An Oracle White Paper
February 2010

# Comprehensive Testing for Siebel With Oracle Application Testing Suite

## Introduction

Siebel provides a wide range of business-critical applications for Sales, Marketing, Contact Center, Self-Service and more. These Siebel applications are highly interactive, intuitive and personalized, delivering the user experience of a Win-32 application within a browser environment. Ensuring Siebel application quality and performance is critical but also challenging. There are many different applications and business transaction flows to test. Siebel's highly distributed application architecture means there are many integrated components to consider, from both a functionality and performance perspective. And periodic Siebel application & infrastructure upgrades and patches need to be managed and result in ongoing testing.

Oracle Application Testing Suite (ATS) helps organizations ensure the quality and performance of their Siebel applications. ATS is part of Oracle Enterprise Manager's Application Quality Management solutions that deliver high quality testing for all tiers of the application stack. ATS provides an end-to-end application testing solution that enables users to manage their test process, automate functional & regression testing and execute load tests to isolate and address performance bottlenecks. ATS can help you ensure that your Siebel applications meet end-user requirements, function as expected and are optimized for performance at go-live, leading to increased adoption and organizational efficiency.

## The Challenges of Testing Siebel

Siebel comprises a wide range of applications and critical business transactions that users rely on. The Siebel application infrastructure is a complex system comprised of multiple components, including a traditional Web-based front end, Siebel application and gateway servers and one of multiple possible databases. As a result, application quality and performance can be difficult to ensure prior to deployment with traditional test methods. Poor application quality and performance can have any number of detrimental effects including lost productivity, failed adoption and dissatisfied users - any of which can result in impact to bottom-line performance.

With the release of Siebel 7, Siebel introduced the Siebel Smart Web Architecture, a zero-footprint implementation that provides high levels of interactivity previously seen only in Windows desktop applications. The Siebel High Interactivity Client accomplishes this by separating data transfer from the user interface refreshes. The Siebel Smart Web Architecture was extended in release 7.7 to include a test API called Siebel Test Automation, making Siebel the first CRM system to be designed for "testability" and to enable automated functional and load testing of Siebel CRM applications.

Testing Siebel applications requires an understanding not only of the Siebel Smart Web Architecture but also of the various constraints inherent in Siebel environments. Some of these constraints include the various business and database rules that transactions must adhere to. Siebel supports many business verticals and organizations with its various application modules. Most companies tailor their Siebel implementation to fit their particular business model. The rules around these modules range from data use and re-use to "consumable" business steps that cannot be repeated or do not allow for concurrency during testing. Therefore, it is vital that subject matter experts be engaged at the beginning of the test project and stay involved throughout the performance and tuning phase. Collaboration between these subject matter experts is critical to the success of the project.

Siebel can also run on multiple operating systems, including Windows and UNIX systems, and various databases. The application contains many server types, and each tier (Web Server, Gateway Server, and Enterprise Server) may actually consist of a set of clustered servers to manage performance. This multi-tier architecture provides additional flexibility, enabling users to customize the application for specific needs. However, this architecture results in an increase in testing complexity and more difficult performance bottleneck identification.

## Siebel End-to-End Testing Methodology

Thorough testing can help identify Siebel application issues prior to deployment. Testing is one of the most challenging and time consuming parts of successfully deploying Siebel, but it is also the most critical to the project's success. Testing is also an ongoing activity that must be repeated as new application functionality, upgrades and patches are deployed.

The figure below from the Siebel Technical Architecture Kit shows Siebel's recommended methodology for end-to-end testing. As you can see, testing covers almost all parts of software implementation and rollout phases. This begins with defining a test strategy during the early phases of development. It then proceeds to functional testing, system testing and integration testing as the application is developed and finally user acceptance and performance testing prior to deployment.

