

# Oracle® Application Testing Suite

Release Notes

Version 9.10 for Microsoft Windows (32-Bit)

E17387-01

April 2010

---

This document includes updated information for Oracle Application Testing Suite version 9.10. The information in this document is more up-to-date than that in the manuals. Many of the issues outlined in this document will be corrected in upcoming releases. If you have any questions or problems, please contact our support group at <http://www.oracle.com/support/index.html>.

## 1 Contents

This document has the following sections:

- [Section 1, "Contents"](#)
- [Section 2, "New Features and Updates in this Release"](#)
- [Section 3, "System Requirements"](#)
- [Section 4, "Installing the Oracle Application Testing Suite"](#)
- [Section 5, "Backwards Compatibility and Upgrading Scripts"](#)
- [Section 6, "Database Migration"](#)
- [Section 7, "Tutorial and Documentation for the Oracle Application Testing Suite"](#)
- [Section 8, "Known Problems, Limitations and Workarounds"](#)
- [Section 9, "Documentation Accessibility"](#)

## 2 New Features and Updates in this Release

This version includes maintenance improvements to the existing features as well as new features outlined below.

### 2.1 Oracle Application Testing Suite 9.10 - New Features and Updates

- **Oracle Fusion/ADF Functional Testing Accelerator** - Enables enhanced support for automated functional testing of Fusion/ADF applications. Extends ATS's Web functional testing capabilities by adding native support for ADF GUI components for object identification, automation, and validation. Oracle ADF is a key technology in Fusion Applications and can be used to build custom applications.
- **Oracle Fusion/ADF Load Testing Accelerator** - Enables enhanced support for automated load testing of Fusion/ADF applications. Extends ATS's Web/HTTP load testing capabilities by adding a custom load testing correlation library and related scripting enhancements for Fusion/ADF load testing.

- **Script Assets Manager** - Standardizes management of scripts and related assets like Data Bank files, Object Libraries, and Script Functions.
- **Enhanced Script Functions Support** - Provides an easier way for users to create shared functions through the OpenScript user interface and call those functions from any test script to create more modular, reusable script components.
- **Oracle Real User Experience Insight (RUEI) Script Import** - Allows users to generate Oracle Application Testing Suite load test scripts from the Oracle Real User Experience Insight product by first exporting real user session transaction data from RUEI and then importing it into OpenScript to generate a load test script.
- **Oracle E-Business Suite R12 Test Starter Kit / Sample Scripts** - Provides a test starter kit with sample test scripts for Oracle EBS R12 applications based on the sample EBS "Vision" database.
- **OpenScript Tree View Debugger** - Provides debugging capabilities from the OpenScript Tree View including the ability to set breakpoints, step through script nodes, step over, pause, and watch variables. Previously, script debugging was only available from OpenScript's Java Code View.
- **Download Manager** - Simplifies Oracle Application Testing Suite load test scripts by filtering out embedded HTTP URL requests for Web page resources (i.e. images, js, css, etc.) during script recording in OpenScript and then giving users the option of whether to automatically find and request those resources on playback in OpenScript and Oracle Load Testing.
- **Result Code Verification** - Makes it easier to adjust the behavior of an OpenScript test script based on the outcome of script commands by providing a result code that can be queried after command executes.
- **Oracle Load Testing Reporting Enhancements** - Provides drill-down capabilities that link OLT's performance summary report to corresponding load test graphs for both real-time and post-run reporting.
- **Enterprise Manager J2EE Middleware Diagnostics Integration** - Provides integration between Oracle Load Testing and Enterprise Manager that enables users to view J2EE middleware performance diagnostics during load tests for identifying bottlenecks under load.
- **Oracle Database ServerStats Profile** - New Oracle Load Testing ServerStats monitoring profile for Oracle Database to capture key performance metrics during a load test.

## 2.2 Oracle Application Testing Suite 9.10 - Additional Notes

- **Oracle Weblogic Server Support** - Oracle Application Testing Suite is certified with Oracle Weblogic Server as the application server for Oracle Load Testing and Oracle Test Manager. Users of Oracle Load Testing and Oracle Test Manager will receive a free restricted use license of Oracle Weblogic Server for use with these products. Oracle Weblogic Server is automatically installed and configured with ATS and replaces JBoss Server used in prior versions.
- **Oracle Database 11g Enterprise Edition Support** - Oracle Application Testing Suite is certified with Oracle Database 11g Enterprise Edition as the database repository for Oracle Load Testing and Oracle Test Manager. Users of Oracle Load Testing and Oracle Test Manager will receive a free restricted use license of Oracle Database 11g Enterprise Edition for use with these products. Oracle Application

Testing Suite also supports Oracle Database 10g. An Oracle 10g Express Edition (XE) database can be automatically installed and configured with ATS for demo purposes and replaces Microsoft Access used in prior versions.

