

1. LISA Installation and Configuration Guide	2
1.1 Preinstallation	2
1.1.1 LISA Server Architecture	2
1.1.1.1 LISA Server Components	3
1.1.1.2 Data Flow Within LISA Server	4
1.1.2 Licensing Approaches	6
1.1.3 System Requirements and Prerequisites	8
1.1.3.1 LISA Workstation System Requirements	10
1.1.3.2 LISA Server System Requirements	10
1.1.3.3 LISA Virtualize System Requirements	10
1.1.3.4 LISA Pathfinder System Requirements	11
1.1.3.5 LISA Database Usage	11
1.1.4 LISA Download Site	11
1.1.5 Downloading the Installer	13
1.2 Installing and Configuring LISA Workstation	14
1.2.1 Installing LISA Workstation on Windows	15
1.2.2 Installing LISA Workstation on UNIX	19
1.2.3 Configuring License Settings for LISA Workstation	24
1.2.4 Using an HTTP Proxy Server with LISA Workstation	24
1.2.5 Using the License Manager	25
1.2.6 LISA Workstation Installation Notes	26
1.3 Installing and Configuring LISA Demo Server	26
1.3.1 Installing the Demo Server	27
1.3.2 Starting the Demo Server	28
1.4 Installing and Configuring LISA Server	28
1.4.1 Installing LISA Server on Windows	29
1.4.2 Installing LISA Server on UNIX	34
1.4.3 Configuring License Settings for LISA Server	39
1.4.4 Using an HTTP Proxy Server with LISA Server	40
1.4.5 Starting LISA Server	40
1.4.6 Project Directory Structure	40
1.4.7 Calculating Simulator Instances	40
1.4.8 Load and Performance Server Sizing	41
1.4.9 Uninstalling LISA Server	41
1.5 Configuring Third-Party Tools	41
1.5.1 Installing Performance Monitor (Perfmon)	41
1.5.2 Installing and Configuring SNMP	42
1.5.2.1 Installing Microsoft SNMP Agent	42
1.5.2.2 Configuring Microsoft SNMP Agent	45
1.5.3 Running TCPMon	46
1.5.3.1 Using TCPMon as an Explicit Intermediate	47
1.5.3.2 Using TCPMon as a Request Sender for Web Services	49
1.5.4 Installing and Running the HP Test Director Plug-in	49
1.5.5 Installing and Running the IBM Rational Quality Manager Plug-in	53
1.6 Upgrading LISA from a Previous Version	56
1.6.1 LISA 4.x to LISA 5.0	56
1.6.2 LISA 5.0 to LISA 6.0	57
1.6.2.1 Database Schema Changes	57
1.6.2.2 Exporting Legacy Service Images	57

LISA Installation and Configuration Guide

This guide describes how to install and configure LISA.

Preinstallation: Provides an overview of the LISA Server architecture, describes the licensing approaches, lists the system requirements and default port numbers, and describes how to access and download the LISA installers.

Installing and Configuring LISA Workstation: Describes how to install and configure LISA Workstation.

Installing and Configuring LISA Demo Server: Describes how to install and configure LISA Demo Server. This is optional, but is required if you plan to install the examples and work through them as detailed in the [User Guide](#).

Installing and Configuring LISA Server: Describes how to install and configure LISA Server.

Configuring Third-Party Tools: Describes how to configure various third-party tools that can be used with LISA.

Upgrading LISA from a Previous Version: Includes details on steps needed to upgrade from one LISA version to another.

Preinstallation

This chapter provides an overview of the LISA Server architecture, describes the licensing approaches, lists the system requirements and default port numbers, and describes how to access and download the LISA installers.

The following topics are available.

[LISA Server Architecture](#)
[Licensing Approaches](#)
[System Requirements and Prerequisites](#)
[LISA Download Site](#)
[Downloading the Installer](#)

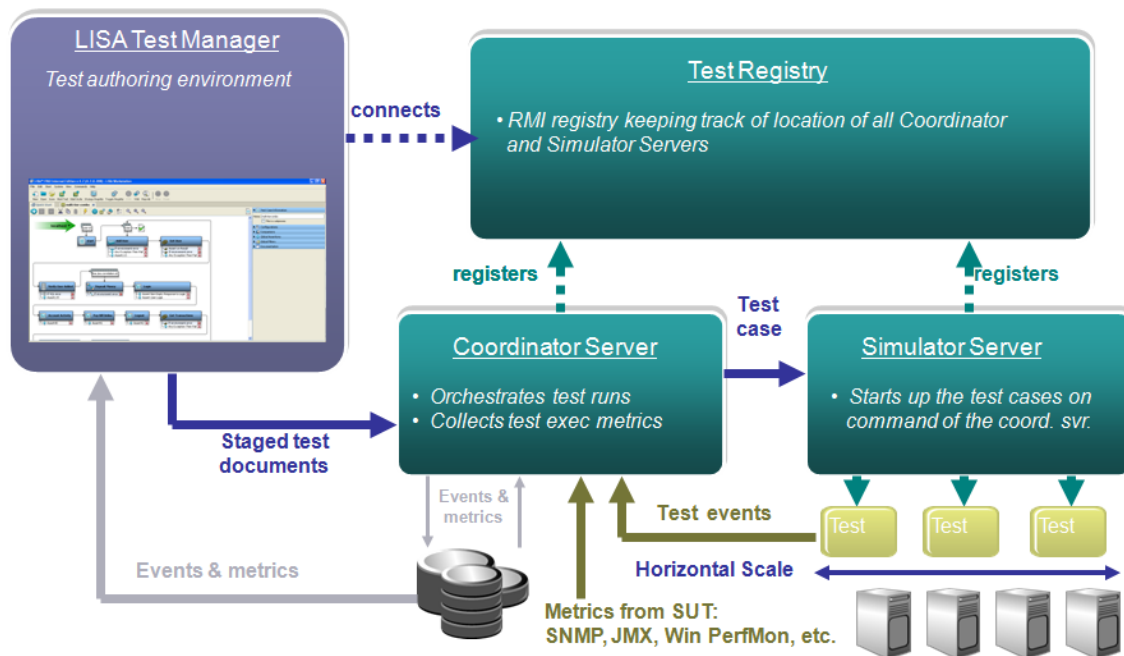
LISA Server Architecture

LISA Server consists of the following major components:

- LISA Workstation
- Registry
- Coordinator Server
- Simulator Server

LISA Workstation is used to create and monitor the tests, but the test cases are run in the LISA server environment. A coordinator server and a simulator server are embedded in LISA Workstation.

The following diagram shows the LISA server architecture.



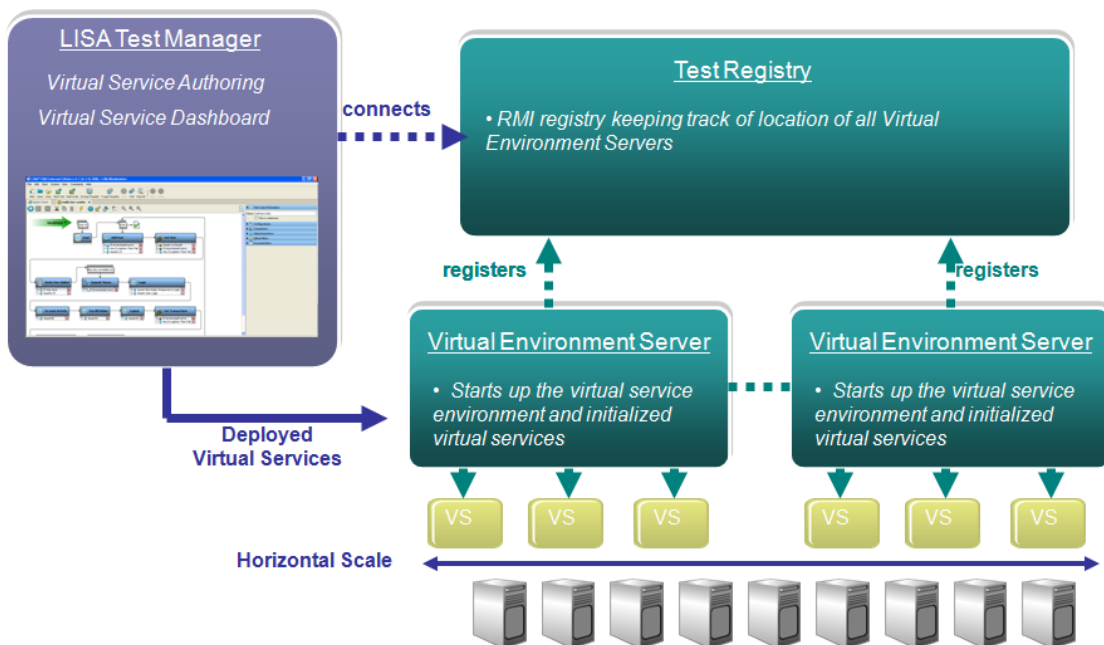
All the tests here are run by the virtual users (or simulators) spawned by the simulator server under the supervision of a coordinator in the coordinator server.

Each simulator connects to and invokes actions on the system under test. A load test results when the virtual users are running in a parallel mode.

A LISA server can be configured to run in the same operating system or computer, including the one that LISA Workstation is using, or on a large distributed environment incorporating many CPUs. Your testing requirements dictate the server architecture to be used. LISA can be scaled for large testing environments by distributing the different components onto different hardware/operating systems.

All LISA components (Workstation, Coordinators, Simulators, Virtual Services Environment) connect to a central registry. A registry is required for LISA Workstation and Server.

The following diagram shows the LISA architecture with LISA Virtualize.



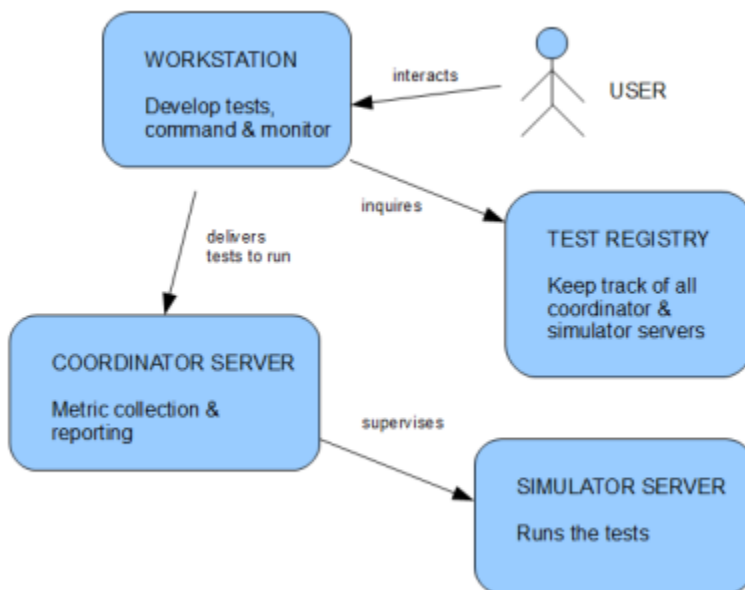
LISA Server Components

LISA Server Components

In LISA Server, the tests are run in the Server environment. LISA Workstation connects to the server to deploy and monitor tests that were developed in LISA Workstation.

LISA Server consists of the following major components:

- **LISA Workstation:** An IDE where test case assets and virtual models are created, edited, and can be run locally in the workstation or staged for remote execution. LISA Workstation needs to be installed on desktop computers for users who will be authoring LISA test and virtual model assets. Any number of workstations can attach to the registry and share the server environment. For more information, see [LISA Workstation](#).
- **Registry:** Provides a central location for the registration of all LISA Server and Workstation components. For more information, see [Registry](#).
- **Coordinator Server:** Receives the test run information in the form of documents, and coordinate the tests that are run on one or more simulator servers. For more information, see [Coordinator Server](#).
- **Simulator Server:** Runs the tests under the supervision of the coordinator server. For more information, see [Simulator Server](#).
- **Virtual Service Environment (VSE):** Used to deploy and run virtual service models. For more information, see the [Virtualize Guide](#).



The registry, coordinator server, simulator server, and Virtual Service Environment are "headless" Java applications that run in separate virtual machines.

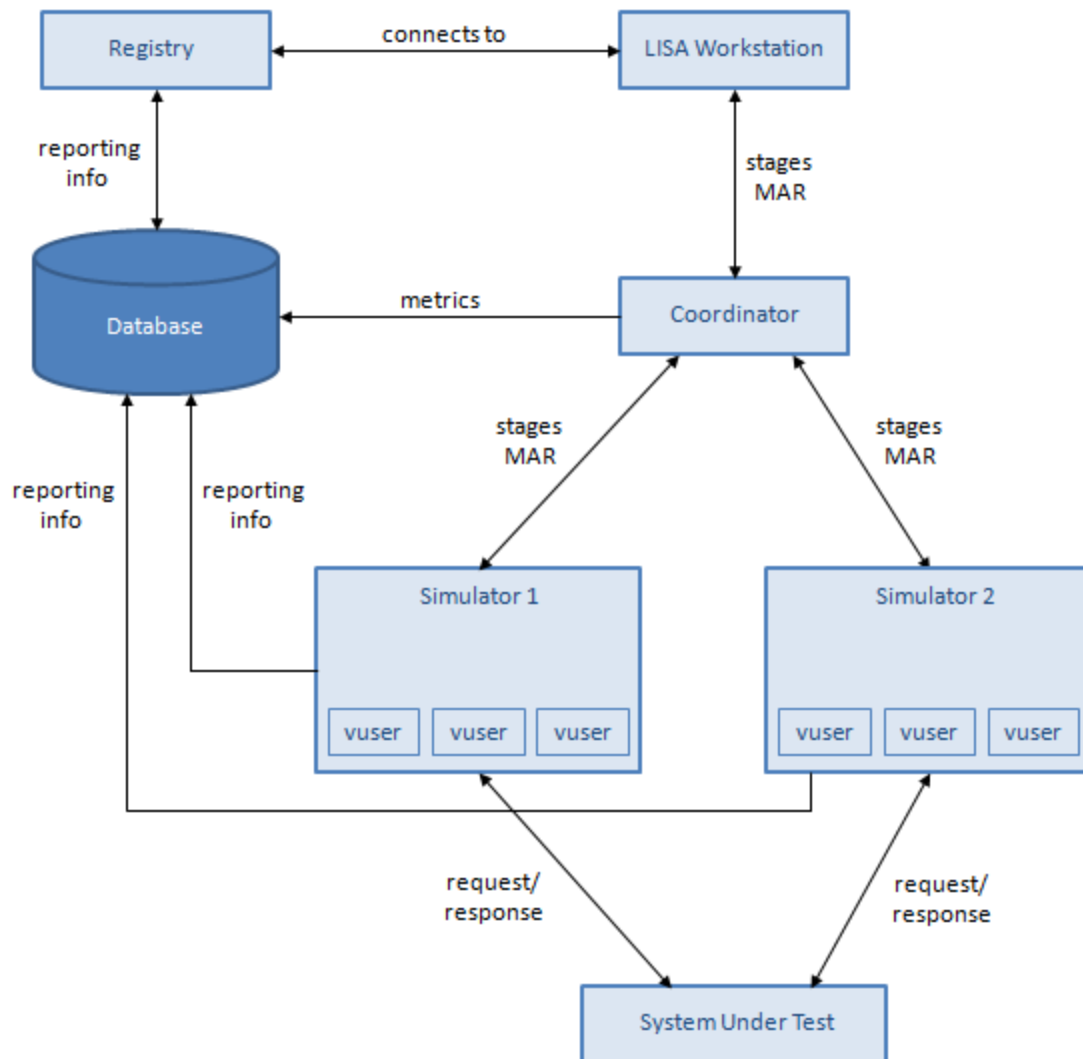
A minimal LISA Server configuration must include at least one of each of these components. There can be as many instances of each type as is needed for a specific testing environment.

Typically, a LISA Server configuration will have one registry, one coordinator server, and multiple simulator servers.

LISA Virtualize is a server-level service. It can co-exist with a registry that has a coordinator and simulator attached to it, but the simulator and coordinator are not mandatory to run VSE.

Data Flow Within LISA Server

The following diagram shows the data flow among the registry, LISA Workstation, coordinator server, simulator servers, and database.



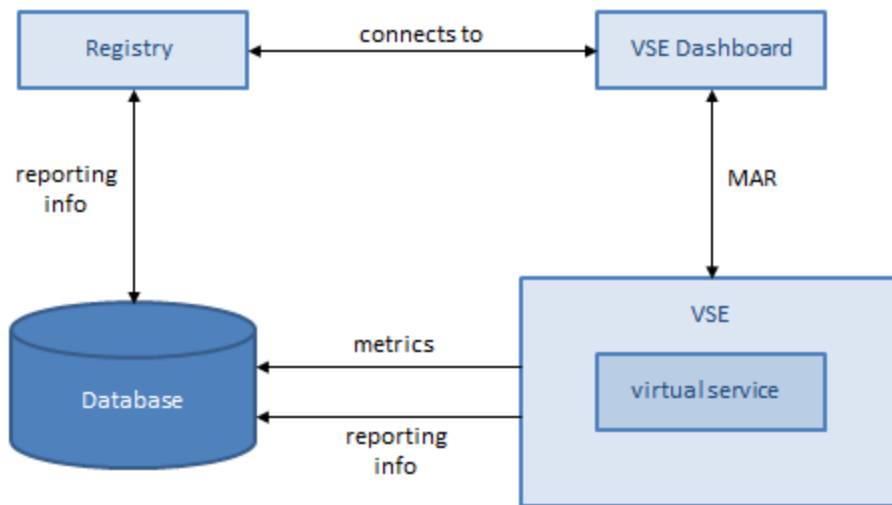
The coordinator server sends test cases to one or more simulator servers in the form of Model Archives (MARs).

The simulator servers interact with the system under test. The types of data that is exchanged between these components can vary enormously. The data could be simple HTTP requests with HTML responses, web service calls, database calls, and so on.

Various components send metrics and reporting information to the database.

The registry hosts the reporting portal, so it talks to the database to retrieve reporting data.

The following diagram shows the data flow among the registry, virtual service environment, VSE Dashboard, and database.



Licensing Approaches

There are two types of licensing approaches:

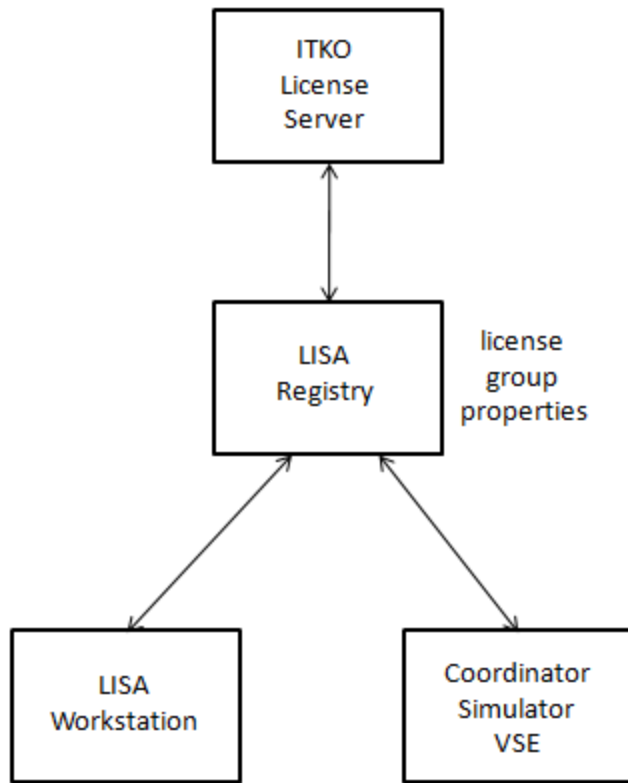
- Registry based
- Server based

The registry-based approach is new in LISA 6.0. The benefit of this approach is that license information does not need to be configured in each install.

Registry-Based Licensing

With the registry-based approach, the registry is the only component that connects to the ITKO license server. The registry uses the license group properties, which must be configured in the **local.properties** file after installation. The registry provides license information as needed to LISA Workstation, coordinators, simulators, and Virtual Service Environments.

The following diagram shows the registry-based approach.



The following table describes the license group properties. The correct values for these properties are provided by ITKO.

Property	Description
laf.server.url	The URL of the ITKO license server.
laf.domain	A three-part identifier that consists of the licensor, the product, and the licensee.
laf.group.name	The name of the license group.
laf.group.password	The password used to access the license group on the license server. When the registry is run for the first time, the property name is changed to laf.group.password.encrypt and the value is encrypted.

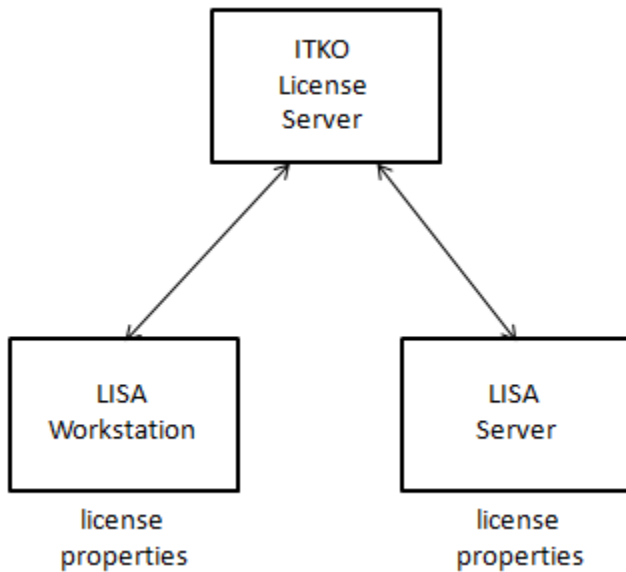
If the registry needs to go through a proxy server to access the ITKO license server, the following properties are also required.

Property	Description
laf.usehttpproxy.server	A flag that indicates whether to use a proxy server to access the license server. The valid values are true and false .
laf.httpproxy.server	The hostname of the proxy server.
laf.httpproxy.port	The port number that the proxy server is listening on.
laf.httpproxy.domain	If the proxy server is NTLM based, then this property specifies the Windows domain to authenticate against.
laf.httpproxy.username	If the proxy server requires authentication, then this property specifies the user name.
laf.httpproxy.password	If the proxy server requires authentication, then this property specifies the password.

Server-Based Licensing

With the server-based approach, LISA Workstation and LISA Server connect directly to the ITKO license server. In each install of LISA Workstation and LISA Server, the license properties must be configured in the **local.properties** file after installation.

The following diagram shows the server-based approach.



The following table describes the license properties. The correct values for these properties are provided by ITKO.

Property	Description
laf.server.url	The URL of the ITKO license server.
laf.domain	A three-part identifier that consists of the licensor, the product, and the licensee.
laf.username	The name of the license.
laf.password	The password used to access the license on the license server. When the registry is run for the first time, the property name is changed to laf.password.encrypt and the value is encrypted.

If the registry needs to go through a proxy server to access the ITKO license server, then the following properties are also required.

Property	Description
laf.usehttpproxy.server	A flag that indicates whether to use a proxy server to access the license server. The valid values are true and false .
laf.httpproxy.server	The hostname of the proxy server.
laf.httpproxy.port	The port number that the proxy server is listening on.
laf.httpproxy.domain	If the proxy server is NTLM based, then this property specifies the Windows domain to authenticate against.
laf.httpproxy.username	If the proxy server requires authentication, then this property specifies the user name.
laf.httpproxy.password	If the proxy server requires authentication, then this property specifies the password.

System Requirements and Prerequisites

This topic lists the requirements for the various LISA components. These general requirements may need to be customized depending on the scope of the project.

- [LISA Workstation System Requirements](#)
- [LISA Server System Requirements](#)
- [LISA Virtualize System Requirements](#)
- [LISA Pathfinder System Requirements](#)
- [LISA Database Usage](#)

LISA JVM System Requirements

ITKO recommendation: 64 bit Java 6

The following are also supported:

- Java 5: Client side only. (You can use LISA running in a Java 5 JRE to test applications on application servers running in an older JRE.)
- Windows installer defaults to 1.6.0_21



Java 1.7 is not supported for LISA Workstation or LISA Server.

Operating System Requirements

ITKO recommendation: 64 bit operating system especially for LISA Server.

The following are also supported:

- Windows 2000/2008/XP/Vista/7, Windows Server 2003 (with latest service pack and all critical updates applied)
- Mac OS X 10.4 and above (Lion not currently supported)
- Red Hat Enterprise Linux 4 and above
- Fedora 8 and above
- AIX 6.1
- Solaris 10
- SUSE Linux

Microsoft Support for Windows 2000 is no longer supported by ITKO. This support ended on July 13, 2010.

Java JDK

LISA is a Java application. A JRE is installed with the 32-bit LISA Windows installer. All other installers require a JRE to be pre-installed.

The minimum supported Java version is Java 1.5. This is a client-side requirement only; LISA running in a Java 1.5 virtual machine (VM) can be used to test applications on application servers running with an older or newer JRE.

The Open JDK is not supported.

For Windows Installer

A JRE is included in the 32-bit installer only. In a Windows 32-bit installation, you can override the JRE used by LISA by setting the system environment variable **LISA_JAVA_HOME** and removing the **LISA_HOME/jre** directory. For all other platforms, the default Java version in the system path or **JAVA_HOME** will be used unless overridden with the **LISA_JAVA_HOME** environment variable.

