Medusa Labs Test Tools Suite

Version 6.0.1

Installation Guide

Notice

Every effort was made to ensure that the information in this document was accurate at the time of printing. However, information is subject to change without notice, and JDSU reserves the right to provide an addendum to this document with information not available at the time that this document was created. This document was published in September 2013.

Copyright

Trademarks

Support and Maintenance

Technical Support:

Telephone:			E-mail: techsupport-snt@jdsu.com
	US Toll Free:	1 (866) 594-2557	
	International:	1 (408) 546-7096	

Sales and Maintenance Support:

Telephone:			E-mail: sales-snt@jdsu.com
	US Toll Free:	1 (888) 746-6484	E-mail: insidesales.snt@jdsu.com
	International:	1 (408) 546-4255	Website: www.jdsu.com/snt/

Contacting Medusa Labs

You can contact Medusa Labs Monday through Friday, 8 a.m. to 5 p.m. Central Time.

Medusa Labs, 800 Paloma Drive, Suite 130, Round Rock, TX 78665

Phone: (512) 670-7300   E-mail: techsupport-medusa@jdsu.com

Fax: (512) 670-5078   Website: www.jdsu.com/go/medusa

Terms and Conditions

Contents

About this Guide

Congratulations on your purchase of the Medusa Labs Test Tools Suite.

This guide provides information about how to install and run the Medusa Labs Test Tools software.

For information on using and operating Medusa Labs Test Tools, refer to the *Medusa Labs Test Tools Suite User's Guide*.

What this Guide Contains

This guide contains the following chapters:

Chapter 1, "Introduction to Installing Medusa Labs Test Tools" describes the test tools contents. It also includes system requirements and information about license servers and licensing.

Chapter 2, "Installing Medusa Labs Test Tools" describes the procedures for deploying the license server and installing the Medusa Labs Test Tools software.

Chapter 3, "Installing Virtual Machine Licenses" provides the details of using Virtual Machine licenses.

Chapter 4, "Using Medusa Labs Test Tools" describes how to launch Medusa Labs Test Tools, update remote systems, troubleshoot issues, and getting help.

Conventions

The following conventions are used in this guide.

Message Formats

This guide uses the following format to highlight special messages:

Note: This format is used to highlight information of importance or special interest.

Important: This format is used to highlight information that you should know.

Caution: This format is used to highlight information that will help you prevent equipment failure or loss of data.

Warning: This format is used to highlight material involving possibility of injury or equipment damage.

Typographical Conventions

This guide uses the following typographical conventions:

bold sans serif	Commands
italics	Directory names, book titles, named keys, for example the *Enter* key.
`courier font`	Screen text, user-typed command-line entries.
`courier italics`	user-supplied variable, argument

Chapter 1

Introduction to Installing Medusa Labs Test Tools

In this chapter:

Medusa Labs Test Tools Suite

The Medusa Labs Test Tools Suite performs data integrity testing, generated signal quality stressing data patterns, and enterprise application simulation. The suite consists of three sets of tools:

- **I/O Tools Pain, Maim, and Sock**—
  - **Pain** is a synchronous I/O tool designed to issue a single pending I/O per worker thread.
  - **Maim** is an asynchronous I/O tool that uses a single worker thread to issue multiple I/Os at once.
  - **Sock** is a TCP and UDP traffic generator. Each thread opens its own socket connections.
- **Catapult Test Tools Automation**—**Catapult** is the target discovery tool that acts as a shell for the I/O tools. You use Catapult to discover targets available to the host system and pass these targets to the other Medusa Labs Test Tools for I/O testing. Catapult can be used to launch tests on remote systems and verify test results. Additional features in Catapult facilitate test scripting and automation.
- **FindLBA and GetKey Utilities**—
  - **FindLBA** is for use when debugging data corruption issues in tests on file systems or logical devices. It is useful in cases where the logical block address (LBA) reported in the I/O tool error logs is not accurate because the tools are not directly referencing areas of the physical media. You can use FindLBA in conjunction with a protocol analyzer to identify the actual LBA corresponding to a file offset reported by Medusa Labs Test Tools. The FindLBA tool sends a periodic "poll" of consecutive reads to a specified offset, which you can identify in a protocol trace. FindLBA is most useful when you need help finding I/O command that resulted in data corruption in a protocol trace capture.
  - **GetKey** is used for remote license checkouts. A system with network access to a license server can perform a license checkout for another system that does not have network access. This utility is particularly useful for temporarily using Medusa Labs Test Tools at an offsite location.

This software installation guide covers the steps to install and run the Medusa Labs Test Tools Suite. For more detailed operating instructions, see the *Medusa Labs Test Tools Suite User's Guide* that you received with the software.

