Oracle® Application Testing Suite

Release Notes

Version 12.1.0.1

E17387-07

April 2012

This document includes updated information for Oracle Application Testing Suite version 12.1.0.1. 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, "Tutorial and Documentation for the Oracle Application Testing Suite"
- Section 7, "Known Problems, Limitations and Workarounds"
- Section 8, "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 12.1.0.1 - New Features and Updates

- **OpenScript Dynamic Loading of Databanks** - The `getDatabank` API provides methods for dynamically loading banks and changing databank settings programmatically within functional test scripts during playback of a script.
- **OpenScript Enumerated List Function Arguments** - OpenScript supports creating custom functions that accept a value from a known list of values. Technical users can create function libraries that include custom functions with arguments that accept values from a user-specified list of possible values.
- **OpenScript List and Map Function Arguments** - OpenScript supports creating custom functions that accept List and Map type function arguments. Technical users can create functions that solicit and return lists of values and key/value pair maps.

- **OpenScript Function Libraries with Code Assist** - OpenScript provides Code Assistance in the OpenScript Java code view for calling methods in dedicated function libraries. The function library must be a newly created 12.1 Function Library or the function library must have been converted to a Function Library using the new "Convert to Function Library" feature.
- **OpenScript ADF Correlation Library Enhancements** - The OpenScript ADF Correlation Library is updated to provide a more reliable way of correlating ADF application component ids and attributes for variables from ADF Rich Responses. The new hierarchical detail view and comparison view provide for enhanced ADF load script debugging.
- **OpenScript JDE Module** - OpenScript includes a a new JDE Load Testing module for creating load testing scripts for testing JDE Enterprise One applications. The JDE load testing scripts include the new JDE Correlation Library enabled by default. Functional testers can use standard Web FT scripts for record/playback of functional test scripts for JDE Enterprise One applications.
- **OpenScript Load Script Authentication Improvements** - OpenScript automatically generates an authentication step for the scripts require an authentication during the recording process for load test scripts that require an authentication scheme such as Basic or NTLM.
- **OpenScript Database SQL Query Result Validation** - OpenScript includes a function to check the data queried from a database connection against the expected data previously set by the user. The user can set the expected value for each cell, check the data, and get the checked result report for a playback using either GUI or API.
- **OpenScript Large Data Handling Improvements** - OpenScript moves the large data in the script to a resource file which can be loaded for playback. OpenScript preferences can set the thresholds at which large amounts of data will be stored as part of a script or stored in a resource file outside of the script. These settings are used in cases where an application-under-test either requires or returns large amounts of data in the request-response transaction such as HTTP post data containing a long list of Name-Value Pairs, Web Table Tests on tables containing thousands of table cells, Database SQL Query Result Validation containing hundreds of cells, and Flex HTTP scripts that use very long action messages.
- **OpenScript Shortcut Key Preferences** - OpenScript includes new preferences for specifying shortcut keys for common script activities such as record, playback, pause, execute, and stop.
- **OpenScript Proxy Command Line Interface** - OpenScript includes a command line interface for the OpenScript proxy to enable the inspection of HTTP traffic originating from clients such as web browsers and Java applets.
- **OpenScript IE 9 Enhancements** - OpenScript includes enhancements to support record and playback of Web Functional scripts with Internet Explorer 9.
- **OpenScript Firefox 4 Enhancements** - OpenScript includes enhancements to support record and playback of Web Functional scripts with Mozilla Firefox 4.0.
- **Oracle Test Manager Test Execution Enhancements** - A new Test Execution tab in Oracle Test Manager lets you define collections of test cases combined in Test Sets. Testers can select or deselect specific tests within a Test Set to execute prior to starting execution of the test.
- **Oracle Test Manager Export Project** - A new **Export Project** option has been added to the **Project** menu for exporting data from the database.

- **Oracle Test Manager Grid View** - The Grid view is no longer a feature of the Oracle Test Manager product.
- **Oracle Test Manager Removal of Schedule Conflict Rules** - In version 12.1.0.1, the thread management of task scheduling is now handled by the Weblogic application server and scheduled tasks run in parallel within the confines of application server resources. Hence there is no longer the concept of a "current task" to be aborted, so the Conflict Rules options have been removed from the New Task dialog box. Other functionality implied by Conflict Rules can be found in other features in the product: ordered execution of tests can be performed by ordering tests within a Test Set and programming a delay between tasks can be performed by scheduling tasks at the desired time interval.
- **Oracle Load Testing Virtual User Logging Enhancements** - Oracle Load Testing Virtual User Display is replaced with a new Virtual User Logging feature set which can be used to easily discover the life-cycle of each VU, filtered logs and better diagnosis of the root cause of errors.
- **Oracle Load Testing Self-Contained Scenario Files** - Oracle Load Testing includes a new file type that includes all scripts in a scenario and optionally recorded data, results, and databank files in a single .zip file. The self-contained zip file are exported using the **Export** option on the **Tools** menu. The self-contained .zip file can be opened directly in Oracle Load Testing using the **Open** option on the **Scenario** menu.
- **Oracle Application Testing Suite Support on Linux Platform** - Oracle Application Testing Suite supports installing the server components (Oracle Load Testing, Oracle Test Manager, and the Administrator) on Linux platforms. OpenScript and functional test scripts are not supported on Linux platform in 12.1.0.1.
- **Data Migration Utility** - In versions prior to 12.1.0.1, the Data Migration Utility provided a graphical user interface to migrate data from previous versions of the testing tool that did not use Oracle databases. The Data Migration Utility is no longer included with the Oracle Application Testing Suite product. The alternate method for migrating data from older Oracle databases is to use the database Export and Import utilities. The procedures are described in the Oracle Load Testing and Oracle Test Manager online help, as follows:

  Oracle Load Testing

  `http://localhost:8088/olt/pages/help/how2_exp_imp_olt_databases.htm`

  Oracle Test Manager

  `http://localhost:8088/otm/Help_en/exp_imp_databases.htm`

3 System Requirements

Oracle Application Testing Suite has the following system requirements:

3.1 Oracle Functional Testing/OpenScript

Oracle Functional Testing's OpenScript scripting platform has the following system requirements:

- Operating System (32-bit and 64-bit versions): Windows XP, Windows Vista, Windows 2003, Windows 7, Windows 2008

- Memory: Minimum 1 GB
- System: x86, 32-bit or 64-bit processor, 2.6 GHz or faster
- Disk Space: 4 GB minimum
- Browser: Internet Explorer 7.x, 8.x., 9.x and Firefox 3.5/3.6, 6.x