- **Oracle Functional Testing Legacy Scripting Platform** - Oracle Functional Testing's legacy, Visual Basic-based scripting platform is included with the ATS 9.1 release, but will no longer be included with Oracle Application Testing Suite starting with the version 9.2 release. Customers are encouraged to upgrade to the new Oracle Functional Testing – OpenScript scripting platform. Customers using Oracle Functional Testing legacy scripting platform can continue to use Oracle Application Testing Suite version 9.1 or older versions under Oracle's Lifetime Support Policy. Customers using Oracle Functional Testing and Visual Basic for Applications can continue to use Oracle Application Testing Suite version 8.4 or older under Oracle's Lifetime Support Policy. Oracle Application Testing Suite version 8.5 and newer versions do not include the VBA module (see below).
- **Visual Basic for Applications (VBA)** - Microsoft's Visual Basic for Applications (VBA) is no longer available in Oracle Application Testing Suite (since version 8.5) as Microsoft has announced the discontinuation of this technology. See <http://vba.summsoft.com/>. VBA and associated features can no longer be enabled in Oracle Functional Testing. However, Visual Script extensibility in Oracle Functional Testing is still available using Test Scriptlets (VBScript). Existing customers that require VBA support can continue to use ATS 8.4 or 8.3 under Oracle's Lifetime Support Policy at <http://www.oracle.com/support/lifetime-support-policy.html>.
- **Oracle Functional Testing Proxy Certificate** - Oracle Functional Testing proxy recording uses a pre-packaged certificate for each session. Users can view the certificate when navigating to any site with SSL/TLS. The installation does not create unique certificates for each installation. The same certificate is used for all installations.
- **Testing HTTPS Sites** - When testing HTTPS sites, Oracle Functional Testing's "Proxy Recorder" feature should only be used to record scripts against trusted Web sites.

## 3 System Requirements

Oracle Application Testing Suite has the following system requirements:

### 3.1 Oracle Functional Testing

Oracle Functional Testing/OpenScript has the following system requirements:

- Operating System (32 bit versions only): Windows XP, Windows Vista, Windows 2003, Windows 7 (OFT-OpenScript only), Windows 2008 (OFT-OpenScript only)
- Memory: Minimum 1 GB
- System: x86 32-bit processor, 1.5 GHz or faster
- Disk Space: 4 GB minimum
- Browser: Internet Explorer 6.x, 7.x, 8.x. and Firefox 3.0/3.5 (OFT-OpenScript only)

### 3.2 Oracle Load Testing

Oracle Load Testing has the following system requirements:

- Operating System (32 bit versions only): Windows XP, Windows Vista, Windows 2003, Windows 7 (OFT-OpenScript/Java Agent only), Windows 2008 (OFT-OpenScript/Java Agent only)
- Memory: Minimum 2 GB.
- System: x86 32-bit processor, 1.5 GHz or faster
- Disk Space: 10 GB minimum free (at least 3 GB free on the system drive)
- Browser: Internet Explorer 6.x, 7.x, 8.x, Firefox 3.0/3.5
- Database: Oracle Database 10g or 11g
- Application Server: Oracle WebLogic 11g (10.3.2.0)
- Oracle Load Testing Agent: Linux 32-bit

---

**Note:** These are minimum requirements only and actual requirements for the Oracle Load Testing Server will vary depending on the size and configuration of your load test. If you are running larger load tests, of greater than 1000 Virtual Users, you should consider deploying Oracle Load Testing on a faster server class machine with additional RAM recommended.

---

---

**Note:** Oracle recommends running the Oracle Load Testing Server and Agents on separate systems for production load testing. The amount of memory required on the Oracle Load Testing Agent systems may increase based on the number of Virtual Users that will be assigned to run on each Agent. For more information, visit the QA/Testing Technology Center on Oracle Technology Network or contact your Oracle representative.

