Analytical debugging methods and problem analysis OC 4.5.3/RDC 4.5.3/TMS 4.6/AERS 4.6/Siebel Clinical environments

Sunil G. Singh

DBMS Consulting

14 October 2008

Administration & Configuration Management

Session 05

Acknowledgements

- Many thanks to the OCUG for this opportunity to present for the OCUG A&CM group.
- Many thanks to the OCUG Planning and Review Committee and OCUG A&CM Focus Group Chairs for their infinite patience in receiving and expeditious review of this presentation
- Many thanks to everyone who participated in the development of presentation.

Assumptions/Scope/Disclaimer

- Assumption: Audience has a basic understanding of the OLS 4.5.x architecture
- Scope: OC 4.5.3/RDC 4.5.3/TMS 4.6/AERS 4.6.x/Siebel 8.x.
- Disclaimer: These methods are for debugging production environment problems. They are not intended for bypassing security measures or regulatory policies, and nothing in this presentation should be construed as intended for such purposes.

Requirements for Debugging and Problem Analysis Within Production OLS environments.

- OLS production environments, especially those running RDC or with a global user base, have close to 24x7 usage and availability requirements.
- This type of environment increasingly presents problems and issues which must be debugged and analyzed in the production environment due to
  - critical time constraints
  - dependency on production infrastructure or components which are part of the issue
  - dependency on production data which does not exist elsewhere

Requirements for Debugging and Problem Analysis Within Production OLS environments. (2)

- While creating refreshed copies/clones of the production environment and reproducing a production issue is the best analytical method, it is not always practical because:
  - Production environments are more complex and sometimes can not be duplicated 100% in their entirety in a test environment (such as load balanced servers, public-facing network components, secure data)
  - Sufficient infrastructure (such as disk space and servers) may not exist to make copies of the production environment
  - Downtime may be required to create production copies which may not be available
  - System Administrator/DBA/Application Administrator resources and availability may not exist or may not be available in time

Requirements for Debugging and Problem Analysis Within Production OLS environments. (3)

- Executing many debugging techniques as documented can:
  - Cause short outages/downtimes which impact multiple users
  - Negatively impact performance for all users
  - Generate an excessive amount of debugging information/large logs, making it difficult to isolate a problem
- Having a way to debug a specific user's issue without effecting the production environment performance, causing downtime, or generating excessive debug files is a tremendous advantage in a production environment

Summary of Previously Discussed Debugging Methods

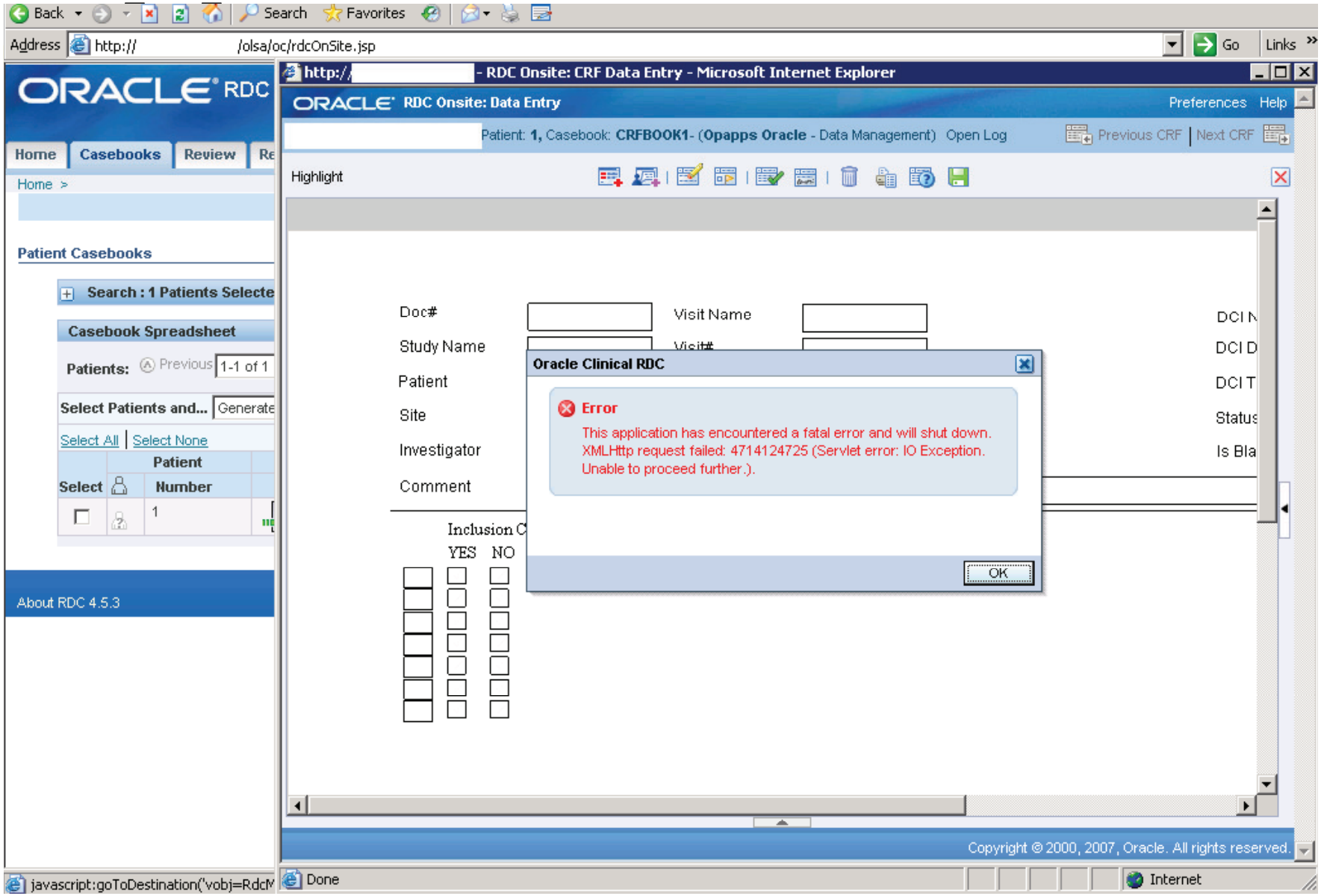
- Database Level RDBMS tracing for a running user's session
  - Useful for tracing an already running session
- Static HTML file generation for individual testing of Forms-related configuration changes, not at the system-wide formsweb.cfg level
  - Useful for testing changes related to forms parameters or tracing forms which crash for a specific user
- Setting environment variables for specific sessions, not at the registry level
  - Useful for hiding forms such as OS Password, Job Scheduling and Reports Queue Monitoring

Summary: Previously Discussed Debugging Methods (2)

- Using User Logon Triggers, used when:
  - an entire job submitted by a user needs to be traced
  - the user's session fails during the login process
  - a session modifiable parameter needs to be changed/tested at the user level
    - Very useful in performance tuning, when combined with local schema objects
- Cross Referencing Desktop Client sessions to Application Tier to RDBMS
  - Forms 6i-based, use of .rti files