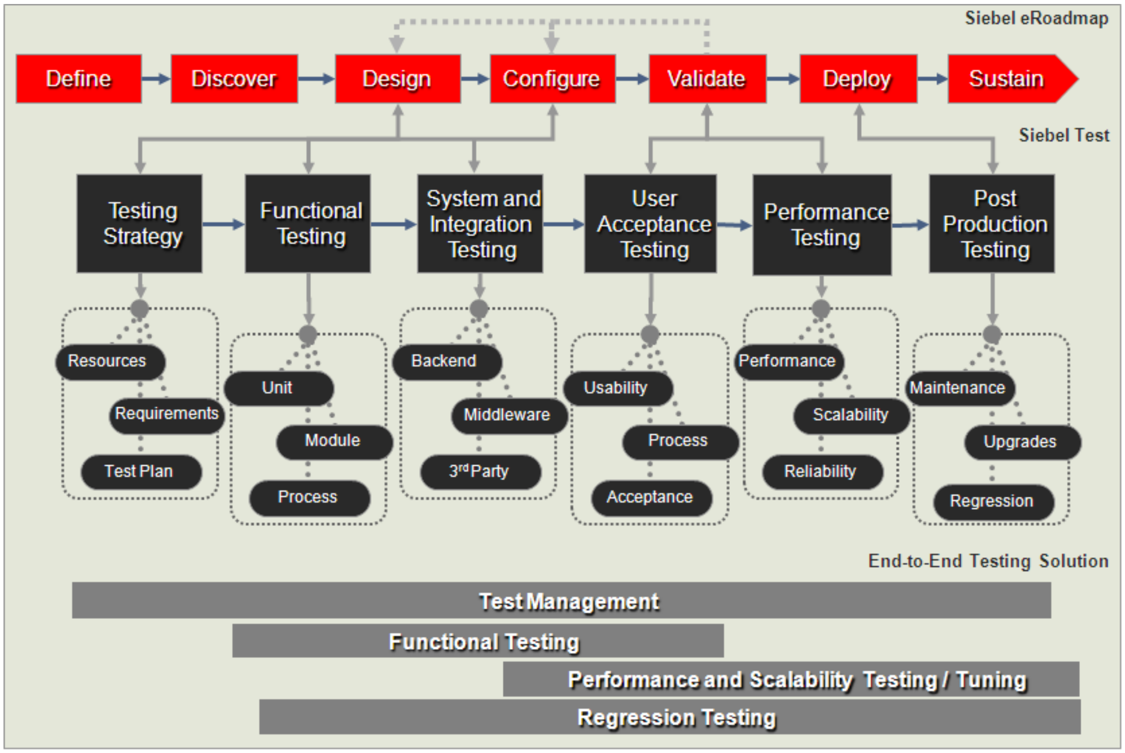


**Figure 1: Siebel Test Methodology Diagram from Siebel 8.1 Technical Architecture Kit – Test Automation Architecture Overview**

## Oracle Application Testing Suite

Oracle Application Testing Suite is a powerful, integrated test solution for ensuring application quality, performance and reliability. Application Testing Suite (ATS) includes a suite of products for automated functional testing, load testing and test management of Web, packaged and Service Oriented Architecture-based applications. By enabling you to automate your test cases, test and tune application performance and better manage your test processes – ATS helps you deliver higher quality applications while also increasing the efficiency of your testing team.

There are three separately licensed products in the Oracle Application Testing Suite (ATS):

- Oracle Test Manager for documenting and managing the overall test process including test requirements, test cases and issues.

- Oracle Functional Testing for automating functional and regression testing of Web applications, packaged applications and Web Services.
- Oracle Load Testing for automated load testing of Web applications, packaged applications and Web Services.

Built from the ground up to be the best-in-class solution for functional and performance testing of Web applications, the Oracle Application Testing Suite has been extended for testing of Siebel CRM. The ATS Siebel Testing Accelerators are fully integrated with Siebel Test Automation to enable more efficient, optimized functional testing and load testing for your Siebel applications. Additional features like ATS's integrated, Java-based scripting platform, collaborative Web-based load testing interface, built-in performance diagnostics and intuitive test management capabilities further differentiate ATS from the generic test tools prevalent in the marketplace.

## Test Planning and Management

Whether you are testing a major new Siebel deployment or doing regression testing as part of an upgrade, having a good test strategy in place sets the stage for success. Too often the test strategy phase is overlooked, usually creating issues later on in the development process.

There are multiple components to consider in your test strategy including:

- Test Requirements – which define what components of the application need to be tested, key transactions, and success criteria
- Test Plan – which defines the overall framework for test execution including which test cases will be executed, resources & timing
- Test Cases – which detail the specifics of how each test will be executed including individual test steps and expected results, and may consist of both manual test cases and automated test scripts
- Issues – which document any failures or problems identified during testing so they can be resolved prior to deployment

As you plan your Siebel test strategy, it's important to first determine what needs to be tested through discovery with application developers, business analysts and end users and decide what the priorities will be. You can then define test requirements and critical use cases which should be reviewed with key stakeholders to get consensus on testing objectives and identify any critical gaps.

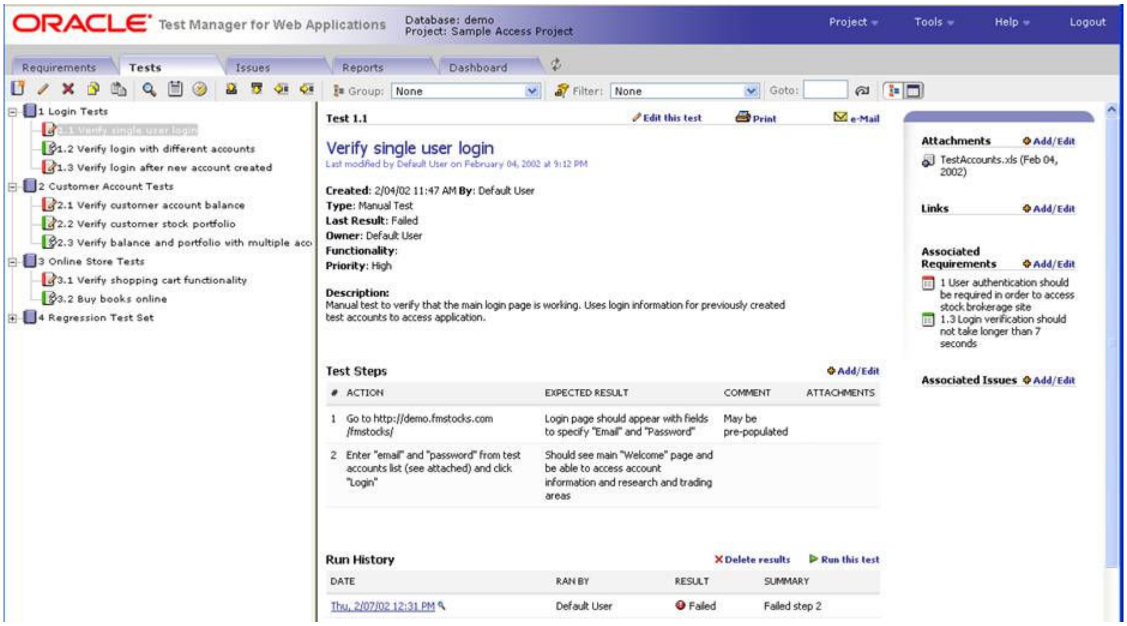
With this information you can then document specific test cases based on these test requirements & uses cases and map your tests to the requirements to ensure full requirements coverage. It's also important to agree on general logistics up front such as when will you test (i.e. for major deployments, or after every upgrade), who will test (i.e. users, business analysts, dedicated QA/test team) and how the tests will be run (i.e. manual testing, automated test tools). You should then publish a test plan and make it accessible to the entire team and centralize all test assets to promote collaboration. Finally, once testing is underway a centralized test management system will enable you to report regularly on testing status and overall application readiness