---

### 3.3 Oracle Test Manager

Oracle Test Manager has the following system requirements:

- Operating System (32 bit versions only): Windows XP, Windows Vista, Windows 2003, Windows 7, Windows 2008
- Memory: Minimum 2 GB.
- System: x86 32-bit processor, 1.5 GHz or faster
- Disk Space: 10 GB minimum free (at least 3 GB free on the system drive)
- Browser: Internet Explorer 6.x, 7.x, 8.x, Firefox 3.0/3.5
- Database: Oracle Database 10g or 11g
- Application Server: Oracle WebLogic 11g (10.3.2.0)

---

**Note:** These are minimum requirements only and actual requirements for Oracle Test Manager will vary depending on the quantity of test assets stored and number of concurrent users accessing the application. If you have a large quantity of test cases, requirements or issues stored and/or are deploying Oracle Test Manager for a team greater than 10 users, you should consider deploying Oracle Test Manager on a faster server class machine with additional RAM recommended.

---

### 3.4 Additional Notes

The following are additional notes about system requirements:

- The ATS installer can automatically install and configure an Oracle 10g Express Edition (XE) database for use with Oracle Load Testing and Oracle Test Manager. Although this is useful for demos, for production usage switching to Oracle's 11g Enterprise Edition database is highly recommended. Oracle 11g Standard Edition and Oracle 10g EE or SE are also supported.
- System memory requirements do not take into account other processes that may be running and using memory on your system, reducing free uncommitted memory available.
- When installed on the same machine, Oracle Load Testing and Oracle Test Manager will run on the same server and use shared system resources. Oracle Functional Test/OpenScript are separate applications that have their own memory and CPU requirements.

## 4 Installing the Oracle Application Testing Suite

This section explains the installation procedures for the Oracle Application Testing Suite.

---

**Caution:** The machines where the OpenScript product is installed should be strictly used for testing. The security features of the browsers have been disabled on this machine to enable recording and playback operations. The browsers cannot be used for secure browsing.

---

### 4.1 Installing the Applications

There are two ways that you can install Oracle Application Testing Suite:

From the Oracle Web site: <http://www.oracle.com/>

1. Select and download the Oracle Application Testing Suite (*oats-version.zip*) from the web site and save it to a temporary directory on your hard disk.
2. Unzip the downloaded file and run the executables: *oats###.exe* for Oracle Application Testing Suite, *openScript###.exe* for OpenScript. If you plan to install both, you should install Oracle Application Testing Suite (*oats###.exe*) first as this will affect the default repository locations.
3. Follow the setup instructions to install the Oracle Application Testing Suite and/or OpenScript.

From the Oracle Application Testing Suite CD:

1. Insert the Oracle Application Testing Suite CD into your computer's CD-ROM drive.
2. In Windows, click **Start -> Run** and browse to the drive letter that corresponds to your computers CD-ROM drive.
3. Run the executables: oats###.exe for Oracle Application Testing Suite and openScript###.exe for OpenScript in the root directory of the CD-ROM. If you plan to install both, you should install Oracle Application Testing Suite (oats###.exe) first as this will affect the default repository locations.
4. Follow the setup instructions to install the Oracle Application Testing Suite and/or OpenScript.

#### 4.1.1 Manually Configuring the Installation

The above installation procedure for Oracle Application Testing Suite (oats###.exe) installs the applications, database, and Web server. If during the installation procedure you select the **Do nothing, I will configure Oracle Application Testing Suite later (advanced)** option you will need to manually create the database schema.

To manually create the database schema:

1. Open a command window and run the DBConfig utility using this command:  
`<installdir>\jdk\jre\bin\javaw.exe -Djava.library.path=<installdir>\install\lib -jar DBConfiguration.jar`
2. Click the third radio button, enter your database parameters, and click Save. Make a note of the OATS schema username (i.e. the user table into which the schema will be created).
3. Edit `<installdir>\bin\config_ds_offline.py`. Change the line:  
`cmo.setValue('oats')`

so that the value set is the same as the user noted in step 2.

4. Set the environment variables as follows:  
OATS\_HOME: your install folder  
OATS\_PASS: the 'master password' you entered during installation  
OATS\_DBPASS: the password for the OATS schema user noted in step2  
OATS\_DBURL: the database URL in the following format:  
`jdbc:oracle:thin:@//<hostname>:<port>/<servicename>`
5. Run `<installdir>\bin\recreateDomain.bat`. This will delete your previous OATS domain, if any, and recreate it.

## 4.2 Installing the Oracle Load Testing Remote Agent

Oracle Load Testing allows you to distribute your Virtual Users to run from remote Agent machines. The Oracle Load Testing Server will connect to Oracle Load Testing Agent systems to start and run your Virtual Users on those machines. The Oracle Application Testing Suite Remote Agent is a component of the Oracle Application Testing Suite installation that enables Virtual Users to be distributed to these Agent systems. Users can either install the full Oracle Application Testing Suite installation or

just the Remote Agent install component on their Agent machines to enable this functionality.