For Mac and UNIX Installer

For operating systems other than Windows, make sure you have the correct JDK installed, because it is not included in the LISA Installer.

If you are running UNIX and do not have an appropriate version of Java installed, download it from <http://www.oracle.com/technetwork/java/index.html> and follow the installation instructions.

If you are running Mac OS X and do not have an appropriate version of Java installed, obtain it from Apple and follow the installation instructions.

Remember that LISA requires a JDK. A JRE is not sufficient because LISA requires the **tools.jar** file that is included only in the JDK.

After you install Java, set the **JAVA_HOME** environment variable to point to the Java installation directory.

After you install LISA, copy the **tools.jar** file that came with your Java installation into the **LISA_HOME/hotDeploy** directory.

LISA Locks Folder

With LISA 6.0, the following folder within the user's home directory (on UNIX, Linux, and Mac) or Documents and Settings (Windows) requires read-write permission:

- **lisatmp_6.x** (if it exists)

In general, the rest of **LISA_HOME** can be restricted with read-only permissions.

LISA Readme File

The readme file is located in the **LISA_HOME/doc** directory and contains information about environment settings, known issues, LISA editions, and so on.

Read this file before starting LISA.

LISA Workstation System Requirements

LISA Workstation is required for a user desktop system for test case creation/execution. This is where the tests and virtual services are created and maintained.

The minimum requirements are:

- **CPU:** 1 dual core CPU
- **RAM:** 2 GB or more
- **Disk Space:** 2 GB free space
- **Operating System:** Windows 2000, 2003, 2008, 7, XP, Vista, Linus, Solaris, AIX 6.1

LISA Server System Requirements

LISA Server is required for maintaining a central reporting repository and for load and performance testing. In a LISA Server configuration, at a minimum, there will be one registry, one coordinator server, and one simulator server (but could have more than one simulator server).

The minimum requirements are:

- **CPU:** 2 dual core CPUs
- **RAM:** 4 GB
- **Disk Space:** 20 GB
- **Operating System:** Recommended: 64 Bit operating system. Also supported: Windows 2000, 2003, 2008, 7, XP, Vista, Linux, SUSE Linux, Solaris, AIX 6.1
- **Database:** Not required but recommended to use MySQL, Oracle, IBM DB2, or Microsoft SQL Server (SQL Server) for production usage. The database can reside on a different system and should have at least 10 GB of storage.

For load and performance testing, the following resources are recommended.

- 250 virtual users/simulator
- 1 processor core and 2GB RAM / simulator

Example for 4000 concurrent virtual users: 16 simulators; 16 processor core; 32 GB RAM (for LISA)

Per data center:

- 1 test registry and 1 coordinator
- 1 processor core/process = 2 processor cores
- 2GB RAM/each = 4GB (for LISA)

LISA Virtualize System Requirements

LISA Virtualize is required for maintaining a virtualization environment. As LISA VSE is a server-level service, it can co-exist with a registry that has a coordinator and simulator attached to it, but the simulator and coordinator are not mandatory to run VSE.

The following are baseline requirements only.

- **CPU:** 2 GHz or faster
- **RAM:** 2 GB or more
- **Disk Space:** 5 GB free space
- **Operating System:** Recommended: 64-bit operating system. Also supported: Windows 2000, 2003, XP, Vista, 7, Linux, SUSE Linux, Solaris, AIX 6.1 (only with LISA 5.0)
- **Database:** Not required but recommended to use MySQL, Oracle, DB2, or Microsoft SQL Server for production usage. The database can reside on a different system and should have at least 10 GB of storage.

The following requirements are recommended for larger scale VSE deployments.

- 256 virtual service threads per VSE instance
- 1 processor core and 2GB RAM per VSE instance

Example for 1,000,000 transactions per day: 2 threads/service to support functional tests, ~6 threads/service to support virtualization for load and performance tests, 8 cores = 2048 concurrent virtual service threads, 16GB RAM (for LISA)

LISA Pathfinder System Requirements

LISA Pathfinder is required for exposing transactions and tracing them step-by-step across the application architecture.

The minimum requirements are:

- **CPU:** 8 cores
- **RAM:** 8 GB
- **Disk Space:** 5 GB local disk storage; 50 GB database storage (recommended databases are MySQL, Oracle, DB2, or Microsoft SQL Server)
- **Operating System:** Windows 2000, 2003, 2008, 7, XP, Vista, Linux, Solaris, AIX 6.1

LISA Database Usage

LISA Virtualize (VSE) Server

- Database is used for the storage of legacy virtual service images only.
- Recommend a dedicated "non-production" database.

LISA Server

- Most of the database is used for the storage of report results, which can be exported to other formats as needed.
- The database is also used for Continuous Validation schedules and for access control (ACL).

Pathfinder

- Most of the database is used for the storage of paths, including request and response data, SQL statements, application logs, and others.
- The database is also used for cases (a collection of paths to troubleshoot a specific defect or audit record).

LISA Download Site

The download site for the latest release is located at [6.0.7 Release](#).

To access the site, you must enter a user name and password.

The user name is the last node of your license domain, followed by a forward slash, followed by your license user name. The password is your license password.

For example, assume that the following is true:

- Your license domain is ITKO/LISA/ABC.
- Your license user name is EnterpriseWorkstation
- Your license password is z5RiW4NNr.

In this scenario, the user name is ABC/EnterpriseWorkstation and the password is z5RiW4NNr.



This information should be in your license email. If you have not received this email, contact your sales representative.

The download site includes the following sections:

- **STOP AND READ**
- **Build Information**
- **LISA Education Materials**
- **Installers**
- **Deployable Example Server**
- **Other**

STOP AND READ

Before you start the installation, read the content in this section.

Build Information

This section provides the latest build information, and a link to the Release Notes.

LISA Education Materials

This section contains a link to the educational materials on the Customer Support Portal.

Installers

This section includes the following types of installers.

Windows Workstation with samples

This installer installs the Windows version of LISA Workstation and deploys a JBoss Demo Server on your system that is pre-configured and ready to run LISA examples.

An installer bundles some examples to be used with the LISA Demo Server. The most common LISA Workstation installation is the **Windows Workstation with samples**.

This installer has sample test cases, which can be referred to while you get started with LISA.



The other workstation installers do not include the Demo Server, and you must download it separately. See [Installing and Configuring LISA Demo Server](#).

Windows

Workstation - lisa_wrk_win.exe – Choose for LISA Workstation installation (about 273 MB)

Server - lisa_svr_win.exe – Choose for LISA Server installation (about 385 MB)

Windows (64 bit)

Server - lisa_svr_win_x64.exe – Choose for LISA Server installation (about 349 MB)

Mac OS X

Workstation - lisa_wrk_osx.dmg – Choose for LISA Workstation installation (about 248 MB)

Server - lisa_svr_osx.dmg – Choose for LISA Server installation (about 360 MB)

Linux/Solaris/UNIX

Workstation - lisa_wrk_unix.sh – Choose for LISA Workstation installation (about 236 MB)

Server - lisa_svr_unix.sh – Choose for LISA Server installation (about 348 MB)

To run the command-line version:

```
./lisa_svr_unix.sh -c
```

For UNIX users, if this does not work, try the [generic .tar file](#), [lisa_unix.tar.gz](#). After you expand this tar file, you must run **configure.sh**.

After you decide on the installer operating system (Windows, Mac OS X, or Linux/Solaris/Unix), click View to see detailed download instructions. Choose the appropriate "Installer type" and download the installer, by clicking the appropriate **Download** link.

For more detailed instructions on the install, see [Downloading the Installer](#).

Deployable Example Server

This section lets you download the LISA Demo Server if you choose an installer that does not include the LISA Demo Server.

Other

The following plug-ins can be run with LISA tests:

- IBM Rational TestManager for Windows
- Mercury's Quality Center TestDirector for Windows

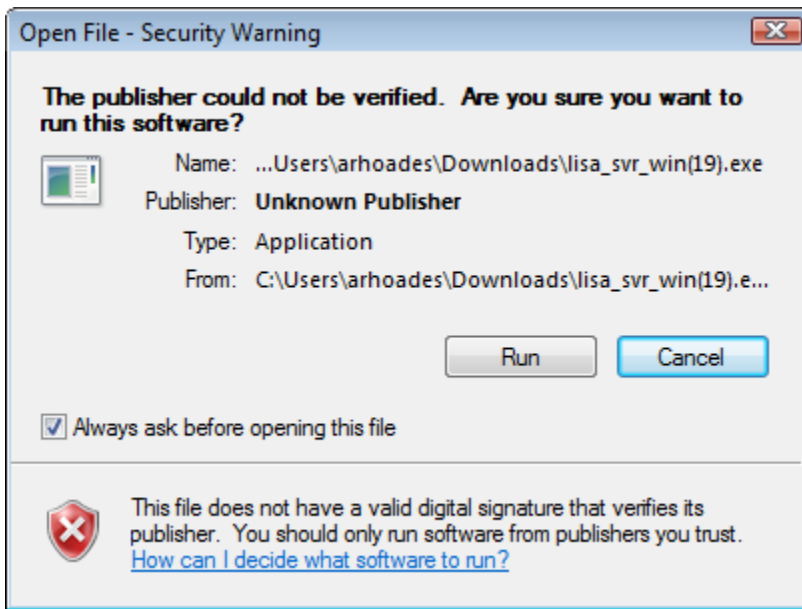
Click the appropriate **Download the plugin** link.

Downloading the Installer

The installers are located on the [LISA download site](#).

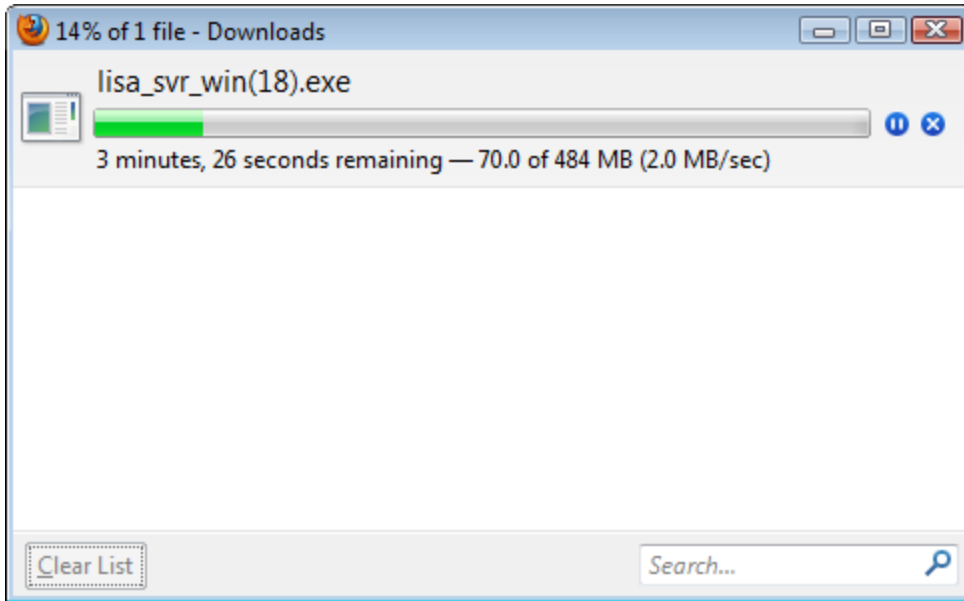
Before installation, make sure that you have enough disk space.

When you click the **Download** link for the appropriate installer based on operating system, the **File Download - Security Warning** dialog opens.

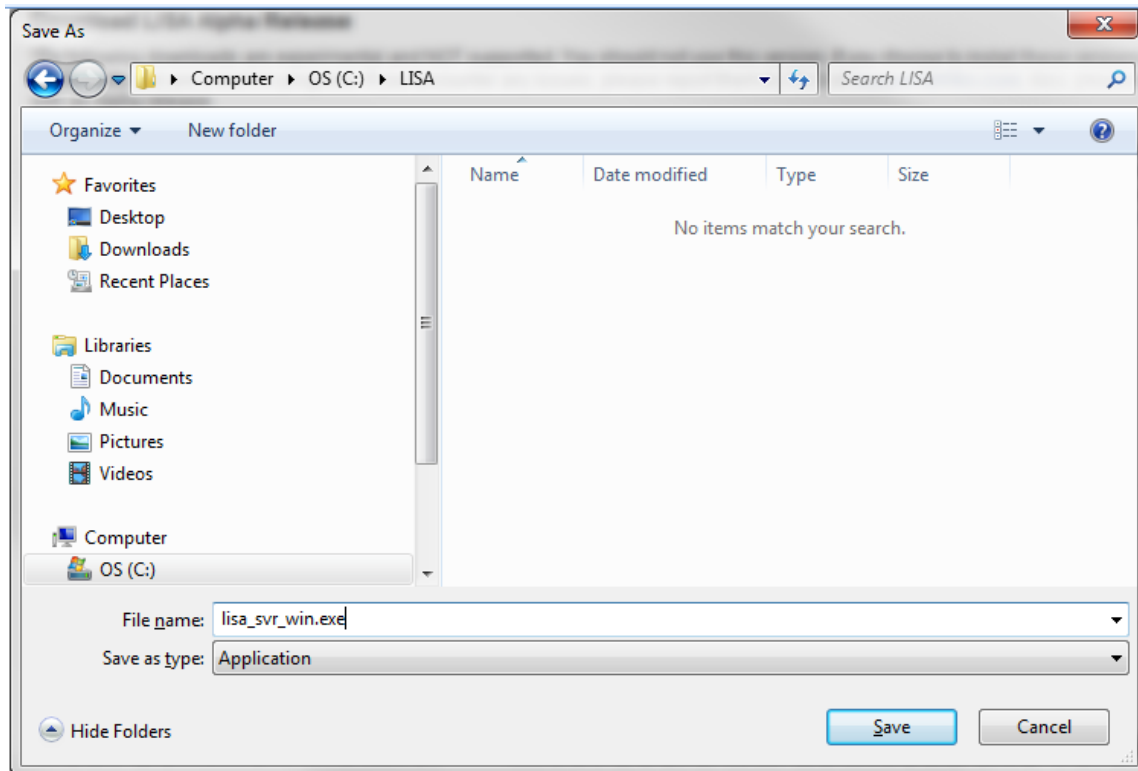


To download and save the installer, click Save. To run the installer without saving it, click Run.

If you click Run, the following screen opens. The installer file is downloaded to a temporary folder.



If you click Save, the following screen opens so the download location can be entered.



Browse to the directory where you want to save the installer and click Save. The installer will be downloaded to the specified location.

After the installer is downloaded to the specified location, double-click the installer file.

The Security Warning dialog opens. Click Run to start the installer.

For detailed steps on using the installer, see the appropriate section:

- [Installing and Configuring LISA Workstation](#)
- [Installing and Configuring LISA Demo Server](#)
- [Installing and Configuring LISA Server](#)

Installing and Configuring LISA Workstation

This chapter describes how to install and configure LISA Workstation.

After you install and configure LISA Workstation, you can log in to LISA Workstation by following the steps in [Opening LISA Workstation](#). Be aware that if your computer has an installation of LISA Workstation rather than LISA Server, then you need to specify a registry that is running on a remote computer. LISA Workstation installations do not include a local registry.

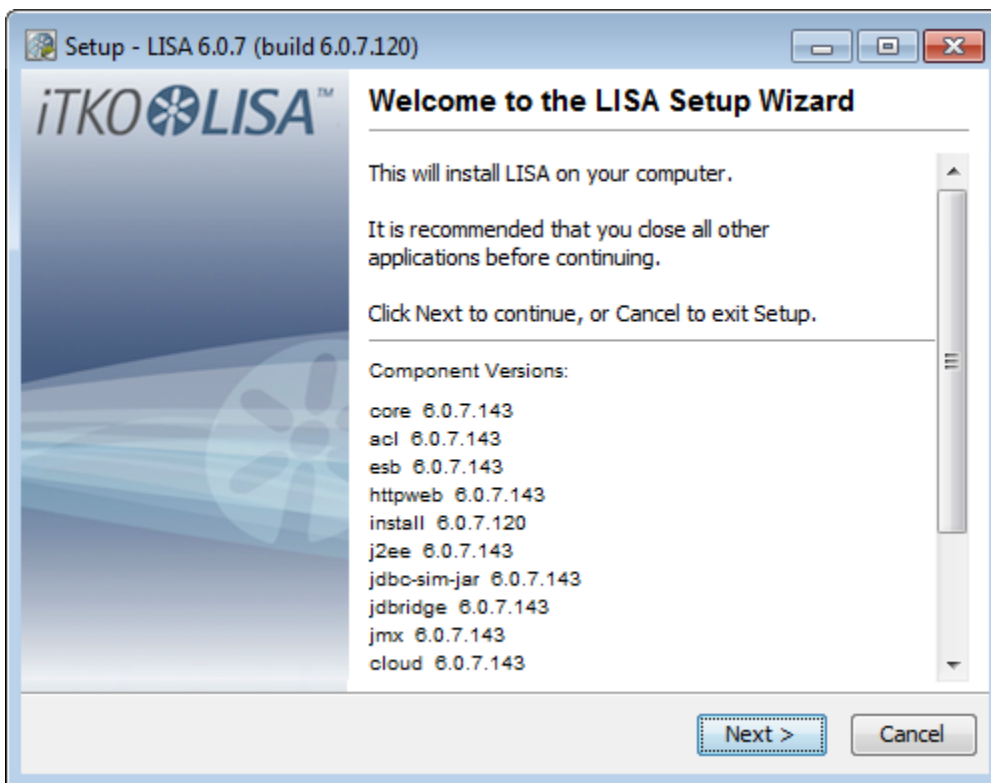
The following topics are available.

[Installing LISA Workstation on Windows](#)
[Installing LISA Workstation on UNIX](#)
[Configuring License Settings for LISA Workstation](#)
[Using an HTTP Proxy Server with LISA Workstation](#)
[Using the License Manager](#)
[LISA Workstation Installation Notes](#)

Installing LISA Workstation on Windows

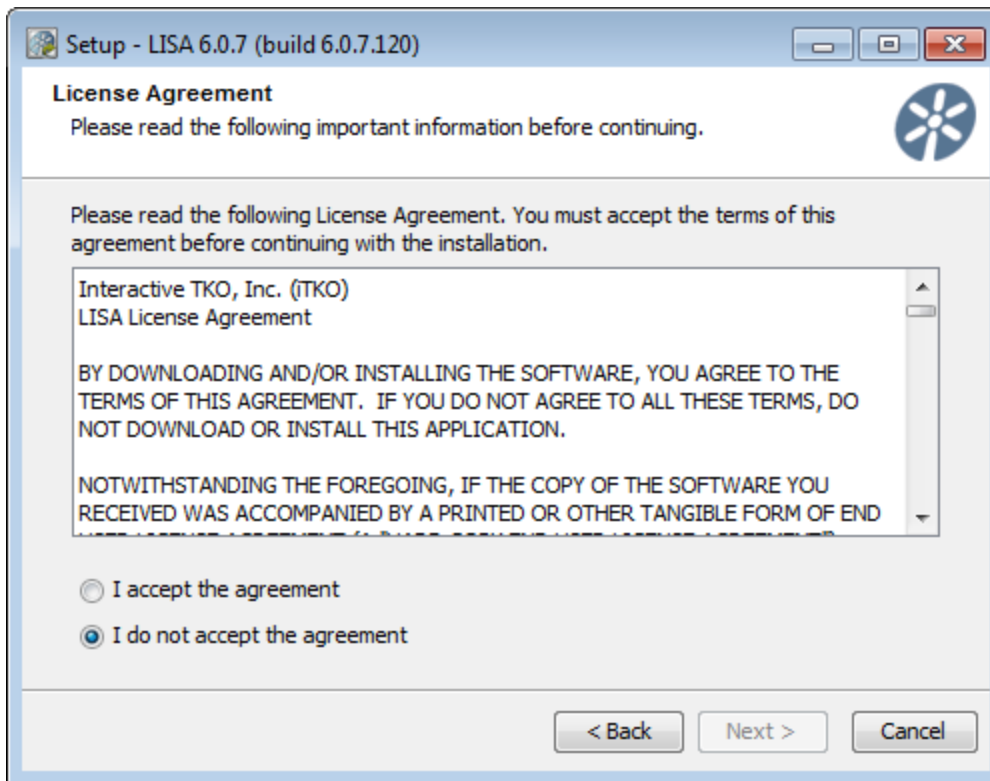
When you [download and run the installer](#) for LISA Workstation, the LISA Setup wizard appears.

The first step lists the components that will be installed.

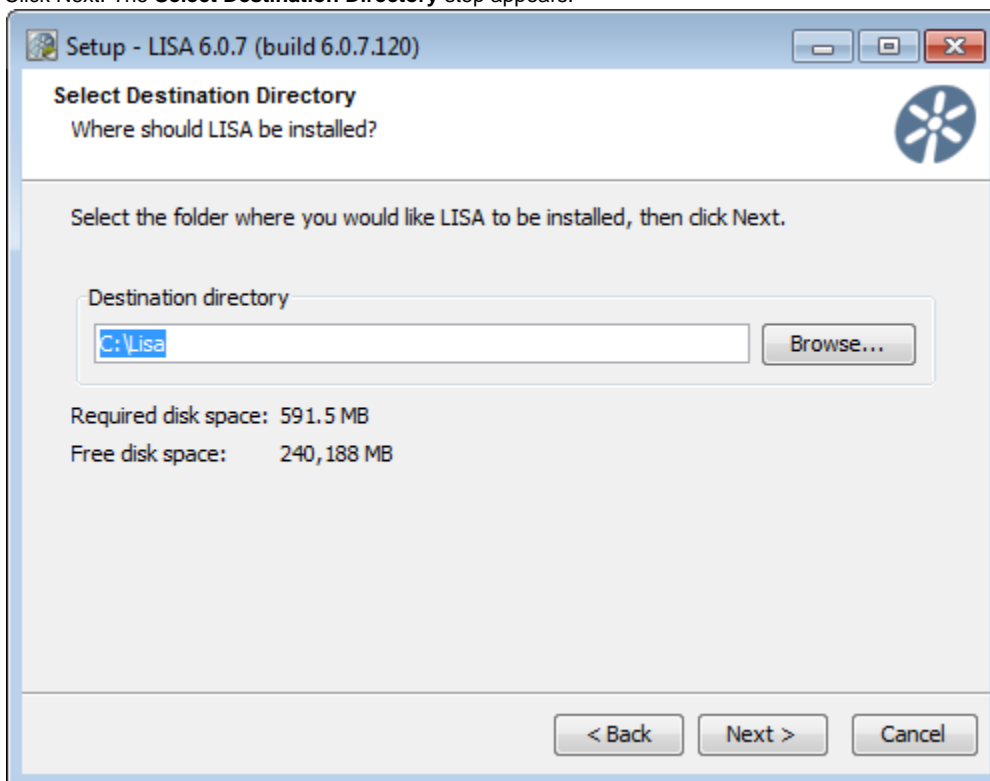


To install LISA Workstation on Windows

1. Click Next. The **License Agreement** step appears.



2. Read the license agreement carefully. The license agreement contains terms and conditions that you must agree to before installing.
3. When you are done, select the **I accept the agreement** check box.
4. Click Next. The **Select Destination Directory** step appears.

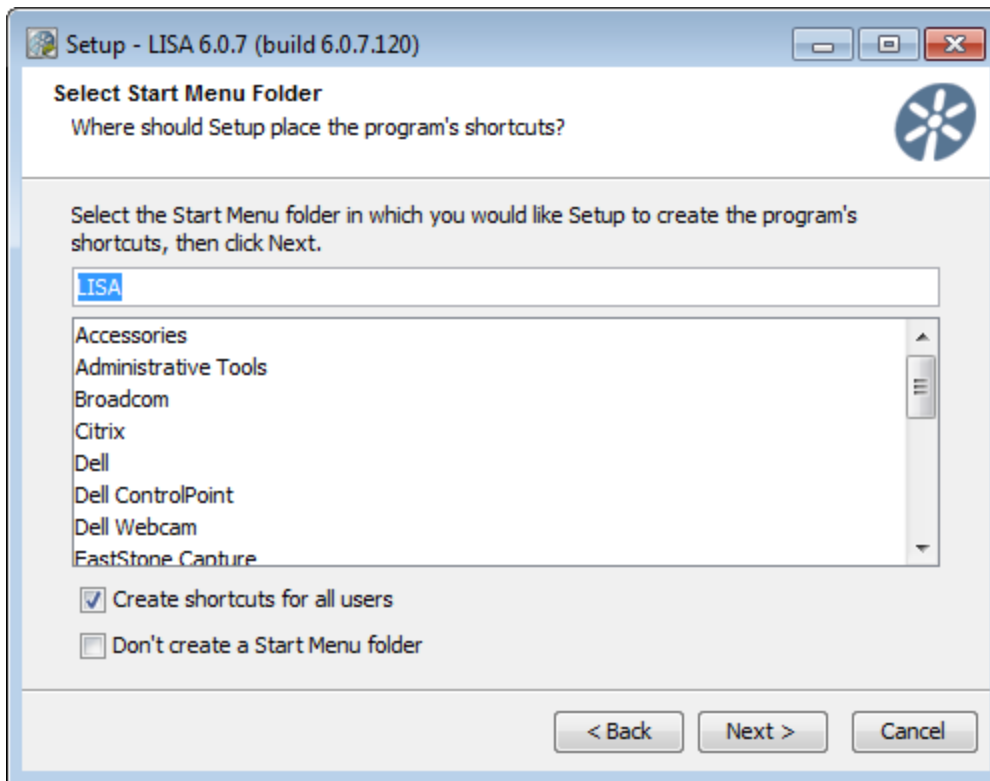


5. Specify the directory where LISA will be installed. You can use a directory that contains spaces (for example, **C:\Program Files**).