3.2 Oracle Load Testing

Oracle Load Testing has the following system requirements:

- Operating System (32-bit and 64-bit versions): Windows XP, Windows Vista, Windows 2003, Windows 7, Windows 2008. Oracle Enterprise Linux 5.1 64 bit, 5.2 64 bit, 4u8 64 bit
- Memory: Minimum 2 GB.
- System: x86, 32-bit or 64-bit processor, 2.6 GHz or faster
- Disk Space: 10 GB minimum free (at least 3 GB free on the system drive)
- Browser: Internet Explorer 7.x, 8.x, 9.x, Firefox 3.5/3.6, 6.x
- Database: Oracle EE 10*g* or 11*g*, XE 10*g*
- Application Server: Oracle WebLogic 11*g* (10.3.5.0). Oracle Application Testing Suite requires the standard WebLogic Server installation (which includes jrocket jdk.). It does not support generic versions of WebLogic Server installation (such as those that use `wls1035_generic.jar`).
- Oracle Load Testing Agent also supported on Linux 32-bit & 64-bit, in addition to Windows 32-bit and 64-bit operating systems listed above

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 and 64-bit versions): Windows XP, Windows Vista, Windows 2003, Windows 7, Windows 2008. Oracle Enterprise Linux 5.1 64 bit, 5.2 64 bit, 4u8 64 bit
- Memory: Minimum 2 GB.

- System: x86, 32-bit or 64-bit processor, 2.6 GHz or faster
- Disk Space: 10 GB minimum free (at least 3 GB free on the system drive)
- Browser: Internet Explorer 7.x, 8.x, 9.x, Firefox 3.5/3.6, 6.x
- Database: Oracle EE 10*g* or 11*g*, XE 10*g*
- Application Server: Oracle WebLogic 11*g* (10.3.5.0). Oracle Application Testing Suite requires the standard WebLogic Server installation (which includes jrocket jdk.). It does not support generic versions of WebLogic Server installation (such as those that use `wls1035_generic.jar`).

> **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 10*g* 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 11*g* Enterprise Edition database is highly recommended. Oracle 11*g* Standard Edition and Oracle 10*g* 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 is a separate application that have its own memory and CPU requirements.
- The ATS products will run in 32-bit compatibility mode when installed on 64-bit operating systems. It is possible for Oracle Application Testing Suite to be installed on an existing WebLogic server. However, if the pre-installed WebLogic server is JRockit plus WebLogic (64-bit) environment, some functions in Oracle Application Testing Suite will not work properly. Oracle Application Testing Suite can be installed on a 32-bit WebLogic, but not a 64-bit.
- The legacy Oracle Functional Testing, VB-based scripting platform and associated components (such as Job Scheduler) are no longer included with the ATS installer starting with version 9.2. Customers that still require this product should use ATS 9.1 or older versions supported under Oracle's Lifetime Support Policy.

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 on Windows Machines

You can install Oracle Application Testing Suite from the product download zip file.

From the Oracle Web site:
`http://www.oracle.com/technetwork/oem/app-test/index.html`

1. Select and download the Oracle Application Testing Suite zip file(s) from the web site and save it to a temporary directory on your hard disk.
   - oats-full-*version*.zip: Full installation plus all prerequisites (Oracle Application Testing Suite, Product documentation PDF files, WebLogic Server 11gR1 (10.3.3) developer edition, Oracle Database Express Edition 10.3.2.4, Oracle Instant Client 11.1.0.7.0, .NET Framework 2.0, Eclipse 3.4.1, Sun JRE 1.6.0_07, Oracle JRockit 4.0.0.1.6.0.R28)
   - oats-micro-*version*.zip: Oracle Application Testing Suite and product documentation PDF files without prerequisites.
   - oats-upgrade-*version*.zip: Oracle Application Testing Suite and Weblogic server. This file can be used to upgrade from Oracle Application Testing Suite version 9.1x/9.2x/9.3x.
   - oats-docs-*version*.zip: Oracle Application Testing Suite documentation PDF files.
2. Unzip the downloaded file and run the `setup.bat` file for Oracle Application Testing Suite.

   Note: Important: Windows Vista, 2008 or Windows 7 users, select **Run as Administrator** from the right-click shortcut menu. This is required process for the Operating Systems that have User Account Control (UAC) enabled.

3. Follow the setup instructions to install the Oracle Application Testing Suite.

   During the installation you will be asked to provide a master password. *Remember this password*. The master password specified during installation is used to set all of the following user passwords:
   - Oracle Test Manager users "default" and "administrator".
   - Oracle Load Testing users "default" and "administrator".
   - User "JMSAdmin" used for Oracle Load Testing agent to controller authentication.
   - User "oats" for Weblogic Server - this is a Weblogic superuser.
   - User "oats-agent" in Weblogic Server for JMS communication between agent and controller - this is *not* a Weblogic superuser.
   - Database users "oats", "olt", "otm".

- Database users (XE's defaults) "system" or "sys".

Note: When installing Oracle Application Testing Suite server components on a machine that:

1. has no previous version of Oracle Application Testing Suite installed, AND
2. has Oracle Enterprise Edition database installed,

make sure to choose the "Custom Installation" option and specify the connection credentials for your Oracle Enterprise Edition database. If you choose "Complete Installation", Oracle Application Testing Suite will install Oracle XE database, which may not run properly alongside Oracle Enterprise Edition database on the same machine.

4.1.1 Manually Configuring the Installation

The above installation procedure for Oracle Application Testing Suite setup.bat in oats-full-*version*.zip 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.

If there is no database installed and you will not be using a remote database, you will need to install Oracle XE or EE on the local machine. Install an Oracle Database or acquire the hostname, port, instance, and credentials for an existing one.

If WebLogic Server is not installed on the local machine, you must download Weblogic 11gR1P2 (10.3.5) from download.oracle.com and install it. Install Weblogic 11gR1P2 (10.3.5) on your local machine or acquire the location on your local machine of a Weblogic instance of the same version.

To manually configure Weblogic and create the database schema:

1. Create the following environment variables:

   WL_HOME=The location of your Weblogic home (typically C:\Oracle\Middleware\wls if it was installed with default options).

   MW_HOME=The location of your Weblogic root folder (typically C:\Oracle\Middleware if it was installed with default options).

   WLS_PW=Your master password. This will be used to set your admin password for Weblogic, as well as to create your Default and Administrator accounts in Oracle Load Testing and Oracle Test Manager.

   DB_HOST=The location of your database.

   DB_NAME=The name of the database that Oracle Application Testing Suite will use. For XE installations, the name is typically XE. For EE installation this will depend on your installation settings.