System Requirements

Medusa Labs Test Tools are designed to utilize system resources as efficiently as possible. However, performance and stress testing is by nature resource intensive. Specific system requirements vary with the architectures under test. Generally, in order to achieve full duplex wire speed levels of throughput with data integrity checking on topologies such as Fibre Channel and Gigabit Ethernet, an enterprise class system is desired. We recommend the following as a minimum for the hardware:

- 1 GHz or greater processor
- 1 GB or greater RAM

Medusa Labs Test Tools takes advantage of multiple processors.

See the *Medusa Labs Test Tools Suite User's Guide* for additional information about system limitations, memory utilization, processor utilization, PC Bus limitations, and operating system restrictions.

License Server

The Medusa Labs Test Tools Suite uses a license server to issue license keys and enforce product licensing policies. The license server software is distributed with Medusa Labs Test Tools but you must install it separately. The license server runs as a Microsoft[1] Windows service. Refer to Figure 1.

Installation instructions are included with the license server software. Medusa Labs provides the USB security dongle required to activate the license manager installation. It is best to install the license server software on a dedicated machine before installing Medusa Labs Test Tools on the systems where you will run Medusa Labs Test Tools.

Figure 1: Medusa Licensing Model

---
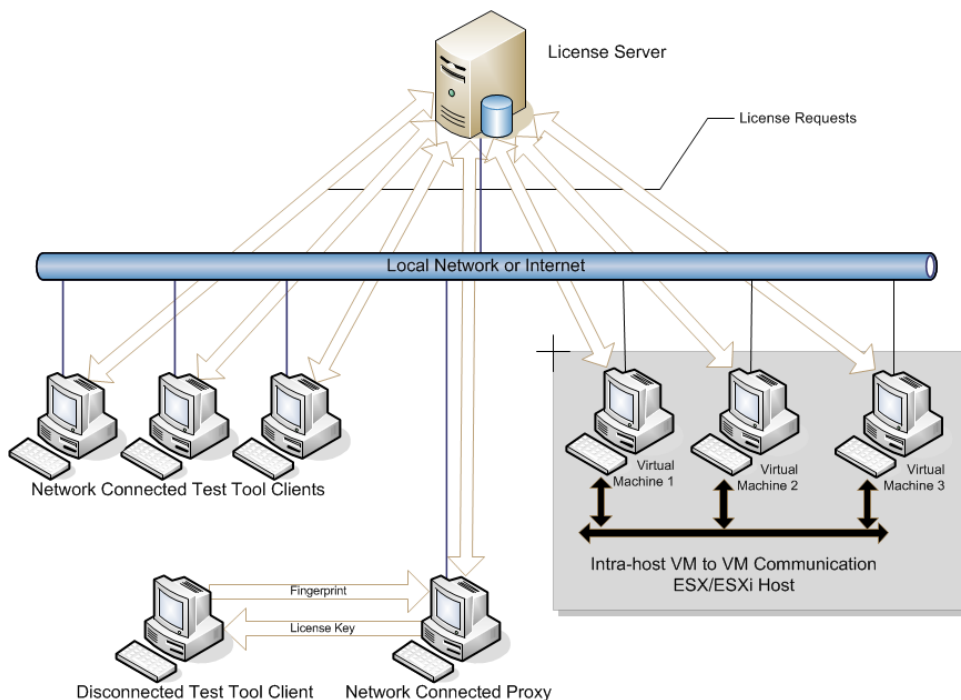
[1] Microsoft and Windows are registered trademarks of Microsoft Corporation in the United States and/or other countries.

Licensing Types

Medusa Labs Test Tools are licensed on a "per seat" subscription basis. This means that the tools are licensed for a period of time (usually one year). A certain number of "seats" are licensed under the subscription. A seat is any client system that is currently running Medusa Labs Test Tools. You can use the license seats on any system, but concurrent usage is limited to the number of seats purchased. This methodology allows you to use the tools where they are needed, without being restricted to particular systems. When the test is finished, the seat can be checked back in to be used by other client machines.

In addition to Regular Licensing, Medusa Labs Test Tools supports Virtual Machine Licensing that is described in depth in the *Medusa Labs Test Tools Suite User's Guide*. Refer to Figure 1 on page 5.

License usage is regulated by a license server that issues license keys in response to checkout requests by client systems, up to the number of seats purchased. The license server software is provided for on-site installation of the license server. Medusa Labs provides a USB security dongle which is a HASP security device that is plugged into the USB port to enable the license server.

Chapter 2
Installing Medusa Labs Test Tools

In this chapter:

Before Installing Medusa Labs Test Tools

> **Important:** The Medusa Labs Test Tools Suite uses a license server to issue license keys and enforce product licensing policies. The license server software is distributed with the application but you must install it separately. The license server runs as a Windows service.
>
> Installation instructions are included with the license server software. Medusa Labs provides the USB security dongle required to activate the license manager installation. We recommend that you install the license server software on a dedicated machine before installing Medusa Labs Test Tools on the systems where you will run the application.

The installation is a three-step process.