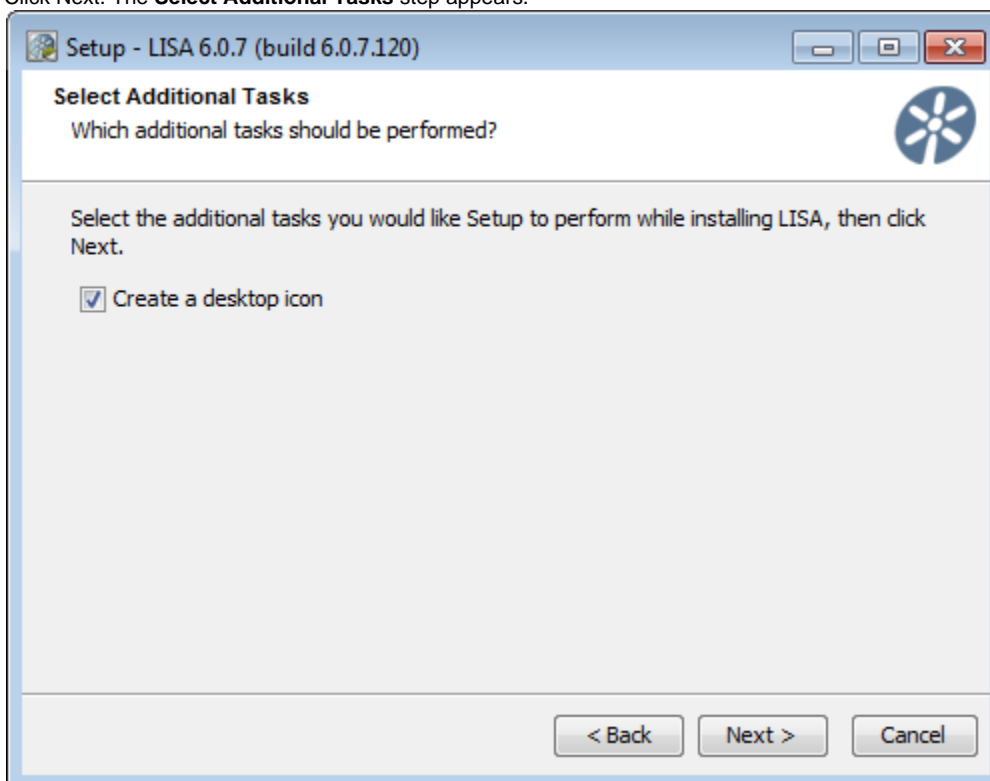


The documentation uses the term **LISA_HOME** to refer to the installation directory.

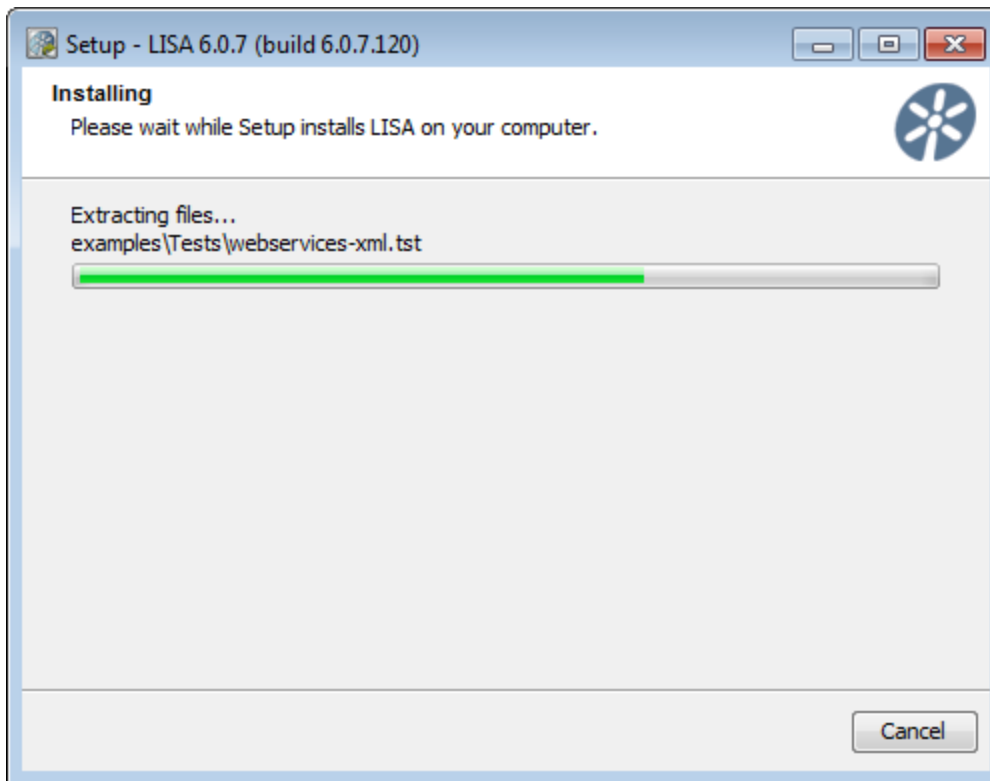
6. Click Next. The **Select Start Menu Folder** step appears.



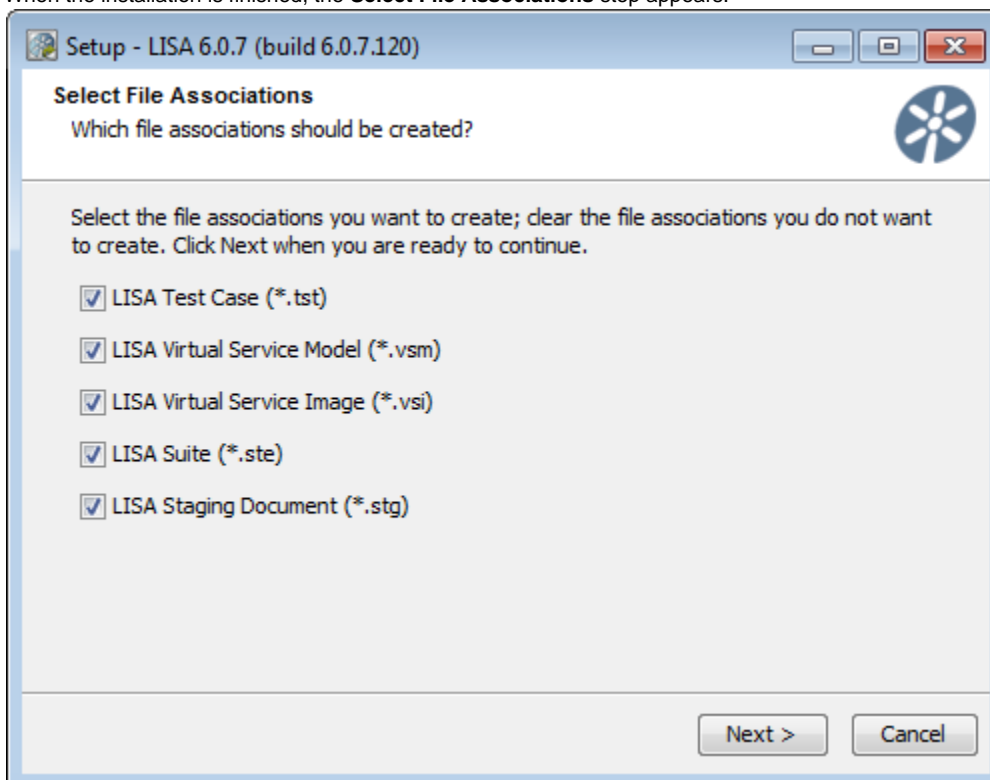
7. Enter the name of the Start menu folder. The default name is **LISA**.
8. Select Create shortcuts for all users to create shortcuts for all users using LISA.
9. Select Don't create a Start Menu folder to prevent the installer from creating a Start menu folder.
10. Click Next. The **Select Additional Tasks** step appears.



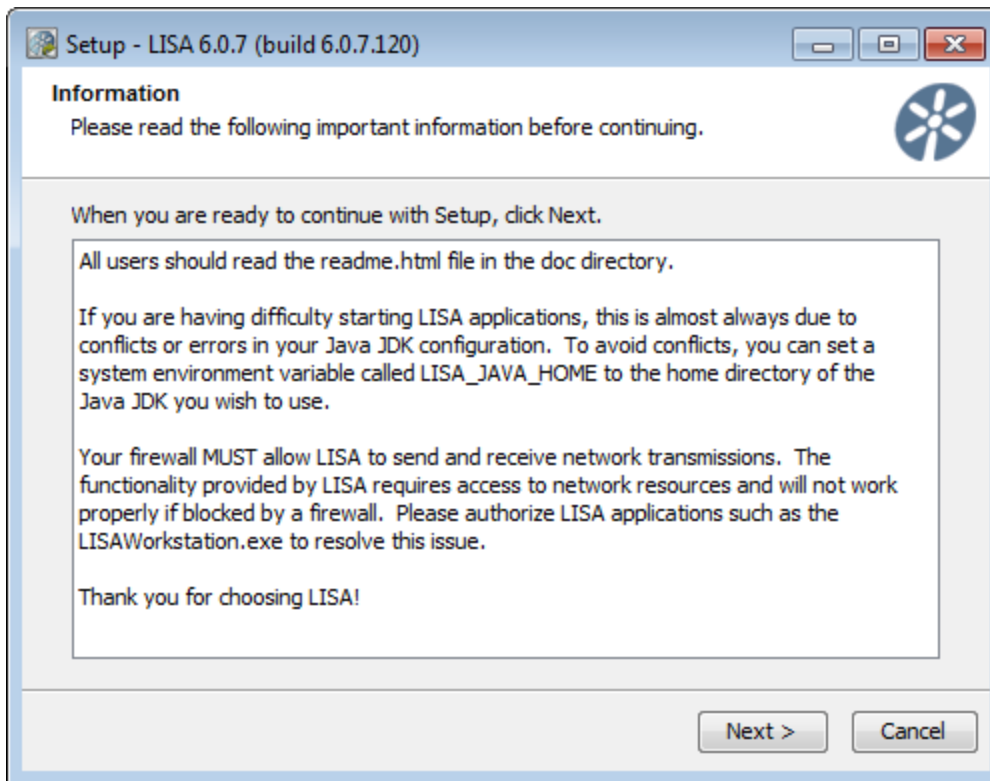
11. If you want to create a desktop icon for LISA, leave the Create a desktop icon check box selected.
12. Click Next to start the installation.



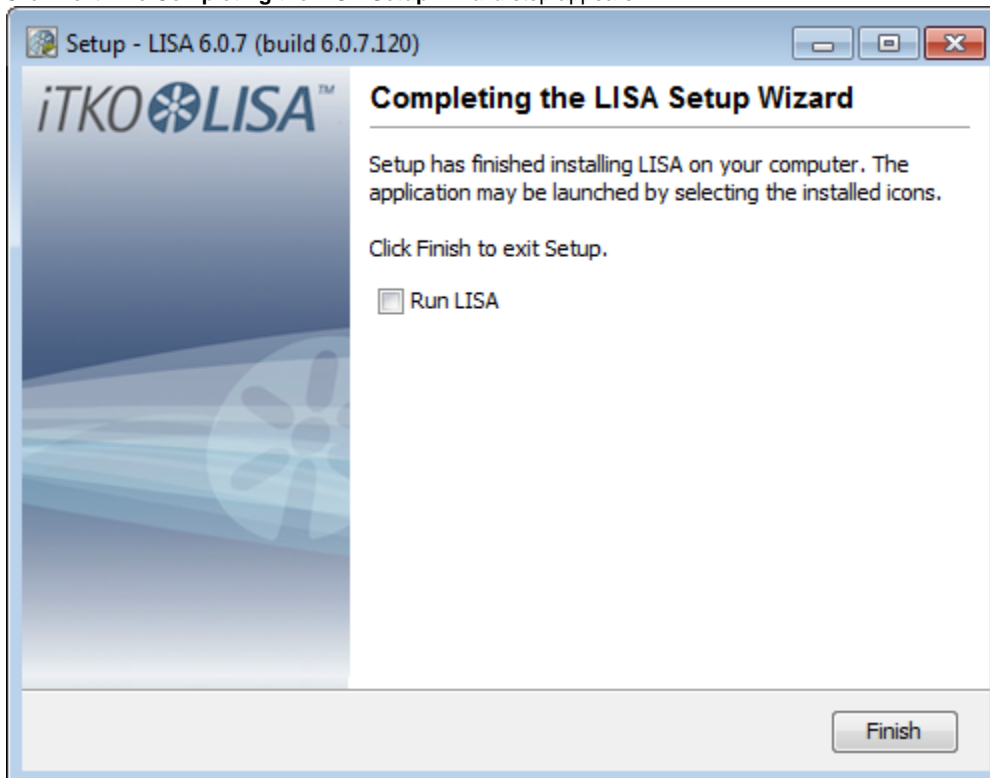
13. When the installation is finished, the **Select File Associations** step appears.



14. Select the file extensions that you to want to associate with LISA. By default, all the file extensions (.tst, .vsm, .vsi, .ste, .stg) are selected.
15. Click Next. The **Information** step appears.



16. Read the information carefully.
17. Click Next. The **Completing the LISA Setup Wizard** step appears.



18. If you want to open LISA Workstation immediately, select the Run LISA check box. LISA Workstation requires you to specify a registry. LISA Workstation installations do not include a registry, so you will need to use a registry that is running on a remote computer.
19. Click Finish.
You are now ready to enter your [license credentials](#).

Installing LISA Workstation on UNIX

This topic assumes that you have [downloaded the installer](#) from the download site.

You might need to change the permissions of the installer file so that it can be executed.

To install LISA Workstation on UNIX

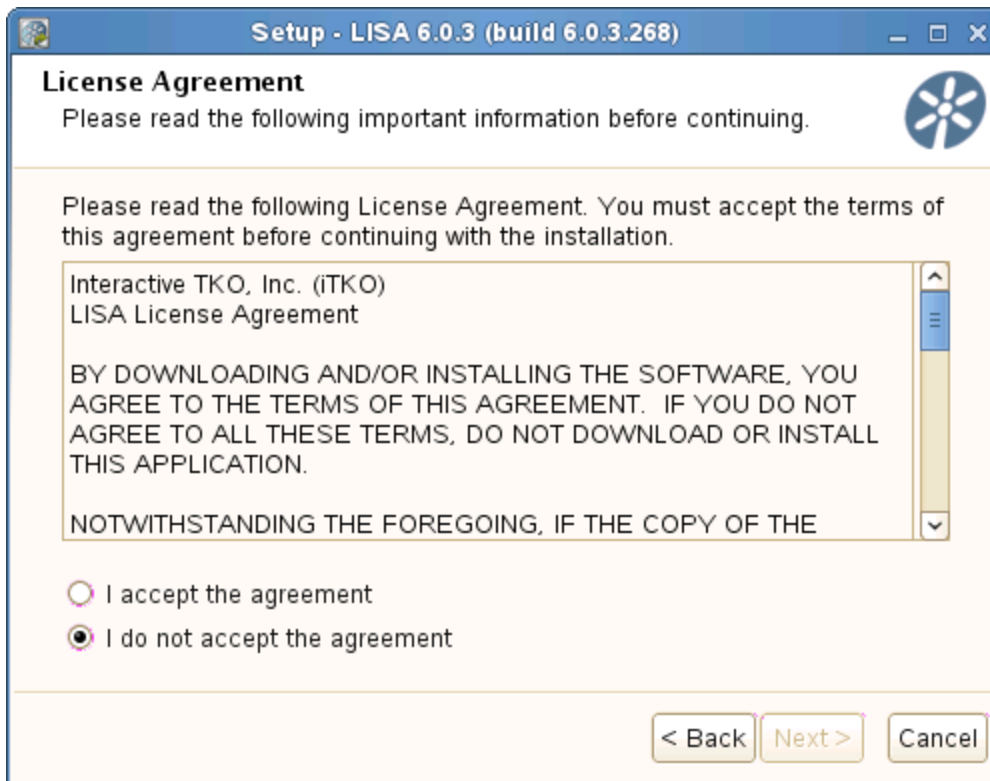
1. In a terminal window, navigate to the directory where the installer file is located and run the installer file. For example:

```
./lisa_wrk_unix.sh
```

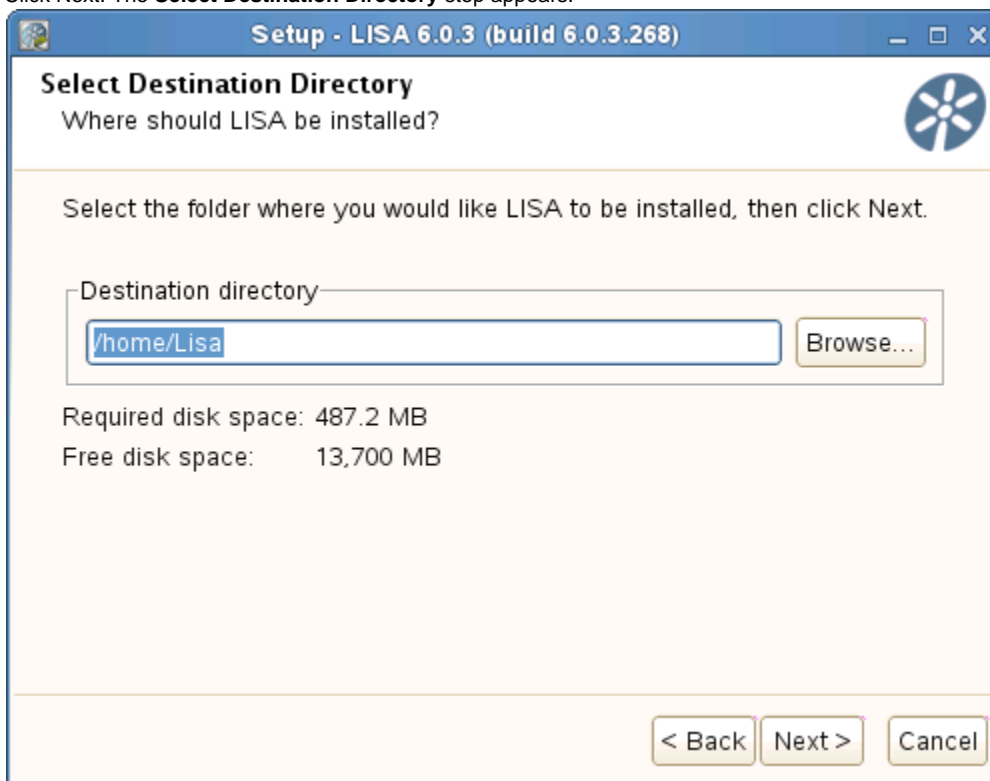
The LISA Setup wizard appears. The first step lists the components that will be installed.



2. Click Next. The **License Agreement** step appears.



3. Read the license agreement carefully. The license agreement contains terms and conditions that you must agree to before installing.
4. When you are done, select the **I accept the agreement** check box.
5. Click Next. The **Select Destination Directory** step appears.

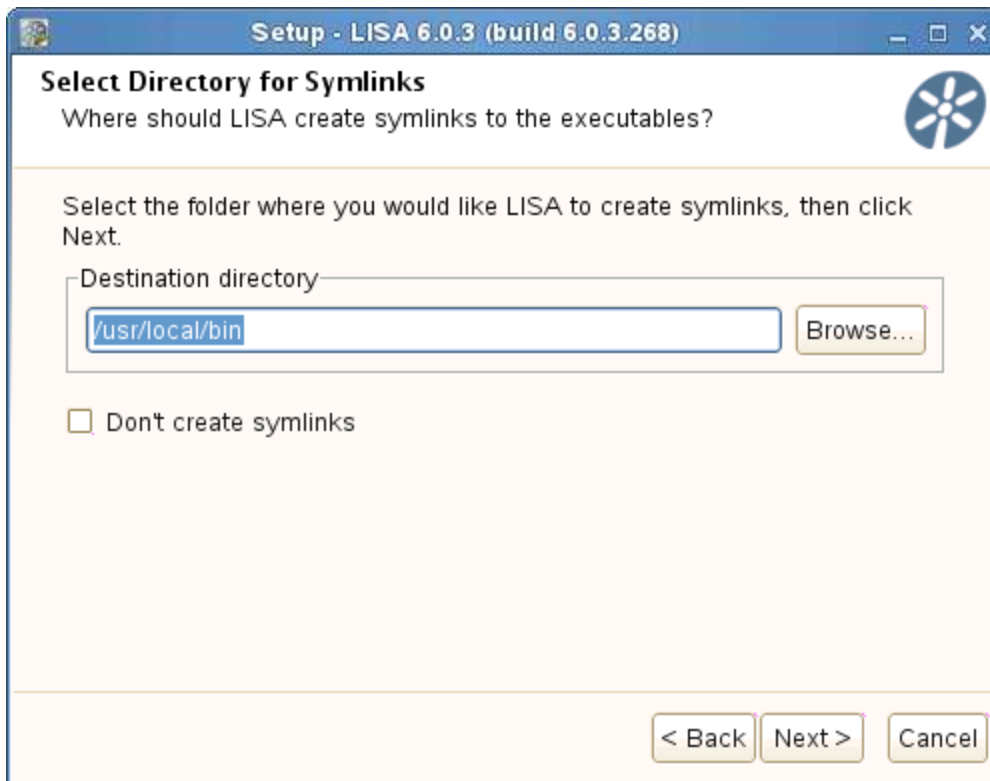


6. Specify the directory where LISA will be installed. Do not use a directory that contains spaces.

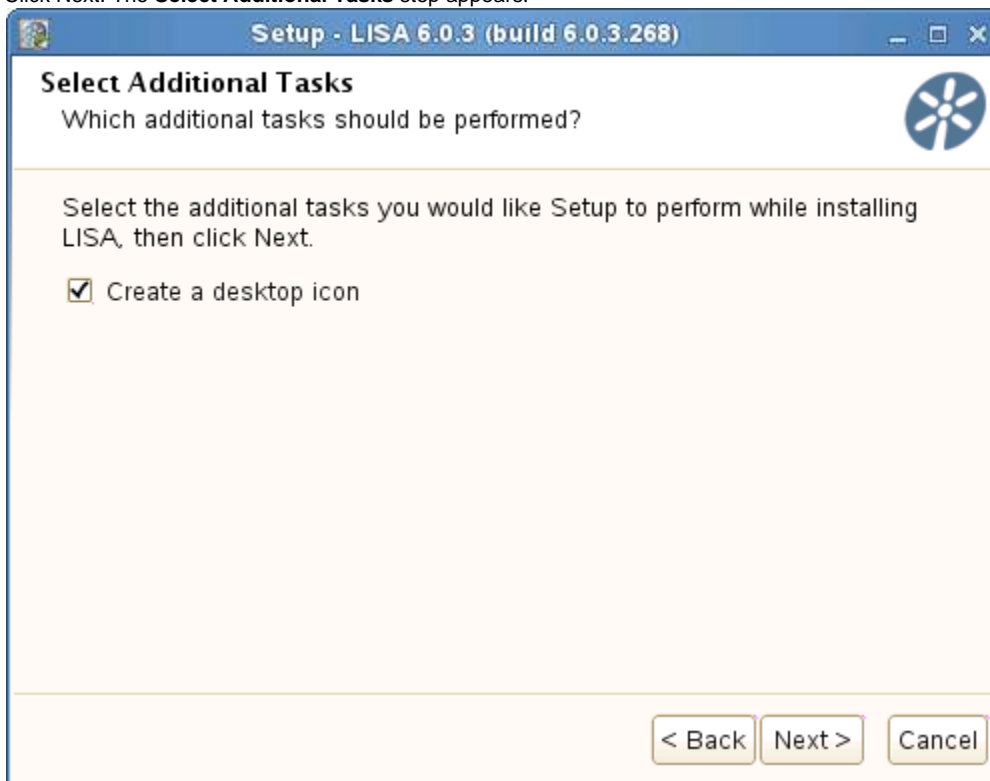


The documentation uses the term **LISA_HOME** to refer to the installation directory.