## Oracle Test Manager

Oracle Test Manager lets you manage your test process from a centralized Web-based console. Its intuitive interface provides comprehensive traceability from test requirements, to test cases and to issues identified during testing. You can use Oracle Test Manager to document both manual & automated test cases and execute these test cases directly from your test plan. The product provides three integrated modules:

- The Requirements module allows testers to document and manage their test requirements before the testing process begins.
- The Tests module enables testers to create detailed test plans, document automated and manual test cases, execute these test cases and store results.
- The Issues module allows testers and developers to capture and share issues found during the testing process and then track their resolution.

Oracle Test Manager's integrated reporting interface helps you provide end-to-end visibility into your test process. It's also fully customizable to fit your test methodology. And because it is built on a scalable enterprise architecture based on Oracle Database and Weblogic Server, it provides unlimited scalability to meet your expanding needs.



Database: demo
Project: Sample Access Project

1 Login Tests
1.1 Verify single user login
1.2 Verify login with different accounts
1.3 Verify login after new account created
2 Customer Account Tests
2.1 Verify customer account balance
2.2 Verify customer stock portfolio
2.3 Verify balance and portfolio with multiple acc
3 Online Store Tests
3.1 Verify shopping cart functionality
3.2 Buy books online
4 Regression Test Set

Test 1.1

Verify single user login
Last modified by Default User on February 04, 2002 at 9:12 PM

**Created:** 2/04/02 11:47 AM **By:** Default User
**Type:** Manual Test
**Last Result:** Failed
**Owner:** Default User
**Functionality:**
**Priority:** High

**Description:**
Manual test to verify that the main login page is working. Uses login information for previously created test accounts to access application.

Test Steps

| # | ACTION | EXPECTED RESULT | COMMENT | ATTACHMENTS |
|---|---|---|---|---|
| 1 | Go to http://demo.fmstocks.com /fmstocks/ | Login page should appear with fields to specify "Email" and "Password" | May be pre-populated | |
| 2 | Enter "email" and "password" from test accounts list (see attached) and click "Login" | Should see main "Welcome" page and be able to access account information and research and trading areas | | |

Run History

| DATE | RAN BY | RESULT | SUMMARY |
|---|---|---|---|
| Thu, 2/07/02 12:31 PM | Default User | Failed | Failed step 2 |

Attachments
TestAccounts.xls (Feb 04, 2002)

Links

Associated Requirements
1 User authentication should be required in order to access stock brokerage site
1.3 Login verification should not take longer than 7 seconds

Associated Issues

**Figure 2: Oracle Test Manager Helps You Document and Manage Siebel Test Cases**

# Functional & Regression Testing

Functional testing helps you validate that your application's functionality meets end user business requirements. Whether you are deploying a new Siebel application or upgrading an existing application, functional testing is critical to ensure the application works as expected and to identify and address any issues prior to deployment.

There are several key questions customers must consider before deploying their application:

- Will the Siebel application work properly and meet user requirements once deployed?
- Have all the critical Siebel transaction flows been validated?
- What is the impact of customizations and other changes that were made to the base Siebel deployment?
- Will any existing functionality break as a result of this Siebel upgrade or patch?

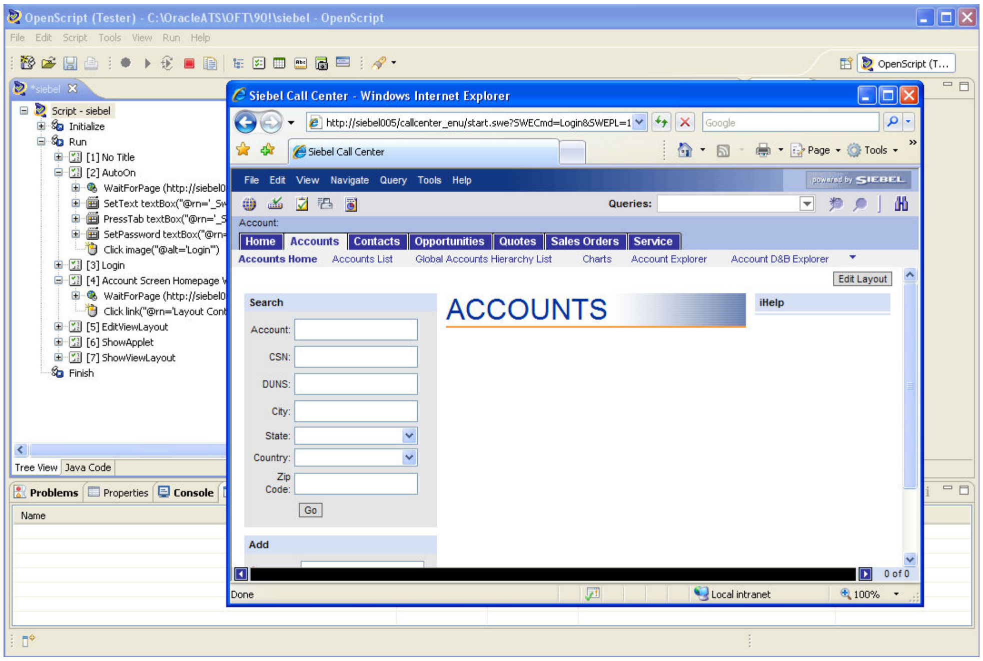
Functional testing helps you determine the answers to these questions. Manual testing accounts for the majority of functional & regression testing performed for all applications today, including Siebel. This testing may be performed by developers working on the application, dedicated testers, business analysts and possibly even end users - all of whom are tasked with stepping through the application and checking key functionality prior to deployment. This can be very time consuming, expensive and inefficient - especially in the face of a major application rollout or a series of repeated application upgrade cycles that require continuous ongoing testing.