1   First, download the software from the JDSU web site.
Refer to "Step 1. Download the Software" on page 9

2   First, install the License Server.
Refer to "Step 2. Install the License Server" on page 10.

3   Next, install the Medusa Labs Test Tools software.
Refer to "Step 3: Install the Medusa Labs Test Tools Suite" on page 13.

Firewalls

In Medusa Labs Test Tools 6.0.1 or later, the application dynamically manages the firewall rules. This means that under most situations, it is no longer necessary to disable the firewall for the application.

However, sometimes this automatic firewall management may not be possible due to site policies or specific user configurations. In such cases, you will need to provide manual firewall configurations if you want to use the Medusa Labs Test Tools networking features, such as remote testing.

In Windows, the preferred method is to add the Medusa Labs Test Tools executable files to the firewall exceptions. The executable files to be added to the exception list to allow incoming network connections are the "<install_dir>\Test Tools\bin*.exe" files.

In Unix, application-based rules are generally not available and the firewall rules must be port-based. Because Medusa Labs Test Tools uses dynamic ports, it is difficult to define a port-based rule. Therefore, if the new dynamic rule management feature of the application is not possible on your system, the firewall may need to be deactivated.

However, under typical default conditions, the new dynamic firewall rule management feature should be sufficient for hands-off operation.

Step 1. Download the Software

When you received your shipment that includes the USB security dongle, there was also a Thank You card in the shipment that advised you to download the software at: http://www.jdsu.com/go/medusa

When you go to this website, click on the **Downloads** tab and you will see the following selections:

1   **Medusa Labs Test Tools Software Package (Complete Software Package)**
2   **Windows Installation Files (Individual Software Installer)**
3   **Linux Installation Files (Individual Software Installer)**
4   **UNIX Installation Files (includes Solaris and HP-UX) (Individual Software Installer)**
5   **License Server Installation Files (Individual Software Installer)**
6   **Trial License Procedure (Evaluation Software Instructions)**

Each selection will open the FTP root site on the ftp.jdsu.com page and allows you to download the appropriate Medusa Labs Test Tools software.

Selecting the **Complete Software Package** link is recommended for first time users of Medusa Labs Test Tools. This allows you to download the complete .ISO package.

Step 2. Install the License Server

The Medusa License Management Server (MLMS) is a hardware/software combination.

Platform Support

The MLMS should install and function on all Windows platforms (32- and 64-bit CPUs) listed below and supported by the USB security dongle.

• Windows XP (with Service Pack 3), Windows XP x64 Edition (with Service Pack 2)

• Windows Server[1] 2003 R2 (with Service Pack 2)

• Windows Server 2008 (with Service Pack 2)

• Windows Server 2008 R2

• Windows Vista[2] (with Service Pack 2)

• Windows 7

Installation

1   Make sure the USB security dongle is not plugged in.

2   Install the MLMS software using one of the two installer executable files provided:

   • `Medusa License Manager 1.2.2 (x86).exe` on 32-bit systems

   • `Medusa License Manager 1.2.2 (x64).exe` on 64-bit systems

The only parameter that you may change during installation is the installation directory.

➡   **Note:** The version number `1.2.2` is not permanent and is incremented as required.

There may be a pause of about 30 seconds while the USB security dongle driver software is installed.

3   After the MLMS software installation has finished, plug in the USB security dongle.

4   The USB security dongle must be programmed.

This process is described in "Programming the USB Security Dongle" on page 11.

---

[1] Windows Server is a registered trademark of Microsoft Corporation in the United States and/or other countries.
[2] Windows Vista is a registered trademark of Microsoft Corporation in the United States and/or other countries.

Installed Items

The MLMS software installation process installs the following:

1 USB Security Dongle

This is the hardware protection key for MLMS. The MLMS will not function without the USB security dongle.

2 License Server

The MLMS software is installed and executed as a Windows Service. The registered display name is **Medusa License Manager Service** in the Windows Service Control Manager application.

The short name is **mlms**, which can be used from the command line. For example, you can:

Start the service by entering: `net start mlms`
Stop the service by entering: `net stop mlms`

3 Utilities

The general utility program, mlmadmin.exe, is included. Its uses are outlined in "Administration" on page 12. This program is also installed as a part of Medusa Labs Test Tools.

Programming the USB Security Dongle

The USB security dongle must be programmed before the MLMS can manage the licenses. This must be done when:

- The MLMS is first deployed for the first time.
- Each time the customer license expires and is renewed.
- If more seats are purchased.

1 Create a USB security dongle ID file:`mlmadmin –getid <idfile name>`

This generates an encrypted ID file named `idfile` (The file name is an example only).
In general, a name that can easily identify the deployment site, such as customer name, should be used.

For example: `mlmadmin -getid server.id`

➡️   **Note:** The USB security dongle must be plugged in while the ID file is created.