   DB_PORT=The port of your database (typically 1521).

   DB_USER=The admin user of your database (typically "system"),

   DB_PASS=The password of the database user.

   DB_URL=`jdbc:oracle:thin:@//`*`<hostname>`*`:`*`<port>`*`/`*`<servicename>`*

2. Run `<installdir>\bin\install.bat`. This will delete your previous OATS domain, if any, and recreate it. (If the install is fresh you do not need to run this .bat file.)
3. Change to directory `%OATS_HOME%\bin` and issue the following commands (in order):

```
initConfig
dbsetup
config_ds
unpack
storeUser
installSvc
restartSvc "OracleATSServer"
deploy %OATS_HOME%\install\oats.ear
config_sec
```

4. Validate your installation by going to http://localhost:8088/olt and verify that you can successfully login (i.e. with credentials Default/master password) and use the product.

4.1.2 Uninstalling the Applications from Windows Machines

To uninstall the Oracle Application Testing Suite applications:

1. Select **Uninstall Oracle Application Testing Suite** from the Oracle Application Testing Suite Start menu.
2. Select OATS in the tree.
3. Click **Remove**.

4.2 Installing the Applications on Linux Machines

Oracle Application Testing Suite server components (Oracle Load Test/Oracle Test Manager) and Agent components (Oracle Load Test agent/Data Collector) can be installed on Linux via a separate installer; however Oracle OpenScript is Windows only. Oracle Application Testing Suite components have been tested with Oracle Enterprise Linux 5.1 64 bit, 5.2 64 bit, 4u8 64 bit. Note that Windows metrics cannot be used on a Linux Data Collector (Perfmon, Ping, COM+, and the SSH version of Virtual Agent).

The Oracle Application Testing Suite Linux installer is based on the Oracle Universal Installer (OUI) which requires UI based access. To install the Oracle Application Testing Suite server components on Linux you will need to access XTerm. XTerm, most typically, can be accessed over VNC.

4.2.1 Installing Oracle XE Database

The Oracle Application Testing Suite installer requires a working instance of Oracle XE database. If you have not installed and configured an Oracle XE database, install and configure an Oracle XE database before installing the Oracle Application Testing Suite server components.

To install the Oracle XE database:

1. Open a terminal window and install XE database, by en follows:

```
sudo rpm -ivh oracle-xe-univ-10.2.0.1-1.0.i386.rpm
```

A progress bar displays during the installation which is followed by the final message: You must run '/etc/init.d/oracle-xe configure' as the root user to configure the database.

2. Configure the database, as follows:

   ```
   sudo /etc/init.d/oracle-xe configure
   ```

3. When prompted, enter a password for database accounts. Enter: `oracle.`
4. For the remainder of the configuration, click **Enter** to accept the default values for most common use-cases.

4.2.2 Installing Oracle Application Testing Suite Components

From the Oracle Web site:
http://www.oracle.com/technetwork/oem/app-test/index.html

1. Select and download the Oracle Application Testing Suite zip file(s) from the web site and save it to a temporary directory on your hard disk.
   - oats-docs-*version*.zip: Oracle Application Testing Suite documentation PDF files.
   - oats-linux64-full-*version*.zip: Full Linux installation plus all prerequisites (XE database and WebLogic Server) for 64-Bit machines.
   - oats-linux64-micro-*version*.zip: Linux installation without prerequisites for 64-Bit machines.
   - oats-linux-full-*version*.zip: Full Linux installation plus all prerequisites (XE database and Weblogic Server) for 32-Bit machines.
   - oats-linux-micro-*version*.zip: Linux installation without prerequisites for 32-Bit machines.
2. Unzip the downloaded file for the Oracle Application Testing Suite.
3. Enter the installation folder where you unzipped the download file.
4. Set appropriate permissions for the installation shell script to execute, as follows:

   ```
   chmod u+rwx,g+rx,o+x setup.sh
   ```

5. Edit the installation Shell script `setup.sh` and change ORACLE_HOME to any designated location under your scratch folder (for example: `/scratch/<myFolder>/oats12`, as follows:

   ```
   #!/bin/bash
   INSTALL_DIR="'dirname $0'" cd "$INSTALL_DIR" INSTALL_DIR="'pwd'"
     cd - INSTALL_DIR=$( echo "$INSTALL_DIR" | sed -e 's/ /\\ /g' )

   ##Adding Chmod to add execute permissions for files.

   chmod -R 777 $INSTALL_DIR/ 1> /dev/null 2>&1 $INSTALL_DIR/oui/bin/runInstaller
     -force -noconsole ORACLE_HOME="/scratch/myFolder/oats12"
     ORACLE_HOME_NAME="OATS" SHOW_EXIT_CONFIRMATION=false
     SHOW_SUMMARY_PAGE=true INSTALL_ROOT=$INSTALL_DIR

   ##COMPONENT_LANGUAGES={"$LOCALE"} SELECTED_LANGUAGES={"$LOCALE"}
   ```

6. Run the `setup.sh` file to start the GUI installation for the Oracle Application Testing Suite.

```
sh setup.sh
```

You cannot install Oracle Application Testing Suite as root, so do not use `sudo sh setup.sh`.

7. Follow the setup instructions to install the Oracle Application Testing Suite.

   During the installation you will be asked to provide a master password. *Remember this password*. The master password specified during installation is used to set all of the following user passwords:

   - Oracle Test Manager users "default" and "administrator".
   - Oracle Load Testing users "default" and "administrator".
   - User "JMSAdmin" used for Oracle Load Testing agent to controller authentication.
   - User "oats" for Weblogic Server - this is a Weblogic superuser.
   - User "oats-agent" in Weblogic Server for JMS communication between agent and controller - this is *not* a Weblogic superuser.
   - Database users "oats", "olt", "otm".
   - Database users (XE's defaults) "system" or "sys".