Automated test tools can help you automate many of your Siebel test cases, extend test coverage to a greater percentage of your application and reduce the need for manual testing which saves you both time and money. However, automated testing of Siebel has its own challenges. In addition to acquiring and learning to use automated test tools, users must deal with the complexity of automating Siebel application transactions when creating automated test scripts. Siebel uses a combination of Standard Interactivity, Web-based user interface components and High Interactivity, Active-X based components within its various applications. The Siebel High Interactivity components in particular are a major challenge to automate since they cannot be natively identified by most test tools. As a result these tools must rely on coordinate-based identification to automate and validate these components. Siebel Test Automation APIs combined with Oracle Functional Testing address this challenge.

## Oracle Functional Testing

Oracle Functional Testing is an automated functional & regression testing solution for Web, SOA and Oracle packaged applications. The Functional Testing Accelerator for Siebel extends Oracle Functional Testing to provide an efficient and accurate way to automate functional and regression testing of Oracle's Siebel CRM applications. Oracle Functional Testing allows users to create test scripts that automate complex end-user transactions against their Web applications through an intuitive visual scripting interface combined with a powerful Java IDE. Users can validate application functionality, parameterize their scripts for data-driven testing and create regression suites to test new versions of their application in an automated fashion.

The Oracle Functional Testing Accelerator for Siebel extends these capabilities by integrating with Siebel Test Automation to simplify automation and validation of Siebel's rich client user interface components. Oracle Functional Testing supports both Standard-Interactivity (SI) and High Interactivity (HI) Siebel controls. It records user interactions with these controls to create automated functional test scripts as users interact with their Siebel application in the browser. It then provides custom test cases to validate Siebel application content during playback. With Oracle Functional Testing Accelerator for Siebel, users can effectively introduce automation into their functional test process to ensure the quality of their Siebel applications.



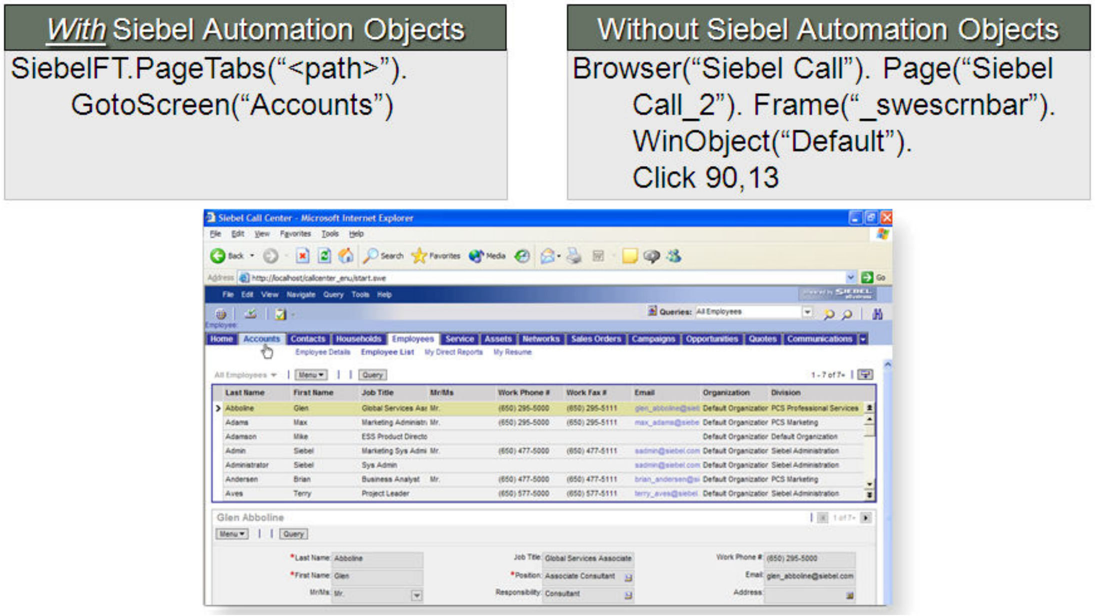
```
Script - siebel
  Initialize
  Run
    [1] No Title
    [2] AutoOn
      WaitForPage (http://siebel0
      SetText textBox("@rn='_Sw
      PressTab textBox("@rn='_S
      SetPassword textBox("@rn=
      Click image("@alt='Login'")
    [3] Login
    [4] Account Screen Homepage V
      WaitForPage (http://siebel0
      Click link("@rn='Layout Cont
    [5] EditViewLayout
    [6] ShowApplet
    [7] ShowViewLayout
  Finish
```

**Figure 3: Oracle Functional Testing Helps You Automate Siebel Functional & Regression Testing**

## Siebel Test Automation Integration

Test automation for Siebel CRM is a major challenge due to the rich nature of Siebel's High-Interactivity (HI) user interface controls. For most test tools, the only way to identify these rich controls is using windows coordinates to identify them by location. This is unreliable since coordinates can change due to issues like screen resolution, browser sizing or when even minor changes are made to the application.