---

2 Send the ID file as an email attachment to the JDSU support team at the following email address: **techsupport-medusa@jdsu.com**

3 The JDSU support team will create a USB security dongle update file that matches the ID file that you provided and will send the file to you as an email attachment.
The update file is named: `dongleupdate.dat`

4 Plug in the USB security dongle and program it using the command:
`mlmadmin -update dongleupdate.dat`

➡️   **Note:** The USB security dongle must be plugged in while it is being programmed.

The license operations (check-in, check-out) should be functional from client machines.

Administration

Display the current MLM server state: `mlmadmin -reportserverstate`

Display the current MLM client state: `mlmadmin -reportclientstate`

This utility shows the currently deployed license state of the server (number of seats and expiration dates) and each client system with a checked-out license.

Migrating the MLM License Server

To migrate the MLM License Server to a different machine requires the following backup/restore procedure in order to avoid having a corrupted server state. This procedure must be performed directly at the old and new license server systems because it requires physically removing and plugging in the USB security dongle.

At the current license server system:

1 Open a command window.

2 Stop the MLM License Server by running: `net stop mlms`

3 Unplug the USB security dongle.

⚠ **Warning:** Do not plug the USB security dongle back in on any system until this backup/restore procedure is completed.

4 Backup the MLM License Server state by running: `mlmadmin -backup`

This creates the `mlms.backup` file.

5 Copy the `mlms.backup` file to the new system.

At the new license server system:

1 Install the MLMS software on the new system.

⚠ **Warning:** Do not plug the USB security dongle back in on any system until this backup/restore procedure is completed.

2 Open a command window.

3 Use `cd` to open the directory where the `mlms.backup` file from the old system was copied.

4 Restore the MLM License Server state by running: `mlmadmin -restore`

5 Plug in the USB security dongle.

6 Start the MLMS service in the new system by running: `net start mlms`

7 Make sure the Medusa Labs Test Tools client machines are configured to use the new MLM License Server.

Step 3: Install the Medusa Labs Test Tools Suite

The procedure to install Medusa Labs Test Tools differs based on the operating system that you are using. Select the correct procedure from the procedures listed below.

If you are installing Medusa Labs Test Tools on operating systems running:

- Windows operating systems:

  Install Medusa Labs Test Tools using the procedure for the appropriate mode:

- Other operating systems, start installing Medusa Labs Test Tools. Depending on your operating system and preferred mode, continue at the appropriate section:

Additional Windows Information

The Medusa Labs Test Tools Suite for Windows operating systems uses a Windows installation wizard with a self-extracting archive program.

- Run "Setup_x86_6.0.1.<build_number>.exe" for 32-bit Windows.
- Run "Setup_x64_6.0.1.<build_number>.exe" for 64-bit Windows.
- Run "Setup_ia64_cmd_6.0.1.<build_number>.exe" for Windows for Itanium[1].

Alternatively, you can bypass the Windows installation wizard by installing Medusa Labs Test Tools in silent mode.

Important: Make sure to uninstall the previous version of Medusa Labs Test Tools before you upgrade the Medusa Labs Test Tools Suite for Windows operating systems. If you decide to use the command-line only version of the application, uninstall the Windows version first.

Microsoft .NET Framework

Microsoft .NET Framework 3.5 is included in the Medusa Labs Test Tools software and will be installed if it is not already on the system.

Note: Microsoft .NET Framework 3.5 is not supported on Windows 2003 Server for Itanium processors and the Medusa Labs Test Tools 6.0 (and higher versions) Graphical User Interface (GUI) cannot run without .NET Framework 3.5. Therefore, for Itanium systems, you must upgrade to Windows 2008 Server (or higher) to use the GUI version of Medusa Labs Test Tools 6.0 (and higher versions). The command-line only interface of Medusa Labs Test Tools 6.0 (and higher versions) can still be run using Windows 2003 Server for Itanium processors.

---

[1] Itanium is a trademark of Intel Corporation in the U.S. and/or other countries.

Installing Medusa Labs Test Tools on Windows

1 Select **Install Test Tools** from the Medusa Labs Test Tools Welcome Window.

The license agreement is displayed.

2 Select **I accept the terms in the License Agreement** and click **Next** to continue with the installation.

3 Note the **License Manager Server Information** dialog box information and ensure the Port number is set to 5033. See Figure 1.

This version of Medusa Labs Test Tools uses the Medusa License Manager server (that was installed previously in "Step 1. Download the Software" on page 9) for issuing and enforcing product license policies. You can check out and check in the license keys directly in a TCP/IP network or off-line with a remote checkout.

Figure 1: License Manager Server Information Dialog Box

4 Choose the setup type. You are provided the following choices:

Typical	Installs all Medusa Labs Test Tools program features in the default installation folder.
Custom	Installs only the command line version of Medusa Labs Test Tools, allows you to choose if the graphical user interface version is also installed, and designate the installation folder.

5   If you selected **Typical**, go to the next step.
       If you selected **Custom**, make your selections in the **Custom Setup** window and click **Next** to proceed.