8. Specify Inventory Directory and credentials, as follows:

   ```
   Enter full path of inventory directory: /scratch/<myFolder>/orainventory

   Specify Operating System group name: dba
   ```

9. Click **Next**.
10. Select Installation Type: Complete and click **Next**.
11. Specify Home Details, as follows:

    ```
    Name: Home1

    Path: /scratch/<myFolder>/oats12
    ```

12. Click **Next**.
13. Enter Install Location, as follows:

    ```
    Folder: /scratch/<myFolder>/oracleats
    ```

14. Click **Next**.
15. Set the Master Password by entering and confirming the password, for example:

    ```
    Enter password: oracle123

    Confirm password: oracle123
    ```

16. Click **Next**.
17. Specify Oracle Database Installation, as follows:

    ```
    Configure an existing Oracle XE or EE Database
    ```

18. Click **Next**.
19. Specify Oracle Database Information, as follows:

```
DB Hostname: localhost
Port: 1521
Service Name: XE
Db User: system
DB Password: oracle
```

20. Click **Next**.
21. Review the Summary details and click Install. The installation will continue automated install until you see the following message:

    ```
    Execute Configuration Scripts
    The following configuration scripts need to be executed as the "root" user

    Scripts to be executed:
    1. /scratch/<myFolder>/oraInventory/orainstRoot.sh
    2. /scratch/<myFolder>/oats12/root.sh
    ```

22. Open a terminal window and execute the configuration scripts, as follows:.

    Script 1: `orainstRoot.sh`

    ```
    sudo sh /scratch/<myFolder>/oraInventory/orainstRoot.sh
    ```

    Script 2: `root.sh`

    ```
    sudo sh /scratch/<myFolder>/oats12/root.sh
    ```

23. Enter the Oracle Application Testing Suite Agent Manager password: (the same as the Master installation password entered and confirmed in the previous step above).

    The script execution will complete with following confirmation messages: "OATS Agent Manager (OracleATSAgent) service is now running!" "Oracle Application Testing Suite Linux Agent is now installed!"

24. Exit the terminal window and click **OK** in the Execute Configuration Script pane of the installation GUI.
25. Click **Exit** on the End of Installation pane to complete the installation.
26. Navigate to `http://<linux-host-name>:8088/` to start using the OATS application.

4.2.3 Starting the Applications on Linux Machines

This section lists how to start Oracle Application Testing Suite applications and utilities on Linux machines. It also lists how to restart and stop the application service.

To start the Oracle Application Testing Suite Administrator:

```
http://<machine>:8088/admin or http://localhost:8088/admin
```

To start the Oracle Test Manager application:

```
http://<machine>:8088/otm or http://localhost:8088/otm
```

To start the Oracle Load Testing application:

```
http://<machine>:8088/olt or http://localhost:8088/olt
```

To start the Database Configuation utility:

```
<installdir>/bin/DbConfig.sh
```

To start the Oracle Load Testing Agent Authentication Manager utility:

```
<installdir>/jdk/jre/bin/java -jar <installdir>/agentmanager/AMAuthManager.jar
```

To restart the Oracle Application Testing Suite service:

```
<installdir>/bin/restartSvc.sh [OracleATSServer|OracleATSAgent]
```

To stop the Oracle Application Testing Suite service:

```
<installdir>/bin/stopSvc.sh [OracleATSServer|OracleATSAgent]
```

To create a support package for troubleshooting purposes (OATSSupport.zip):

```
<installdir>/bin/oats_support.sh
```

4.2.4 Uninstalling the Applications from Linux Machines

To uninstall the Oracle Application Testing Suite server components or remote agent:

1. Run the `setup.sh` file to start the GUI installation for the Oracle Application Testing Suite.

   ```
   sh setup.sh
   ```

   You cannot uninstall Oracle Application Testing Suite as root, so do not use `sudo sh setup.sh`.

2. Follow the screen instructions for uninstalling the server components or remote agent.

4.3 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.

4.3.1 Installing the Remote Agent

To install the Remote agent:

1. Download the appropriate .zip file from the download Web site:
   - For Windows agents, use oats-full-*version*.zip and run `setup.bat`.
   - For Linux agents, use oats-linux-full-*version*.zip for 32-bit or oats-linux64-full-*version*.zip for 64-bit and run `setup.sh`. See Section 4.2, "Installing the Applications on Linux Machines" for additional information.
2. Unzip the downloaded file and run the installation file.
3. Follow the setup instructions to install just the Remote Agent software.
4. Verify network access from the Controller workstation to the Agent workstations and configure the Agent Workstations as explained in the following section.

4.3.2 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.

4.3.3 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.

4.4 Preparing for a "Clean" Installation

In some cases, you may want to prepare a system for "Clean" installation of the Oracle Application Testing Suite product. This may be required in the following cases:

- There is a previous beta version of the Oracle Application Testing Suite product installed on the system.

- It is necessary to downgrade the product from a newer version.

- An installation failure occurred for some reason and it is necessary to clean the environment to a fresh start.

The following sections provide step-by-step instructions to perform a clean installation of the Oracle Application Testing Suite product on your system.

Make sure to backup all scripts and databases before proceeding. See Section 4.5, "Installing over beta versions or earlier releases of Oracle Application Testing Suite" for additional information.

4.4.1 Step 1: Uninstall Oracle Application Testing Suite Products

To uninstall Oracle Application Testing Suite product:

1. Run the Oracle Application Testing Suite `Setup.bat` file to launch "Oracle Universal Installer".
2. Click **Deinstall Products**. The Inventory window opens.
3. Expand the Oracle Homes tree and select the "OATS" checkbox.
4. Click **Remove**. The Remove window opens and starts the deinstallation procedure.
5. Wait for the Remove window to automatically dismiss itself. After the successful deinstalltion, the Inventory window may show "There are no installed products" if there no other Oracle products installed.
6. Close the Inventory window.
7. Click **Cancel** to close the Oracle Universal Installer.

4.4.2 Step 2: Uninstall Oracle XE Database

To uninstall Oracle XE Database:

1. Select Control Panel from the **Start** menu.
2. Open **Add/Remove Programs** or **Program and Features** depending upon your Windows version.
3. Find and select **Oracle Database 10*g* Express Edition**.
4. Click the Remove or the Uninstall/Change button from the tool bar, or double-click the item to launch the InstallSheild wizard.
5. Click **Yes** in the dialog box to remove the database.
6. Wait for the uninstallation to complete successfully.
7. Click **Finish** to close the InstallShield wizard.

4.4.3 Step 3: Remove or Rename Installation Folders

To remove or rename installation folders:

1. Backup OpenScript scripts and script asset files in repositories or folders under the installation directory (in *installdir\workspace*! directories or *installdir\directoryname* directories).
2. Backup OpenScript scripts and script asset files in repositories under the installation directory (in *installdir\workspace* directories).
3. Remove or rename the Oracle Application Testing Suite installation directory (the default is C:\OracleATS).