#### **4.2.1 Installing the Remote Agent**

To install the Remote agent:

1. Download the Oracle Application Testing Suite from the download Web site.
2. Unzip the downloaded file and run the oats9xx.exe executable.
3. Follow the setup procedure to the **Select Components and Installation Directory** screen.
4. Select to install the full Oracle Application Testing Suite or select just the **Remote Agent** check box and clear all other boxes to install just the Remote Agent software.
5. Verify network access from the Controller workstation to the Agent workstations and configure the Agent Workstations as explained in the following section.

#### **4.2.2 Configure Remote Agent Service Login**

To specify the user login for the Oracle Load Testing Agent Service, open the Services control panel on the Agent machine and change the login credentials for the "Oracle Load Testing Agent Service". By default, the Oracle Load Testing Agent Service will run under the Local System account.

#### **4.2.3 Verify Network Access to Agent Systems**

Once you have the Oracle Load Testing Server and Agent software installed on the individual systems, you should verify network access between the Oracle Load Testing Server system and each Remote Agent system. This section provides basic tips and techniques to make sure the Oracle Load Testing Server system can successfully communicate with each Remote Agent system.

- Make sure that you have the Oracle Load Testing Agent software loaded on the Agent system(s) and that it is the same version as the Oracle Application Testing Suite software that is loaded on the Oracle Load Testing Server system. The systems you plan to use as Agents must have either the Oracle Load Testing Agent software or the full Oracle Application Testing Suite installed to work as agents.
- Make sure you can successfully Ping all of the Agent systems from the Oracle Load Testing Server system. The machine names you use to Ping the systems are the same names that you will specify for the Agent systems in the Oracle Load Testing server. You can also use the IP addresses of the agent systems. If you cannot successfully Ping the Agent systems, contact your network administrator to resolve the issue. If you cannot Ping the Agent systems from the Oracle Load Testing Server system, you will not be able to run the Agents from the server.
- Make sure that the same user is logged in on both the Oracle Load Testing server system and all of the Agent systems. All of the Agent systems must have a user logged in to be controlled by the Oracle Load Testing Server system. You may be able to log in as a different user on the Agent systems as long as the user login has the same administrative privileges as the user logged in on the server system.
- From the server system, try mapping a drive on each of the Agent systems using Windows Explorer. Depending on how your network is setup, the server system may not be allowed to start up processes on the Agent systems. The easiest way

around this is to map a drive to the Agent system in order to authenticate with Windows.

- In the Oracle Load Testing server add a Visual Script to the Scenario Profiles list. Enter the machine name or IP address of the Agent system where you want to run the Visual Script into the Systems Manager and select that machine in the Systems field on the Build Scenario tab of Oracle Load Testing.

### 4.3 Installing over beta versions or earlier releases of Oracle Application Testing Suite

If you have a 9.10 beta or Early Access version of the Oracle Application Testing Suite installed, you should un-install the earlier release of the Oracle Application Testing Suite before installing this version.

If you are installing over a previous version of Oracle Application Testing Suite, you may consider making backup copies of the following before installing this version:

- OpenScript scripts in repositories or folders under the installation directory (in *installdir\workspace!* directories or *installdir\directoryname* directories).
- Visual Script files in repositories under the installation directory (in *installdir\workspace!* directories). Scripts created in earlier versions of Oracle Application Testing Suite can be used in 9.10 without modification.
- Oracle Load Testing reports database.
- Oracle Test Manager database.

If you are installing or upgrading over a previous version of OpenScript, you should close any open scripts in the earlier version before upgrading.

## 5 Backwards Compatibility and Upgrading Scripts

The introduction of Script Assets (in Script Properties) requires pre-version 9.10 scripts to be migrated to the current version of 9.10 or higher. This section provides information about backwards compatibility of OpenScript scripts and upgrading OpenScript scripts.

### 5.1 Statement of Backwards Compatibility

Scripts created in older versions of OpenScript will always run in new versions of the product without modification from the command-line, Oracle Load Testing, and Oracle Test Manager.

Older OpenScript scripts may not be opened or played back in the newer version of the OpenScript User Interface without upgrading them first. See [Section 5.2, "Upgrading Scripts to the New Release"](#) below.

Previously published script API functions are supported in the latest release. Some published API may be marked as deprecated, but will still work in the new release in order to maintain backwards compatibility.