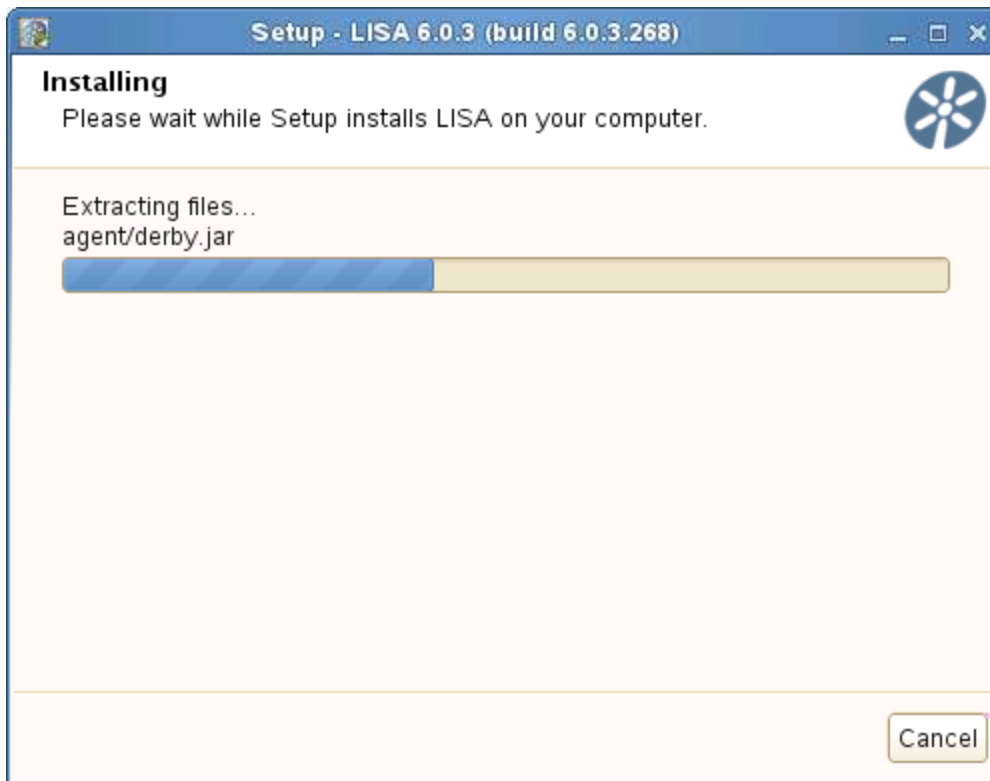
7. Click Next. The **Select Directory for Symlinks** step appears.



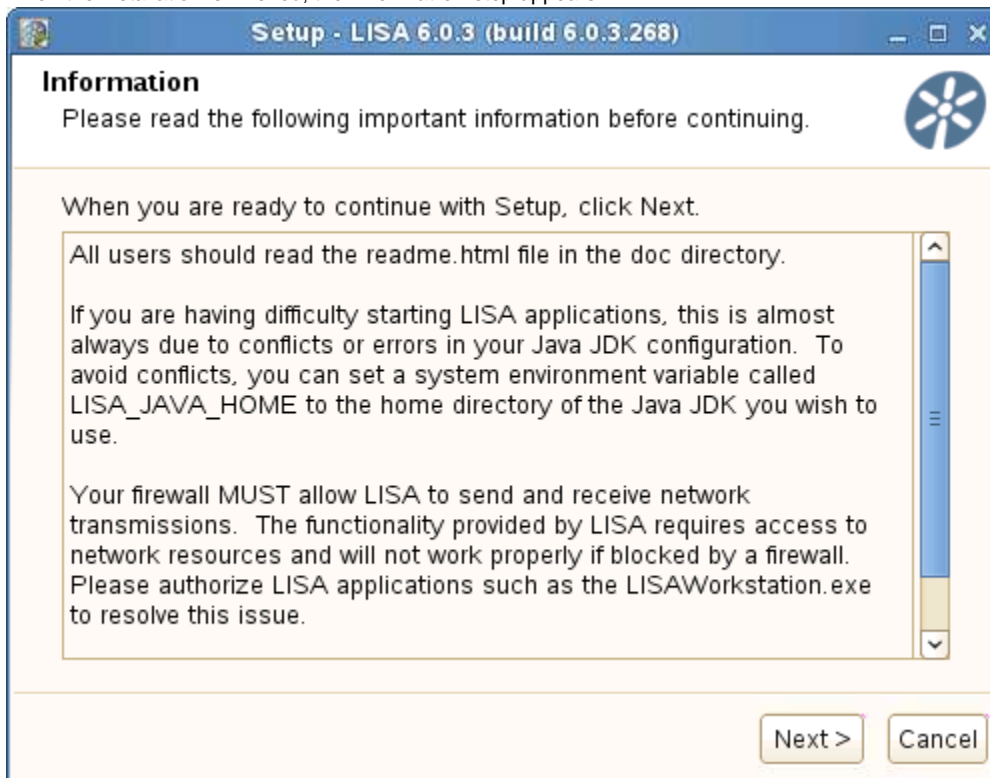
8. Specify the directory where LISA will create symbolic links to the executable files. You must have the required permissions to write to the directory. If you do not want symbolic links to be created, then select the check box.
9. Click Next. The **Select Additional Tasks** step appears.



10. If you want to create a desktop icon for LISA, leave the **Create a desktop icon** check box selected.
11. Click Next to start the installation.



12. When the installation is finished, the **Information** step appears.



13. Read the information carefully.
14. Click Next. The **Completing the LISA Setup Wizard** step appears.



15. If you want to open LISA Workstation immediately, select the Run LISA check box. LISA cannot start up unless you have a registry started already.
16. Click Finish.
You are now ready to enter your [license credentials](#).

Configuring License Settings for LISA Workstation



This topic is for organizations that are using the server-based licensing approach. If your organization is using the registry-based licensing approach, you can ignore this topic. For more information, see [Licensing Approaches](#).

After LISA Workstation has been installed, you must add the license information to the **local.properties** file in the **LISA_HOME** directory.

To create the **local.properties** file, make a copy of the **_local.properties** file in the **LISA_HOME** directory. Change the name of the copy to **local.properties**.

Open the **local.properties** file in a text editor. In the license properties section, remove the comment symbols in front of the properties. Set the property values to the server URL, domain, username, and password from your license email.

```
# =====  
#license properties  
# =====  
laf.server.url=https://license.itko.com  
laf.domain=iTKO/LISA/YOURCO  
laf.username=YOURUSERNAME  
laf.password=YOURPASSWORD
```

Save the **local.properties** file.

When you run LISA for the first time, the name of the **laf.password** property is changed to **laf.password.encrypt** and the password is encrypted.

Using an HTTP Proxy Server with LISA Workstation

If you are using an HTTP proxy server, you will have to configure the following section of the **local.properties** file in the **LISA_HOME** directory. Be sure to remove the comment symbols in front of the properties.


```
# =====  
#license properties if using an http proxy server  
# =====  
#laf.usehttpproxy.server=true  
#laf.httpproxy.server=my_proxyserver.com  
#laf.httpproxy.port=3128  
# === if your proxy server requires credentials - leave blank to use native NTLM authentication  
#laf.httpproxy.domain=if needed for NTLM  
#laf.httpproxy.username=if needed  
#laf.httpproxy.password=if needed
```

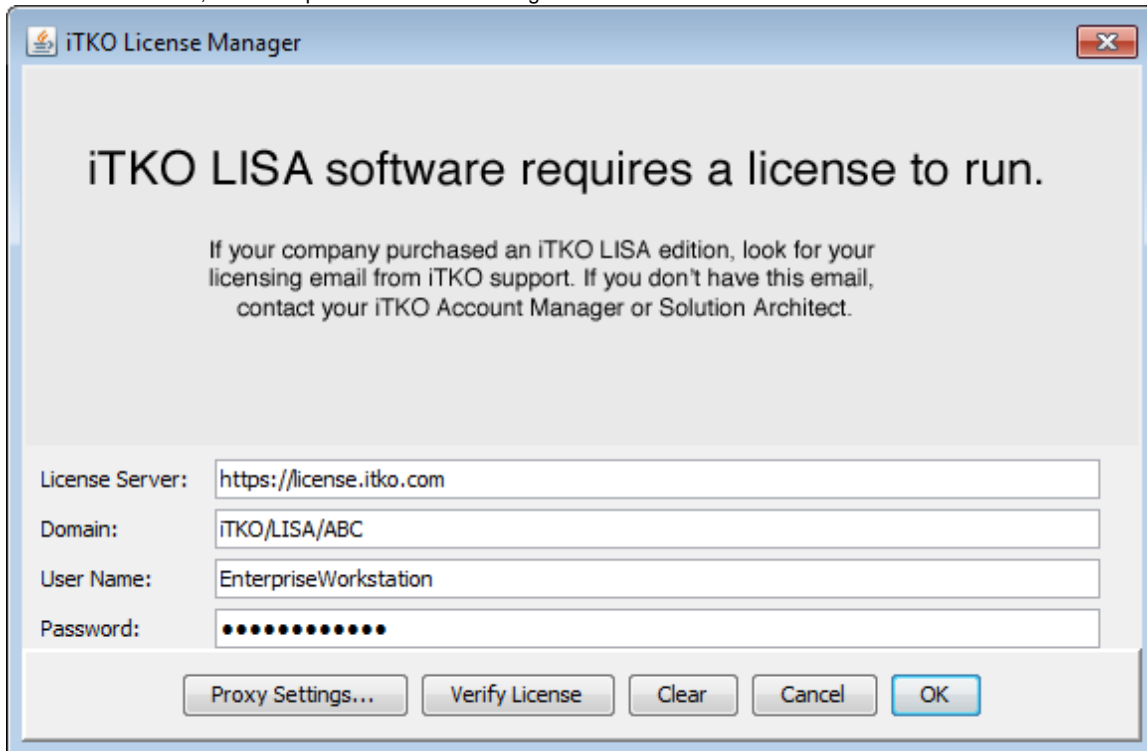
Using the License Manager

As an alternative to configuring license settings in the **local.properties** file, you can use the License Manager dialog in LISA Workstation. With this approach, the settings are added to the LISA GUI preferences file.

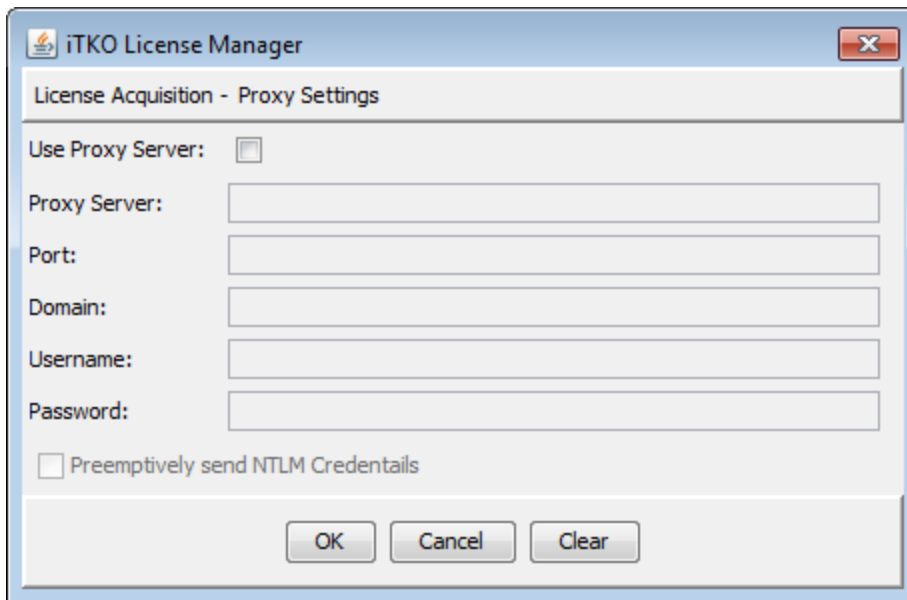
After you switch to using the License Manager dialog, you cannot change the license settings in the **local.properties** files and override what is in the preferences file.

To use the ITKO License Manager

1. From the main menu, select Help > LISA License Settings.



2. Enter the license server URL, domain, user name, and password.
3. The license server is accessed over HTTPS on port 443 and can support most proxies if your network requires proxy access to public Internet sites. If you need a proxy to reach the Internet, click **Proxy Settings** and enter your proxy-related information.



4. After you enter all the information, click OK.

LISA Workstation Installation Notes

Using LISA with Your Java Environment

You can replace the default JRE used by the LISA installation with your own Java environment. To do so, you must understand how LISA selects the JRE to use.

The following priority is used to select what Java VM to use when launching LISA applications:

1. The LISA-installed JRE (in a directory named **jre** in the LISA install directory)
2. LISA_JAVA_HOME environment variable
3. JAVA_HOME environment variable
4. JDK_HOME environment variable

To use your Java environment:

1. Rename the JRE directory in the LISA installation directory (for example, to **jre_default**).
2. Point the LISA_JAVA_HOME environment variable to your Java installation directory.

Environment Settings

LISA documentation mentions a token named **%LISA_HOME%** or **\$LISA_HOME**. This is the location where LISA was installed.

On all supported operating systems, an environment variable is set with this name automatically from the launch scripts or programs.

For example, if you installed into **C:\Lisa**, that would be the value of **%LISA_HOME%**. LISA Workstation also has access to the value of this variable in a property named **LISA_HOME**.

If you need to put additional JARs, zips, or directories in LISA's classpath, you have two options:

- Define the environment variable **LISA_POST_CLASSPATH** and set the resources you want there.
- Put them in the **%LISA_HOME%\hotDeploy** directory.

For more information about environment settings, see [Common LISA Properties and Environment Variables](#).

Generic UNIX Installer

If you have any problems using the Mac OS X or UNIX installers, we provide a Generic UNIX tar file on the [download site](#).

1. Download and expand the tar file.
2. Run **configure.sh**. You may have to change the permissions on this file.
3. Copy the **tools.jar** file that came with your Java install into either the **LISA_HOME/lib** directory or the **LISA_HOME/hotDeploy** directory. The JAR files in the **LISA_HOME/lib** directory are loaded only once. The JAR files in the **LISA_HOME/hotDeploy** directory are loaded whenever the contents of the directory change.

Installing and Configuring LISA Demo Server

LISA Demo Server includes a variety of [examples](#) that help you get started with LISA.

There are two ways to access the Demo Server to view LISA samples:

- Installing it locally on your system
- Accessing it remotely from the ITKO website

Local Demo Server

ITKO provides a bundle of the entire server hosted on examples.itko.com that you can download and run.

We recommend that you download the LISA Demo Server with your LISA installation, so that you can launch the demos from the desktop shortcut or the Windows Start menu.

If you download the **Windows Workstation with samples** installer as described in [LISA Download Site](#), the JBoss Demo Server is already included in that installation. The JBoss Demo Server is included in all LISA Server installers.

The demo server database, **lisa-demo-server.db**, gets created when the demo server is started for the first time. It is not present until then. It is located at **LISA_HOME/DemoServer/lisa-demo-server/jboss/server/default/data**.



The demo server uses port 1529, which should not be in use by any other application. If the demo server startup script is able to ping a Derby database on port 1529, it will not start our script because it assumes the script is already running.

Remote Demo Server

The same local demo server, which demonstrates LISA's features and capabilities, is also available on the ITKO website.

You can run the examples directly from the ITKO website at <http://examples.itko.com/>.

However, several of these examples utilize non-web ports. For example, many J2EE Servers use port 1099 to look up JNDI names. If you are having success with the website examples but not the others, then you are probably behind a firewall that is preventing you from accessing those services.

This chapter describes how to install and run the LISA Demo Server locally.

The following topics are available.

[Installing the Demo Server](#)
[Starting the Demo Server](#)

Installing the Demo Server

You can download the LISA Demo Server from the [LISA download site](#).

This site is password protected. If you have any questions about your login credentials, contact your [ITKO sales representative](#).

Enter the given **Username** and **Password** and click OK, to open the LISA GA download page.

Scroll down to **Deployable Example Server** section.

Deployable Example Server

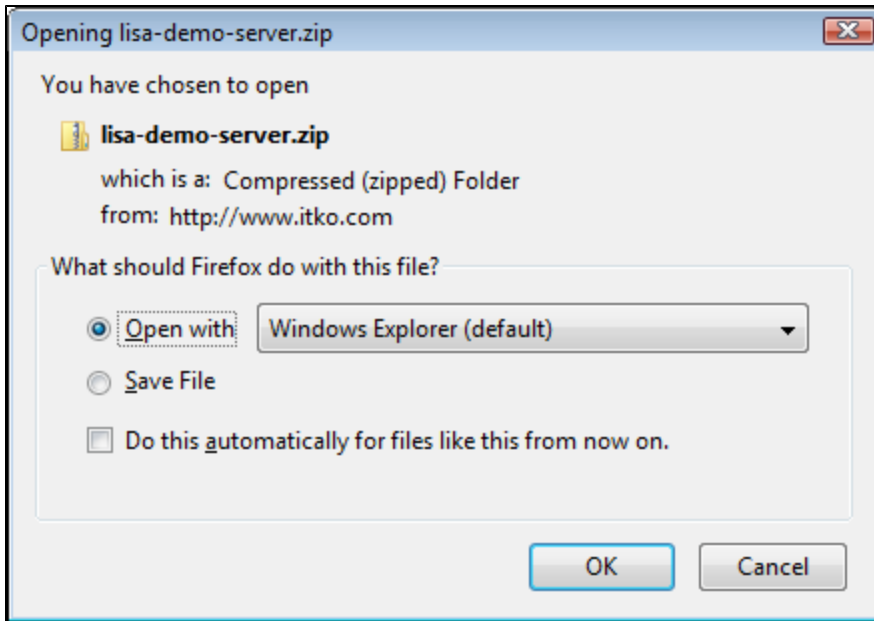
JBoss: ITKO provides a hosted demo server you can connect to with example test cases included in your download. If you are behind a firewall you may not be able to run some of the examples. For that reason we have created a self-running version of JBoss you may download.

[Download a self-running version of JBoss](#)

Includes source for the examples.

By using or downloading any of the above software, you agree to abide by ITKO's Terms Of Use for the software and this site. LISA Automated Testing is a registered trademark of ITKO (Interactive TKO), Inc., InstallAnywhere is a registered trademark of Zero G Software, Inc. Mac OS is a registered trademark of Apple Computer, Inc. Solaris and Java are trademarks of Sun Microsystems, Inc. Windows is a registered trademark of Microsoft Corporation. All other marks are properties of their respective owners.

Click the **Download a self-running version of JBoss** link to start the download.



The JBoss Demo Server download file **lisa-demo-server.zip** is about 117 MB and includes the JBoss Server with all the LISA examples.

To install the JBoss Demo Server

After the zip file is downloaded, extract the zip file contents into a local directory on your computer.



- Make sure you set an environment variable **JAVA_HOME**. This is required for JBoss to compile and run JSP files.
- Do not put the JBoss Server directory on your desktop or any path with spaces in it. JBoss will not be able to compile the JSPs if there is a space in the path to its directory.

Starting the Demo Server

The **readme.txt** file included with the LISA Demo Server contains operating environment-specific instructions for starting the demo server.



The demo server runs the LISA Agent by default, reporting as much information as possible back to LISA Pathfinder. To turn off this reporting, use the **-noagent** flag. To turn off heap / stack information only, use the **-noheapss** flag.

For more information about starting the demo server and running some of the sample test cases packaged with LISA, see [Tutorials](#).

Installing and Configuring LISA Server

This chapter describes how to install and configure LISA Server.

The procedure for installing LISA Server is similar to the procedure for installing LISA Workstation.

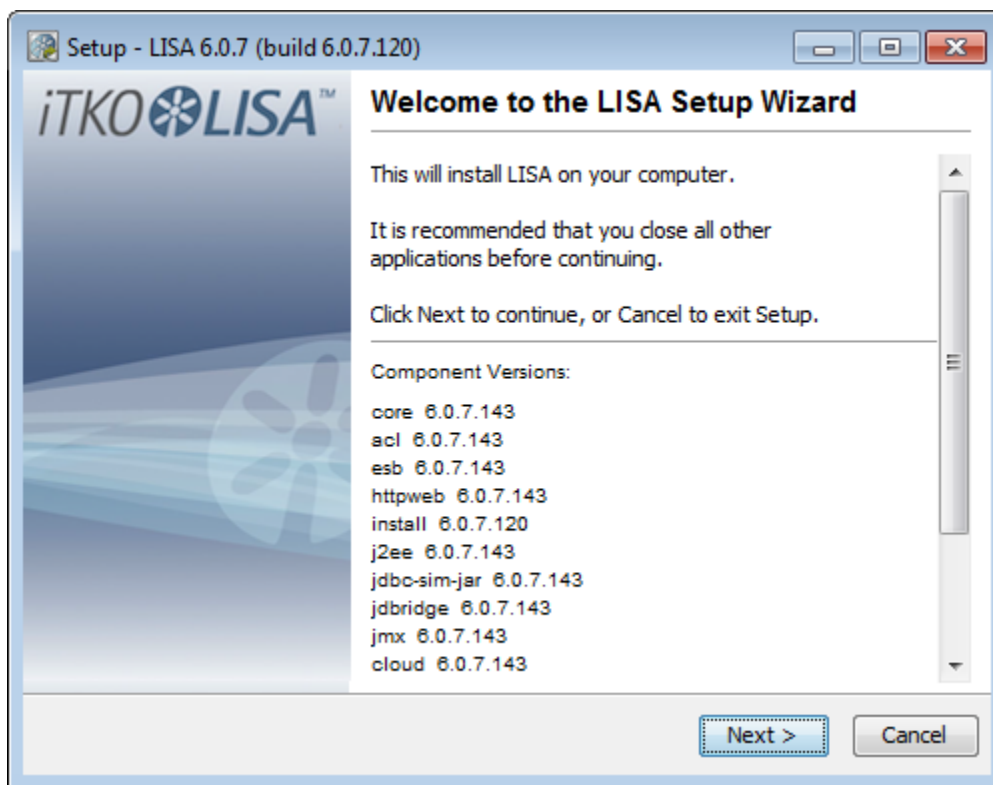
The following topics are available.

Installing LISA Server on Windows
Installing LISA Server on UNIX
Configuring License Settings for LISA Server
Using an HTTP Proxy Server with LISA Server
Starting LISA Server
Project Directory Structure
Calculating Simulator Instances
Load and Performance Server Sizing
Uninstalling LISA Server

Installing LISA Server on Windows

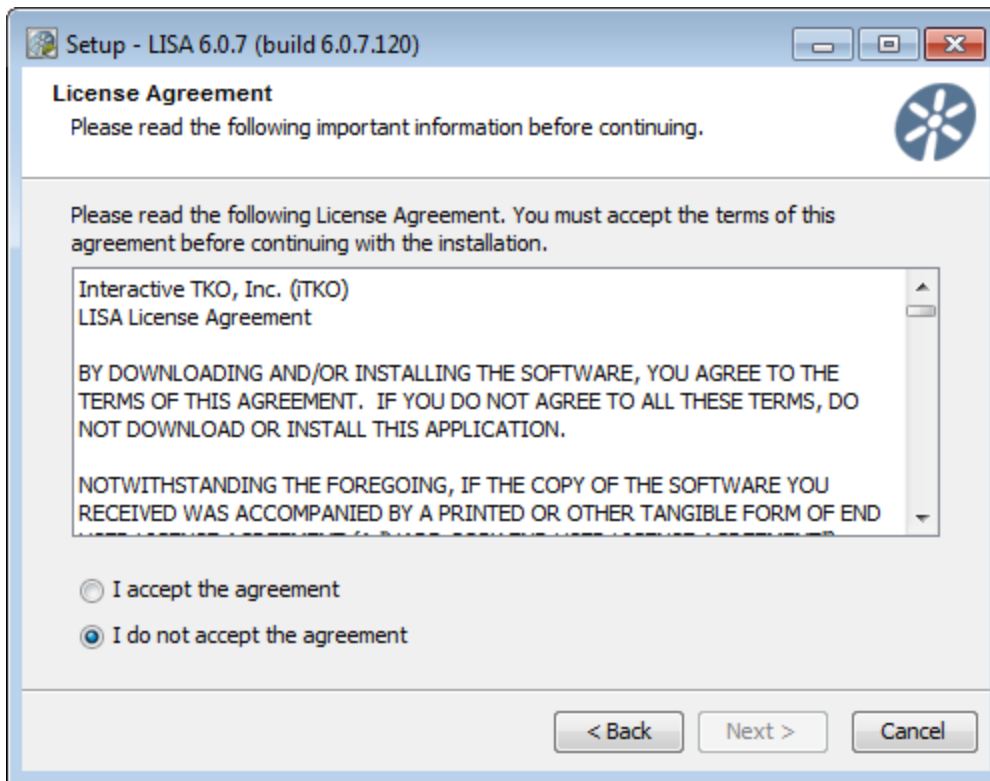
When you download and run the installer for LISA Server, the LISA Setup wizard appears.

The first step lists the components that will be installed.