Cross Referencing Desktop Client session to Application Tier to RDBMS

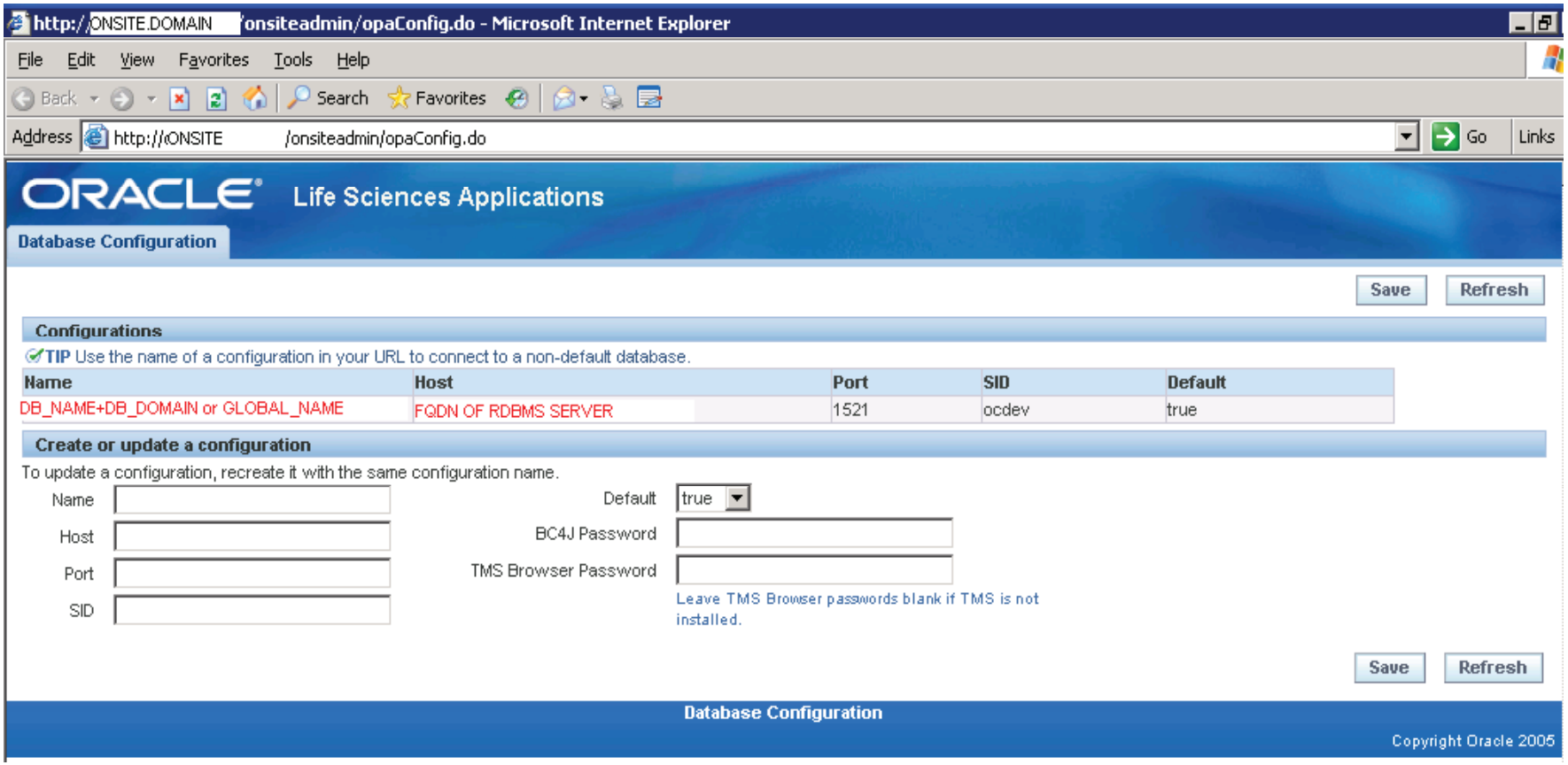
- The Windows process ID is the same as the first part of the process column in v$session in the RDBMS level.
- When querying v$session, be sure to include the username, process, program AND terminal column in environments where multiple Middle Tiers can connect to the same instance. This allows the correct identification of the Middle Tier and the correct forms executable process
- The second part of the process column is the Windows thread id. The thread ID can NOT be seen through the normal Task Manager process list and different utilties are required to see the thread ID.
- The OS-level process ID can be identified by joining PADDR in v$session to ADDR in v$process. The SPID is the process ID of the corresponding TNS listener process on the RDBMS Server if Dedicated Listener is being used.



[C] RDC 4.5.3 Individual URL debugging

- All parameters listed here are from: **Metalink DocID: 400219.1: Configuration/Debug for RDC OnSite**
- The general URL is of the form:
  - http://<Middle_Tier.domain>/olsa/oc/rdcLogin.do?event=doSetup&db=<OPA_CONFIG_NAME>&debug=<options>
- This url also support additional debug parameters with &debug added to the end of the URL:
  - dcapi (Debug DCAPI, similar to OPA_DCAPI_PDF_DEBUG registry key = Y)
    - %OPA_HOME%\log\dcapiHtml_<userid>_<ts>.dbg
    - %OPA_HOME%\log\dcapiJaveDebug_<userid>_<ts>.dbg
  - surround (Debug the Servlet, similar to setting the debug in the web.xml file)
    - %OPA_HOME%\log\RdcOnsite0.xml
  - all (Enable both dcapi and surround and opa_trace)

[X] Configuration Name for RDC OnSite Debugging



[X] Debug Files on the RDC ONSITE Server

```
rdconsite0.xml.14Oct2008
RdcOnsite0.xml
DcapiHtml_ops$opapps_122398240660 9.dbg
18694957506167757 04.log
DcapiJava_ops$opapps_122398240660 9.dbg
delog0.log
```

- RdcOnsite0.xml: This is a continuous log with all errors encountered in the RDC OnSite spreadsheet historically kept.
  - A backup copy is required to open the .xml file in wordpad while OnSite is running

[X] Debug Files on the RDC ONSITE Server (2)

- Delog0.xml: Contains J-Initiator like servlet-errors, also historically kept, usually related to disconnections of a data entry page
  - A backup copy is required to open the .xml file in wordpad while OnSite is running
- olsardcapi.dll authentication log: Contains specific arguments to the calls to the Java Servlet functions and their result

[X] Errors from DcapiHtml and DcapiJava Debug file

DcapiHtml_ops$opapps_1223982406609 - Notepad

File Edit Format View Help

```
Tue Oct 14 06:06:52.250 2008 : ------- In populateCrfDetails
Tue Oct 14 06:06:52.281 2008 : Error returned from database : Error - No record found
---- In validate_pdf_dcapi_call, status is 1, strlen of combinedresponsestring is 0
     Function getCrfDetails returned error -1:Error returned from database : Error - No record
found
 of type 0.
 ---- End of validate_pdf_dcapi_call, PdfDcapiFailure is 1, PdfDcapiWarning is 0

 ---- In handle_pdf_dcapi_status ..., PdfDcapiFailure is 1, PdfDcapiWarning is 0,
PdfDcapiReturnDiscrep
is 0, DcapiRedisplayVal is 0
     In handle_pdf_dcapi_status combinedresponsestring is 80getCrfDetails - -1:Error returned
from
database : Error - No record found
.
---- End of handle_pdf_dcapi_status ret_status is 1
```

DcapiJava_ops$opapps_1223982406609 - Notepad

File Edit Format View Help

```
[Tue, Oct 14, 2008 at 06:06:52 CDT] [DCAPI Debug 90381] Calling sendAndReceiveMessage
[Tue, Oct 14, 2008 at 06:06:52 CDT] [DCAPI Debug] Inside sendAndReceiveMessage
[Tue, Oct 14, 2008 at 06:06:52 CDT] [DCAPI Debug] Inside sendData
[Tue, Oct 14, 2008 at 06:06:52 CDT] [DCAPI Debug] Failed to send message(Connection reset by peer:
socket write error). Will try to reconnect and send message again
[Tue, Oct 14, 2008 at 06:06:53 CDT] [DCAPI Debug] Inside method - disconnect (force = true)
[Tue, Oct 14, 2008 at 06:06:53 CDT] [DCAPI Debug 90381] Calling sendAndReceiveMessage
[Tue, Oct 14, 2008 at 06:06:53 CDT] [DCAPI Debug] Inside sendAndReceiveMessage
[Tue, Oct 14, 2008 at 06:06:53 CDT] [DCAPI Debug] Inside sendData
[Tue, Oct 14, 2008 at 06:06:53 CDT] [DCAPI Debug] Inside close of DcapiMessenger
```

[C] RDC 4.5.3 Individual URL debugging (2)