### 5.2 Upgrading Scripts to the New Release

OpenScript requires that scripts be upgraded to the latest release in order to open them in the OpenScript User Interface. You are not required to upgrade a script to the new version unless you want to open the script in the OpenScript User Interface. Older

versions of OpenScript scripts can be run without modification from the command-line, Oracle Load Testing, and Oracle Test Manager.

---

**Caution:** Version 9.10 and higher scripts cannot be played back in earlier versions of OpenScript, Oracle Load Testing, and Oracle Test Manager. If you want to maintain pre-version 9.10 scripts, you should make a back up copy of your scripts *before* opening and saving them in version 9.10 or higher. OpenScript automatically migrates any pre-version 9.10 scripts when the script is opened and saved in OpenScript version 9.10 or higher.

---

### 5.2.1 Opening Older Scripts in OpenScript

OpenScript automatically prompts you to upgrade older version scripts to the current version whenever the script is opened in the OpenScript User Interface. When opening an older script, you can choose not to open the script and the script will not be upgraded.

When prompted to upgrade a script, if the script depends on any child scripts or function libraries, OpenScript provides an option to upgrade the child scripts or function libraries to the new version also.

Once a script is upgraded to a new release, the script cannot be opened or run using older versions of Oracle Application Testing Suite (OpenScript, Oracle Load Testing, or Oracle Test Manager).

### 5.2.2 Migrating Older Scripts in OpenScript

If you wish to upgrade scripts without opening them individually in OpenScript, you can use the **Migrate Scripts** upgrade option on the **Tools** menu. The Migrate Scripts tool lets you select which scripts to migrate to the current version and find any child scripts that also need to be migrated.

Version 9.10 does not permit absolute paths for repositories or script assets in scripts that will be used with Oracle Load Testing. However, version 9.0x scripts permitted absolute paths. If your version 9.1x scripts use absolute paths, you can run the version 9.0x scripts, unmodified, in version 9.10 Oracle Load Testing. As soon as you upgrade the 9.0x scripts to 9.10 in OpenScript User Interface or the Migrate Script tool, the script will not playback in Oracle Load Testing until the absolute paths are changed to relative paths. The Migrate Scripts tool does not migrate absolute paths to relative paths or to repository paths. The absolute paths must be changed in the scripts.

## 5.3 Running Mixed Versions of Scripts

You are advised not to run mixed versions of "job" scripts where a parent script calls child scripts or function libraries. This may happen in cases where you may have 9.1x "parent" scripts that run 9.0x "child" scripts or function libraries. Although this configuration has been tested and is supported, the combination of mixed versions scripts may lead to unpredictable results and some confusion as to which scripts are the latest version. In addition, mixed version job scripts may not be able to take advantage of certain new version 9.10 improvements, such as:

- Version 9.10 provides an option to visually inspect and add child script functions into a parent script. If child scripts are not upgraded to 9.10, OpenScript will not display their available functions in the user interface options.

- Version 9.10 scripts no longer require that parent scripts add all child script databanks as their own databanks. If child scripts are not upgraded to 9.10, then parent scripts still must have child script databanks added as their own databanks.

## 5.4 Upgrade Details

When an OpenScript script is upgraded from an older version to version 9.10, the following changes are applied:

- The modules.properties file is updated to reflect the new version numbers of the modules.
- The META-INF/MANIFEST.MF file may be updated to reflect new bundles required by the newer version of the product.
- A new assets.xml file is created containing similar data as in the script.xml file. The original script.xml file is now unused, but remains in the script folder for troubleshooting purposes.
- The versions.txt file is updated with the new version.
- The script.JWG file is updated with the new files.
- The script.java file is not modified.

## 6 Database Migration

A database migration utility is now available for users to enable migration of data from MS Access, SQL Server, and Oracle database into an Oracle database.

Documentation for the database migration utility can be found in  
`<installDir>/bin/DataMigration/help`.

## 7 Tutorial and Documentation for the Oracle Application Testing Suite

The Oracle Application Testing Suite includes product documentation in Adobe Acrobat .PDF format and online help for each application and utility. The documentation includes electronic versions of the Getting Started Guide, OpenScript User's Guide, Oracle Functional Testing User's Guide, Oracle Load Testing User's Guide, and Oracle Test Manager User's Guide. The PDF files are installed to the `<installDir>/OFT` directory.

You need the Adobe® Acrobat® Reader v4 (or newer) to open and view the documents. You can download the reader from the Adobe web site at  
<http://get.adobe.com/reader/>.