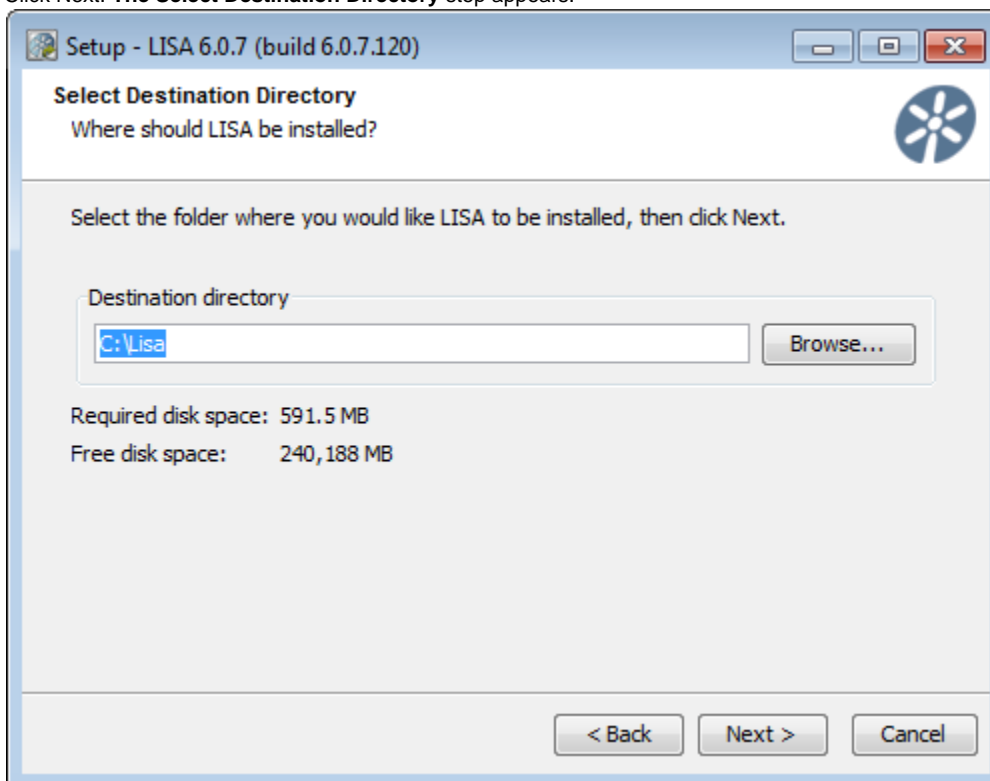


To install LISA Server on Windows

1. Click Next. The **License Agreement** step appears.



2. Read the license agreement carefully. The license agreement contains terms and conditions that you must agree to before installing.
3. When you are done, select the **I accept the agreement** check box.
4. Click Next. **The Select Destination Directory** step appears.

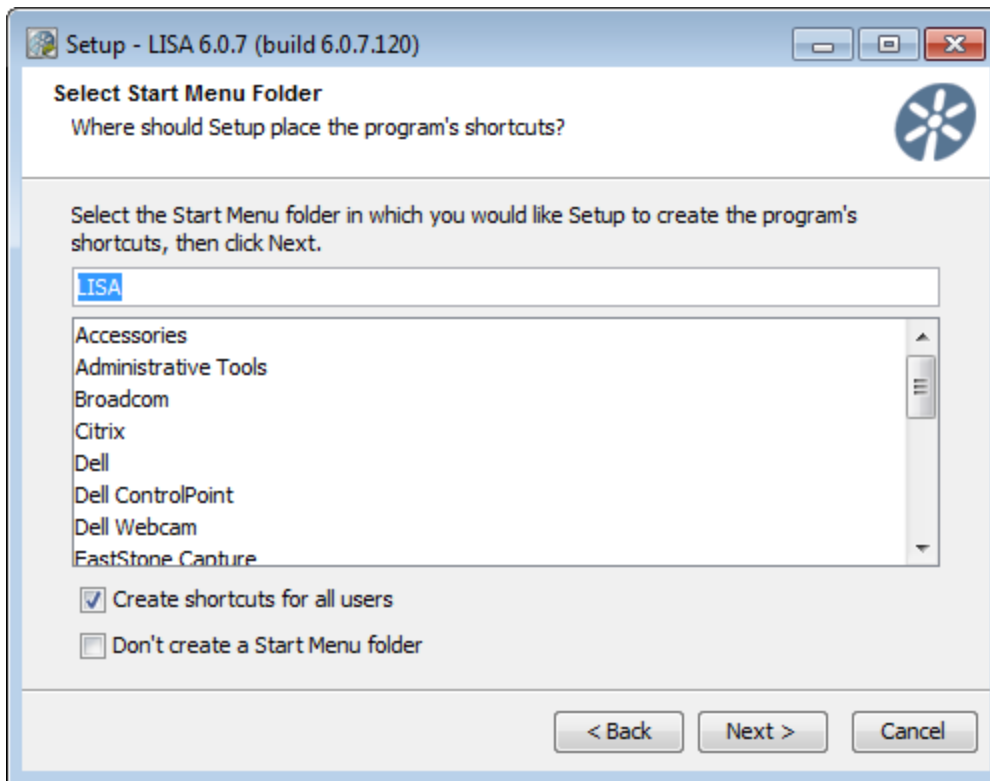


5. Specify the directory where LISA will be installed. You can use a directory that contains spaces (for example, **C:\Program Files**).

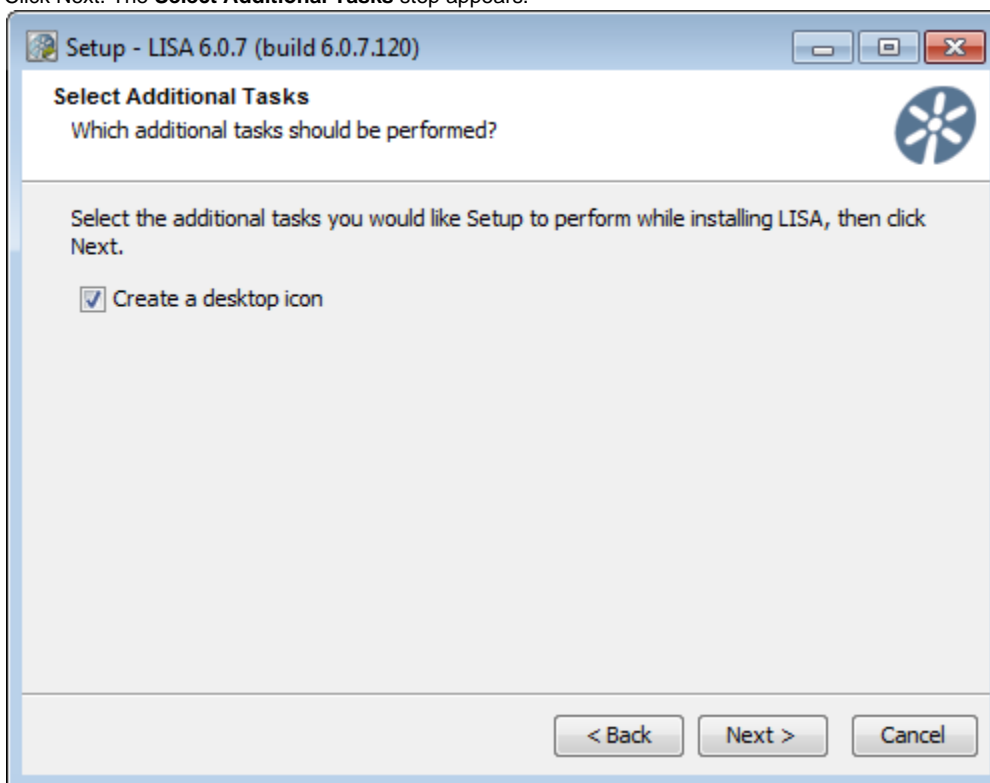


The documentation uses the term **LISA_HOME** to refer to the installation directory.

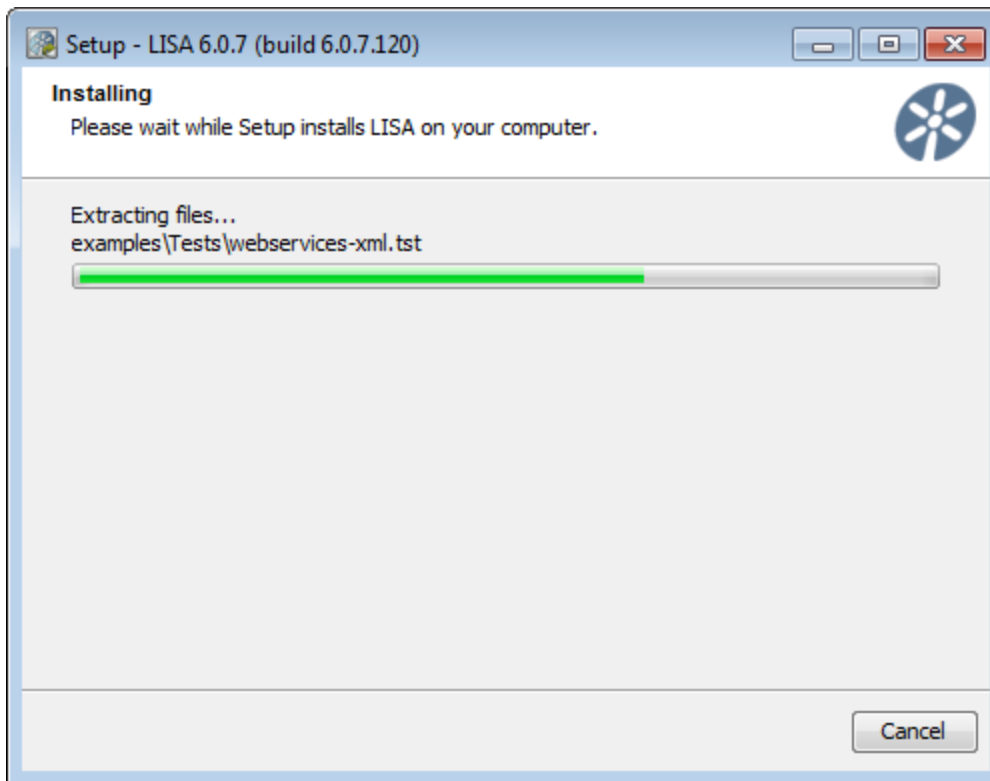
6. Click Next. **The Select Start Menu Folder** step appears.



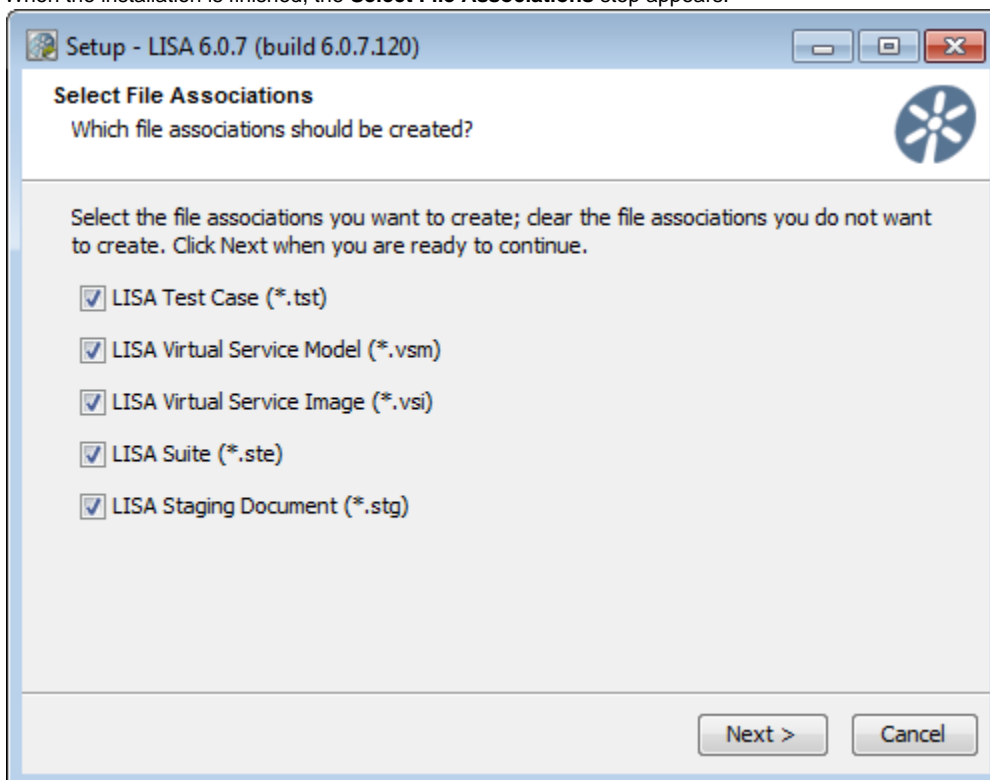
7. Enter the name of the Start menu folder. The default name is **LISA**.
8. Select **Create shortcuts for all users** to create shortcuts for all users using LISA.
9. Select **Don't create a Start Menu folder** to prevent the installer from creating a Start menu folder.
10. Click Next. The **Select Additional Tasks** step appears.



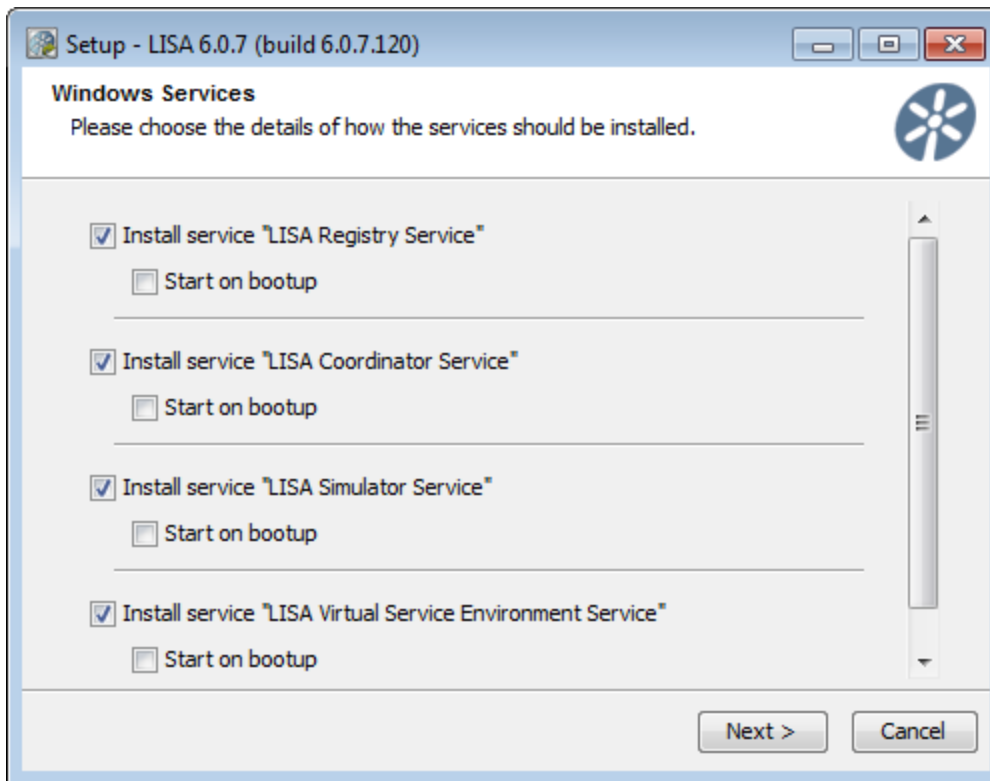
11. If you want to create a desktop icon for LISA, leave the **Create a desktop icon** check box selected.
12. Click Next to start the installation.



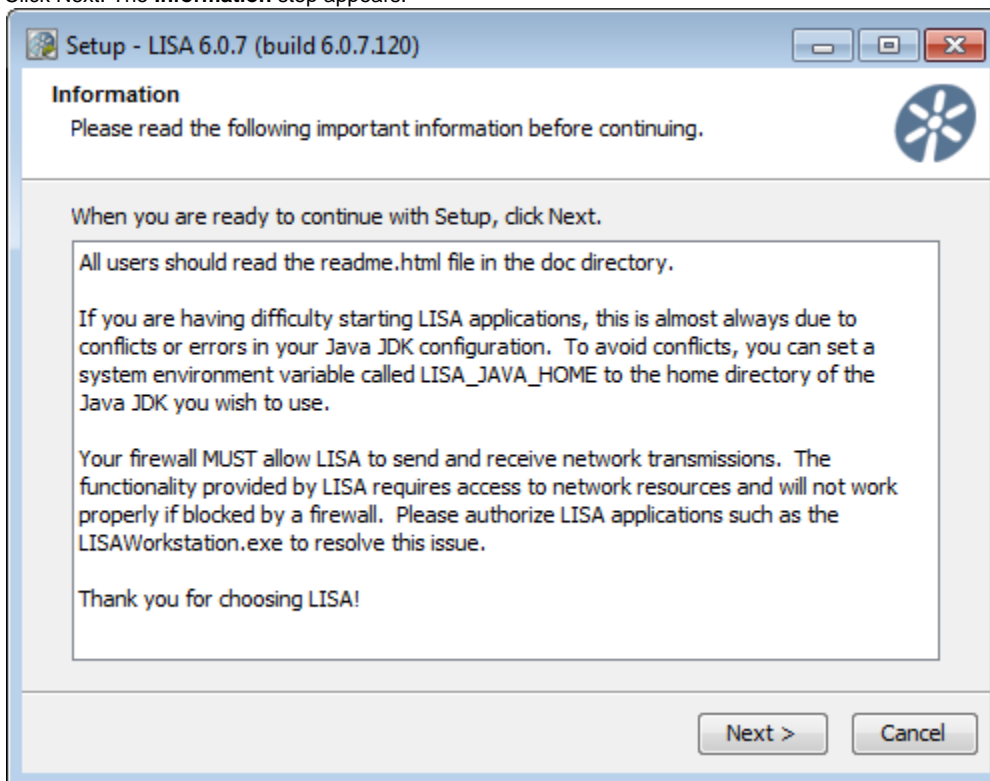
13. When the installation is finished, the **Select File Associations** step appears.



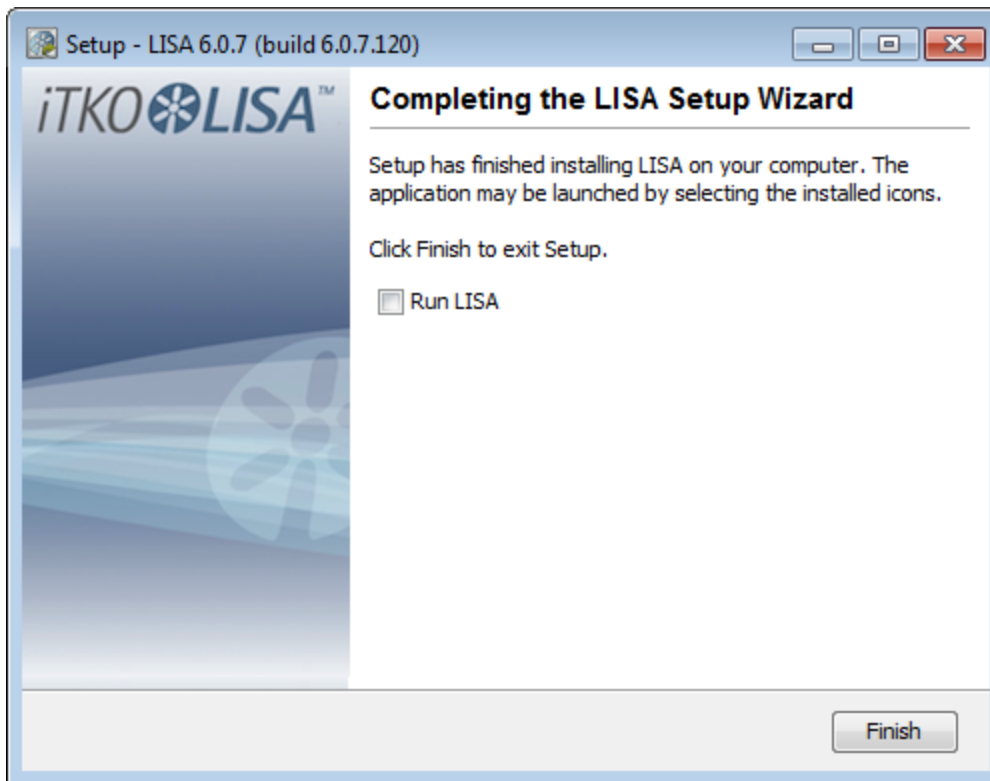
14. Select the file extensions that you to want to associate with LISA. By default, all the file extensions (.tst, .vsm, .vsi, .ste, .stg) are selected.
15. Click Next. The **Windows Services** step appears.



16. You can install any of the following server components as Windows services: Registry, Coordinator, Simulator, and Virtual Service Environment. By default, all of the components are selected. If you want a service to start on bootup, select the appropriate check box.
17. Click Next. The **Information** step appears.



18. Read the information carefully.
19. Click Next. The **Completing the LISA Setup Wizard** step appears.



20. If you want to open LISA Workstation immediately, select the Run LISA check box. LISA Workstation requires you to specify a registry. LISA Server installations include a registry. You can either start the registry locally, or use a registry that is running on a remote computer.
21. Click Finish.
You are now ready to enter your [license credentials](#).

Installing LISA Server on UNIX

This topic assumes that you have [downloaded the installer](#) from the download site.

You might need to change the permissions of the installer file so that it can be executed.

To install LISA Server on UNIX

1. In a terminal window, navigate to the directory where the installer file is located and run the installer file. For example:

```
./lisa_svr_unix.sh
```

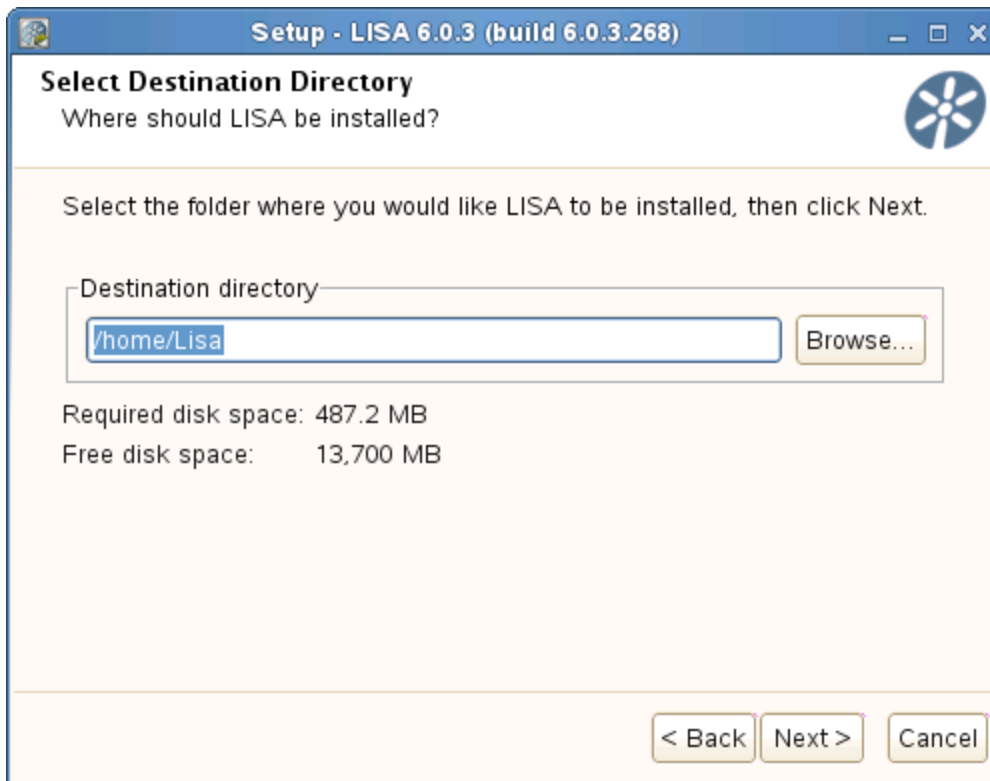
The LISA Setup wizard appears. The first step lists the components that will be installed.



2. Click Next. The **License Agreement** step appears.



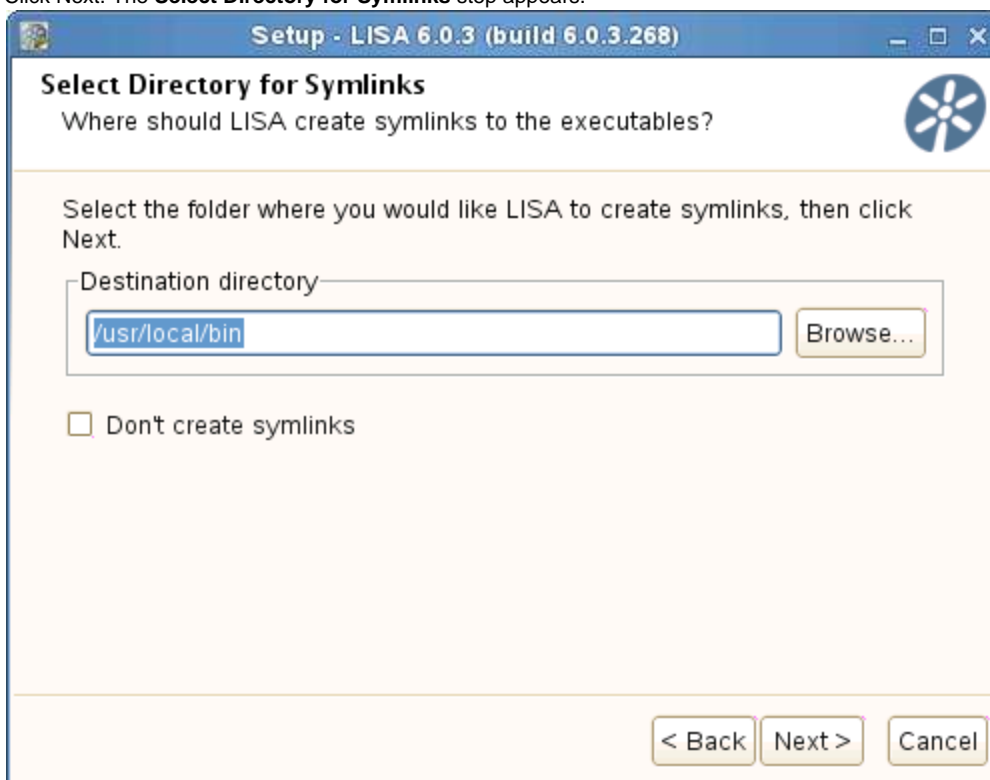
3. Read the license agreement carefully. The license agreement contains terms and conditions that you must agree to before installing.
4. When you are done, select the **I accept the agreement** check box.
5. Click Next. The **Select Destination Directory** step appears.