Oracle Functional Testing integrates with Siebel Test Automation's Client Automation Services (CAS) API to record and play back user interactions with these controls. Once CAS is enabled on the Siebel Server, Oracle Functional Testing is able to capture more detailed information on the Siebel HI controls so they can be identified by more reliable attributes like component name. The diagram below shows the difference between test script actions captured with Oracle Functional Testing and Siebel Test Automation API vs. those captured by a generic, coordinate based test tool. As you can see, Siebel Test Automation makes test scripts more reliable and easier to understand since they actually leverage the logical UI name of the Siebel objects instead of just screen coordinates.



*With* Siebel Automation Objects

```
SiebelFT.PageTabs(“<path>”).
    GotoScreen(“Accounts”)
```

Without Siebel Automation Objects

```
Browser(“Siebel Call”). Page(“Siebel
    Call_2”). Frame(“_swescrnbar”).
    WinObject(“Default”).
    Click 90,13
```

Siebel Call Center - Microsoft Internet Explorer

Address: http://localhost/callcenter_enu/start.swe

Queries: All Employees

1 - 7 of 7+

| Last Name | First Name | Job Title | Mr/Ms | Work Phone # | Work Fax # | Email | Organization | Division |
|---|---|---|---|---|---|---|---|---|
| Abboline | Glen | Global Services Ass | Mr. | (650) 295-5000 | (650) 295-5111 | glen_abboline@siebl | Default Organization | PCS Professional Services |
| Adams | Max | Marketing Administr | Mr. | (650) 295-5000 | (650) 295-5111 | max_adams@siebe | Default Organization | PCS Marketing |
| Adamson | Mike | ESS Product Directo | | | | | Default Organization | Default Organization |
| Admin | Siebel | Marketing Sys Admi | Mr. | (650) 477-5000 | (650) 477-5111 | sadmin@siebel.com | Default Organization | Siebel Administration |
| Administrator | Siebel | Sys Admin | | | | sadmin@siebel.com | Default Organization | Siebel Administration |
| Andersen | Brian | Business Analyst | Mr. | (650) 477-5000 | (650) 477-5111 | brian_andersen@si | Default Organization | PCS Marketing |
| Aves | Terry | Project Leader | | (650) 577-5000 | (650) 577-5111 | terry_aves@siebel | Default Organization | Siebel Administration |

Glen Abboline

1 of 7+

*Last Name: Abboline

*First Name: Glen

Mr/Ms: Mr.

Job Title: Global Services Associate

*Position: Associate Consultant

Responsibility: Consultant

Work Phone #: (650) 295-5000

Email: glen_abboline@siebel.com

Address:

**Figure 4: Siebel Test Automation Integrated with Oracle Functional Testing Makes Test Scripts More Reliable**

## Load & Performance Testing

Load testing can help you ensure that your Siebel application will perform and scale under real user workloads once it’s deployed to production. This can help you ensure that it will be able to withstand the expected number of concurrent users while maintaining acceptable performance and response times. It can also help you identify and address critical bottlenecks prior to deployment. Stress testing can be performed to test beyond the limits of normal operation and helps you assess the capacity and scalability of your application infrastructure.

Some key questions to consider in terms of application performance are:

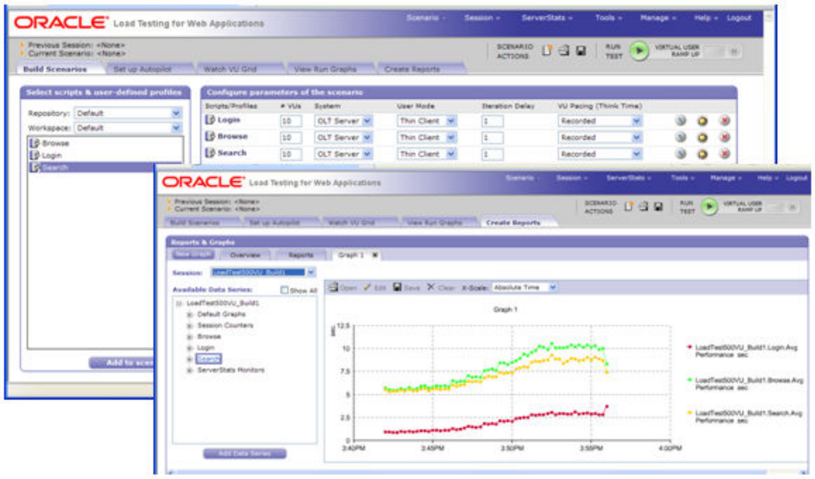
- Will the Siebel application be able to handle the number of concurrent users we expect?
- What will the response times be like during peak usage periods?
- Are there any bottlenecks in the application or infrastructure that will limit performance?
- Did we size the hardware appropriately and do we have the capacity to meet future growth?