All of the components of the Oracle Application Testing Suite include comprehensive online Help. You can access the online help using the Help menu options inside of each product.

## 8 Known Problems, Limitations and Workarounds

This section lists known issues and workaround solutions for the components in the Oracle Application Testing Suite.

## 8.1 Oracle Load Testing

The following are known issues and workaround solutions for Oracle Load Testing.

- If your script fails to playback successfully in Oracle Load Testing due to Windows authentication or other user permissions related issues with your application, this may be due to the login authentication of the Oracle Load Testing Agent Service. The Oracle Load Testing Agent Service which runs all the Virtual Users on a given Agent system will by default be installed to run under the Local System account. This may cause issues for certain applications if they block access to this account. To address this, open the Oracle Load Testing Agent Service in the Services panel and change the login to run under a specific user with appropriate application permissions.
- In order to use the Oracle Load Testing Java Client "User Mode" option, users must first save a Java version of their test script saved in Oracle Functional Testing. There is an option available in Oracle Functional Testing that enables it to compile and save the test script in Java for use by the Java Agent. In the Oracle Functional Testing Advanced Options dialog under JavaAgent, check **Create a Java Agent script after saving a script**. The next time you save your script, Oracle Functional Testing will save the standard script files plus the Java Agent files. This script can then be run in Oracle Load Testing using the Java Client option.
- When using the SNMP data source, new SNMP MIBs that users specify are uploaded to the Oracle Load Testing Server by default and can be accessed only through the Local Oracle Load Testing Data Collector running on that Server. Remote Oracle Load Testing Data Collectors cannot access these MIBs and users need to copy those over manually.
- To use the JMX Data Source, you may need to manually copy the server-specific class files to the `<installdir>\DataCollector\classes` directory on the Data Collector machine. These class files are required for each J2EE Application supported server. The jar files should be same version as the J2EE Application Server. You should get the class files from your Application Server installation. The list of required jar files is contained in the Oracle Load Testing "agents.properties" file located in `<installdir>\DataCollector\properties`.

To do JMX Discovery on WebLogic 10.3.1, doing the following:

1. Copy `<installdir>\lib\wlfullclient.jar` to `<installdir>\DataCollector\classes\weblogic9.1`
2. Open `<installdir>\DataCollector\properties\agents.properties` and modify it so `Agents.JMX.Server.weblogic91.ClassPath = classes/weblogic9.1/wlfullclient.jar` and save the file.
3. Log in to Oracle Load Testing.
4. Select **Metrics** from the **ServerStats** menu then select **New**.
5. Select the JMX Data Source then click **Discover Counters**.
6. Select the local system and click **Edit**.
7. Select Oracle WebLogic 9.1 as the server (port 8088), username oats and the password (the password specified when installing the Oracle Application Testing Suite product).
8. Click **OK**.
9. Click **OK** and the discovery process begins. After a few moments, you will see a full list of the MBeans on the server.

The ServerStats JMX discovery caches counter information. The side effect is that it also caches error messages. For example, failure to copy the correct jar files to data collector's classes directory prior to attempting the discovery would result in an error. When the discovery is attempted after copying the jar files, the discovery still may not work because of the cache. The recommended workaround is to re-start the Oracle Load Testing Server and retry the discovery.

## 8.2 Oracle Test Manager

The following are known issues and workaround solutions for Oracle Test Manager.

- When creating an Oracle Functional Testing type Test Case, the script file must be saved as a package file (.otmpkg) in Oracle Functional Testing in order to attach it to the Oracle Functional Testing Test Case in Oracle Test Manager.
- The icons in the Tree View for Requirements, Tests and Issues are automatically assigned and cannot currently be configured by the Administrator.
- After configuring a new database in the Oracle Test Manager Database Configuration dialog, users must restart the Oracle Application Testing Suite Application Service in order for that database to appear in the Database dropdown list on the Oracle Test Manager login page.
- Troubleshooting OpenScript Script Execution in Oracle Test Manager

The following additional steps are required in order to run the following types of OpenScript scripts from Oracle Test Manager:

- Siebel Functional
- Oracle Forms Functional
- Web Functional scripts that rely on system input events, such as key press or mouse click

It is necessary to run these scripts using an interactive desktop of a named Windows user account that is always logged in.