6. Specify the directory where LISA will be installed. Do not use a directory that contains spaces.

 The documentation uses the term **LISA_HOME** to refer to the installation directory.

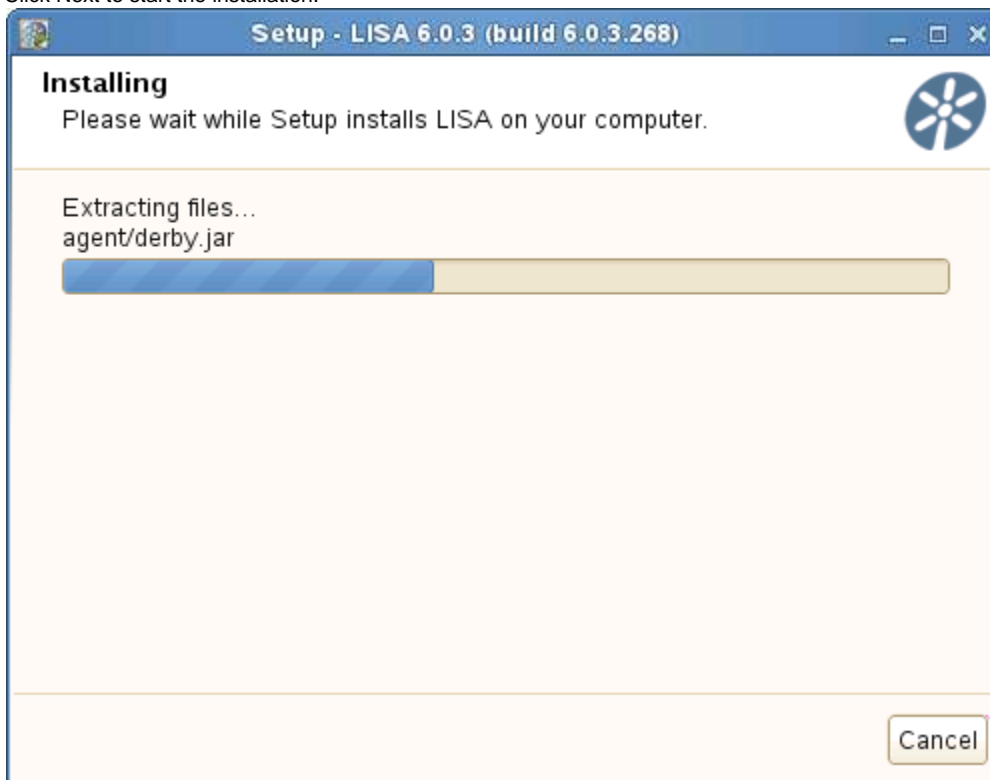
7. Click Next. The **Select Directory for Symlinks** step appears.



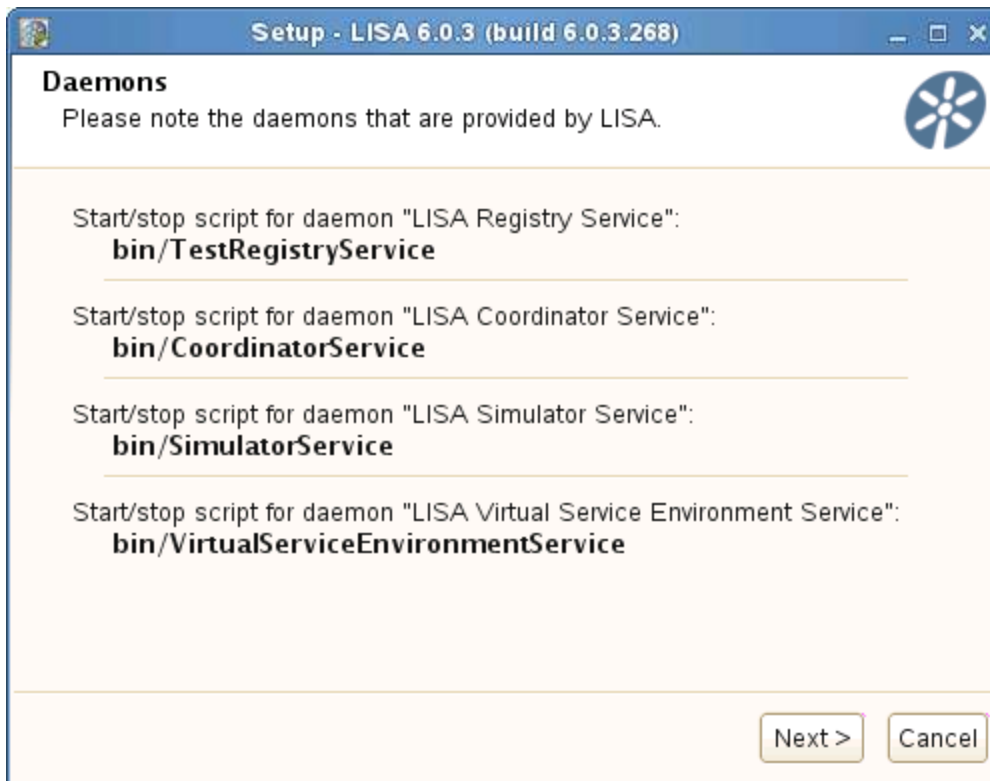
8. Specify the directory where LISA will create symbolic links to the executable files. You must have the required permissions to write to the directory. If you do not want symbolic links to be created, then select the check box.
9. Click Next. The **Select Additional Tasks** step appears.



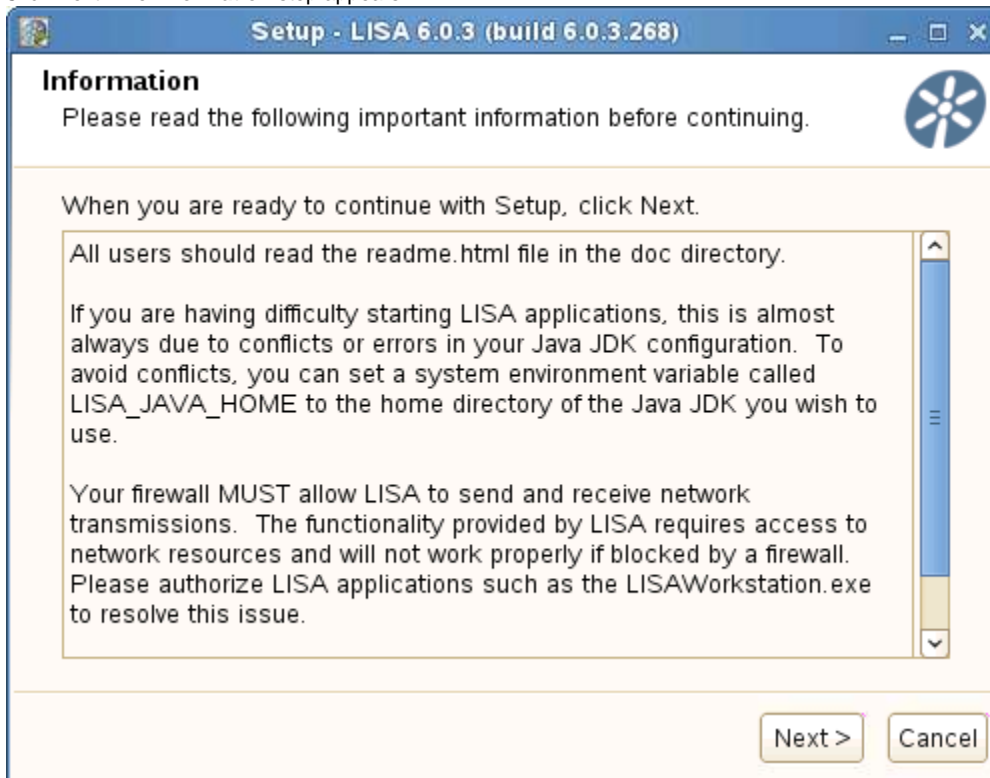
10. If you want to create a desktop icon for LISA, leave the **Create a desktop icon** check box selected.
11. Click Next to start the installation.



When the installation is finished, the **Daemons** step appears.



12. Review the daemons that were installed for the server components.
13. Click Next. The **Information** step appears.



14. Read the information carefully.
15. Click Next. The **Completing the LISA Setup Wizard** step appears.



16. If you want to open LISA Workstation immediately, select the Run LISA check box. LISA cannot start up unless you have a registry started already.
17. Click Finish.
You are now ready to enter your [license credentials](#).

Configuring License Settings for LISA Server

After LISA Server has been installed, the license information must be added to the **local.properties** file in the **LISA_HOME** directory.

To create the **local.properties** file, make a copy of the **_local.properties** file in the **LISA_HOME** directory. Change the name of the copy to **local.properties**.

The properties depend on whether your organization is using the registry-based licensing approach or the server-based licensing approach. For more information about each approach, see [Licensing Approaches](#).

Registry-Based Licensing

Open the **local.properties** file in a text editor. Add the properties from your license email.

```
laf.server.url=https://license.itko.com
laf.domain=iTKO/LISA/YOURCO
laf.group.name=YOURGROUPUSERNAME
laf.group.password=YOURGROUPPASSWORD
```

Save the **local.properties** file.

When you run LISA for the first time, the name of the **laf.group.password** property is changed to **laf.group.password.encrypt** and the password is encrypted.

Server-Based Licensing

Open the **local.properties** file in a text editor. In the license properties section, remove the comment symbols in front of the properties. Set the property values to the server URL, domain, username, and password from your license email.

```
# =====
#license properties
# =====
laf.server.url=https://license.itko.com
laf.domain=ITKO/LISA/YOURCO
laf.username=YOURUSERNAME
laf.password=YOURPASSWORD
```

Save the **local.properties** file.

When you run LISA for the first time, the name of the **laf.password** property is changed to **laf.password.encrypt** and the password is encrypted.

Using an HTTP Proxy Server with LISA Server

If you are using an HTTP proxy server, you must configure the following section of the **local.properties** file in the **LISA_HOME** directory. Be sure to remove the comment symbols in front of the properties.

```
# =====
#license properties if using an http proxy server
# =====
#laf.usehttpproxy.server=true
#laf.httpproxy.server=my_proxyserver.com
#laf.httpproxy.port=3128
# === if your proxy server requires credentials - leave blank to use native NTLM authentication
#laf.httpproxy.domain=if needed for NTLM
#laf.httpproxy.username=if needed
#laf.httpproxy.password=if needed
```

Starting LISA Server

Typically, you start the following LISA server components in the order shown:

1. Registry: For instructions, see [Registry](#).
2. Coordinator Server: For instructions, see [Coordinator Server](#).
3. Simulator Server: For instructions, see [Simulator Server](#).

After the registry has started, you can [open LISA Workstation](#).

Project Directory Structure

It is a best practice to make test assets (for example, projects) available to the server components using them.

From a server perspective, this is accomplished by setting the **lisa.projects.home** property in the **local.properties** file.

The value of the property can be a local path on the computer, or it can be a UNC path (remember to use forward slashes or escape backslashes if you must use them).

To manage access to the test assets, the requirements are:

- Use naming standards. Different teams may be using the same server environment, so having naming standards to differentiate ownership and purpose are important for maintaining order.
- Project names must be unique. On the server environment, if two deployed projects have the same name, unexpected things will happen.
- All projects must be under **lisa.projects.home**.

By default, the **lisa.projects.home** property is set to **LISA_HOME**. The server will recursively search the directory looking for projects, so the narrower you can make it, the better. For example, **lisa.projects.home=C:/** is probably not a good idea, but **lisa.projects.home=C:/lisa_projects** would be fine.

Make sure that anything you want the server to be able to run is stored under that directory.

Calculating Simulator Instances

To calculate the number of instances for a given simulator, do the following analysis:

1. Start LISA Workstation and note the memory usage from Help > LISA Runtime Info.
2. Run the test suite locally and note the memory usage from Help > LISA Runtime Info.
3. Take the difference between the memory usage in step 2 and step 1.
4. Multiply your available RAM by 60%.
5. Divide the available RAM in step 4 by memory usage in step 3.

The resulting number in step 5 is a good starting estimate of the number of virtual users (instances) that you need configured in your simulator server.

If the coordinator server and registry also run on the same server as the simulator server, then multiply available RAM by 40% in step 4 instead of 60%. (The coordinator server collects all reports and metrics and therefore consumes RAM.)

This technique will give you a starting point. Use several iterations and other intuitive methods to get to the correct number of instances per simulator.

Load and Performance Server Sizing

There is no easy way to calculate how many simulation servers are needed for a particular load test.

It will depend on many factors, including:

- Server host configuration (number of CPUs, amount of RAM)
- Test case footprint (number of test steps, type of test steps)
- Other test requirements (number of reports, size of data sets)

A recommended practice is to make several test runs of your performance test, just to collect data that can be helpful in determining the configuration of your LISA Server environment. Collecting metrics and monitoring memory usage and CPU usage will be invaluable for estimating the number of virtual users you can safely use on a given simulator server.

The registry is lightweight and requires very few computing resources. It can be run from virtually any computer in your network.

The coordinator server requires resources, so although it does not require its own computer, it is a common practice to install it on a separate computer. This is especially true if there are lots of metrics being collected and many reports requested.

Simulator servers are used to simulate thousands of virtual users. It is a recommended practice to run one simulator server per physical server. Technically, a single simulator server can be started with as many instances as you want. However, the number of instances per simulator is generally limited by server memory size and speed. A good upper limit is around 250 virtual users.

Vertical or horizontal scaling can be used for server sizing. In vertical scaling, you increase CPU speed and available memory, which are generally limited. In horizontal scaling, you add more servers. To increase the number of virtual users, horizontal scaling is recommended.

Because the number of instances per simulator depends on a lot of factors, it is not possible to use a simple rule to calculate the maximum number of instances.

Uninstalling LISA Server

The top level of the **LISA_HOME** directory includes an uninstall application. You can launch this application from the Start menu on Windows operating systems.

The uninstaller does not remove the directory where the main [log files](#) are located. After the uninstallation procedure is finished, you can remove this directory.

Configuring Third-Party Tools

This chapter describes how to configure third-party tools that can be used with LISA.

The following topics are available.

[Installing Performance Monitor \(Perfmon\)](#)
[Installing and Configuring SNMP](#)
[Running TCPMon](#)
[Installing and Running the HP Test Director Plug-in](#)
[Installing and Running the IBM Rational Quality Manager Plug-in](#)

Installing Performance Monitor (Perfmon)

Performance Monitor (Perfmon) is a utility that demonstrates monitoring the performance of the local or remote system. It demonstrates how to monitor system performance using performance counters.

To use Perfmon to monitor a Windows system's performance:

- You must have version 2.0 of the Microsoft .NET framework installed.
- From a command prompt, you must run the **setup-wperfmon.bat** file located in the **LISA_HOME/bin** directory.
- On a Vista / 2008 computer or later, the command prompt must be "Run as Administrator".

In addition, make sure the following are true:

- The user ID is the same on both computers.
- The user ID has administrator privileges on both computers.
- File and Printer sharing is turned ON.
- Simple File sharing is turned OFF.
- The default C\$ and/or ADMIN\$ shares are enabled.

Sometimes the firewall on the computer to be monitored has to be stopped.

You can verify that remote monitoring is working by selecting **Start >Control Panel >Administrative Tools >Performance** and adding a monitor to the computer that you want to observe.

LISA and Windows use the same technology to do remote monitoring. Because of this, if Windows monitoring works, then LISA monitoring should work.

To use Perfmon to gather metrics within LISA, see [Windows Perfmon Metrics](#).

Installing and Configuring SNMP

The Microsoft Windows implementation of the Simple Network Management Protocol (SNMP) is used to configure remote devices, monitor network performance, audit network usage, and detect network faults or inappropriate access.

SNMP support on Windows

Windows 2000, XP, and Vista provide an agent that is able to answer SNMP requests and send Traps.

[Installing Microsoft SNMP Agent](#)
[Configuring Microsoft SNMP Agent](#)

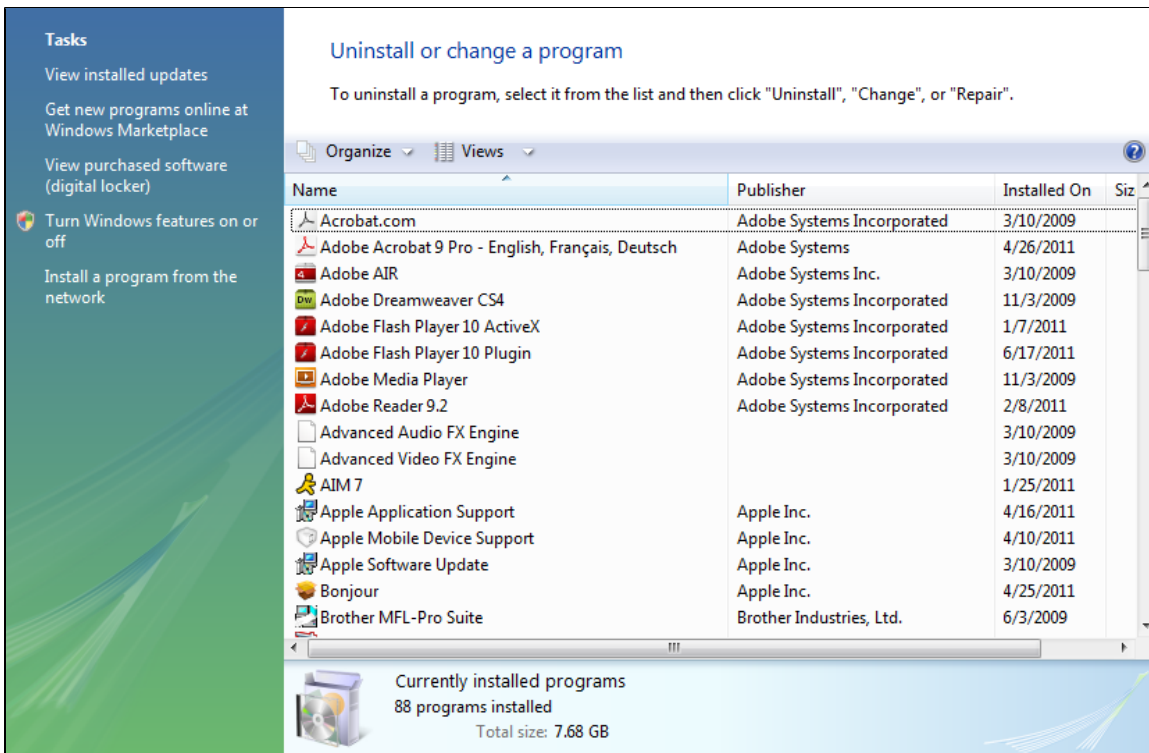
SNMP support on UNIX

This is available from your operating system vendor, or you can try the Net-SNMP open source SNMP package. See its accompanying documentation for installation and configuration directions.

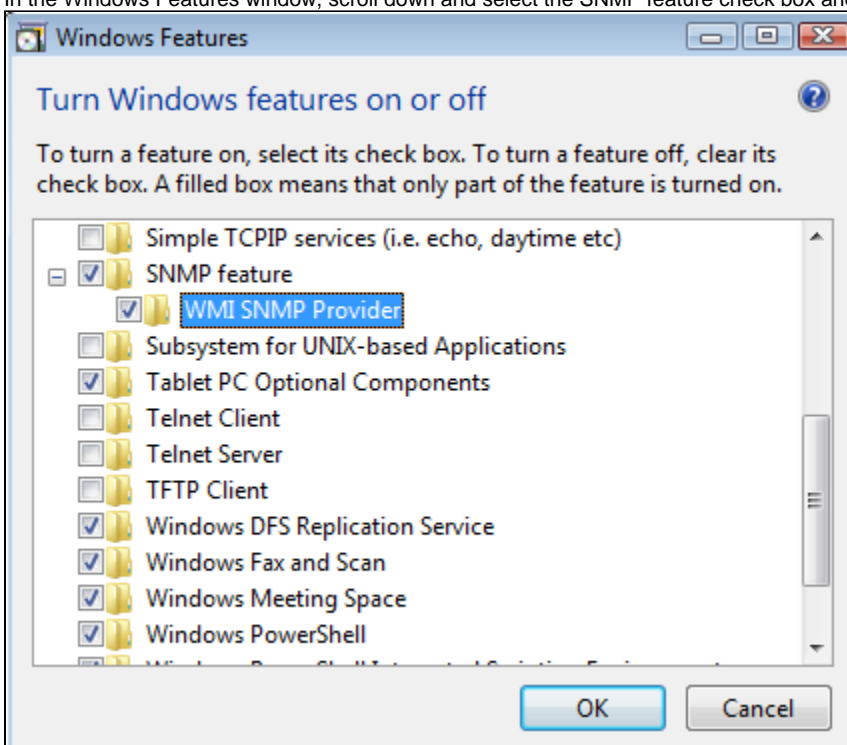
Installing Microsoft SNMP Agent

Installing Microsoft SNMP Agent on Windows Vista and Windows 7

1. Open the Windows **Control Panel** and double-click Programs and Features.
2. The **Uninstall or change a program** window appears.



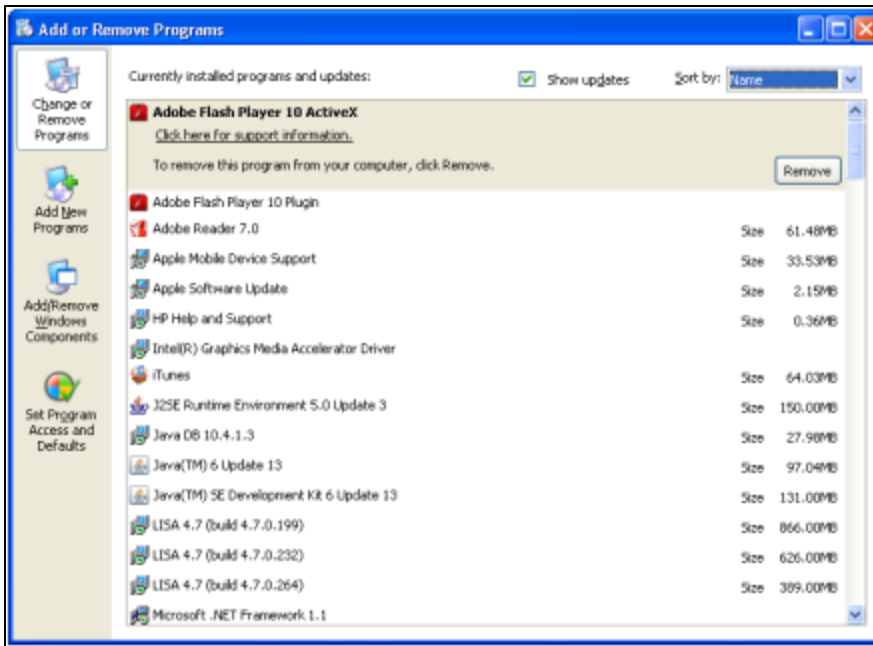
3. Click **Turn Windows features on or off** in the left pane. If UAC prompted, then click Yes.
4. In the Windows Features window, scroll down and select the SNMP feature check box and click OK.



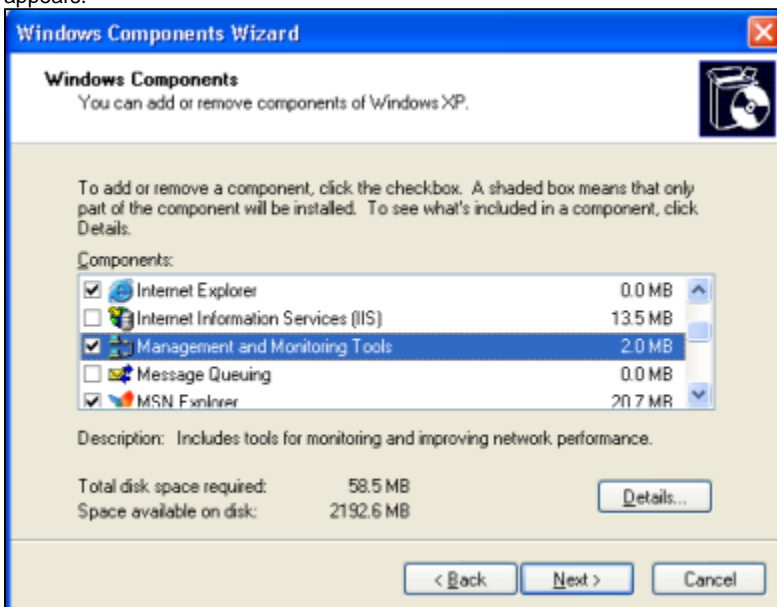
5. Wait for some time for SNMP to be installed.