- OPA_TRACE package can be invoked with:
  - http://<Middle_Tier.domain>/olsa/oc/rdcLogin.do?event=doSetup&db=<OPA_CONFIG_NAME>&opaTrace=TRUE&debug=<options>
- Populates entries in the OPA_DEBUG table (see opadebug_verbose.log)
- Enable session level SQL Tracing
  - http://<Middle_Tier.domain>/olsa/oc/rdcLogin.do?event=doSetup&db=<OPA_CONFIG_NAME>&sqlTrace=TRUE&debug=<options>
  - May not have bind variables available
  - Useful since RDC 4.5.3 and TMS 4.6 create multiple sessions for each new page/form

[X] Logon Triggers for more precise SQL tracing

- Compile trigger for a specific user but make the trigger disabled
- Log into RDC/TMS/AERS with a normal URL: NO DEBUGGING OPTION
- BEFORE opening page or form with error, ENABLE the trigger
- AFTER opening page or form, DISABLE the trigger
- Copy the .trc file immediately to prevent further writes

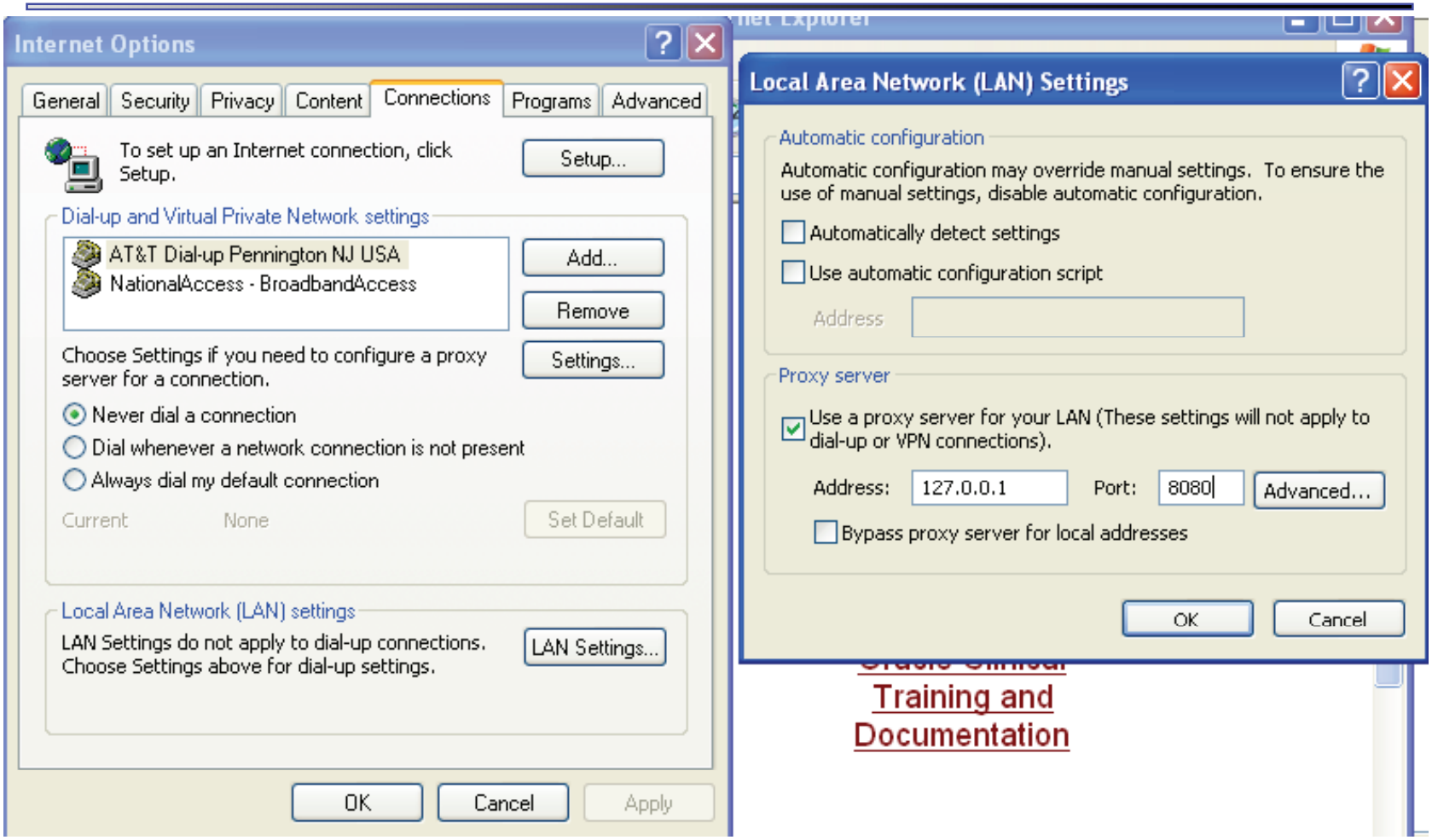
Java Console Debugging

- For 9i Forms, located in the Control Panel for 1.3.x Java versions (TMS 4.6/AERS 4.6)
- Console and Show Exception dialogue can be enabled:

Network-Based http debugging

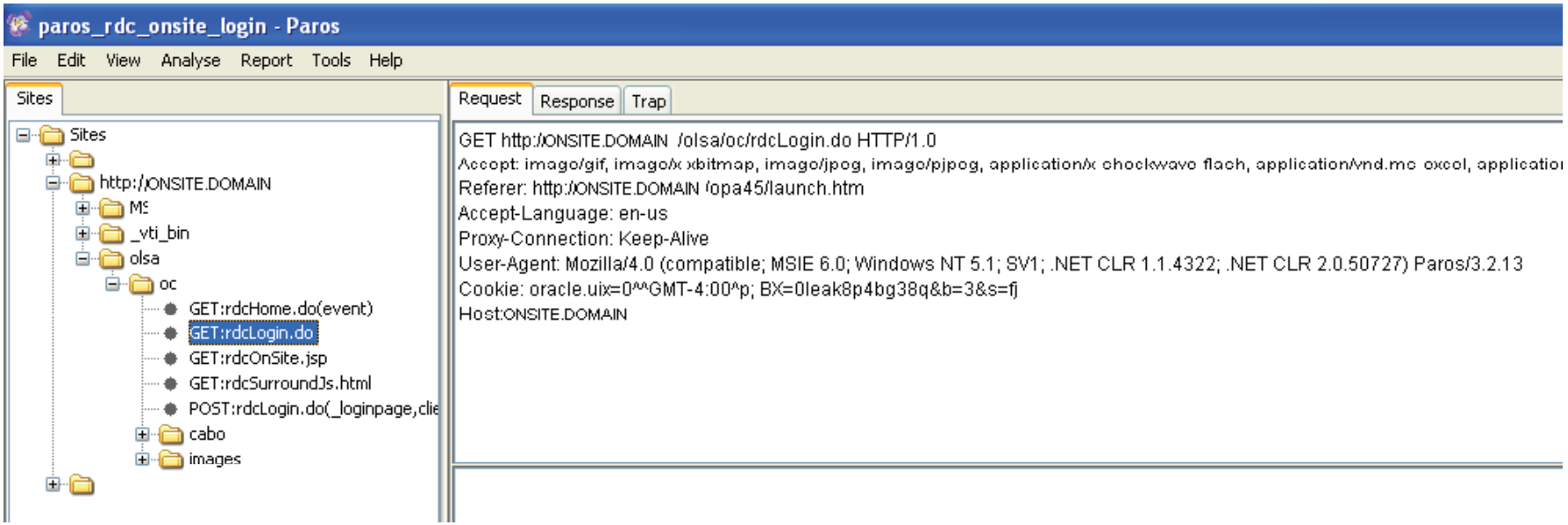
- More sophisticate network debugging can be used with http level trapping tools such Paros Proxy
- Each argument for servlet communication can be analyzed when examining an issue, for example, a specific CRF page not opening in RDC Onsite
- Combine with the debug options to determine if the issue is related to the:
  - Study Design
  - Data
  - Application
- The key is to set the IE Browser to have a proxy of 127.0.0.1 while Paros is running
  - This traps the browser traffic and then sends it to the application tier destination