1. For Siebel and Oracle Forms, the named user's account must have visited the Siebel or Oracle Forms site at least once to ensure that all necessary ActiveX controls and plug-ins are installed in the named user's browser.
2. On the Oracle Test Manager agent machine that will run the scripts, stop the "Oracle Application Testing Suite Agent Service" and configure it to start manually.
3. On the Oracle Test Manager agent machine, login as the named Windows user account that will run the scripts. From a command prompt, run the following all as one command. The .conf file is an argument for the .exe.

```
C:\OracleATS\agentmanager\bin\AgentManagerService.exe -c  
C:\OracleATS\agentmanager\bin\AgentManagerService.conf
```

where C:\OracleATS is the OATS installed folder.

4. The named user account must remain logged into the system at all times that scripts will be run.

### 8.3 Oracle Database 10g Express Edition

The following are known issues and workaround solutions for Oracle Database 10g Express Edition.

- In some cases, the Oracle Database 10g Express Edition service does not start completely. If you receive a 404 Not Found error when starting Oracle Load Testing or Oracle Test Manager, restart the Oracle database service and the Oracle Application testing service, as follows:
  1. Open the Control panel and the Administrative Tools.
  2. Open Services.
  3. Select the Oracle Application Testing Suite Application Service and click **Stop the service**.
  4. Select the OracleServiceXE service and click **Restart the service**.
  5. Wait for the OracleServiceXE service Status to indicate "Started".
  6. Select the Oracle Application Testing Suite Application Service and click **Start the service**.
  7. Wait for the Oracle Application Testing Suite Application Service Status to indicate "Started".
  8. Restart Oracle Load Testing or Oracle Test Manager.

### 8.4 Oracle Functional Testing - OpenScript

The following are known issues and workaround solutions for Oracle OpenScript.

- The Java editor does not pause at breakpoints outside of the current script. Playback will only pause at any breakpoints added to the current script. To pause at a breakpoint in a child script, it is necessary to add a breakpoint before the child script function. Then, use the "Step Into" feature to step into the child script function. Only after stepping into the child script, playback will pause at the child script's breakpoint.
- Automatic proxy configuration (**Preferences - OpenScript - Record - HTTP Module**) for Firefox may not correctly configure the browser proxy. To record through Firefox, you should configure the browser proxy settings manually (Default: host=localhost, port=7777)
- Certain systems may have trouble launching the OpenScript Help (**Help - Help Contents**). This can be resolved by configuring the help to open in an external browser window:
  1. From the Tester Perspective, select **View - OpenScript Preferences**.
  2. Click the "Clear" icon next to the filter text field in the upper left. Click the Help tree node and check the "Use external browser" box in the Help preferences.
  3. Click **OK** and relaunch OpenScript Help.
- Scripts created in OpenScript will not work in user defined profiles in Oracle Load Testing. Instead, users can build jobs using the Initialize, Run, and Finish sections of a script created in OpenScript.

#### **8.4.1 Web Functional Test Recording with Firefox Troubleshooting**

The following section describes Web Functional Test Recording with Firefox Troubleshooting techniques.

- When using the Firefox browser, you should disable or not install any add-ons into the OpenScript-created Firefox profiles.
- The first time recording it is possible that Firefox has not completely initialized the OpenScript plug-ins. Try stopping recording and trying again. If Firefox still fails to record:
  1. Run <install dir>/OpenScript/UninstallBrowserHelpers.bat
  2. Run <install dir>/OpenScript/InstallBrowserHelpers.bat

### **8.5 Oracle Functional Testing**

The following are known issues and workaround solutions for Oracle Functional Testing.

- Scripts that playback successfully in Oracle Functional Testing may still fail when run in Oracle Load Testing Thin Client or Java Client modes. This is because Oracle Functional Testing is playing back scripts at the GUI level by automating browser DOM events and Thin/Java Client are playing back at the HTTP protocol level which may have different playback issues. Use the Oracle Functional Testing Navigation Editor to verify and troubleshoot script playback in Oracle Load Testing Thin and Java Agents.
- When adding a Text Matching Test to your script it is possible that it may be added to the wrong page or frame depending on what you have selected in your tree, resulting in a failure. Be sure to verify that your Text Matching Test was added to the correct page and frame if you are seeing unexpected failures.
- Double clicking on the HTML node of a frame with sub-frames will not display the HTML of the recorded page. Only pages with no sub-frames can be displayed in this manner.
- The Navigation Editor does not allow you to iterate over multiple records using databanks when playing back a script. Select the "Playback with Databank" option and specify which record to use to playback with that record.
- Some proxy servers will always return a HTTP return code 200 ('OK') whether or not the requested item is available. In these cases Resource Validation will incorrectly report that a missing resource is available.
- In Options (Current Script or New Script) - Advanced - Siebel Support: The Reset Siebel State button is used in conjunction with the Regenerate Thin Client Script button. The Reset Siebel State button removes all of the previously found sources in the Thin Client script so that the Regenerate will operate on a script free of the previously found sources. The only effect of the Reset Siebel State button is to change the Thin Client script generated by the Regenerate Thin Client Script button.
- On systems without Microsoft Office installed, the Navigation Editor in Oracle Functional Testing may not function correctly. To remedy, install the Microsoft ActiveX Control Pad from  
<http://msdn.microsoft.com/library/default.asp?url=/library/en-us/dnaxctrl/html/cpad.asp> which will install a needed .dll file.