Installing Microsoft SNMP Agent on earlier Windows versions

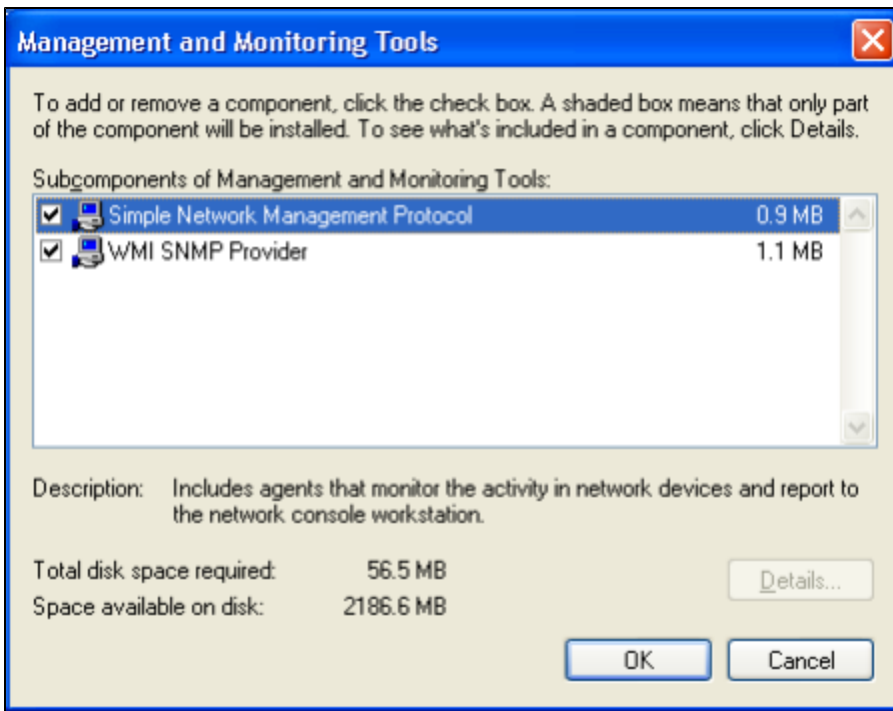
1. Open the Windows **Control Panel** and double-click the Add or Remove Programs icon.
2. The **Add or Remove Programs** window appears.



3. Click the **Add/Remove Windows Components** icon on the left side of the window. The Windows Component wizard appears.
4. Select **Management and Monitoring Tools** in the **Components** list and click Details. The Management and Monitoring Tools window appears.



5. Select **Simple Network Management Protocol** from the **Subcomponents of Management and Monitoring Tools** list and click OK.

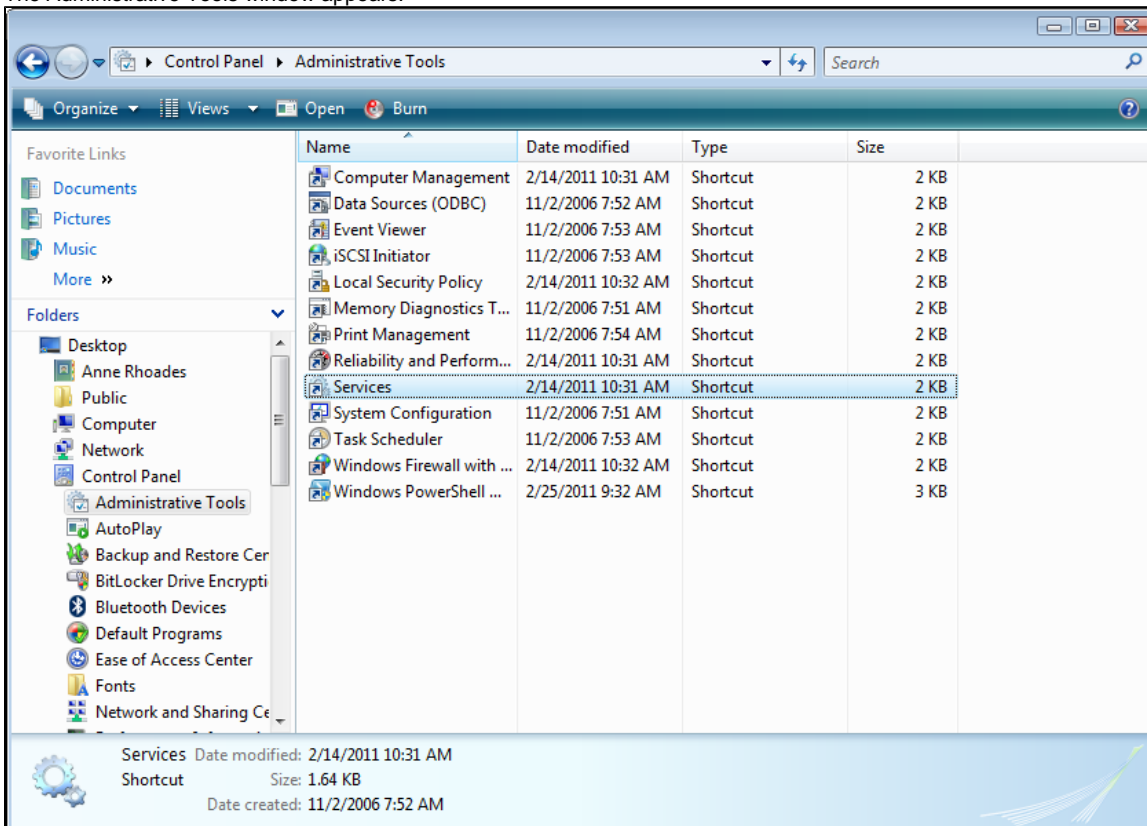


6. Click Next. The Windows Components Wizard installs the Microsoft SNMP agent.
7. When complete, click Finish.

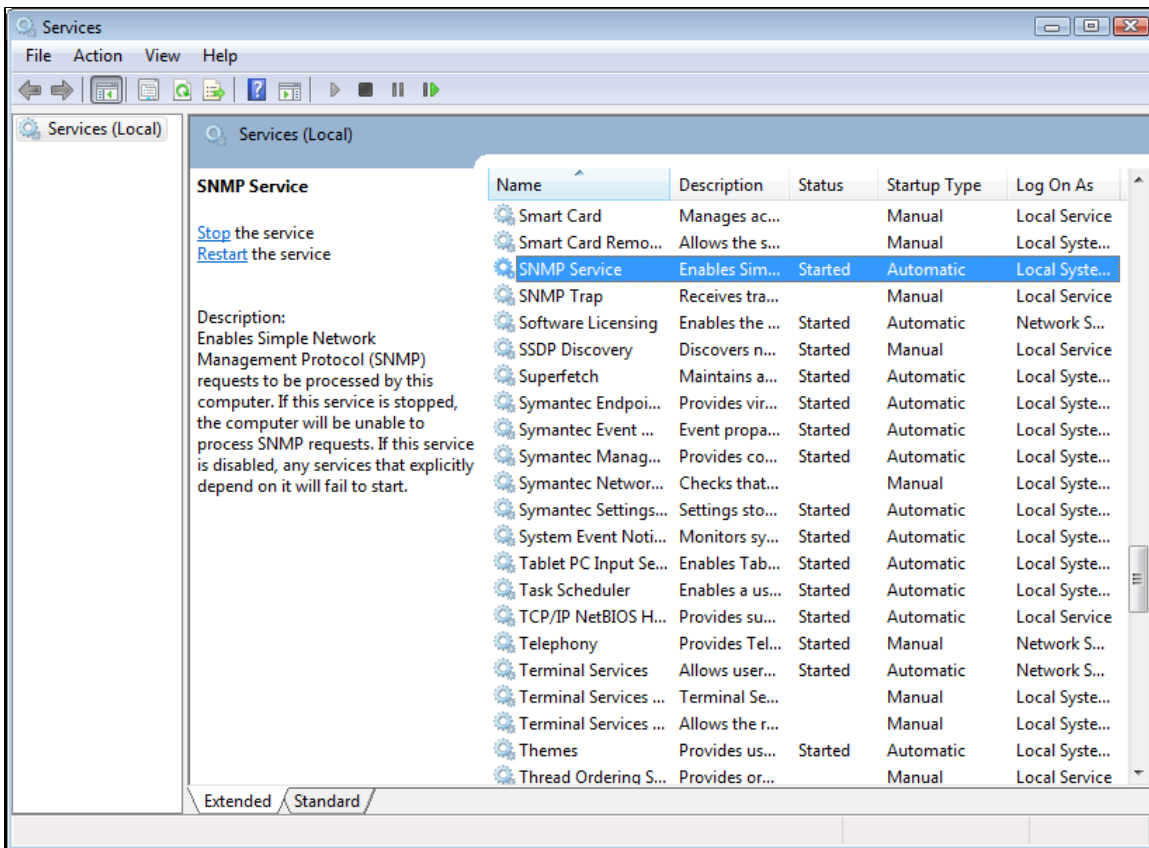
Configuring Microsoft SNMP Agent

To configure the Microsoft SNMP Agent

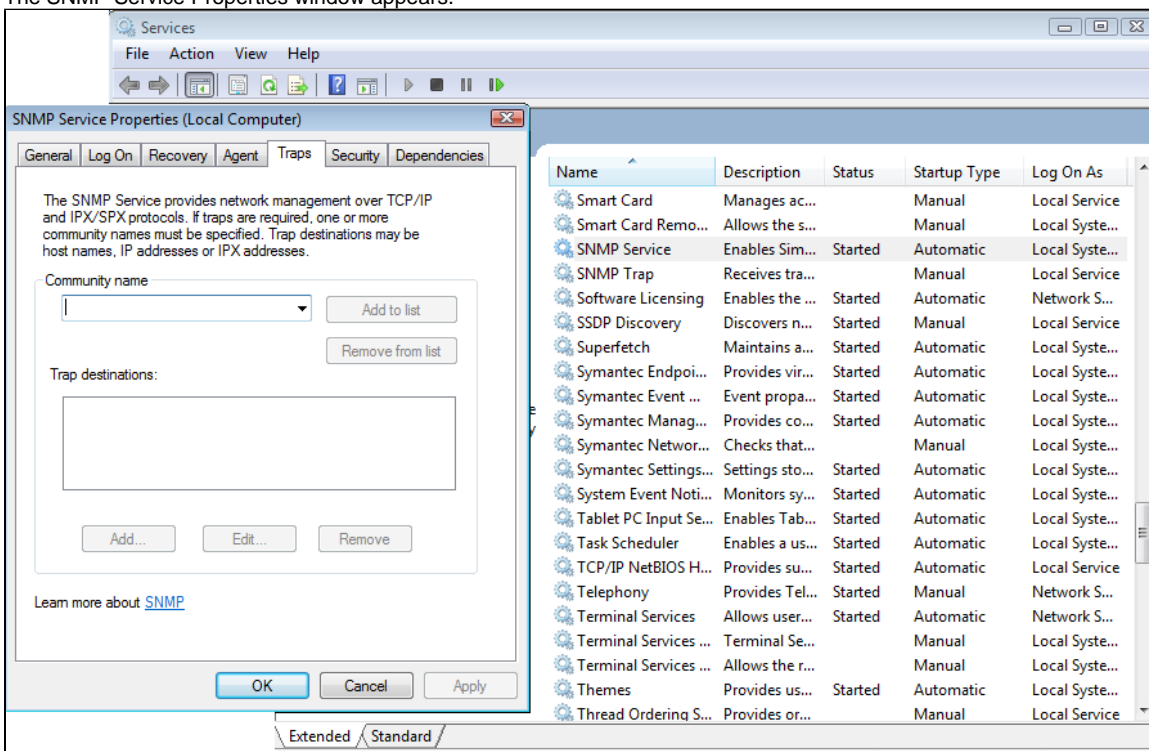
1. Open the Windows **Control Panel display** and double-click the Administrative Tools icon. The Administrative Tools window appears.



2. From the **Administrative Tools** window, double-click the Services icon. The Services window appears.



3. In the **Services** window, double-click the SNMP service. The SNMP Service Properties window appears.



4. In the **General** tab of the SNMP Service Properties window, change **Startup Type** to **Automatic**. This configures the SNMP service to start the Microsoft SNMP agent on system startup.
 5. Click the Traps tab.
 6. In the **Community Name** box, type the community name to which your computer will send trap messages and click the Add to list button.
 7. Click Apply and OK.
 8. Click OK.
- To use Windows SNMP to gather metrics, see the [User Guide](#).

Running TCPMon

TCPMon is a utility that lets you monitor the messages passed in a TCP-based conversation.

TCPMon consists of:

- **For Windows:** A .jar file, a .bat file
- **For UNIX:** A shell script

To run TCPMon

Double-click the **.bat** file on Windows or execute the shell script on UNIX.

There is a **tcpmon.bat** file in the **LISA_HOME/bin** directory. You can get the latest version of TCPMon from:
<http://ws.apache.org/commons/tcpmon/>.



This section documents the TCPMon version from Apache. It contains a **Sender** tab that is not available in the TCPMon version currently distributed with LISA.

Using TCPMon as an Explicit Intermediate
Using TCPMon as a Request Sender for Web Services

Using TCPMon as an Explicit Intermediate

The most common usage pattern for the TCPMon is as an intermediary. It is specified as *explicit* because the client has to point to the intermediary, rather than the original endpoint, to monitor the messages.



To start the TCPMon in this configuration, you must provide the listen port, the host name, and the port for the listener in the **Admin** tab.

TCPMonitor

Admin

Create a new TCP/IP Monitor...

Listen Port #

Act as a...

☒ Listener

Target Hostname

Target Port #

☐ Proxy

Options

☐ HTTP Proxy Support

Hostname

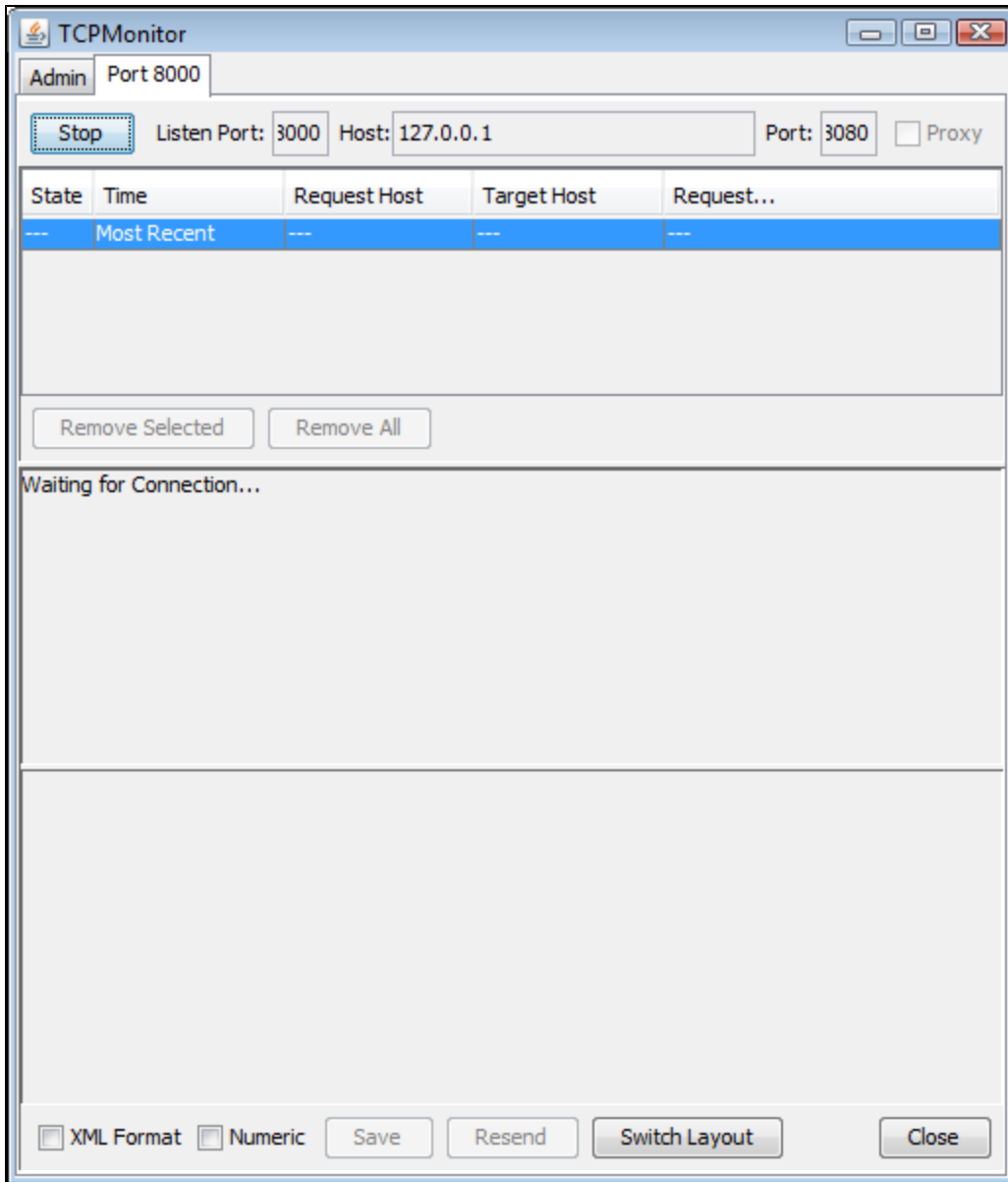
Port #

☐ Simulate Slow Connection

Bytes per Pause

Delay in Milliseconds

Click Add to open up a new tab (Port 8000) that allows the messages to be seen.



Requests should now point to the listener port of the TCPMon instead of the original endpoint.

As shown, all messages passed to and from localhost:8080 will be monitored.

- We set the listener to port 8000 which may be any unused port in the local computer.
- We added a listener with host as **localhost** and port as **8080**.
- Point the browser to **localhost:8000** instead of localhost:8080.

Using TCPMon as a Request Sender for Web Services

TCPMon can also be used as a request sender for web services.

- The request SOAP message can be pasted into the Sender screen and then sent directly to the server.
- The web service endpoint is entered in the connection Endpoint text box.

For more information about TCPMon, see [TCPMon Tutorial](#).

Installing and Running the HP Test Director Plug-in

HP Test Director for Quality Center™ is a single, web-based application for all essential aspects of test management: requirements management, test plan, test lab, and defect management.



The latest version is named HP Quality Center.

ITKO provides an HP Test Director for Quality Center plug-in. You can download the plug-in from the [LISA download site](#).

The plug-in lets you load and run LISA test cases as Quality Center tests from within the HP Quality Center suite. You can import into and run LISA tests from Quality Center so you can take advantage of all Quality Center features while harnessing the power of LISA testing. By loading a LISA test case into Quality Center, you get real-time execution of LISA tests, with full capture of the test results and LISA callbacks returning from any system under test. LISA tests are executable within the workflow of Quality Center, and they report back results to maintain the context and status of the testing process.

When you install the QC Plugin on a machine that has LISA installed, it will add two files to the LISA install directory: **LisaQCRunner.exe** and **LisaQCGLue.dll**. **LisaQCRunner.exe** is the COM server that gets registered as part of the install process. LISA is invoked using QC's VAPI-XP interface. The VAPI-XP script creates an instance of our COM object. That in turn stages the test to be executed and listens for the test results. Finally, the COM object takes the results and updates the QC instance with the results from LISA.

This topic contains the following sections:

- [System Requirements](#)
- [Installation](#)
- [Usage](#)
- [Command Line Interface](#)
- [Troubleshooting](#)

System Requirements

The following software must be installed:

- LISA 6.0
- LISA HP Test Director for Quality Center Plug-in
- HP Quality Center 9.2, 10, or 11
- HP Quality Center Client Side Components
- .NET 2.0 Runtime

Installation

To install the plug-in

1. Ensure that you have the .NET runtime installed on the client computer.
2. Install a supported version of LISA, and the LISA QC plug-in (both are available on the LISA download site).
3. Install the HP Quality Center Client Side Components.

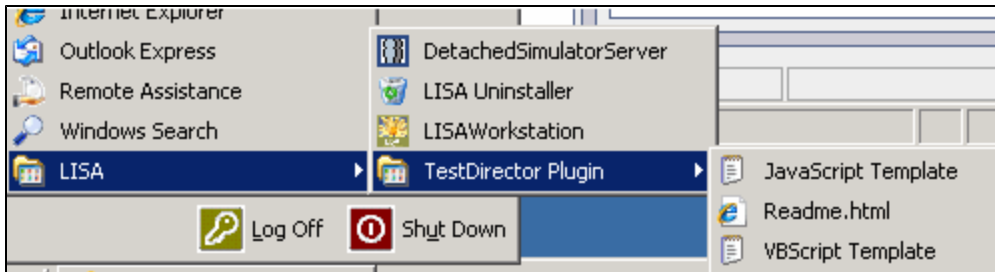
Usage

The plug-in uses VAPI-XP to integrate with Quality Center.

To use LISA with Quality Center, create VAPI-XP tests like you typically would in Quality Center.

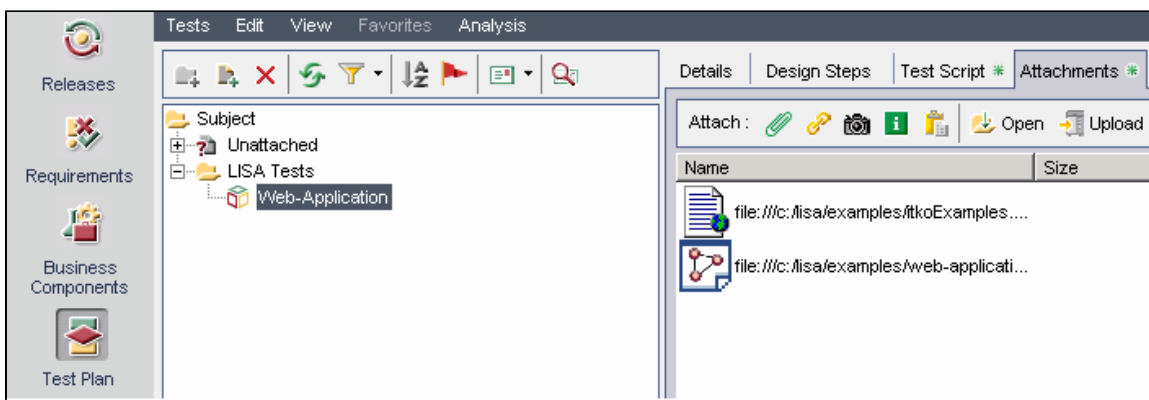
The screenshot shows a standard Windows-style dialog box titled "Create New Test". It has a close button (X) in the top right corner. Inside the dialog, there are two labels: "Test Type:" and "Test Name:". The "Test Type:" label is followed by a dropdown menu that currently displays "VAPI-XP-TEST". The "Test Name:" label is followed by an empty text input field. At the bottom of the dialog, there are three buttons: "OK", "Cancel", and "Help".