It's impossible to analyze application performance and identify bottlenecks without simulating production-level loads. As with functional testing, it is important to carefully plan your load testing efforts in advance to ensure success. This involves analyzing the application and determining which business transactions to test, deciding what level of load you need to scale to, and defining what your success criteria for acceptable performance is. This also involves building out your test environment and ensuring that your Siebel application will be a reasonable replica of production environment (if you aren't able to test against the true production environment).

Unlike for functional testing, there is no realistic manual alternative for Siebel load testing. However, automated load testing for Siebel has its own challenges. Load test scripts are difficult to create due to highly dynamic nature of Siebel applications. Load test scripts typically automate transactions at the protocol level for maximum scalability, however Siebel requests are very dynamic and if the recording tool captures hard-coded data values then they will not play back. Manual parameterization is often required which is tedious and time-consuming, plus it requires extensive Siebel expertise. Once you get your tests running, the highly distributed Siebel architecture makes performance bottlenecks difficult to identify.

## Oracle Load Testing

Oracle Load Testing lets you run realistic load tests for Web, SOA and Oracle packaged applications – helping you simulate thousands of concurrent users and analyze the impact of production load levels on application performance. The Load Testing Accelerator for Siebel extends Oracle Load Testing to enable automated load and performance testing of Oracle's Siebel CRM applications. Oracle Load Testing lets you simulate thousands of virtual users accessing the Siebel application simultaneously to measure the effect of user load on application performance. During performance testing, Oracle Load Testing measures end-user response times as well as the performance of the underlying application infrastructure to help you analyze performance and identify bottlenecks.



Previous Session: <None>
Current Scenario: <None>

Repository: Default
Workspace: Default

- Browse
- Login
- Search

| Scripts/Profiles | # VUs | System | User Mode | Iteration Delay | VU Pacing (Think Time) |
|---|---|---|---|---|---|
| Login | 10 | OLT Server | Thin Client | 1 | Recorded |
| Browse | 10 | OLT Server | Thin Client | 1 | Recorded |
| Search | 10 | OLT Server | Thin Client | 1 | Recorded |

Session: LoadTest500VU_Build1

Available Data Series: □ Show All

- LoadTest500VU_Build1
  - Default Graphs
  - Session Counters
  - Browse
  - Login
  - Search
  - ServerStats Monitors

X-Scale: Absolute Time

**Figure 5: Oracle Load Testing Helps You Test Siebel Application Performance**

## Siebel Load Test Scripting

Oracle Application Testing Suite provides a unique integrated scripting platform in Oracle Functional Testing for both functional test automation and script creation for load testing. The Load Testing Accelerator for Siebel extends these capabilities by integrating with Siebel Test Automation to simplify load script generation for Siebel's complex and dynamic Web transactions. Oracle Load Testing then allows you to configure and run these scripts across thousands of virtual users to assess application performance.

Scripts are created in the Oracle Functional Testing component of ATS by capturing user interactions with the system in real time, so there is no need for complicated programming or arcane domain expertise. In addition, Oracle Functional Testing is used to create both functional test scripts and load test scripts, so users don't have to learn multiple test tools with different scripting languages. As a result of this approach, Application Testing Suite delivers the best time-to-value in functional test automation and performance testing. Additionally, with the integration of the Siebel Test Automation libraries, scripting against standard and highly customized Siebel CRM installs has been greatly simplified. The same record/playback approach that has been so effective when working with other Web applications has been extended for Siebel scripting against the Siebel High Interactivity Client.

Creating working test scripts is typically one of the biggest challenges in load test automation. Load testing solutions automate user requests at the protocol level to simulate the application's interactions with the back-end servers and generate scalable loads of hundreds or thousands of virtual users. For Web applications, these http-based protocol requests typically contain dynamic parameters to manage user session and state information that load test tools must account for and correlate in order to get their scripts working. Siebel applications are extremely dynamic, given the complex business logic and extensive personalized functionality the applications employ. Manually correlating application requests and programming load tests scripts to get them working, which is difficult for most Web applications, is even more challenging for Siebel without in-depth knowledge of how Siebel applications work. As a result, most load test solutions on the market will not provide adequate capabilities for Siebel load testing.

Oracle Functional Testing, which is used for creating both load and functional test scripts, integrates with Siebel Test Automation's Siebel Correlation Library to allow users to create performance testing scripts in minutes by recording real interactions with Siebel applications. Siebel Correlation Library automatically identifies dynamic Siebel request parameters and converts them into Oracle Load Testing script variables. This eliminates the need to identify dynamic parameters and correlate these requests manually, which results in a dramatic reduction in scripting time. In the cases when out-of-the-box support is insufficient, the load testing scripts may be edited in the same scripting interface, without resorting to programming. This again enables the performance test team to spend less time scripting and more time testing and identifying and resolving performance issues.

## Collaborative Load Testing Interface

Testing and tuning the performance of Siebel CRM applications is a complex and iterative process. As with most load testing exercises, "breaking" the application and identifying performance failures is not the desired outcome; finding bottlenecks and tuning the application to maximize performance are. This is a difficult challenge and consequently testers, developers, architects, and others must work together to analyze load test results and tune the application. The most effective way to identify and resolve performance issues is through collaboration with these domain experts. A load test tool at its core simply applies load and takes performance measurements. The performance test team must find the performance bottlenecks and resolve them. The key to successful collaboration during a performance test is having access to both real-time performance measurements and a team of domain experts to help analyze and identify bottlenecks. Oracle Load Testing for Siebel provides a platform to enable such collaboration.