4. Remove or rename the directory: C:\Documents and Settings*username*\osworkspace (depending upon your Operating System, it may be similar to: C:\Users\username\osworkspace).
5. Remove or rename the directory: C:\Program Files\Oracle\Inventory (depending on your Operating System, it may be like: C:\Program Files (x86)\Oracle\Inventory).
6. Remove or rename the file: C:\Windows\rsw.ini

4.4.4 Step 4: Remove the Services (Optional)

To remove Oracle Application Testing Suite services:

1. Select **Run** from the **Start** menu.
2. Type `cmd` and click **OK**.
3. Type the command: `sc delete` *`servicename`*, where *`servicename`* is each of the following:
   - `OracleATSAgent`
   - `OracleATSServer`
   - `OracleATSHelper`

   If the message "The specified services does not exist as an installed services" is returned from the program, the service was successfully removed by the previous steps.

4.4.5 Step 5: Remove the Registry Keys (Optional)

To remove Oracle Application Testing Suite Registry keys:

1. Select **Run** from the **Start** menu.
2. Type `regedit` and click **OK**.
3. Delete the following Registry keys using the Registry Editor:
   - `HKEY_LOCAL_MACHINE\SYSTEM\CurrentControlSet\Services\OracleATSServer`
   - `HKEY_LOCAL_MACHINE\SYSTEM\CurrentControlSet\Services\OracleATSAgent`
   - `HKEY_LOCAL_MACHINE\SYSTEM\CurrentControlSet\Services\OracleATSHelper`

   If you do not find the keys in the Registry, they were successfully removed by the previous steps. Note: The keys may be found in the different location depends on your Operating System (search by the key to find the correct location).

4.4.6 Step 6: Restart the System (Optional)

While not required, it is a best practice to restart the system after uninstalling applications.

- Select **Shutdown** from the **Start** menu, then select **Restart**.

4.4.7 Step 7: Install Oracle Application Testing Suite

To install Oracle Application Testing Suite:

1. Make sure the installing user has administrator privilege on the system.

2. Unzip the oats-full-*version*.zip product download file in a temporary directory.
3. Using Windows Explorer, double-click the `setup.bat` to start the Oracle Universal Installer. See Section 4.1, "Installing the Applications on Windows Machines" for additional information.

> **Note:** Important: Windows Vista, 2008 or Windows 7 users, select **Run as Administrator** from the right-click shortcut menu. This is required process for the Operating Systems that have User Account Control (UAC) enabled.

4. Follow the setup instructions to install the Oracle Application Testing Suite.

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

If you have a 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 and script asset files in repositories or folders under the installation directory (in *installdir\workspace*! directories or *installdir\directoryname* directories).
- Script and script asset files in repositories under the installation directory (in *installdir\workspace* directories).
- 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.

4.6 Installing OLT/OTM after OpenScript Only Install

The Oracle Application Testing Suite 12.1.0.1 component based installer will install Oracle Load Testing and Oracle Test Manager over installed OpenScript only installations (of either version or pre 12.1.0.1). The installer will upgrade the OpenScript installation and will install other Oracle Application Testing Suite components.

5 Backwards Compatibility and Upgrading Scripts

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. The introduction of Script Assets (in Script Properties) requires pre-version 9.1 scripts to be migrated to the current version if they are to be edited in the OpenScript User Interface.

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 9.20 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 9.20 scripts, you should make a back up copy of your scripts *before* opening and saving them in version 9.20 or higher. OpenScript automatically migrates any pre-version 9.20 scripts when the script is opened and saved in OpenScript version 9.20 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 migrate pre-version 9.10 scripts to the current version without having to open scripts individually.

The Migrate Scripts tool provides options for migrating top-level scripts and locating all dependent child scripts. 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.

Since version 9.10, scripts that will be run in Oracle Load Testing may not specify absolute paths for their repositories or script assets. However, if your pre-9.10 scripts use absolute paths, you may continue to run the same scripts, unmodified, in the current version of Oracle Load Testing, without issue. As soon as you upgrade the pre-9.10 scripts to the current version using either the 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 manually.

5.2.3 Copying Older Scripts to New Repositories

For scripts created in releases before 9.1, you should maintain the same rigid, legacy Repository/Workspace folder structure as repositoryLocation/workspace!/script. Changing the repository folder structure within which pre-9.1 scripts exist, such as by adding multiple sub-folders within the repository, or by creating sub-folders without an exclamation mark "!" at the end, may prevent certain pre-9.1 scripts from playing back successfully.

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 9.2x "parent" scripts that run pre-9.1 "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 some newer features, such as:

- The ability to visually inspect and add child script functions into a parent script. If pre-9.1 child scripts are not upgraded to the current version, OpenScript will not display their available functions in the user interface options.
- Scripts upgraded to version 9.1 or later no longer require that parent scripts add all child script databanks as their own databanks. If pre-9.1 child scripts are not upgraded to the current version, 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 the current version, 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.
- If the script being upgraded were created in a pre-9.1 release, then 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 replaced by the script.properties file.
- The script .JWG file is updated with the new files.
- The script.java file is not modified.

6 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 Load Testing User's Guide, Oracle Test Manager User's Guide and Release Notes. The PDF files are installed to the <*installDir*>/docs 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.

7 Known Problems, Limitations and Workarounds

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

> **Note:** On Windows 7/Vista (UAC systems), it is necessary to run commands from an elevated command line in order to restart services. If you try to run it from the **Start** menu, you will see the following errors in the cmd window:
>
> ```
> System error 5 has occurred.
> Access is denied.
> ```
>
> Run `services.msc` from the command line and restart the services from there. Windows 7/Vista (UAC systems) require admintrator privileges to run commands from an elevated command line in order to stop and restart services.

7.1 Oracle Application Testing Suite

The following are know issues and workaround solutions for Oracle Application Testing Suite.

- **Problem**: Non-administrator user experiences problems using Internet Explorer 9 in Oracle Load Testing or Oracle Test Manager.

  Resolution: Non-administrator users should disable protected mode if using Internet Explorer 9.

  1. Close Oracle Application Testing Suite applications.
  2. Select **Internet Options** from the **Tools** menu in IE9.
  3. Select the **Security** tab.
  4. Select **Local Intranet**.
  5. Select the**Enable Protected Mode** checkbox.
  6. Click **OK**.
  7. Restart IE9/Oracle Applicaiton Testing Suite applications.

- **Problem**: If the Oracle Application Testing Suite Application Service does not start and the log file <*installdir*>\oats\servers\AdminServer\logs\AdminServer.log contains the following error:

  ```
  weblogic.store.PersistentStoreException: [Store:280073]The file store "WLS_
  DIAGNOSTICS" could not be opened because it contained a file with the invalid
  version 1. A file of version 2 was expected.
  ```

  Resolution: Delete the .DAT file in: <*installdir*>\oats\servers\AdminServer\data\store\diagnostics\ and then restart the server.