The setup program is ready to install Medusa Labs Test Tools on your machine at this point. If you want to change any of the settings you specified in the previous steps, click **Back** to review or modify the settings.

> ➡ **Note:** This is your final opportunity to cancel the installation process.

6   Click **Install** to start the installation process.

When the installation process begins, a progress bar displays to indicate activity.

The setup program performs the following actions (where: `<install_dir>` is the installation directory you have chosen).

- Copies the Medusa Labs Test Tools executable files into the `<install_dir>\Test Tools\bin` subdirectory.

- Copies the documentation and license agreement files into the following subdirectory: `<install_dir>\Test Tools\docs`

- Creates the necessary Registry entries.

- Creates the license manager configuration file (which you can edit with a text editor at any time): `<install_dir>\Test Tools\config\MedusaTools.cfg`

- Makes new system environment variable entries necessary for the proper operation of Medusa Labs Test Tools.

- Registers the Medusa Labs Application Agent Service: `<install_dir>\Test Tools\bin\maagent.exe` and starts the service. (The agent service must be running for Medusa Labs Test Tools to operate properly; therefore, the proxy runs as a service and automatically starts during each system reboot.)

7   Click **Finish** in the final dialog box to close it. After installation, the Medusa Labs Test Tools suite will appear in the Windows Control Panel add/remove programs application as **Medusa Labs Test Tools**.

Installing Medusa Labs Test Tools on Windows in Silent Mode

The options for the Medusa License Manager are `MLM_SERVER` and `MLM_PORT`.

> ⓘ **Important:** Make sure to uninstall any older versions of Medusa Labs Test Tools before you upgrade to the Medusa Labs Test Tools Suite for Windows operating systems.

To use the Medusa License Manager, manually run the Windows setup program in the \windows directory on the CD using the /qb switch.

For example:

```
Setup_x86_6.0.1.<build_number>.exe /qb MLM_SERVER=<server_name>
MLM_PORT=<port> MLTT_DEFAULT_CHECKOUT=<days>
```

where, you use:

- The appropriate setup command from the following list whose name represents the architecture of your target system:
  - `Setup_x86_6.0.1.<build_number>.exe` for 32-bit Windows
  - `Setup_x64_6.0.1.<build_number>.exe` for 64-bit Windows
  - `Setup_ia64_cmd_6.0.1.<build_number>.exe` for Windows for Itanium
- `<server_name>` is the name of the license server
- `<port>` is the port to connect to the license server (the default is `5033`)
- `<days>` is the number of days that the license will be checked out (the default is 5)

Medusa Labs Test Tools is installed in the standard installation directory, *C:\Program Files\Medusa Labs* without further user intervention.

Before Running Medusa Labs Test Tools on Windows

When the application is successfully installed, the test machine environment is properly configured to run Medusa Labs Test Tools. However, if you have any currently running console sessions, you will not have the proper environment to run the application. You must open a new Command window to run the application.

Installing Medusa Labs Test Tools on Linux, Solaris, or HP-UX

The Medusa Labs Test Tools Suite for Linux[1], Solaris[2], and HP-UX[3] operating systems is distributed as a TAR archive file with an installation script.

- Extract "linux-x86_6.0.1.<build_number>.tar.gz" for 32- and 64-bit x86 Linux.
- Extract "linux26-ia64_6.0.1.<build_number>.tar.gz" for Linux for Itanium.
- Extract "solaris-sparc_6.0.1.<build_number>.Z" for Solaris for 32- and 64-bit SPARC[4].
- Extract "solaris-x86_6.0.1.<build_number>.Z" for Solaris for 32-and 64-bit x86.
- Extract "hpux-ia64_6.0.1.<build_number>.Z" for HP-UX for Itanium.

To install the software, execute the setup.sh script and follow the on-screen directions that guide you through the installation process described in the following example. In this example, the pound symbol # represents the shell prompt where you enter commands. The actual shell prompt might differ on each system.

1 Log on as a root user.

2 Execute the `setup.sh` script from the directory where it resides:

```
# ./setup.sh
```

If you are not logged in as the `root` user, the setup script will prompt for the root password:

```
This program must be run as root user. Please enter the root password.
Password:
```

Enter the root password to continue. (This step is not necessary if you are already logged in as root.)

The installation process begins by displaying the welcome screen shown in Figure 2.

Figure 2: Medusa Labs Test Tools Installation Welcome Screen

---
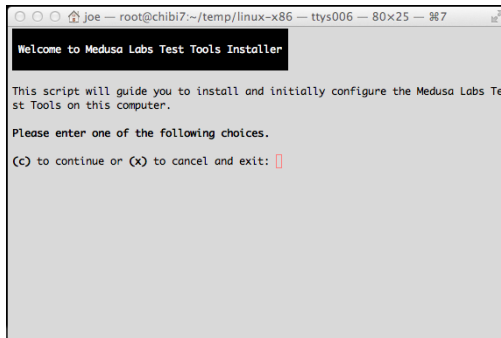
[1] Linux is a registered trademark of Linus Torvalds.
[2] Solaris, Sun, Sun Microsystems, Java, SQL*Plus, and Oracle are either trademarks or registered trademarks of Oracle Corporation in the United States and/or other countries.
[3] HP-UX is a registered trademark of Hewlett-Packard Development Company, L.P.
[4] SPARC is a trademark or registered trademark of SPARC International in the United States and/or other countries.