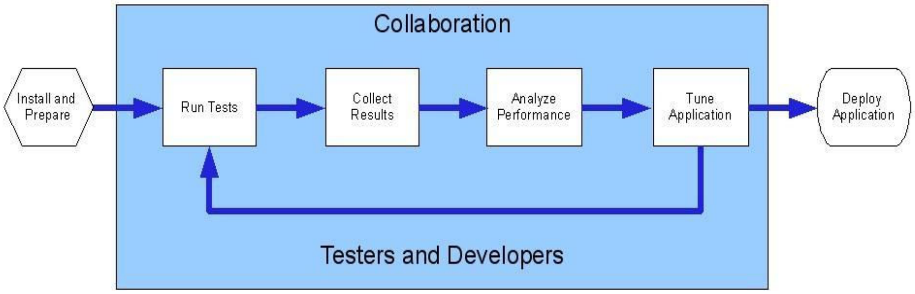


**Figure 6: Collaborative load testing process**

The load test controller in the Oracle Application Testing Suite runs using a standard Weblogic application server platform and is accessed via any standard Web browser. The benefits of this architecture are two-fold. First, the Web interface provides greater accessibility to both enable and encourage collaboration during performance testing. Test team participants are no longer limited to a single console with a single operator. Each domain expert involved in a performance test may view the test in progress and generate the reports that are specifically relevant to their area of expertise. This results in faster and more efficient identification of performance bottlenecks during the test and more effective testing overall. Second, the platform is much more scalable than the standard desktop install, resulting in higher available processing power and throughput. This enables higher volumes of generated load without resorting to using multiple instances of the load controller. As a result, there is no need to correlate reports and concatenate data after the test, again enabling much greater testing efficiency and, as a result, better application performance.

### Built-in Performance Diagnostics for Siebel

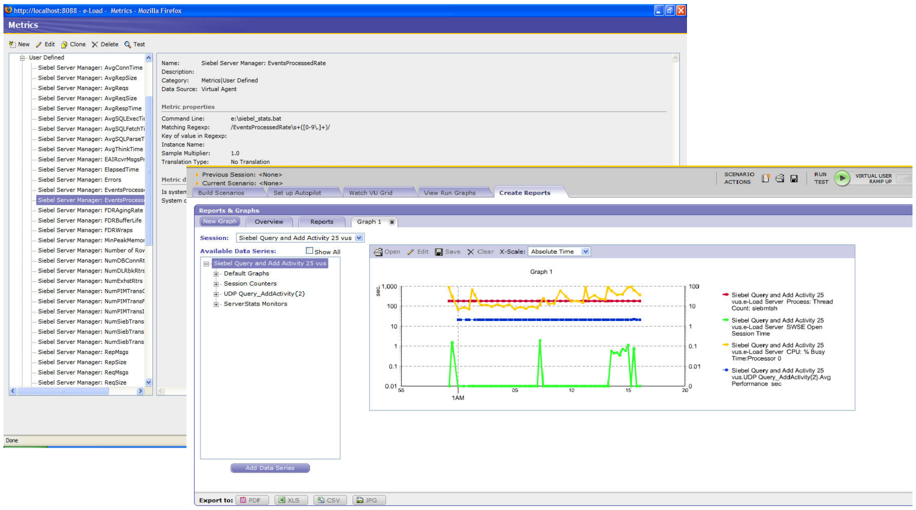
Another feature of the Oracle Application Testing Suite important for performance testing of Siebel CRM is the built-in capability to measure the performance of the application components under test. This includes the ability to measure operating system and hardware metrics like CPU utilization and memory consumption, database performance information from Oracle or other databases, and middleware and application server performance metrics. The Siebel-specific capabilities include using the Oracle Load Testing Virtual Agent data collector to gather performance information from the Siebel Server Manager, SARM, or the SWSE Administration page. It should be noted that these server monitoring capabilities are a part of the default Oracle Load Testing product installation and do not carry any additional licensing costs – which is also unique in the marketplace.

With this additional level of monitoring, poorly performing components of the Siebel architecture can be quickly identified during a load test. Further, this data can be quickly correlated to the end-user response time measured during the test to fully understand the bottleneck impact on performance and allow the testing team to correctly prioritize optimization efforts. Finally, this and all other data is available through the Web-based load test console interface as discussed earlier, and test team members at disparate locations may still take an active part in performance testing.

Monitoring the components of your Siebel environment during load testing is extremely important. These components include Siebel Servers, Web servers, Gateway name server, Reports Server, the database, load balancing, and any integrated external system or integration points. In addition, the infrastructure resources such as disks, CPUs, networks, database servers, and authentication and directory servers, also need to be monitored on each component in the Siebel environment, since they can potentially impact application operation and performance. Finally, metrics like memory usage (both physical and virtual), CPU utilization, available disk space, and overall network access, bandwidth, and traffic also need to be measured during the load test.

Siebel offers several interfaces for collecting provided health and availability metrics. ServerStats, a unique feature of Oracle Load Testing, can make use of the command-line program srvrmgr (Server Manager) to monitor the run-time status of the Siebel Enterprise Server, individual Siebel Servers, Component groups, Components, Server Tasks, and user sessions. Along with srvrmgr, Siebel generates log files to record data for each Siebel Server deployed as part of a Siebel Enterprise. Information contained in the Siebel Server component log files can indicate deadlocks reached or exceeded threshold values. ServerStats can gather all these metrics using a standard log file monitor. Deeper diagnostics provided by Siebel ARM (SARM) can also be analyzed during a load test or accessed through Enterprise Manager's Siebel Management Pack.