[X] Set the local proxy Server to 127.0.0.1:80

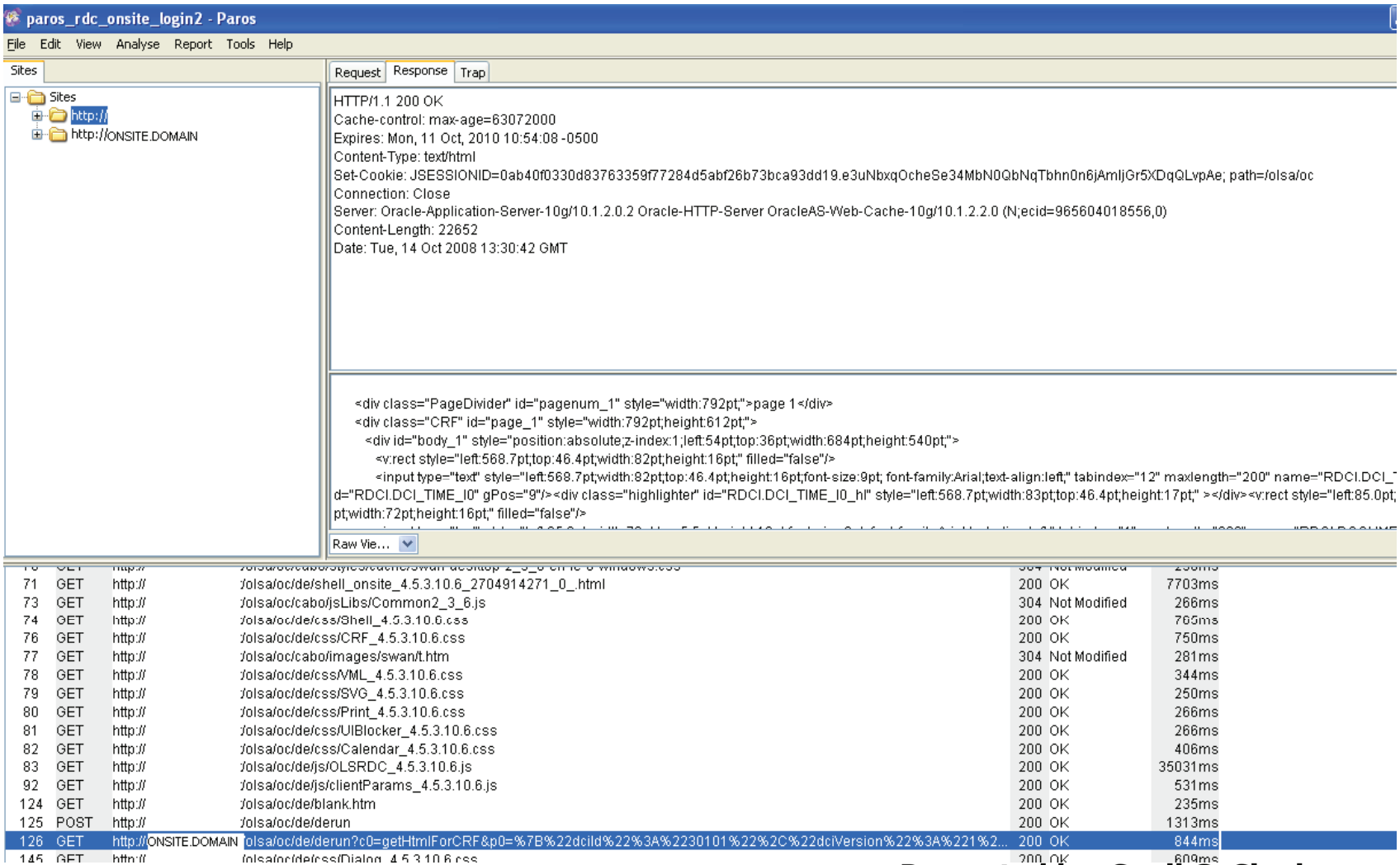


[X] Sample Output from Paros

- Each http request and response is logged with the FULL URL actually passed to OLS
- These requests can be stored or modified to debug specific URL-related issues



[X] Identify specific failing request and response



Forms 9i Debugging

- Enable trace with:
  - http://<Middle_Tier.Domain>/forms/frmservlet?config=opa46&record=forms&tracegroup=0-199
- Perform actions which require tracing
- Examine forms_<PID>.trc in %ORA_10gMT_HOME%\forms\trace

Forms 9i Debugging (2)

- Parse the trace files with:
  - set ORACLE_HOME=<ORA_10gMT_HOME>
  - set CLASSPATH=%ORA_10gMT_HOME%\forms\java\frmxlate.jar; %ORA_10gMT_HOME%\forms\java\f90all.jar
  - FOR XML: %ORA_10gMT_HOME%\jdk\bin\java oracle.forms.diagnostics.Xlate Datafile=forms_<PID>.trc Outputfile=forms_<PID>.xml OutputClass=WriteOut
  - FOR HTML: %ORA_10gMT_HOME%\jdk\bin\java oracle.forms.diagnostics.Xlate Datafile=forms_<PID>.trc Outputfile=forms_<PID>.html OutputClass=WriteOutHTML

MDR troubleshooting

- For AS10g Infra Home (controls OID and the OEM HTTP listener)
  - OID can be manually restarted with oidctl
  - A new instance number can be set if oidctl does not start correctly. It could also be an issue with the ODS schema account in the MDR
  - Ldapbind must be run manually afterward
  - Then opmnctl startall should work

MDR troubleshooting (2)

- For the AS10g MT Home (controls OC4J olsardc applications)
  - MDR can be resynched with the Application Tier files or Application Tier can be resynched with MDR
  - dcmctl in shell mode is very useful for extended help
  - updateconfig -force –d <IAS_INSTANCE>
    - Usually, force is required for updating .xml config files
  - resynchinstance -force –d <IAS_INSTANCE>
    - Usually, force is required for updating MDR
  - olsardc can be removed and reinstalled
    - dcmctl removeComponent -co olsardc

Using 10g Grid Control for Monitoring RDC 4.5.3 and Siebel Clinical 8.0

- Since RDC 4.5.3 runs solely on Oracle AS 10g R2 without additional Plug-Ins, it is a true J2EE application running with Oracle Containers for Java or OC4J
- Siebel Clinical can also Optionally deployed with the same technology stack for its application servers, although this is not required
- Oracle has introduced OEM 10g Grid Control with extensions to natively monitor and control both OC4J applications as well as Siebel RDBMS and application servers.
  - OEM requires that the Siebel Application Pack for OEM be installed on top of OEM 10g Grid Control to monitor Siebel
- Additionally, Siebel itself has some detailed logging configuration options available

Using 10g Grid Control for Monitoring RDC 4.5.3

- As shown in the next example, any J2EE application running in OC4J, such as RDC 4.5.3, can be monitored and controlled once a OEM 10g Grid Control agent is installed on the same Application Server running RDC 4.5.3.
- The Oracle Enterprise Manager Concepts Guide describes some possible Monitoring options and alerts show below