At each step of the installation process, you can enter:

p to return to the previous step

c to proceed to the next step
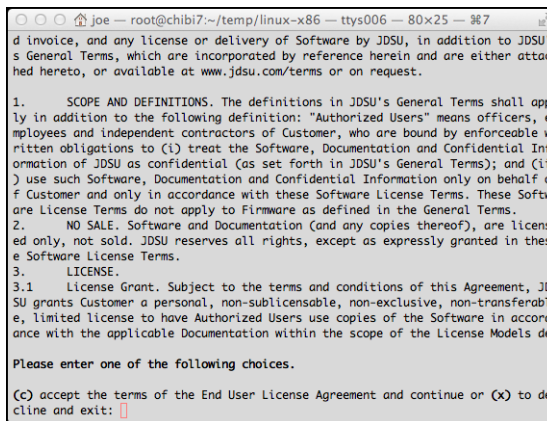
x to cancel and exit the installation process

Then press **Return** or **Enter**.

If you cancel the installation process, the setup program makes no change to the system.

3 Enter c and press **Return** to continue with the installation.

To install and use the software, you must agree to the terms and conditions of the JDSU End User License Agreement. The setup program displays the license agreement text. Use the more page command by pressing the space bar to scroll through the text or press q to end the display. At the end of the license agreement text, you can accept or decline the license agreement (Figure 3).

Figure 3: End User License Agreement Screen

4 Enter c to accept the license agreement and continue.

You can choose to cancel the installation, leaving the system untouched. The setup program does not allow you to continue the installation process if you do not accept the license agreement.

5 Specify the installation directory where to install the software.

The various components of the software are installed in the *test_tools* subdirectory within the installation directory you specify (Figure 4).

Figure 4: Installation Directory Screen
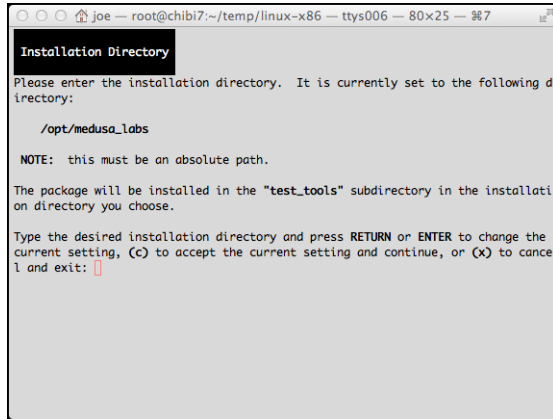
6 If you want to choose a different installation directory, type the absolute path name of the directory and enter c to continue to the next step (Figure 5).

If you have installed the license management server, you can provide the license manager configuration parameters now. Alternatively, you can proceed without specifying the information and edit the configuration file later. See the "Editing the License Server Configuration" on page 22 for more about editing the license server configuration.

Figure 5: License Manager Information Screen
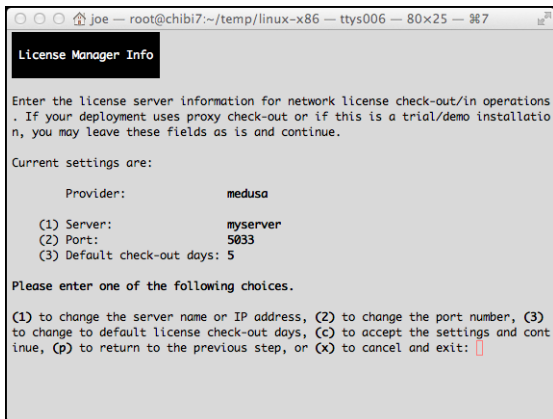
The license management provider to use should be set. To set the Medusa License Manager, use medusa.

The "Server" field should be set to the host name or IP address of the machine running the license manager server.

You can change the "Port" number, the TCP/IP server port the license manager uses, only if you change the default port number on the license manager server.

7 Enter `c` to continue.

Or you can enter:

> `1` to change the server name
>
> `2` to change the port number
>
> `3` to change the default license checkout days

The setup script prompts you to enter the new setting and returns you to this step after changing the setting.

The setup program is now ready to install Medusa Labs Test Tools on your machine.

8 To install the software, enter `c`.

If you wish to change any of the settings you specified in the previous steps, enter `p` to return to previous steps and review or modify the settings. This is also the final opportunity to cancel the installation process (Figure 6).