  The behavior may occur when starting Oracle Application Testing Suite Application Service and the service is stays in starting mode even after the restarting service command window disappears. This condition may happen when the Weblogic (WLS) service is stopped abruptly - either from a machine reboot or the process is stopped for some reason.

- **Problem**: In some cases, the Oracle Universal Installer (OUI) hangs after installing the Oracle Application Testing Suite. If this situation occurs, you can verify the installation by checking the installation log (typically C:\Program Files\Oracle\inventory\logs\installActions-xxxx.log) for a "OATS installed successfully." entry.

  Resolution: Use the Windows Task Manager, Processes tab to end all running `cmd.exe` processes and, if necessary, end the OUI application process in Task Manager.

- **Problem**: After you install, you can’t login to Oracle Load Testing, Oracle Test Manager, Oracle Administrator, etc.

  Resolution: During install, you will be prompted for a default password to use in the various OATS products where passwords may be required. It’s important that you remember or write down this password as it will be needed to login to these applications (username will typically be "default" or "administrator"). If you don’t remember the password, you will have to re-run the installer and establish a new default password. The passwords can be changed and new user accounts added through the Administrator after install.

- **Problem**: Installation on Japanese systems with existing Oracle database.

  Resolution: In cases where installing Oracle Application Testing Suite on a Japanese system with the option to use an existing Oracle database, you will need to add a system environment variable `NLS_LANG` with value of `Japanese_Japan.JA16SJIS` and restart the system to apply. If the variable already exists with different value on the system, it's not recommended to install ATS on the system because the change will affect other applications which use Oracle database.

7.2 Oracle Load Testing

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

- **Problem**: Load tests on agent machines running Windows XP or Windows 2003 may experience TCP port exhaustion during load testing. When too many TCP ports are opened on the agent machine, virtual users may fail with errors such as:

  ```
  java.net.SocketException: No buffer space available (maximum connections
  reached?)
  ```

and

```
The attempt to connect to the server XXXXX on port NNNN failed. Comparable
WinInet error code: error 12029: ERROR_INTERNET_CANNOT_CONNECT. Caused by:
java.net.BindException occurred. Error Message: Address already in use: connect
```

Resolution: By default, Windows XP and Windows 2003 only allow 5000 TCP ports. This number should be increased according to the steps here:

`http://msdn.microsoft.com/en-us/library/aa560610(v=bts.20).aspx`

In Windows Vista and Windows Server 2008, the default port range is increased from 5,000 to about 65,000 ports:
`http://support.microsoft.com/kb/929851/`

- **Problem**: If a Data Collector process (olt-dc-java-agent.exe) was running and you restart the Oracle Load Testing server and attempt to do something with ServerStats on the server (start a metric, for instance), it will start a second Data Collector process that will not function properly because of the lock established by the first process.

  Resolution: To address this, end the Data Collector process(es) manually, delete the lock file(s) (`$installDir\DataCollector\bin\*.pid`), then repeat the ServerStats task attempted previously.

- **Problem**: 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.

  Resolution: 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.

- 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.

- **Problem**: 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.

  Resolution: To do JMX Discovery on WebLogic 10.3.3, doing the following:

  1. Copy <*installdir*>\lib\wlfullclient.jar to <*installdir*>\DataCollector\classes\weblogic
  2. Log in to Oracle Load Testing.
  3. Select **Metrics** from the **ServerStats** menu then select **New**.
  4. Select the JMX Data Source then click **Discover Counters**.

5. Select the local system and click **Edit**.
6. 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).
7. Click **OK**.
8. 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.

- **Problem**: When bringing up Oracle Load Testing (or Oracle Test Manager) you get the following error reported by the browser: "Unable to connect" or "Can't establish a connection to the server at localhost:8088."

  Resolution: This error indicates that the OATS Application Service is not running. Start (or restart) the OATS Application Service on the server machine where you installed OATS server. You will also want to make sure that the Oracle Database services are also running (this would be OracleServiceXE and OracleXETNSListener if you used the Oracle XE database).

- **Problem**: When bringing up OLT (or OTM) you get the following error reported by the browser: "Error 404--Not Found"

  Resolution: This error indicates that the OATS Application Service is unable to connect to the Oracle Database. You will also want to make sure that the Oracle Database services are also running and start/restart them if necessary (this would be OracleServiceXE and OracleXETNSListener if you used the Oracle XE database). Then restart the OATS Application Service.

- The remote ServerStats Data Collector requires either all of ATS installed or the Remote Agent installed on the remote Data Collector machine.

7.3 Oracle Test Manager

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

- Remote execution of JUnit tests or execution of JUnit tests is not supported on Linux. Support for this is planned for a future release.
- The icons in the Tree View for Test Plans, 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.
- **Problem**: When upgrading from version 9.31 to 12.1, certain pre-existing OTM Filters and Groups may no longer work properly. This issue may be associated with filters based on usernames, such as "Created By", "Owner", "Last Modified By", and with filters based on date and version fields, such as "Last Modified Date" field, and "Version" field.

Resolution: The filters/groups may need to be recreated manually.

- **Problem**: Exceptions occur or application hangs when attempting to export extremely large projects (projects exceeding 11,000 test cases).

  Resolution: For large databases, change Stuck Thread Max Time parameter (default value is 600, i.e. 10 min.) for the server using the following steps:

  1. Close all OATS applications.
  2. Open a browser and go to `http://localhost:8088/console/` (or `http://<server>:8088/console/`).
  3. Log into WebLogic Server (username "oats", password defined during OATS installation).
  4. Expand the **Environment** tree under **Domain Structure** (left pane).
  5. Select **Servers**.
  6. Click the AdminServer(admin) link under Servers (right pane).
  7. Click the **Tuning** tab.
  8. Change the value for parameter "Stuck Thread Max Time" to a larger appropriate value.
  9. Click **Save** then restart the application server.

- **Problem**: Oracle Test Manager fails to playback certain OpenScript scripts completely.

  Resolution: 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
  - Web Functional scripts that display modal dialog windows

  It is necessary to run these scripts using an interactive desktop of a named Windows user account that is always logged in. See also Section 7.6, "Command-Line Script Execution" for additional information.

  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.

Note: For Windows 7 and Vista systems, the command must be run using the elevated command line. From the Start menu, expand the Accessories, right click on the Command Prompt menu option and select **Run as administrator** from the shortcut menu.

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

- **Problem**: When bringing up Oracle Load Testing (or Oracle Test Manager) you get the following error reported by the browser: "Unable to connect" or "Can't establish a connection to the server at localhost:8088."