Automated Monitoring and Alerts

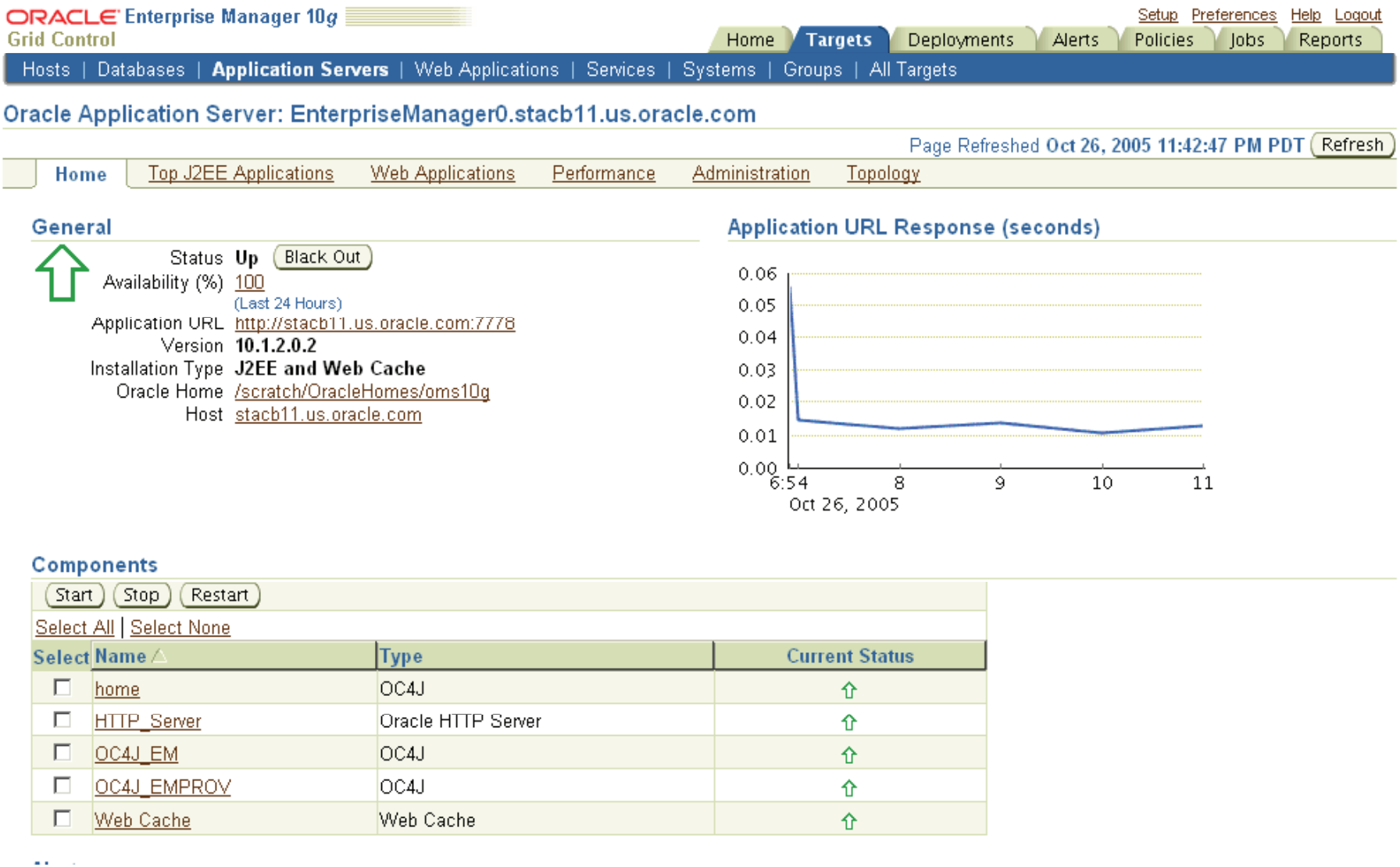
Enterprise Manager provides a comprehensive set of features that facilitates automated monitoring and generation of alerts. The Oracle Management Agent on a host automatically discovers the Oracle Application Server targets on that host, and helps Enterprise Manager perform unattended monitoring of their status, health, and performance.

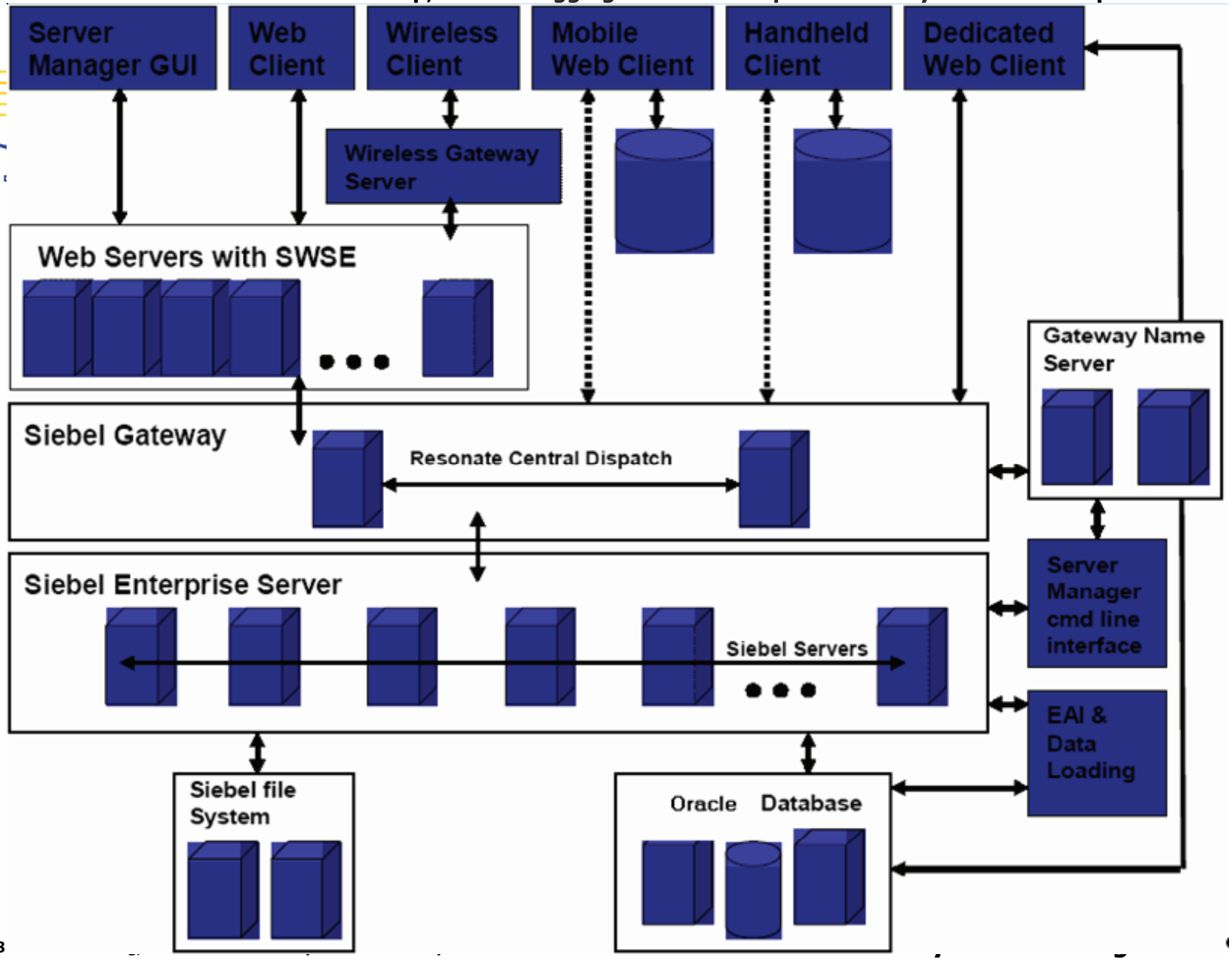
Enterprise Manager gathers and evaluates diagnostic information from these targets distributed across the enterprise, and an extensive array of application server performance metrics are automatically monitored against predefined thresholds.

For example, Enterprise Manager can automatically monitor:

- The CPU or memory consumption of the application server, including detailed monitoring of individual Java Virtual Machines (JVMs) being run by the server's Oracle Application Server Containers for J2EE (OC4J) instances.
- J2EE application responsiveness from the application down through individual servlets and Enterprise JavaBeans (EJBs).
- HTTP Server session volumes, connection duration, and error rates.
- Oracle Application Server Web Cache hit rates and volumes.
- Top servlets based on number of requests, maximum processing time, and highest average processing time.

If an Oracle Application Server or any of its core components go down, or if a performance metric crosses a warning or critical threshold, an alert is generated by Enterprise Manager and a notification is sent to you. Enterprise Manager supports notifications via e-mail (including e-mail-to-page systems), SNMP traps, and/or by running custom scripts.




