS Forms Services – Tracing and Diagnostics

An Oracle White Paper
February 2003

ORACLE

Oracle 9*i*AS Forms Services – Trace and Diagnostics

Oracle 9*i*AS Forms Services – Trace and Diagnostics

INTRODUCTION

This paper provides a framework and guidelines for best practices that developers using Oracle9*i*AS Forms Services 9.0.2 can adopt and adapt.

When you develop and deploy Oracle9*i* Forms applications, it is helpful to have information that allows you to optimize your applications. Tracing and diagnostic tools that are available with Oracle9*i* Forms allow you to analyze the performance and resource consumption of your Oracle9*i* Forms applications at runtime. You can use trace output to diagnose performance and other problems with Oracle9i Forms applications.

The following tools are available to collect trace information for Oracle9*i* Forms:

- **Forms Trace**:
  Replaces the functionality that was provided with Forms Runtime Diagnostics (FRD) and Performance Event Collection Services (PECS), which were available in earlier releases of Oracle9*i* Forms. Forms Trace allows you to trace the execution path through a form, for example, the steps the user took while using the form.
- **Servlet Logging Tools**:
  Enables site administrators to keep a record of all Oracle9*i* Forms sessions, monitor Oracle9*i* Forms-related network traffic, and debug site configuration problems.

FORMS TRACE

Forms Trace allows you to record information about a precisely defined part of forms functionality or a class of user actions. This is accomplished by defining events for which you want to collect trace information. For example, you can record information about trigger execution, mouse-clicks, or both. This section on Forms Trace contains the following information:

- Configuring Forms Trace
- Starting the Trace
- Viewing Forms Trace Output
- List of Traceable Events

- List of Event Details

CONFIGURING FORMS TRACE

You define the events that you want to trace in the ftrace.cfg file or in the URL when you start the trace. An event is something that happens inside Oracle9i Forms as a direct or indirect result of a user action. See "List of Traceable Events" for a list of events and their corresponding event numbers.

ftrace.cfg

The ftrace.cfg file is installed in the %forms_home%/forms90/server directory by default, and the FORMS90_TRACE_PATH environment variable is set in the default.env file. (The FORMS90_TRACE_PATH environment variable specifies the location of the ftrace.cfg file and the location of trace output files.)

In a text editor, edit the ftrace.cfg configuration file to specify named event sets. An event set specifies a set of events that you can trace simply by specifying the event set name rather than each event number individually when you start the trace.

Sample of ftrace.cfg

The following is a sample ftrace.cfg configuration file where two event sets have been specified.

```
//

// example ftrace.cfg file
// This file is used to specify event groups for use
with Forms Trace
//
// The file format is
// name1: event1, event2, ... , event_n
// name2: event1-event_m
//

all: 0-199
errors: 0-3
custom1: 32-46, 65, 66, 96, 194
```

Note the following:

- There must be a blank line between keyword entries.
- Keywords can be any name as long as they do not contain spaces. For example, a_b_c is an acceptable keyword.
- There must be a comma between each event number.

URL PARAMETER OPTIONS

The following command line parameters are used to configure Forms Trace:

Record = forms
This will enable Forms trace

Tracegroup =
Tracegroup indicates which events should be recorded and logged. If Tracegroup is not specified, only error messages are collected.

- Tracegroup is ignored if Forms Trace is not switched on at the command line.
- You can create a named set of events using the Tracegroup keyword, for example:
  Tracegroup=<keyword>, where <keyword> is specified in ftrace.cfg (for example, Tracegroup=MyEvents).
  This lets you log the events in the named set SQLInfo.
- You can log all events in a specified range using the Tracegroup keyword, for example:
  `Tracegroup = 0-3`
  This lets you log all events in the range defined by 0 <= event <=3.
- You can log individual events using the Tracegroup keyword, for example `Tracegroup = 34,67`
- You can combine event sets using the Tracegroup keyword, for example `Tracegroup = 0-3,34,67,SQLInfo`

Log files

The trace output is stored in a log file, which is saved in the directory specified by the FORMS90_TRACE_PATH environment variable. (If this variable is not set, the output is written to the current working directory.)

Users cannot specify the log name for the Forms Trace file in a URL. This behavior prevents a user from accidentally writing a file to an invalid location. If a user specifies log=<filename> in the URL, the URL will be ignored. The file will be named forms_<pid>.trc where <pid> is the process ID on the server.

STARTING THE TRACE

You start a trace by specifying trace entries in the URL or in the Forms configuration file - formsweb.cfg file. Entries should include the grouping of events to collect and the trace collection starts when the form executes.

The following are sample-URLs to start a trace:

```
http://cx-pc/forms90/f90servlet?form=cxl&record=forms&tracegroup
=0-199
```

```
http://cx-pc/forms90/f90servlet?form=cxl&record=forms&tracegroup
=mysql
```

The following are sample configurations in the formsweb.cfg file to start a trace:

```
[Tracing1]
```

```
record=forms
tracegroup=0-199
```

VIEWING FORMS TRACE OUTPUT

Trace data is stored in a binary file with a *.trc extension. To view trace data, you must either:

- Use the Upload/Translate utility to convert the data in the *.trc file to XML format, and then view the data using an XML viewer.
- Use the Upload/Translate utility to upload the data in the *.trc file to database tables.