  Resolution: This error indicates that the OATS Application Service is not running. Start (or restart) the OATS Application Service on the server machine where you installed OATS server.

  You will also want to make sure that the Oracle Database services are also running (this would be OracleServiceXE and OracleXETNSListener if you used the Oracle XE database).

- **Problem**: When bringing up Oracle Load Testing (or Oracle Test Manager) you get the following error reported by the browser: "Error 404--Not Found"

  Resolution: This error indicates that the OATS Application Service is unable to connect to the Oracle Database. You will also want to make sure that the Oracle Database services are also running and start/restart them if necessary (this would be OracleServiceXE and OracleXETNSListener if you used the Oracle XE database). Then restart the OATS Application Service. See the above note about stopping and retsarting services on Windows 7/Vista systems.

7.4 Oracle Database 10*g* Express Edition

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

- **Problem**: In some cases, the Oracle Database 10*g* Express Edition service does not start completely.

  Resolution: 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:

  Note: On Windows 7/Vista (UAC systems), it is necessary to run commands from an elevated command line in order to restart services. If you try to run it from the **Start** menu, you will see the following errors in the cmd window:

  ```
  System error 5 has occurred.
  Access is denied.
  ```

  Run `services.msc` from the command line and restart the services from there. Windows 7/Vista (UAC systems) require administrator privileges to run commands from an elevated command line in order to stop and restart services.

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.

7.5 Oracle Functional Testing - OpenScript

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

- **Problem**: Windows 7 does not permit running OpenScript as non-administrator user. Steps are not recorded in the Windows 7 Secure Desktop environment for non-administrator user when navigating between zones with different security settings. A new process for zones with different security level starts in the browser.

  For example, when starting recording, a browser (brower.launch) starts with the about:blank page – Internet Security Zone. If you start recording the application-under-test – Intranet security zone, a new browser process starts. The OpenScript toolbar is not enabled in the new process and steps performed in the application-under-test are not recorded.

  Resolution: The following workarounds can be used:

  - Set security settings for the Intranet zone to the same as the Internet zone.
  - Add `about:blank` to the Local Intranet zone.

  To set security settings for the Intranet zone:

  1. In Internet Explorer, select **Internet Options** from the **Tools** menu.
  2. Select the **Security** tab.
  3. Select **Internet**.
  4. Note the security level settings for the Internet zone.
  5. Select **Local intranet**.
  6. Set the Security level setting to the same setting as the Internet zone.
  7. Click **OK**.

  To add `about:blank` to Local Intranet zone:

  1. In Internet Explorer, select **Internet Options** from the **Tools** menu.
  2. Select the **Security** tab.
  3. Select **Local intranet**.
  4. Click **Sites**.
  5. Click **Advanced**.
  6. Type `about:blank` into **Add this website to the zone** field.

7. Click **Add**.
8. Click **Close**.
9. Click **OK**.
10. Click **OK**.
11. Use the Intranet zone when recording or playing back scripts in OpenScript.

- **Problem**: Adobe Flex controls do not record during Adobe Flex functional testing.

  > **Note:** The automation libraries/swc files are required for Flex Functional Testing only. This does not apply for Adobe Flex (AMF) load testing which records at the protocol level.

  Resolution: The Flex application must include the Adobe Flex automation libraries either at compile time or at run time. You need at least `automation.swc` and `automation_agent.swc` from the Adobe <flex builder>\sdks\3.5.0\frameworks\libs folder (3.5.0 is an Adobe sdk version). Also, `automation_dmv.swc` is required for charts, advanceddatagrid and olapdatagrid support. See the Creating Applications for Testing section of the Adobe Flex Data Visualization Developer's Guide for additional information about the tasks required to include the Flex automation libraries.

  Creating Applications for Testing:

  http://livedocs.adobe.com/flex/3/html/help.html?content=functest_components2_15.html#178953

  Adobe Flex Data Visualization Developer's Guide:

  http://livedocs.adobe.com/flex/3/html/help.html?content=functest_components2_15.html#178953

  The Oracle OpenScript `openscript_agent.swc` file must be included when re-compiling Flex applications. The Flex application must be linked with the OpenScript Flex agent located in <*installdir*>\OpenScript\plugins\oracle.oats.scripting.modules.flexFT_*version*\flexagent\openscript_agent.swc or equivalent.

- **Problem**: Automatic proxy configuration (**Preferences** - **OpenScript** - **Record** - **HTTP Module**) for Firefox may not correctly configure the browser proxy.

  Resolution: To record through Firefox, you should configure the browser proxy settings manually (Default: host=localhost, port=7777).

- **Problem**: Certain systems may have trouble launching the OpenScript Help (**Help** - **Help Contents**).

  Resolution: 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.

- **Problem**: When recording a functional testing script you get the following error: "Cannot connect to browser".

  Resolution: The browser did not come up in the required amount of time when you clicked the Record button. This could be because the machine was busy or the ATS OpenScript browser add-ons for IE or Firefox were not installed properly. You can try increasing the browser "Startup timeout" setting under OpenScript Preferences, General…Browser section. If that does not work, run the OpenScript Diagnosis Tool under OpenScript Help. The Diagnosis Tool will attempt to correct any browser add-on installation problems, and report any failures it encounters.

- **Problem**: When recording a functional testing script, you do not see any actions added to the script after recording.

  Resolution: The browser add-on is not capturing events and sending them to OpenScript during recording. Run the OpenScript Diagnosis Tool under OpenScript Help. The Diagnosis Tool will attempt to correct any browser add-on installation problems, and report any failures it encounters. You can also try switching from IE to Firefox to see if that works better.

- **Problem**: When you record a Web load test script, you do not see any navigations added to the script after recording.

  Resolution: OpenScript uses an HTTP proxy server between OpenScript and the target application when you record Web load test scripts. If your browser is accessing the Web via a proxy, you may need to chain that proxy through OpenScript's HTTP recording preferences. OpenScript will attempt to chain the proxy automatically for you using the browser's proxy settings. If you do still have problems recording HTTP navigations, check the "HTTP" Record preferences for network interface and proxy settings.

  Also, if you are recording a load test script against a local Web application running on the same machine as OpenScript (like Weblogic Medical Records), you cannot use "localhost" as the server name in the http request. So in the browser, specify your starting URL as: http://systemname:7011/medrec/index.action instead of http://localhost:7011/medrec/index.action. This has to do with the fact that IE bypasses the proxy for localhost addresses and is not an issue for recording in Firefox.

- **Problem**: When recording an Oracle EBS/Forms functional testing or load testing script, the browser appears to hang when loading the Oracle Forms applications and you are not able to continue.