After the test is created, select the Test Script tab of the newly created test and replace the default contents with the contents of the appropriate script template located at Start > Programs > LISA > TestDirector Plugin > <<Script Template>>



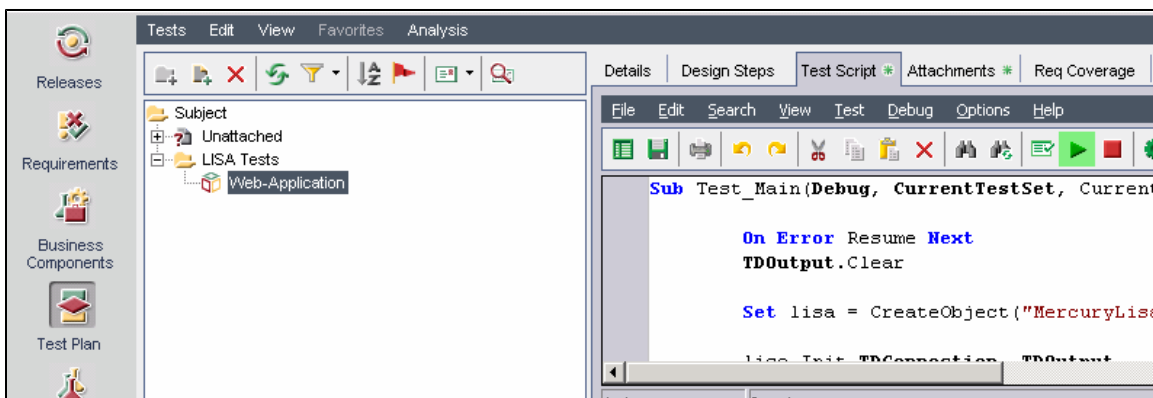
After the script template has been added, the final step is to add references to the test, staging (5.5) and config files in the attachments tab. It is recommended that the test, staging (5.5) and config files be added as links rather than attaching the actual file itself. This approach lets any changes to the LISA test to not have to be uploaded back to Quality Center. To do this, select the Link icon and then file a properly constructed URL to the file. If the file is on the same computer as the QC instance, it will be in a form similar to this:

```
file:/// <drive>:/<path>/?<to>/<file>.<ext>
```

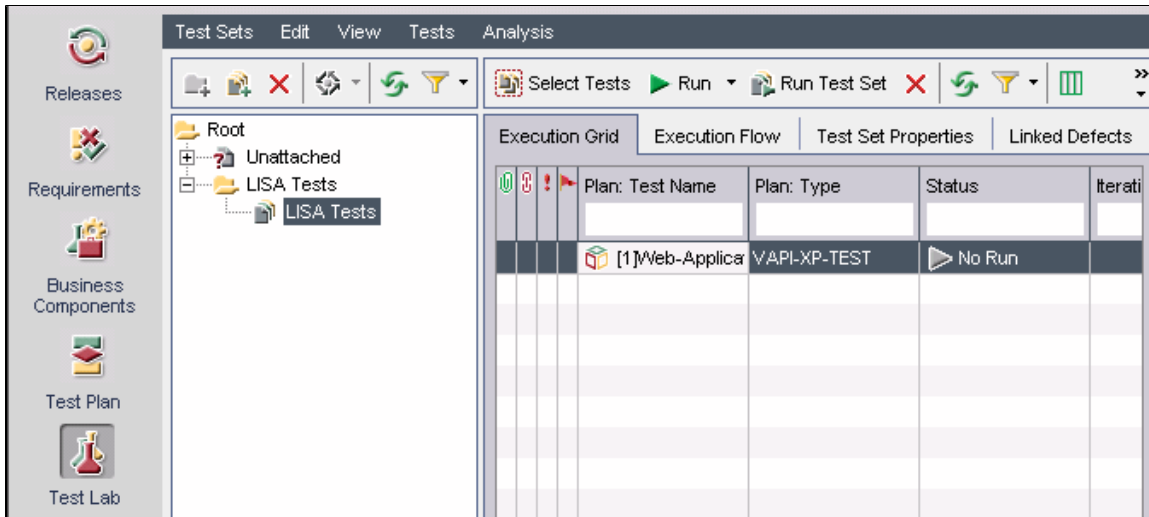


The test can now be run from the Test Plan debug window or the Test Lab's test suite. The first command that needs to be run is Reload. This command populates the test's design steps which are necessary for a successful test run. By default, the Reload command is set to run with each test run. If the file is linked (recommended), then any changes to the test are updated before the test is run. For some tests, the reload process can take some time. If you know that the underlying LISA test file has not changed, then you can comment out that line from the script file and the step will be skipped.

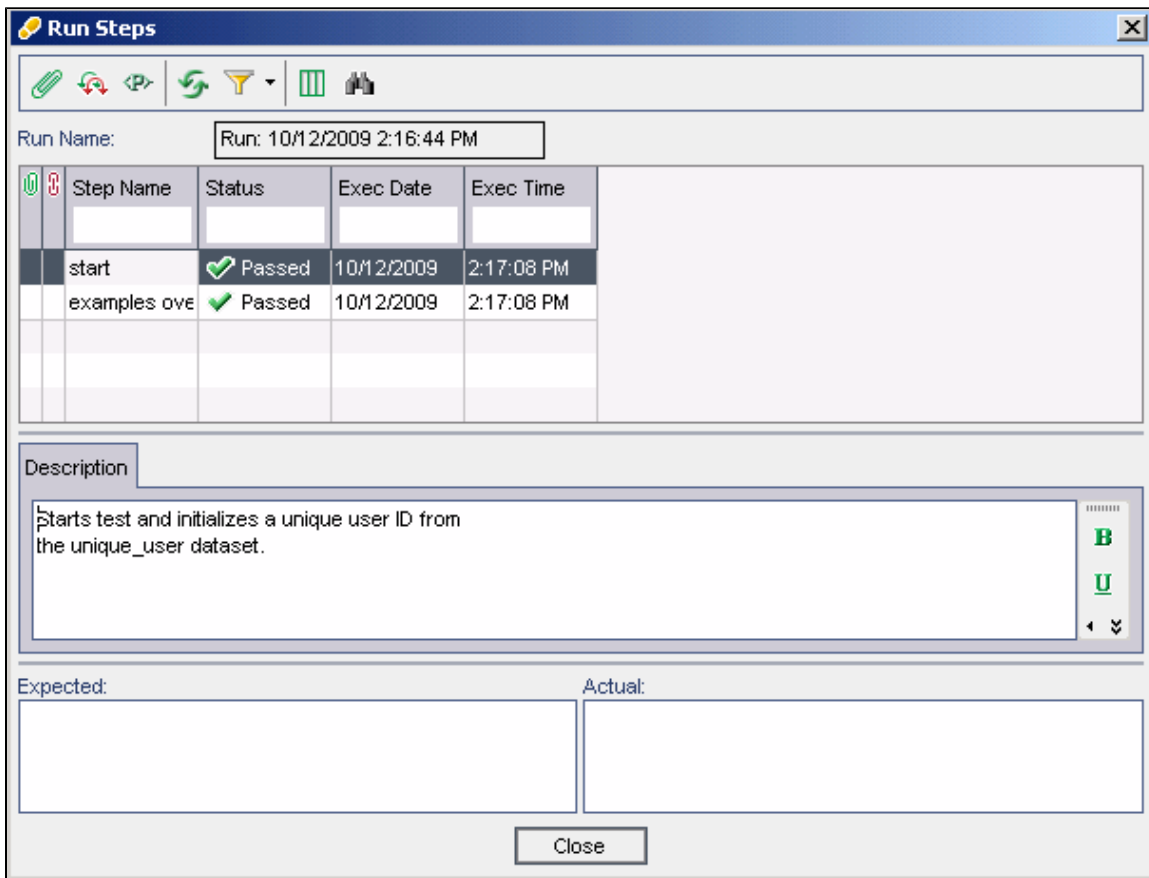
To run the test from the Test Plan (debug mode), locate the test and click the Test Script tab. From there you can click the green arrow. The test will start executing, and its output will appear in the output window. You will be notified when the test has completed.



To run the test from the Test Lab, add the test to a valid test set. From there you can run the single test or the entire test set. After the test has completed, you can check the status of the test from the Test Set window or check the history of the test in the Test Instance Properties window.



Depending on the structure of the test, clicking a test run will show different results. If the test has more than one cycle that is executed, then you will see a list of cycle history results and its pass or fail status. If the test has only one cycle, then you will see a list of steps for that test.



You can now run LISA tests from within the QC interface.

Command-Line Interface

The plug-in lets you debug or run Quality Center LISA tests from the command line and persist results in the Quality Center database for later viewing.

Use the **LisaQCRRunner** executable in the **LISA_HOME/bin** directory.

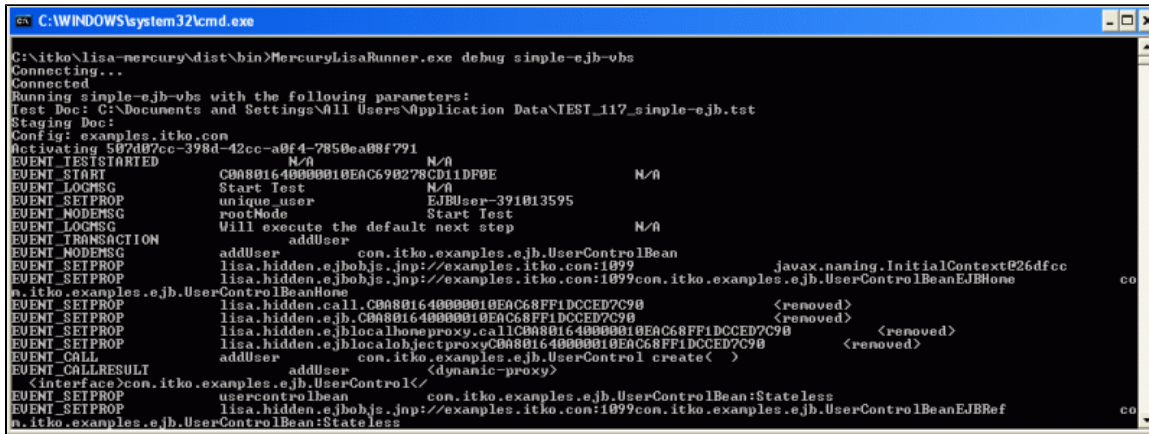
```
LisaQCRRunner [-h <host>][--P <port>][--u <user>][--p <password>][--D <domain>][--l <project>]
<run|debug|reload> <test name|all>
```

The default host is localhost. The default port is 8080. The default user is admin. The default password is admin. The default domain is DEFAULT. The default project is **QualityCenter_Demo**.

The **run** command runs the test name that you specify as an argument.

The **debug** command runs the test name that you specify as an argument in debug mode.

The **reload** command reloads either the test name that you specify as an argument, or all LISA tests (if the argument is **all**).



```
C:\WINDOWS\system32\cmd.exe
C:\itko\lisa-mercury\dist\bin>MercuryLisaRunner.exe debug simple-ejb-vbs
Connecting...
Connected
Running simple-ejb-vbs with the following parameters:
Test Doc: C:\Documents and Settings\All Users\Application Data\IESI_117_simple-ejb.tst
Staging Doc:
Config: examples.itko.com
Activating 507d07cc-398d-42cc-a0f4-7850ea08f791
EVENT_TESTSTARTED N/A N/A N/A
EVENT_START C0A801640000010EAC690278CD11DF0E N/A
EVENT_LOCMGMSG Start Test N/A
EVENT_SETPROP unique_user EJBUser-391013595
EVENT_MODEMSG rootNode Start Test
EVENT_LOCMGMSG Will execute the default next step N/A
EVENT_TRANSACTION addUser con.itko.examples.ejb.UserControlBean
EVENT_SETPROP lisa.hidden.ejbbjbs.jsp://examples.itko.com:1099 javax.naming.InitialContext@26dfcc
EVENT_SETPROP lisa.hidden.ejbbjbs.jsp://examples.itko.com:1099con.itko.examples.ejb.UserControlBeanEJBHome co
n.itko.examples.ejb.UserControlBeanHome
EVENT_SETPROP lisa.hidden.call.C0A801640000010EAC68FF1DCCED7C9B <removed>
EVENT_SETPROP lisa.hidden.ejb.C0A801640000010EAC68FF1DCCED7C9B <removed>
EVENT_SETPROP lisa.hidden.ejblocalhomeproxy.callC0A801640000010EAC68FF1DCCED7C9B <removed>
EVENT_SETPROP lisa.hidden.ejblocalhomeproxyC0A801640000010EAC68FF1DCCED7C9B <removed>
EVENT_CALL addUser con.itko.examples.ejb.UserControl create( )
EVENT_CALLRESULT addUser <dynamic-proxy>
EVENT_SETPROP <Interface>con.itko.examples.ejb.UserControl</
EVENT_SETPROP usercontrolbean con.itko.examples.ejb.UserControlBean:Stateless
EVENT_SETPROP lisa.hidden.ejbbjbs.jsp://examples.itko.com:1099con.itko.examples.ejb.UserControlBeanEJBRef co
n.itko.examples.ejb.UserControlBean:Stateless
```

Troubleshooting

If you have trouble determining the cause of a test failure, the full log of a given run is available at **<Documents and Settings>\All Users\Application Data\<Test name>** with some Perfmon counters.

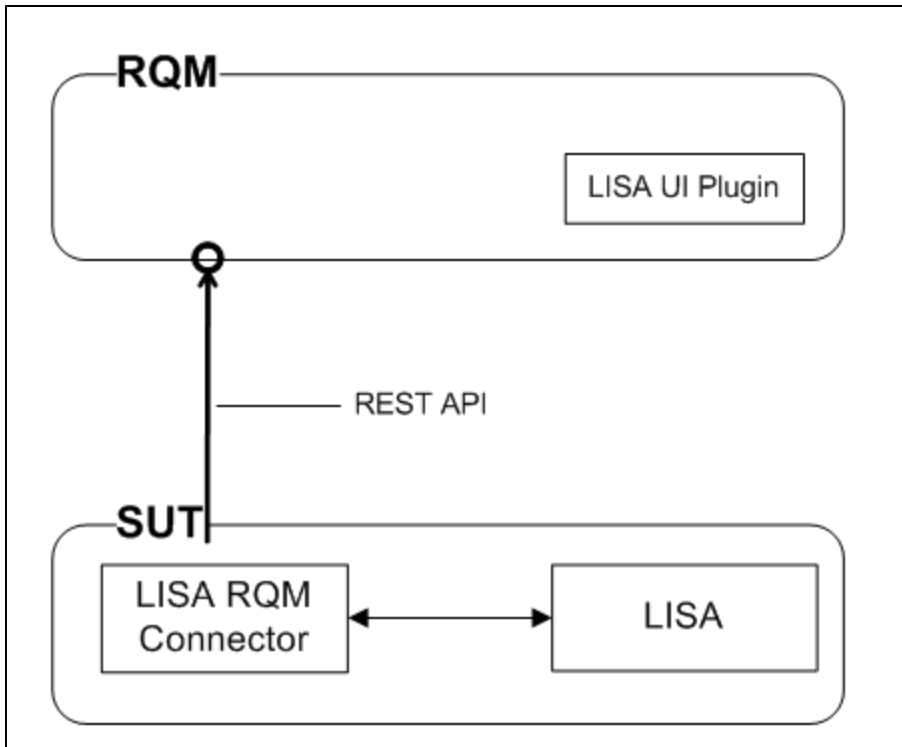
To improve performance, the LISA bridge always keeps a reference to the LISA COM server so that it does not need to be instantiated for each API call. When the process hosting the bridge terminates, this reference is released unless the host is a native app like a web browser so the **LisaQCRRunner.exe** process will stay alive. This is normally not a problem unless something gets in a bad state (for example, because of an abrupt termination). In that case, consider manually terminating the lingering **LisaQCRRunner.exe** process before proceeding.

Installing and Running the IBM Rational Quality Manager Plug-in

IBM Rational Quality Manager is a web-based centralized test management environment for test planning, workflow control, tracking and metrics reporting. Rational Quality Manager is extensible in a number of ways including the ability to use "connectors" or plug-ins to bridge between RQM and external systems. The ITKO LISA RQM plug-in lets you re-use or create LISA test cases and tie them into RQM. Then the LISA test cases can be executed from the RQM interface and results from tests runs are aggregated into RQM's test execution and reporting history.

Implementation

The complete LISA RQM solution is implemented as two components; the ITKO LISA RQM Connector and the web UI extension. The web UI allows for the extension of the out of the box UI and allows for the necessary parameters to be passed and displayed between LISA and RQM. The LISA RQM Connector is used to respond to work tasks scheduled by RQM. The task information collected in the UI is passed to the connector and the connector takes care of invoking TestRunner with the necessary parameters.



The connector takes advantage of TestRunner's -html switch and uses it to generate the HTML output that is later uploaded back to RQM. After the test is run through to completion, the results of the test are uploaded to RQM and associated with the test run. The user can then inspect the results through the RQM interface.

Installing The LISA Adapter UI

1. From the <LISA_HOME>\addons directory locate the zip file named **rqm-adapter.zip**.
2. Unpack the file and locate the file **com.itko.lisa.integration.ibm.rqm.update.site.zip** in the folder **lisa-adapter-ui**.
3. On the RQM server, unzip the file **com.itko.lisa.integration.ibm.rqm.update.site.zip**.
4. Open the file **com.itko.lisa.integration.ibm.rqm.adapter.web.update.ini** and update the reference to point to the location of the update site.
5. Copy the .ini file you just modified to the following location: <RQM install root>/server/server/conf/jazz/provision_profiles/.
6. Use the Rational Quality Manager Server Reset service to update the server configuration. The reset utility is found at the following URL:
 - a. [https://<hostname>:<portnumber>/jazz/admin?internal#action=com.ibm.team.repository.admin.serverReset|https://%3choostname
7. The service will prompt you to log in to the IBM Rational Quality Manager Admin console. The user ID must have administrator privileges.
8. Stop the RQM server.
9. Restart the Rational Quality Manager server.

Running The Command Line Adapter

1. From the <LISA_HOME>\addons directory locate the zip file named **rqm-adapter.zip**.
2. Unpack the file and locate the child folder **lisa-adapter2.0** in the folder.
3. From a command line, start the adapter. The parameters needed are as follows:
 - -repository [https://<rqmserver>:<port>/jazz|https://%3crqmserver%3e%3cport%3e]
 - [https://%3crqmserver%3e%3cport%3e]-user <userid>
 - -password <password>
 - -adapter <adapter>(com.itko.lisa.integration.ibm.rqm)
 - -adapterName <adapterName> (LISA RQM Adapter)## -LISA_HOME <lisaHomePath> (d:\lisa)
 - [-projectArea <project area> (Quality%20Manager)]
 - [-sleepTime <sleep time>(5)]

General Usage Workflow

1. Create a test script in RQM.
2. Associate a LISA test case, staging document and optional config document with the test script.
3. Associate one or more test scripts with a test case, and the test case is ready to run.
4. Select to run the test case, select a running adapter and select OK.
5. The server contacts the SUT through the RQM Connector plug-in, which in turn invokes LISA.
6. Wait for the test to run, then check the run results in RQM.

RQM-LISA Test Script Setup

Test Script: 1: LISA Simple Test - Quality Manage...

Quality Management (Jazz) Project1

Requirements Planning Construction Lab Management Builds Execution Reports Defects

Quality Dashboard View Test Scripts 1: LISA Simple Test

1: LISA Simple Test
Test Script Overview | Snapshots | History

Originator: itko Owner: itko Action: Select Action State: Draft Work Item: Create

Type: ITKO Lisa Adapter

Description: < Click here to enter a description >

Summary

Lisa Script Details

This will create a Test Script that references external test resources. Select the type of test resource to use:

☒ Use test resources from a shared location
Choose test resources from a shared location.
At execution time, the resources will be copied to the test machine.

☐ Use test resources that are local to a test machine
Choose test resources that are on a test machine already.
At execution time, the resources at the location you choose will be used.

Select Adapter

Test Case: C:\LISA\Projects\kr1\Tests\web2simple.tst

Configuration: C:\LISA\Projects\kr1\StagingDocs\supersimple.stg

Staging Doc: C:\LISA\Projects\kr1\StagingDocs\supersimple.stg

Discard Changes Save

RQM-LISA Test Script Execution

Quality Management (Jazz) Project1

Test Case: 1: LISA Tests - Quality Management
https://192.168.11.188:9443/jazz/web/console/Project1#actions:com.ibm.rqm.planning.home.actionDispatcher&subAction=viewTestCa...

Requirements Planning Construction Lab Management Builds Execution Reports Defects

Quality Dashboard View Test Scripts 1: LISA Simple Test View Test Cases 1: LISA Tests

Manage Sections

Overview Test Case Overview

Summary

Test Case Design Formal Review Development Items Requirement Links Risk Assessment Pre-Condition Post-Condition Expected Results Test Scripts Test Case Execution Records Attachments Execution Variables Show All Sections

Originator: itko Owner: itko

Description: < Click here to enter a description >

Summary

Use the theme, category, measure of execution, Categories

Function: Test Phase: Weight: *

Run Test Case

Test Case Execution: LISA Tests

Select New to create a new record or select a pre-existing record from the list

Test Plan: Unassigned

Test Milestone: Unassigned

Test Environment: Unassigned

Test Script: LISA Simple Test

Machine: Win2k8x64-RQM Available

Test Cell:

☒ Create Result without Execution

☐ Modify Execution Variable Value(s)

Host Name: Win2k8x64-RQM

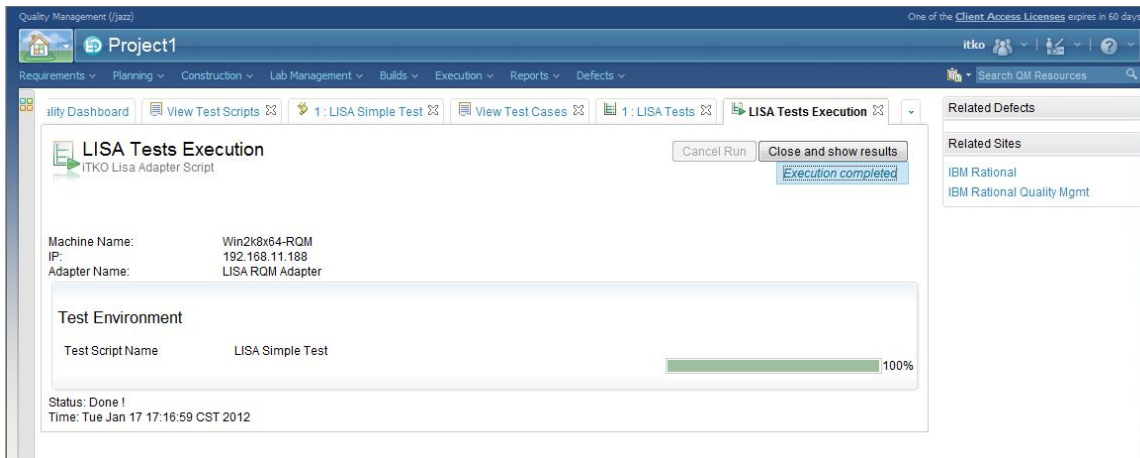
State: Adapter is Available.

Time since last contact: 5 sec

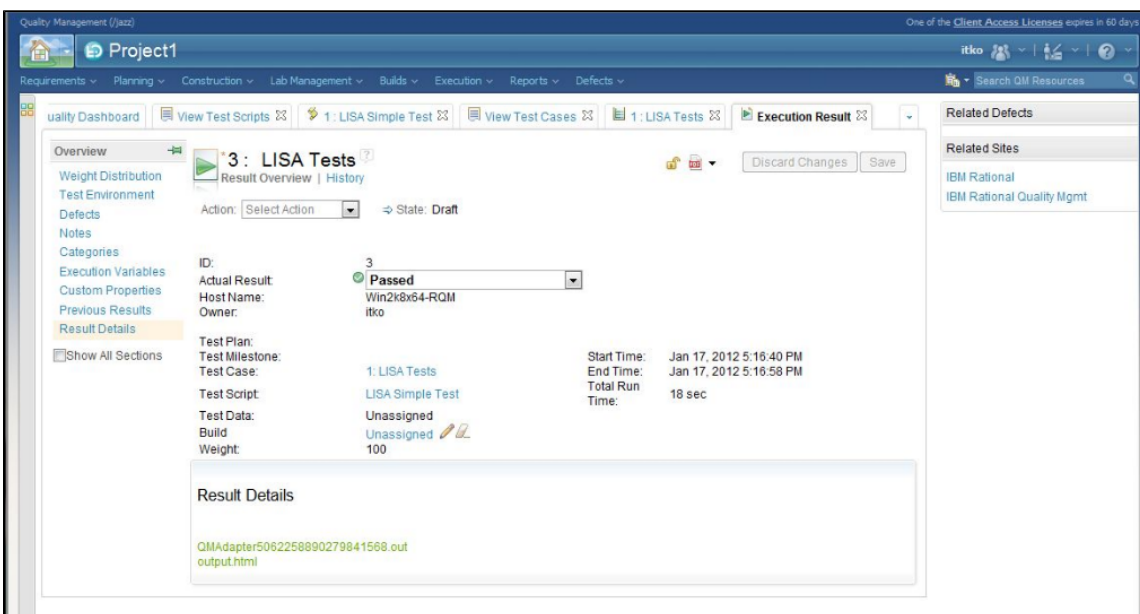
Reserved: Not Reserved

OK Cancel

RQM-LISA Execution Complete



RQM-LISA Test Script Results View



Upgrading LISA from a Previous Version

This section includes details on steps needed to upgrade from one LISA version to another.

LISA 4.x to LISA 5.0
LISA 5.0 to LISA 6.0

LISA 4.x to LISA 5.0

Items to consider if you are upgrading LISA 4.6 to 5.0

Review the [new features](#).

Review the [known issues](#).

Review [LISA 5.0 The JVM Could not start](#).

Virtual Services Edition

See [Virtual Service Image Conversion for 5.0](#).

Replacing 4.x Virtual Web Service with 5.0 WSDL generated VSE.

QC Plugin

[QC9 - 10 Updates](#)

LISA 5.0 to LISA 6.0

Before upgrading to LISA 6.0, we recommend that you review the [new features](#) and the [known issues](#).

Database Schema Changes

The LISA database has undergone the following schema changes in release 6.0:

- A column was added to the HST_STEP_RUN table.
- The name of an existing column was changed in the HST_CYCLE_RUN table.

Database administrators can use one of the following scripts in the **LISA_HOME/database** directory to update an external database:

- db2-Lisa-6-0-upgrade.ddl
- derby-Lisa-6-0-upgrade.ddl
- mysql-Lisa-6-0-upgrade.ddl
- oracle-Lisa-6-0-upgrade.ddl
- sqlserver-Lisa-6-0-upgrade.ddl

For databases that give LISA permission, the updates will be made automatically on startup.



LISA Virtualize no longer uses a database.

Exporting Legacy Service Images

Beginning with LISA 6.0, service images are no longer stored in a database. If you need to use LISA 5.0 service images in LISA 6.0, export them using LISA 5.0 and then import them using the Import item on the project tree's context menu in LISA 6.0.

See [Creating Service Images](#) for information about how to export legacy service images.