Running the Upload/Translate Utility

The Upload/Translate utility performs two functions:

- Converts trace data to XML format.
- Uploads trace data to database tables.

Convert trace data to XML format

To convert trace data to XML format, on the command line, you will have to first set the class path to the jar file containing the Xlate.class:

```
SET CLASSPATH = %FORMS_HOME%\java\f90xlate.jar
```

Then type:

```
java oracle.forms.diagnostics.Xlate datafile=a.trc
xmlfile=myfile.xml
```

to create myfile.xml.

LIST OF TRACEABLE EVENTS

The following table lists the events that can be defined for tracing. In future releases of Forms, more events will be added to this list. Event types are as follows:

- Point event: An event that happens in Oracle9*i* Forms as the result of a user action or internal signal for which there is no discernible duration, for example, displaying an error message on the status line. Each instance of this event type creates one entry in the log file.
- Duration event: An event with a start and end, for example, a trigger. Each instance of this event type creates a pair of entries in the log file (a start and end event).
- Built-in event: An event associated with a built-in. Each instance of this event type creates a greater quantity of information about the event (for example, argument values).

For list of traceable events and event details, please refer to the Deployment Book posted on the OTN web site (http://otn.oracle.com/docs/products/forms/content.html).

SERVLET LOGGING TOOLS

The servlet logging tools available with Oracle9*i*AS Forms Services provides the following:

- A record of all Oracle9*i* Forms sessions, including session start and end times, and the user's IP address and host name (session-level logging)
- Monitoring of Oracle9*i* Forms-related network traffic and performance (session-performance and request-performance-level logging)
- Information for debugging site configuration problems (debug-level logging)

TURNING ON LOGGING

The servlet logging can be done on both the Forms Servlet and on the Forms Listener Servlet. If used with the Forms Listener the servlet logging feature will collect information while connecting to the runtime session. Using the servlet logging with the Forms Listener Servlet will give you information while the session is running.

Turn on logging for the Forms Listener Servlet by:

- Appending one of the strings in the below list "Supported logging capabilities" to the serverURL parameter in the URL that starts the form.
- Appending one of the strings in the below list "Supported logging capabilities" to the serverURL client parameter in the formsweb.cfg file

When you turn on logging, the Listener Servlet writes log messages to the servlet log file.

Specifying Logging in the URL for the Forms Listener Servlet

As an example, to start a performance-level trace, you would start the Oracle9*i* Forms application using a URL as follows:

```
http://yourserver/forms90/f90servlet?serverURL=/forms90/l90servlet/perf
```

Specifying Logging in the formsweb.cfg File for the Forms Listener Servlet

As an example, to start session-level logging for all users, you would change the serverURL entry in the default section of the formsweb.cfg file to the following:

```
serverURL=/forms90/l90servlet/session
```

Turn on logging for the Forms Servlet by:

Appending one of the strings in the below list “Supported logging capabilities” to the URL that starts the form. When you turn on logging, the Listener Servlet writes log messages to the servlet log file.

Specifying Logging in the URL for the Forms Servlet

As an example, you would start the Oracle9*i* Forms application using a URL as follows.

```
http://yourserver/forms90/f90servlet/debug?form=test.fmx
```

Specifying Full Diagnostics in the URL

As an example, to start full diagnostics, you would start the Oracle9*i* Forms application using a URL as follows. Note that if you append /debug to the URL used to invoke the Forms Servlet that servlet will output debug messages to the log file too.

```
http://yourserver/forms90/f90servlet/debug?serverURL=/forms90/l90servlet/debug
```

Supported logging capabilities

String	Description of logging
(none)	No log messages are produced. However, during Forms Servlet initialization, a message is written to the log file stating the name and path of the configuration file being used.
/session	Log messages are written whenever a Forms session starts or ends. These give the host name and IP address of the client (the machine on which the user's web browser is running), the runtime process id, and a unique internal session id number.
/sessionperf	Performance summary statistics are included with the session end message.
/perf	A performance message is written for every request from the client.
/debug	Full debug messages. Other debug messages are written in addition to the messages mentioned above. This logging level is very verbose and is intended mainly for debugging and support purposes.

LOCATION OF LOG FILES

The servlet log file is application.log. It is written to the application-deployments/forms90app directory of the OC4J instance to which Forms is deployed. In Oracle9*i*AS, the full path is:

```
<ORACLE_HOME>/j2ee/ProductGroup2/application-
deployments/forms90app/1_default_island/application.log
```

In Oracle9*i*DS, it is:

```
<ORACLE_HOME>/j2ee/iDS/application-
deployments/forms90app/application.log
```

ORACLE TRACE

Oracle Trace has been desupported for Oracle9i Forms. If you specify record=otrace, the Form will fail to start. This issue will be resolved in the next patch set to Oracle9i Forms.