  Resolution: The browser you are using to record your script may not have the proper JVM version downloaded/installed that is required to run the Oracle Forms applications. Before you start recording Oracle EBS/Forms functional testing or load testing scripts, step through the EBS/Forms transaction once in the browser stand-alone (i.e. while not recording) to make sure you have the proper JVM downloaded/installed and confirm that the Forms applications run properly – then try to re-record your script.

- **Problem**: When recording a Siebel functional testing or load testing script, the browser appears to hang when loading the Siebel High Interactivity applications and you are not able to continue.

  Resolution: The browser you are using to record your script may not have the Siebel High Interactivity Framework Active-X components installed that are required to run the Siebel HI applications. Before you start recording Siebel functional testing or load testing scripts, step through the Siebel transaction once in the browser stand-alone (i.e. while not recording) to make sure you have the proper components downloaded/installed and confirm that the Siebel HI applications run properly – then try to re-record your script.

- **Problem**: When recording a Siebel functional testing script, you do not see any Siebel High Interactivity component actions recorded in your script.

  Resolution: In order to record Siebel functional testing scripts in OpenScript, the Siebel Test Automation Component Automation Services (CAS) must be enabled on Siebel Server before recording. Please review the documentation for information on how to enable Siebel Test Automation for Siebel 7.7, 7.8 and 8.x versions. This is not required for Siebel load testing scripts.

- **Problem**: A database error occurs when trying to generate graphs in Oracle Load Testing.

  Resolution: This may occur if the Oracle XE database was installed separately and not part of the Oracle Application Testing Suite installation. The default setting for allowed connections for the XE process is low and needs to be increased. From a command prompt, run the following command, then restart Oracle XE:

  ```
  alter system set processes=200 scope=spfile;
  ```

7.5.1 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.

7.6 Command-Line Script Execution

The following are tips, known issues, and workaround solutions for running OpenScript scripts from the Command-Line Interface (CLI).

- **Problem**: Playback does not recognize objects outside of the visible window area. (Note: this is also applicable to playback of scripts from the OpenScript UI.)

  Resolution: When playing back scripts, make sure the browser window is the same size as when the script was recorded. One way to ensure that the browser is the same size is to record the script with the browser window maximized, then add a `web.window("path").maximize()` method to the script code before playing back the script. Also, make sure the machine used to play back the script is set to the same screen resolution as the machine used to record the script.

- **Problem**: Playback does not recognize recorded keyboard actions.

  Resolution: When recording scripts for command-line execution, use mouse clicks instead of keyboard actions to interact with the application-under-test. For example, when filling in forms, click the mouse on the form fields before entering text rather than using the Tab key. Also, click the Submit button with the mouse rather than pressing the Enter key.

- **Problem**: Playback does not work with Remote Desktop minimized.

  Resolution: Make sure the Remote Desktop is visible/maximized before playback. Alternatively, you can add the `RemoteDesktop_SuppressWhenMinimized` value to the Registry of the system from which you are running the command-line interface, as follows:

Caution: The following procedures modify your system Registry. Modifying the system Registry carries risk of causing serious problems that may affect the operation of your operating system software. Use the information provided at your own risk.

1. Close any currently open Remote Desktop connections.
2. Select **Run** from the **Start** menu.
3. Type `regedit` and click **OK**.
4. Expand one of the following Registry keys depending upon the system and which user account you want to use for the command-line execution:
   - for 32-bit current user account:

     ```
     HKEY_CURRENT_USER\Software\Microsoft\Terminal Server Client
     ```

   - for 32-bit all user accounts:

     ```
     HKEY_LOCAL_MACHINE\Software\Microsoft\Terminal Server Client
     ```

   - for 64-bit current user account:

     ```
     HKEY_CURRENT_USER\Software\Wow6432Node\Microsoft\Terminal Server Client
     ```

   - for 64-bit all user accounts:

     ```
     HKEY_LOCAL_MACHINE\Software\Wow6432Node\Microsoft\Terminal Server
     Client
     ```

5. Select **New** from the **Edit** menu, then select **DWORD (32-Bit) Value**.
6. Enter `RemoteDesktop_SuppressWhenMinimized` as the name and press Enter.
7. Double-click the `RemoteDesktop_SuppressWhenMinimized` registry entry to open the edit dialog box.
8. Specify `2` in the **Value Data** field and click **OK**.
9. Close the Registry Editor.

- **Problem**: Need to automatically initiate a Remote Desktop connection after a reboot.

  Resolution: You can initiate a Remote Desktop connection after rebooting the test machine using a saved Remote Desktop Protocol (.rdp) file for a Remote Desktop connection.

  To specify and save a Remote Desktop Protocol file:

  1. Select **Accessories** from the **Start** menu, then select **Remote Desktop Connection**.
  2. Click the **Options** button.
  3. Specify the Logon settings in the **General** tab.
  4. If you wish to be able to start the Remote Desktop Connection without being prompted for credentials (user and password), select **Allow me to save credentials**.

5. If you wish to automatically start a program on the Remote Desktop connection, click the **Programs** tab and specify the program to start.
6. Click the **Connect** button.
7. Enter the username and password and click **OK**.
8. Close the Remote Desktop connection.
9. Select **Accessories** from the **Start** menu, then select **Remote Desktop Connection**.
10. Click the **Options** button.
11. Select the **General** tab, if necessary, and click **Save As**.
12. Specify the directory and file name for the .rdp file and click **Save**.
13. If you wish to automatically scale the Remote Desktop when connecting, edit the saved .rdp file in a text editor and add the following setting to the end of the settings:

    ```
    smart sizing:i:1
    ```

14. Schedule the .rdp file to launch automatically after a reboot/restart using Task Scheduler or create a batch file on the test machine using a text editor with the following command:

    ```
    mstsc c:/myRdpFile.rdp
    ```

    where *myRdpFile* is the name you used to save the Remote Desktop connection file.

15. If you use a batch file, schedule the batch file to run using a tool that is capable of launching the batch file automatically after a reboot/restart.

8 Documentation Accessibility

For information about Oracle's commitment to accessibility, visit the Oracle Accessibility Program website at http://www.oracle.com/pls/topic/lookup?ctx=acc&id=docacc.

Access to Oracle Support

Oracle customers have access to electronic support through My Oracle Support. For information, visit http://www.oracle.com/pls/topic/lookup?ctx=acc&id=info or visit http://www.oracle.com/pls/topic/lookup?ctx=acc&id=trs if you are hearing impaired.

Oracle Application Testing Suite Release Notes Version 12.1.0.1
E17387-07