Server Manager GUI
Web Client
Wireless Client
Mobile Web Client
Handheld Client
Dedicated Web Client
Wireless Gateway Server
Web Servers with SWSE
Gateway Name Server
Siebel Gateway
Resonate Central Dispatch
Siebel Enterprise Server
Siebel Servers
Server Manager cmd line interface
EAI & Data Loading
Siebel file System
Oracle Database

Using 10g Grid Control with Oracle Application Management Pack for Monitoring Siebel CRM

New Siebel-Specific Targets

Several new targets, as discussed in Table 1, have been added to Enterprise Manager in order to facilitate the management of Siebel CRM applications. These targets model the entities within a Siebel environment so that they can be managed within Enterprise Manager.

Most of these targets have direct one-to-one mapping with their counterparts in Siebel. Some are created to facilitate specific management capabilities within Enterprise Manager.

Excerpted from
Oracle® Application Management Pack for Siebel Getting Started Guide

Table 1 Siebel-Specific Targets

Enterprise Manager Target	Siebel Entity	Purpose
Siebel Enterprise	Siebel Enterprise	Representation of Siebel enterprise providing access to metrics and associated Siebel servers.
Siebel Server	Siebel Application Server	Representation of Siebel server providing access to related metrics and configuration information.
Siebel Component Group	Siebel Component Group	Representation of Siebel component group providing access to metrics and associated Siebel components.
Siebel Component	Siebel Component	Representation of Siebel component providing access to component metrics and configuration information.
Siebel Required Component Group	-	Representation of all the Siebel components providing mandatory functionality for the proper function of a Siebel server.
Siebel Functional Component Group	-	Representation of all the Siebel components providing functionality that may be used by multiple components (for example, Workflow).
Siebel Database Repository	Siebel Database	Representation of Siebel database providing access to Siebel business metrics.
Siebel Gateway Server	Siebel Gateway Server	Representation of Siebel gateway server
Siebel Application Service (HI)	Employee Facing Siebel Applications (high interactivity)	Aggregated Service providing information about all the Siebel high interactivity applications.
Siebel Application Service (SI)	Customer Facing Siebel Applications (standard interactivity)	Aggregated Service, providing information about all the Siebel standard interactivity applications.

Types of Logging in available in Siebel Clinical: Application Object Manager

Table 16. Common Event Types for Application Object Manager Diagnostics

Event Type Name	Event Type Alias	Log Level Setting	Description
Event to track the flow of a message	MessageFlow	4	Captures messages exchanged between the Application Object Manager (AOM) and Siebel Web Server Extension (SWSE).
Object Manager Session Information	ObjMgrSessionInfo	4	Captures User Session login, logout, and timeout information.
Event Context	EventContext	4	Captures applet and method executed, view names, and screen names that the user navigates to.
		5	Captures username and IP address when the session completes.
Object Manager Data Object Log	ObjMgrDataObjLog	5	Captures data manager object tracking; that is, the creation, use, and deletion of database connections, search specifications, sort specifications, and cursors.
Object Manager Log	ObjMgrLog	5	Captures general AOM events: load license, open SRF, errors, and so on.
Object Manager Business Component Log	ObjMgrBusCompLog	4	Captures Business Component-related events: create and delete.
Object Manager Business Service Log	ObjMgrBusServiceLog	4	Captures Business Service-related events: create, delete, methods invoked, and so on.
Main Thread Events	MainThread	4	Captures task counter, task creates, and task exits (in main Multithreaded Server log).

Types of Logging in available in Siebel Clinical: Application Object Manager (2)

- **Excerpted from Siebel Systems Monitoring and Diagnostics Guide, previous, current, next slides**

Task Related Events	TaskEvents	4	Captures task creation, context, session timeout, and close info.
SQL Parse and Execute	SQLParseAndExecute	4	Captures the SQL insert, update, and delete statements processed by the database connector. It includes the SQL statement and bind variables. The content is similar to the ObjMgrSqlLog event; however, the select statement is not captured by the SQLParseAndExecute event.
Object Manager SQL Log	ObjMgrSqlLog	4	Captures the SQL select, insert, update, and delete statements processed by the AOM data object layer. Includes the SQL statement and bind variables. It also captures the prepare, execute, and fetch time for the SQL cursor.
		5	Captures internal and customer-defined search and sort specifications, the joins processed for queries, as well as a call stack of the operation performed. Setting this event to log level 5 incurs a significant performance impact because a callstack is generated. Only set this event to log level 5 in consultation with Siebel Technical Support.
SQL Profiling	SQLProfiling	4	Captures SQL Profiling information. Helps aid in the diagnosis of a poorly performing component.
SQL Summary	SQLSummary	4	Captures SQL prepare, fetch, and execute times. Provides detailed information regarding the execution of a SQL statement.
SQL Slow Query	SQLSlowQuery	4	Captures SQL Performance— lists ten slowest performing queries.
Security Adapter Log	SecAdptLog	5	Captures security adaptor tracing information to the AOM log file.
Security Manager Log	SecMgrLog	5	Captures security manager tracing information to the AOM log file.

Types of Logging in available in Siebel Clinical

- Log Levels are 1=Most Severe to 6=Informational messages

Components | Parameters | Events

Menu | Query | Enable | Disable 1 - 1 of 1

Component	Alias	Component Group	Enable State
Server Request Broker	SRBroker	System	✔

Each Component in the Siebel Architecture can have several Event Types with different Logging Levels

Events | Parameters

Menu | Query 8 - 17 of 76+

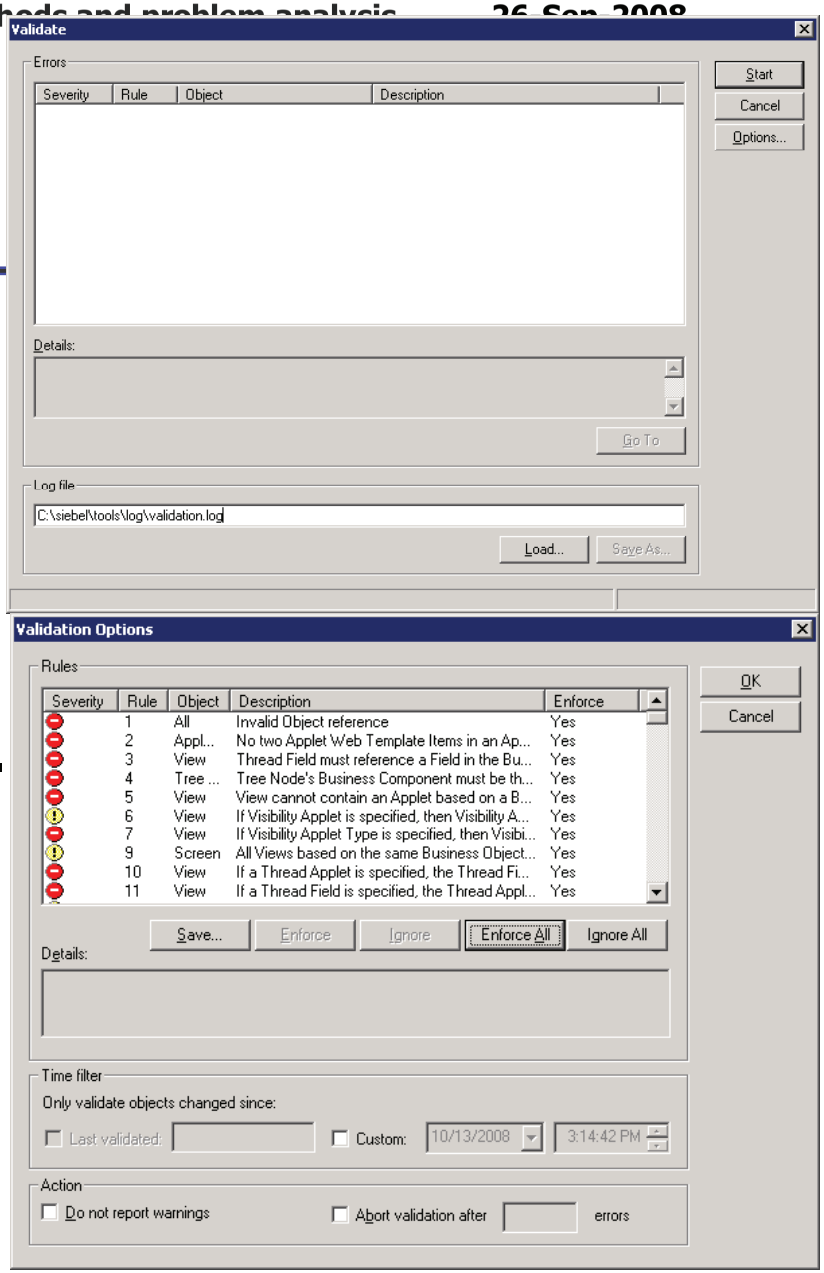
Component Alias	Event Type	Log Level	Description
SRBroker	Error Condition	1	Triggered upon reaching an unhandled error or exception
SRBroker	General Events	1	General event point logging
SRBroker	Param Encryption	1	Param encryption values in the scr layer
SRBroker	Performance Event	2	Event for Performance Measurements
SRBroker	SRM record Synchronize	1	Triggered during the SRM synchronize from gateway to the database.
SRBroker	Component Tracing	1	A trace condition was met (used from LogTrace only)
SRBroker	Task Configuration	1	Configuration of Server Task
SRBroker	Task Configuration at exit	1	Configuration of Server Task at exit