Figure 6: Confirm Installation Settings Screen
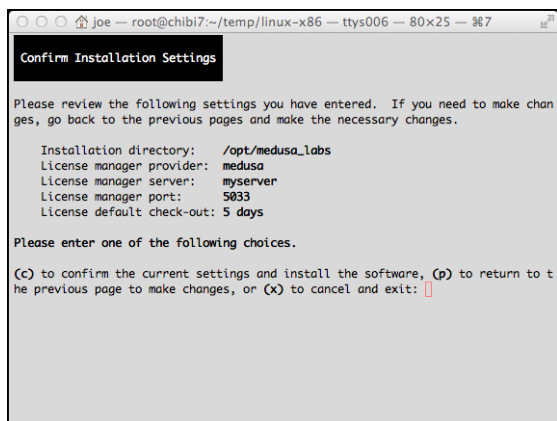
While the setup program is installing the software, it displays the current progress.

During this step, the setup program performs the following actions (where: *<install_dir>* is the installation directory you have chosen).

- Copies the Medusa Labs Test Tools executable files into the following subdirectory: *<install_dir>/test_tools/bin*

- Copies the documentation and license agreement files into the following subdirectory: *<install_dir>/test_tools/docs*

- Creates the license manager configuration file (which you can edit with a text editor at any time): *<install_dir>/test_tools/config/MedusaTools.cfg*

- Installs the startup script to run the Medusa Labs Application Agent daemon in */etc/init.d* and */etc/rc3.d* directories and starts the daemon. (The agent must be running for Medusa Labs Test Tools to operate properly; therefore, the agent runs as a daemon and automatically starts during each system reboot.)

- Adds entries to login startup scripts (for example, `profile`, `.cshrc`, etc.) for the root user.

When the installation completes, the setup program displays the final status message (Figure 7).

Figure 7: Installation Complete

9   Close the terminal window to log out and complete the installation.

Installing Medusa Labs Test Tools on Linux, Solaris, or HP-UX in Silent Mode

Unattended installation allows you to specify the installation parameters as command line parameters in order to install Medusa Labs Test Tools without interacting with the installation script as shown above.

- Extract "linux-x86_6.0.1.<build_number>.tar.gz" for 32- and 64-bit x86 Linux.
- Extract "linux26-ia64_6.0.1.<build_number>.tar.gz" for Linux for Itanium.
- Extract "solaris-sparc_6.0.1.<build_number>.Z" for Solaris for 32- and 64-bit SPARC.
- Extract "solaris-x86_6.0.1.<build_number>.Z" for Solaris for 32-and 64-bit x86.
- Extract "hpux-ia64_6.0.1.<build_number>.Z" for HP-UX for Itanium.

Run in unattended mode with the '`-s`' option: `./setup.sh -s` [other options...]

➡ **Note:** The option key and option values are separated by space.

The options are:

`-s`	Runs the installation script in unattended mode
`-p`	Ignores the `-d`, `-l`, `-c`, and `-w` options (if they are specified) and uses the existing or default settings instead
`-d` <path/directory>	Defines the full path to the installation directory
`-l` <server name/IP address>	Identifies the name or the IP address of the license server
`-c` <integer>	Sets the default number of days to check out licenses
`-w medusa`	Identifies the name of the license provider
Note: Any unspecified `-d`, `-l`, `-c`, `-w` will result in using the existing setting for the option or a default value.	

Before Running Medusa Labs Test Tools on Linux, Solaris, or HP-UX

After you have successfully installed Medusa Labs Test Tools, the test machine environment is properly configured to run the application. However, any currently running terminal session will not have the proper environment to run the application. Either log out and log back in, or start new shells as root user and run the tools. You might need to add *<install_dir>/test_tools/bin* to your path if your login shell is something other than `sh`, `bash`, `csh`, or `tcsh`.

Editing the License Server Configuration

If you skipped configuring the license server manager information, obtain the proper license server configuration information and edit the following file before running Medusa Labs Test Tools:

Windows:

<install_dir>\Test Tools\config\MedusaTools.cfg

UNIX:

<install_dir>/test_tools/config/MedusaTools.cfg

Chapter 3

Installing Virtual Machine Licenses

In this chapter:

- Installing the Virtual Machine License

A Virtual Machine (VM) License is a new license type added to Medusa License Management Server (MLMS) 1.2.1 and Medusa Labs Test Tools 6.0.0 (or newer). The optional VM licenses provide the same capability as regular licenses but are more cost effective on per seat basis. However, VM licenses can only be granted to supported and recognized virtual machines being hosted on VMware[1] ESX and VMware ESXi servers.

VM licenses are considered valid only when at least one VM on the host environment has a regular license checked out. For example, a Medusa Labs Test Tools deployment site has two VMware ESX/ESXi servers with each server hosting ten VMs. Each ESX/ESXi server (with its ten VMs) is considered a "host environment". With the new VM licensing option, you can purchase the less expensive VM licenses. However, because at least one regular license per host environment is required, at least two regular licenses (one checked out to a VM on each server) are needed for this scenario. Refer to Figure 8.