For Windows Vista systems, complete these steps before running the ActiveX Control Pad setup procedure:

1. Right click on setupad.exe and select **Properties**.
2. Click the **Compatibility** tab.
3. Select **Run this program in compatibility mode for:** option and then select **Windows XP (Service Pack 2)**.
4. Click **Apply** and then **OK**.
5. Run the setupad.exe program to install the ActiveX Control Pad.

## 9 Documentation Accessibility

Our goal is to make Oracle products, services, and supporting documentation accessible to all users, including users that are disabled. To that end, our documentation includes features that make information available to users of assistive technology. This documentation is available in HTML format, and contains markup to facilitate access by the disabled community. Accessibility standards will continue to evolve over time, and Oracle is actively engaged with other market-leading technology vendors to address technical obstacles so that our documentation can be accessible to all of our customers. For more information, visit the Oracle Accessibility Program Web site at <http://www.oracle.com/accessibility/>.

### Accessibility of Code Examples in Documentation

Screen readers may not always correctly read the code examples in this document. The conventions for writing code require that closing braces should appear on an otherwise empty line; however, some screen readers may not always read a line of text that consists solely of a bracket or brace.

### Accessibility of Links to External Web Sites in Documentation

This documentation may contain links to Web sites of other companies or organizations that Oracle does not own or control. Oracle neither evaluates nor makes any representations regarding the accessibility of these Web sites.

### Access to Oracle Support

Oracle customers have access to electronic support through My Oracle Support. For information, visit <http://www.oracle.com/support/contact.html> or visit <http://www.oracle.com/accessibility/support.html> if you are hearing impaired.

---

Oracle Application Testing Suite Release Notes Version 9.10 for Microsoft Windows (32-Bit)  
E17387-01

Copyright © 1997, 2010, Oracle and/or its affiliates. All rights reserved.

This software and related documentation are provided under a license agreement containing restrictions on use and disclosure and are protected by intellectual property laws. Except as expressly permitted in your license agreement or allowed by law, you may not use, copy, reproduce, translate, broadcast, modify, license, transmit, distribute, exhibit, perform, publish, or display any part, in any form, or by any means. Reverse engineering, disassembly, or decompilation of this software, unless required by law for interoperability, is prohibited.

The information contained herein is subject to change without notice and is not warranted to be error-free. If you find any errors, please report them to us in writing.

If this software or related documentation is delivered to the U.S. Government or anyone licensing it on behalf of the U.S. Government, the following notice is applicable:

U.S. GOVERNMENT RIGHTS Programs, software, databases, and related documentation and technical data delivered to U.S. Government customers are "commercial computer software" or "commercial technical data" pursuant to the applicable Federal Acquisition Regulation and agency-specific supplemental regulations. As such, the use, duplication, disclosure, modification, and adaptation shall be subject to the restrictions and license terms set forth in the applicable Government contract, and, to the extent applicable by the terms of the Government contract, the additional rights set forth in FAR 52.227-19, Commercial Computer Software License (December 2007). Oracle USA, Inc., 500 Oracle Parkway, Redwood City, CA 94065.

This software is developed for general use in a variety of information management applications. It is not developed or intended for use in any inherently dangerous applications, including applications which may create a risk of personal injury. If you use this software in dangerous applications, then you shall be responsible to take all appropriate fail-safe, backup, redundancy, and other measures to ensure the safe use of this software. Oracle Corporation and its affiliates disclaim any liability for any damages caused by use of this software in dangerous applications.

Oracle is a registered trademark of Oracle Corporation and/or its affiliates. Other names may be trademarks of their respective owners.

This software and documentation may provide access to or information on content, products, and services from third parties. Oracle Corporation and its affiliates are not responsible for and expressly disclaim all warranties of any kind with respect to third-party content, products, and services. Oracle Corporation and its affiliates will not be responsible for any loss, costs, or damages incurred due to your access to or use of third-party content, products, or services.