[x] Troubleshooting CTMS Issues

- Errors resulting from configuration problems in the Repository.
- SQL Execution errors and tracing on the Dedicated Client.
- SQL Execution errors and tracing on the server for Thin Client.
- Running the Siebel Client in debug mode

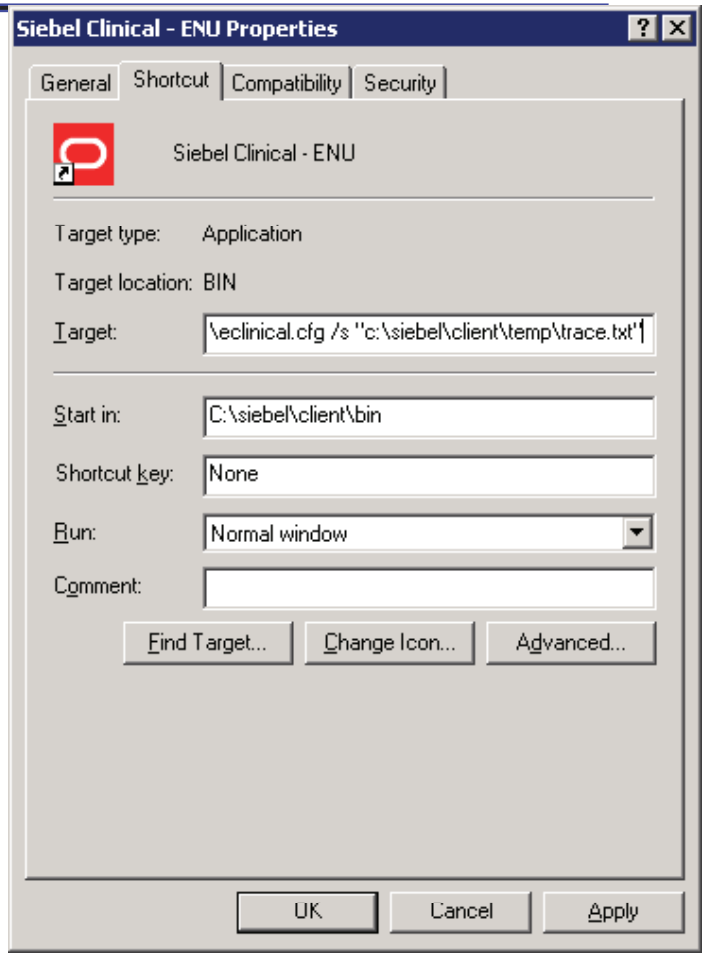
[x]Configuration Troubleshooting

- Validate configuration of Repository Objects before compiling SRF:
- In Siebel Tools, select object(s) of any type, right click, Validate.
- Set output log filename and path.
- Click Options and choose rules to enforce, or "Enforce All".



[X] SQL Query Troubleshooting – Dedicated/Remote Client

- Configuration problems with Repository and/or Database objects can cause SQL errors in the application:
- Edit Shortcut Properties for Siebel Clinical.
- Use the /s command line switch followed by a path and filename for output to specify a SQL trace file which will help identify the query that is causing a problem.
- When an error occurs, open the output file and the last query logged should be the culprit. By analyzing the tables involved, in the context of the screen/view/business object where the error occurred, you should be able to identify the problem Business Component.
- Sometimes running the SQL statement with a third party SQL client (like Toad) can help identify a problem if it is with the physical DB object layer.

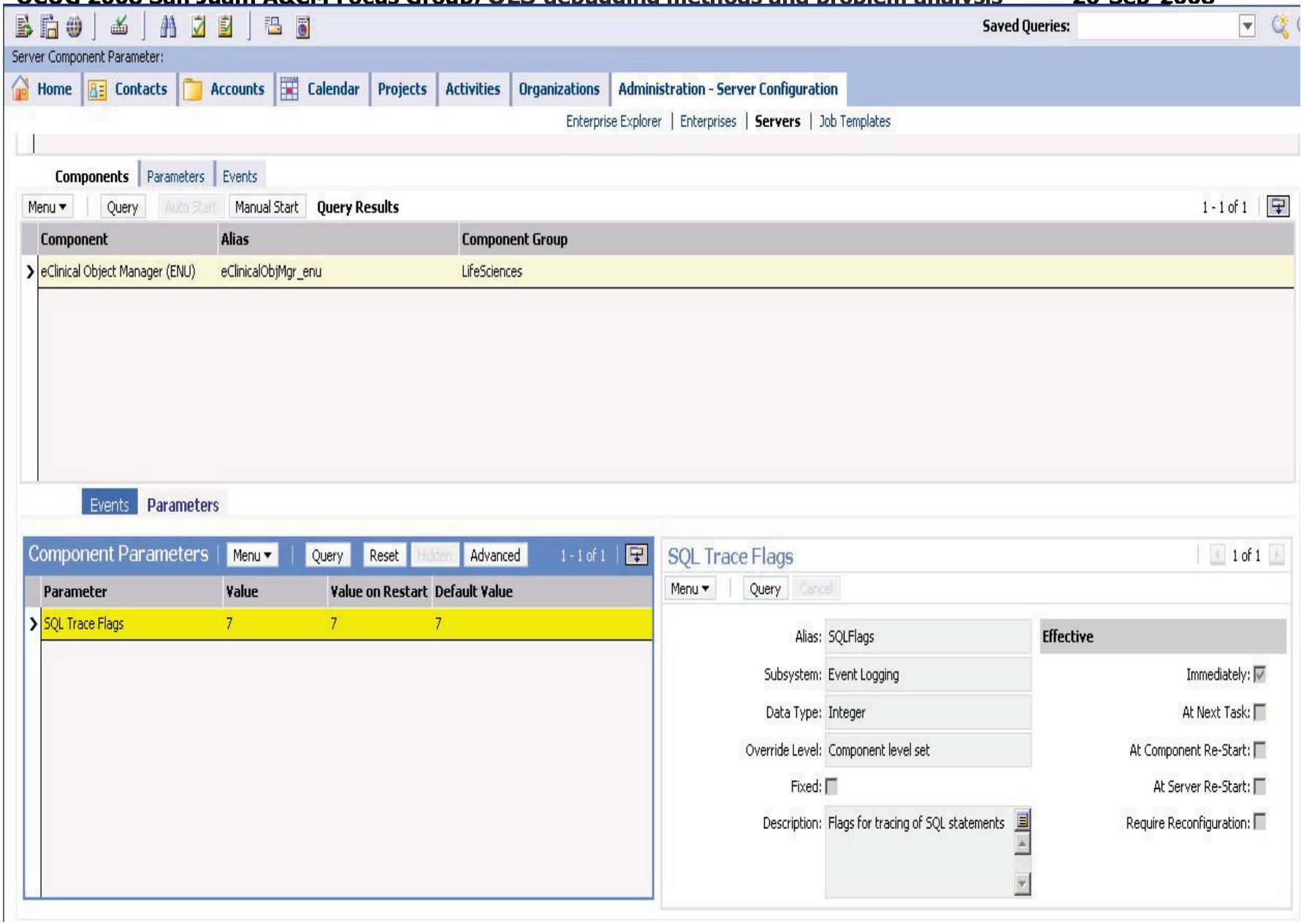


[X] SQL Query Troubleshooting: Server Component (Thin/Web Client)

- To trace the SQL execution and event handling of server components, you need to set certain flags within the Component's parameters:
- Navigate to the Administration – Server Configuration screen, Server Component Parameters view, and query for the eClinical Object Manager component on the correct server(s).
- Click the "Hidden" button on the Component Parameters applet and set the SQL Trace Flags parameter to 7.