APPENDIX

Example batch file to convert trace out put to XML (Win32)

```
@echo off
@echo ***********************************
@echo * Forms Trace to XML converter
@echo ***********************************

:again

set /p file1=Please enter the name of the trace file or
END to quit:
if /i [%file1%]==[END] goto end
if [%file1%]==[] goto again

:again2

set /p file2=Please enter the name of the of xml file or
END to quit:
if /i [%file2%]==[END] goto end
if [%file2%]==[] goto again2

REM Setting up the environment

setlocal

SET
CLASSPATH=C:\ids902\forms90\java\f90xlate.jar;%CLASSPATH
%

REM calling the XML converter for Forms trace

C:\ids902\jdk\bin\java oracle.forms.diagnostics.Xlate
datafile=%file1% xmlfile=%file2%

endlocal

@echo Thank you!!

set /p xxx=Press any key to exit.
goto end

:end
```

Example Output for Each Level of Servlet Logging

The following are examples of the type of output you will get when you use the following levels of logging:

(none)

```
FormsServlet init():
```

```
configFileName: d:\orant9i/forms90/server/formsweb.cfg

testMode: false
```

/session

```
Session start messages (example):

Forms session <10> started for test-pc.mycompany.com (
138.56.98.72 )

Forms session <10> runtime process id = 373

Session end message (example):

Forms session <10> ended
```

/sessionperf

```
Forms session <3> started for test-pc.mycompany.com (
138.56.98.72 )

Forms session <3> runtime process id = 460

Forms session <3> ended

Total duration of network exchanges: 1.041

Total number of network exchanges: 2 (1 "long" ones over
1.000 sec)

Average time for one network exchange (excluding long
ones): 0.030

Total bytes: sent 1,110, received 316
```

/perf

```
Forms session <3> started for test-pc.mycompany.com
(138.56.98.72)

Forms session <3> runtime process id = 460

Forms session <3>: request processed in 1.011 sec.
Received 8 bytes,returned 8bytes.

Forms session <3>: request processed in 0.030 sec.
Received 308 bytes, returned 1,102 bytes.

Forms session <3> ended

Total duration of network exchanges: 1.041

Total number of network exchanges: 2 (1 "long" ones over
1.000 sec)
```

```
Average time for one network exchange (excluding long
ones): 0.030

Total bytes: sent 1,110, received 316
```

/debug

Here is an example run by going to a URL like

http://cpx:8888/forms90/f90servlet/debug&config=ienative&serverURL=/forms90/l90servlet/debug):

```
============== FormsServlet ==============

GET request received, cmd=debug,
qstring=config=ienative&serverURL=/forms90/l90servlet/de
bug

No current servlet session

File baseie.htm not found, looking in
d:\orant9i/forms90/server

The SSO_USERID is: null

GET request received, cmd=startsession,
qstring=config=ienative&serverURL=

/forms90/l90servlet/debug&ifcmd=startsession

No current servlet session

New servlet session started

SSO_USERID in startSession: null

SSO_AuthType in startSession: null

User DN: null

Subscriber DN: null

EM mode in the config file: 0

File default.env not found, looking in
d:\orant9i/forms90/server

envFile = d:\orant9i\forms90\server\default.env

serverURL: /forms90/l90servlet/debug

rewrittenURL:
/forms90/l90servlet/debug;jsessionid=27f6412da05c426ab47
db4ae77636113

============== ListenerServlet ==============
```

```
GET request received, cmd=getinfo,
qstring=ifcmd=getinfo&ifhost=test-
pc.mycompany.com&ifip=130.35.96.71

Existing servlet session, id =
27f6412da05c426ab47db4ae77636113, not from cookie

Creating new Runtime Process using default executable

Starting Forms Server in EM mode

startProcess: executing ifweb90 server webfile=HTTP-
0,0,1

Getting stdin, stdout and stderr of child process

Writing working directory to stdin: d:\orant9i\forms90

New server process created

Forms session <4> started for test-pc.mycompany.com
(138.56.98.72)

**********************************************

Got POST request, length = 8

HTTP request headers:

ACCEPT-LANGUAGE: en

PRAGMA: 1

CONTENT-TYPE: application/x-www-form-urlencoded

ACCEPT: text/html, image/gif, image/jpeg, *; q=.2, */*;
q=.2

USER-AGENT: Mozilla/4.0 (compatible; MSIE 5.0; Win32)

Oracle Trace

HOST:cpx:8888

CONTENT-LENGTH: 8

CONNECTION: Keep-Alive

Existing servlet session, id =
27f6412da05c426ab47db4ae77636113, not from cookie

Forms session <4> runtime process id = 474

Port number is 2791

RunformProcess.connect(): connected after 1 attempts
Connected to ifweb process at port 2791
```

```
Forms session <4>: request processed in 1.032 sec.
Received 8 bytes,
returned 8 bytes.
*****************************************
```

Oracle 9iAS Forms Services – Trace and Diagnostics
Februaruy2003
Author: Jonas Jacobi

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

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