Beyond Siebel specific KPIs, ServerStats can use system-level profiles to gather and measure the physical resources consumed by all components, particularly memory and CPU. As an example, system CPU utilization should be at or below 70% during the test, and the components should be able to accommodate temporary spikes, whether in the number of tasks running concurrently, the amount of memory being used, or the percentage of CPU usage. Monitoring values like these will help ensure that system capacity is being used effectively without overloading.


http://localhost:8088 - e-Load - Metrics - Mozilla Firefox
Metrics
New
Edit
Clone
Delete
Test
User Defined
Siebel Server Manager: AvgConnTime
Siebel Server Manager: AvgRepSize
Siebel Server Manager: AvgReqs
Siebel Server Manager: AvgReqSize
Siebel Server Manager: AvgRespTime
Siebel Server Manager: AvgSQLExecTi
Siebel Server Manager: AvgSQLFetchTi
Siebel Server Manager: AvgSQLParseT
Siebel Server Manager: AvgThinkTime
Siebel Server Manager: EAIRcvrMsgsPr
Siebel Server Manager: ElapsedTime
Siebel Server Manager: Errors
Siebel Server Manager: EventsProcess
Siebel Server Manager: EventsProcess
Siebel Server Manager: FDRAgingRate
Siebel Server Manager: FDRBufferLife
Siebel Server Manager: FDRWraps
Siebel Server Manager: MinPeakMemor
Siebel Server Manager: Number of Rov
Siebel Server Manager: NumDBConnRt
Siebel Server Manager: NumDLRbkRtrs
Siebel Server Manager: NumExhstRtrs
Siebel Server Manager: NumPIMTransC
Siebel Server Manager: NumPIMTransF
Siebel Server Manager: NumPIMTransI
Siebel Server Manager: NumSiebTrans
Siebel Server Manager: NumSiebTrans
Siebel Server Manager: NumSiebTrans
Siebel Server Manager: RepMsgs
Siebel Server Manager: RepSize
Siebel Server Manager: ReqMsgs
Siebel Server Manager: ReqSize
Done

Name: Siebel Server Manager: EventsProcessedRate
Description:
Category: Metrics|User Defined
Data Source: Virtual Agent

Metric properties

Command Line: e:\siebel_stats.bat
Matching Regexp: /EventsProcessedRate\s+([0-9\.]+)/
Key of value in Regexp:
Instance Name:
Sample Multiplier: 1.0
Translation Type: No Translation

Metric d
Is system
System c

Previous Session: <None>
Current Scenario: <None>

Session: Siebel Query and Add Activity 25 vus

Available Data Series: □ Show All

X-Scale: Absolute Time

Graph 1
sec.
1,000
100
10
1
0.1
0.01
100
10
1
0.1
0.01
0
55
05
10
15
20
1AM
Siebel Query and Add Activity 25 vus.e-Load Server Process: Thread Count: siebmtsh
Siebel Query and Add Activity 25 vus.e-Load Server SWSE Open Session Time
Siebel Query and Add Activity 25 vus.e-Load Server CPU: % Busy Time:Processor 0
Siebel Query and Add Activity 25 vus.UDP Query_AddActivity{2}.Avg Performance sec
Export to:
PDF
XLS
CSV
JPG

**Figure 7: Siebel CRM Diagnostics in Oracle Load Testing**

## Conclusion

The Oracle Application Testing Suite is a best-of-breed solution for the performance testing of Web-based applications. The Siebel Testing Accelerators extend these capabilities to provide unparalleled value for ensuring the quality and performance of Siebel CRM applications. Oracle Test Manager provides an intuitive and easy-to-use solution for managing the Siebel test process. Oracle Functional Testing allows users to automate their Siebel test cases and reduce the need for manual testing. The integration of Siebel Test Automation as part of the ATS Siebel Testing Accelerators enables quick and easy scripting, greater scripting accuracy and enhanced validation for both functional testing and load testing. Oracle Load Testing's Web-based performance testing interface offers unmatched scalability and enables performance testing to become a collaborative activity. Its built-in diagnostic capabilities deliver the ability to quickly and efficiently identify poorly performing Siebel infrastructure or application components. With these features, the Application Testing Suite is the recommended solution for organizations looking to improve the performance of their Siebel CRM installations or any other business- and mission-critical applications deployed over the Web.

Oracle is committed to developing practices and products that help protect the environment

Comprehensive Testing for Siebel With Oracle
Application Testing Suite
February 2010

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
oracle.com

Copyright © 2010, Oracle and/or its affiliates. All rights reserved.

This document is provided for information purposes only and the contents hereof are subject to change without notice. This document is not warranted to be error-free, nor subject to any other warranties or conditions, whether expressed orally or implied in law, including implied warranties and conditions of merchantability or fitness for a particular purpose. We specifically disclaim any liability with respect to this document and no contractual obligations are formed either directly or indirectly by this document. This document may not be reproduced or transmitted in any form or by any means, electronic or mechanical, for any purpose, without our prior written permission.

Oracle and Java are registered trademarks of Oracle and/or its affiliates. Other names may be trademarks of their respective owners.

AMD, Opteron, the AMD logo, and the AMD Opteron logo are trademarks or registered trademarks of Advanced Micro Devices. Intel and Intel Xeon are trademarks or registered trademarks of Intel Corporation. All SPARC trademarks are used under license and are trademarks or registered trademarks of SPARC International, Inc. UNIX is a registered trademark licensed through X/Open Company, Ltd. 0110