[X] SQL Query Troubleshooting: Server Component (Thin/Web Client)

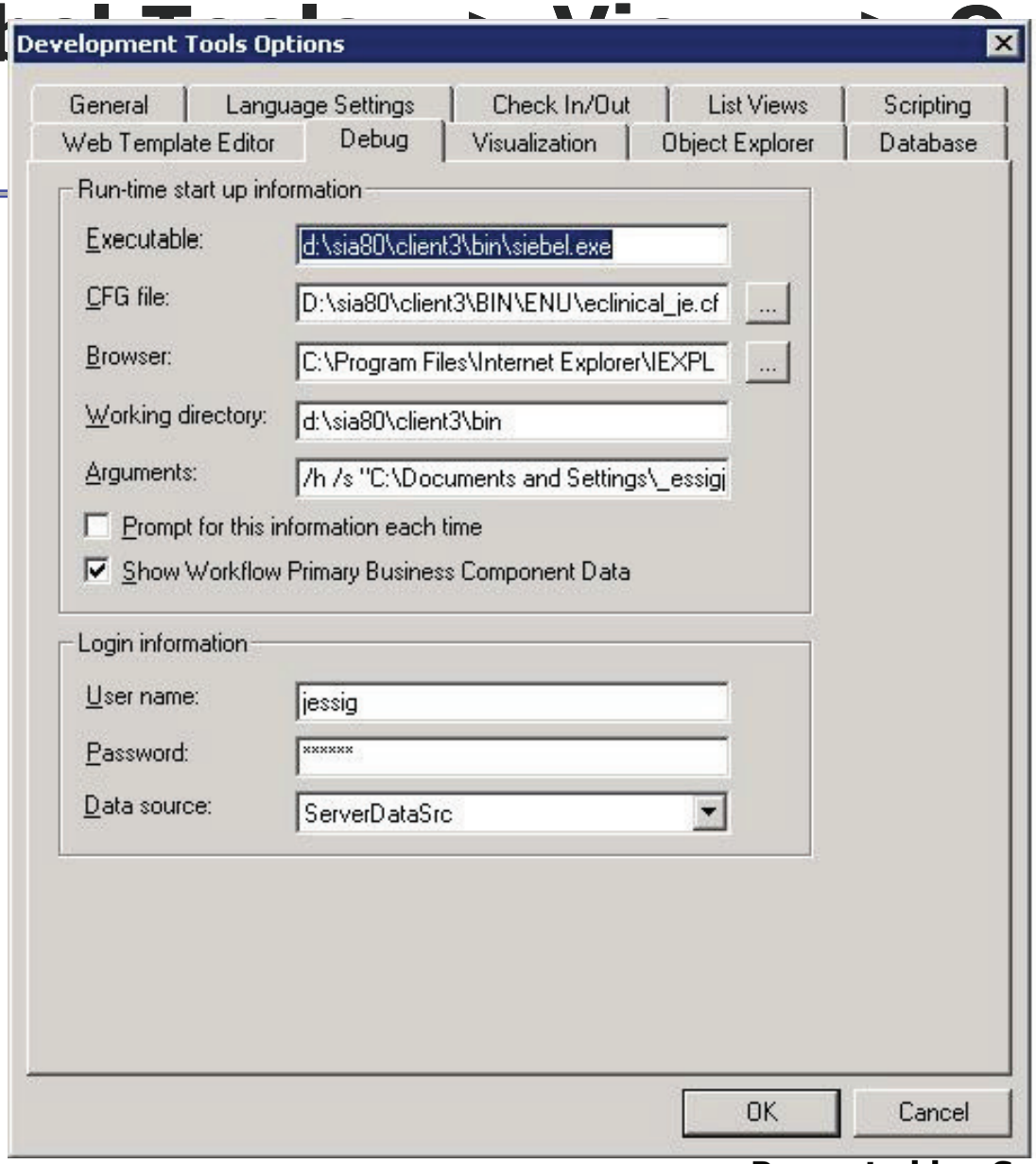
- The log file for the component task, including the trace of executed SQL, can be found at the following path:
- /<siebelroot>/siebsrvr/enterprises/<enterprisename>/<siebelservername>/log
- (Don't forget to turn the trace flags back to 0 when you are finished as tracing produces large output files.)



[X] Running the Siebel Client in Debug Mode

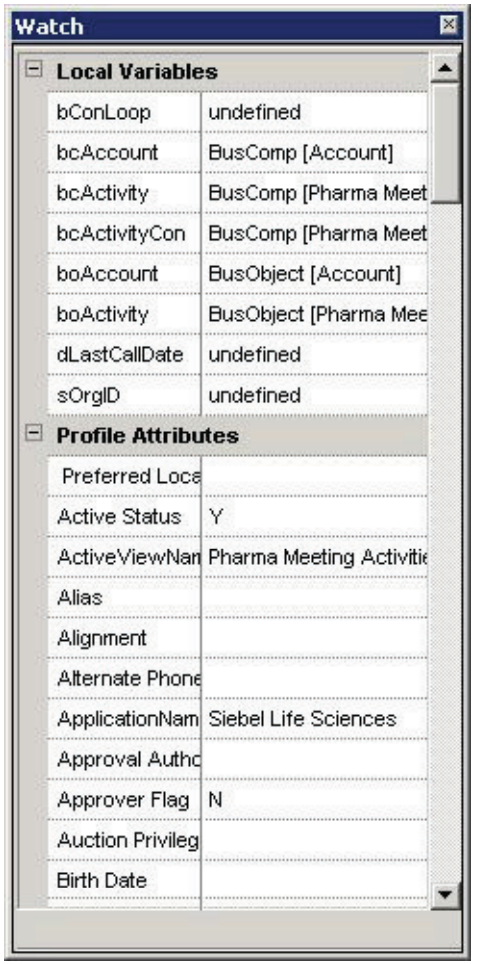
- For the purposes of debugging Siebel eScript or Siebel VB script, you can run a client session in debug mode and set breakpoints in the script(s).
- In Siebel Tools, select View, Options, and edit the information in the Debug tab. Make sure to include the /h switch in the command line "Arguments" (you may also include /s and specify a SQL trace output file).

Siebel Tools => View => Options



[X] Running the Siebel Client in Debug Mode (2)

- To set a breakpoint, right click the left margin of the line and select Toggle Breakpoint.
- Start the client in debug mode by pressing F5, or select Debug, Start, from the menubar.
- When the breakpoint is reached the client will halt and Siebel Tools will now allow you to step through the script with familiar commands like F8, and resume with F5.



Conclusions

- The OLS Application Suite has become more complex but many components are using more current technologies. As a result, there are more options available to analyze and debug production level problems and issues
- It is still possible to utilize the core RDBMS-level and Forms level methods shown to isolate and identify issues in many cases, even when complex architecture or technology stacks are present

Question and Answers

Sunil G. Singh
singh@clinicalserver.com
+1-860-983-5848

Jose Garcia
jgarcia@clinicalserver.com
+1-347-452-9501

Anoop Nair
arnair@clinicalserver.com
+1-917-881-4524

Dr. Letian Liu
lliu@clinicalserver.com
+86-134-0212-4879

Jason Essig
jessig@clinicalserver.com
+1-917-846-7683