Figure 8: VM Licensing Example Diagram

---
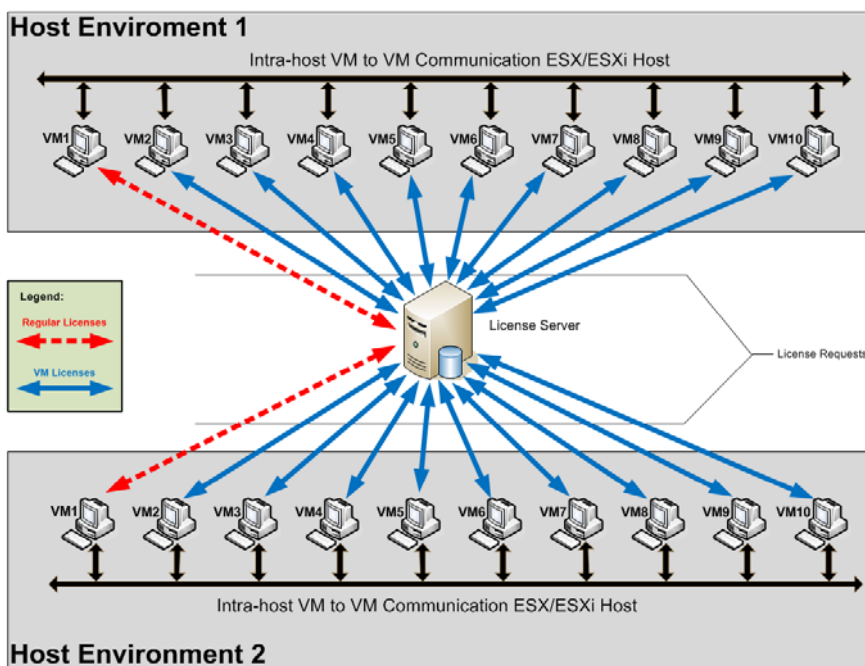
[1] VMware is a registered trademark or trademark of VMware, Inc. in the United States and/or other jurisdictions.

Installing the Virtual Machine License

This section describes:

- Using the VM Licensing
- Disabling VM Licensing
- Troubleshooting VM Licensing

Using VM Licensing

1 Install Medusa Labs Test Tools 6.0.0 (or newer) on VMs.

2 For each host environment, designate one of the VMs running Medusa Labs Test Tools 6.0 (or newer) as the "VM license proxy" by executing the CLI command in the designated VM: "`mlmadmin -vmproxy 1`"

3 Once a VM license proxy is designated and active, all other VMs in the host environment enter the VM licensing state and stay in this state until VM licensing is explicitly disabled for all VMs.

4 Running "`mlmadmin -reportclientstate`" on the VM license proxy displays "`VMLM PROXY: ENABLED`".

5 Be careful not to designate more than one VM in a host environment as the VM license proxy because this will cause more regular licenses to be consumed than is necessary.

Disabling VM Licensing

1 Once a VM enters the VM licensing state, the state persists until it is explicitly cleared.

2 To clear the state, first run "`mlmadmin -vmproxy 0`" on the VM license proxy then run "`mlmadmin -vmproxy 0`" on every other VM in the host environment.

3 If more than one VM was set as the VM license proxy by mistake:

a First run "`mlmadmin -vmproxy 0`" on each VM license proxy one at a time.

b Then run "`mlmadmin -vmproxy 0`" on each VM license proxy once again.

c Finally run "`mlmadmin -vmproxy 0`" on every other VM in the host environment.

Troubleshooting VM Licensing

There are some scenarios in which a VM may end up with both a regular license and a VM license when only VM licensing is desired. For example, if the VM had checked out a VM license before the VM licensing state is cleared without first checking in the VM license.

The "`mlmadmin -reportserverstate`" command can help to identify which systems might have both types of licenses checked-out. For each client system with a license, the output displays:

"Checked-out Product:" of "MedusaLabs Test Tools version 1" for regular license
   - and -
"Checked-out Product:" of "MedusaLabs Test Tools version 1 for VM" for VM license

An example of the output returned by the "`mlmadmin -reportserverstate`" command is shown in Figure 9.

Figure 9: `mlmadmin -reportserverstate` Command Output

```
mlmadmin v1.2.1:  Medusa License Management administration tool.
Copyright (c) 2008-2012 JDS Uniphase, Corp


SERVER: 10.23.1.222
CLIENT: localhost

____GLOBAL SERVER STATE: Good
____LICENSEE:           ACME Widgets
____SERVER ID:          1836456269
____SERVER VERSION:     1.2.2.133553
____SERVER LIB VERSION: 1.2.2
____CLIENT LIB VERSION: 1.2.1
    -------------------------------------------------------------------------
____PRODUCTS
    -------------------------------------------------------------------------
    Product:                MedusaLabs Test Tools version 1
    Expiration:             2014/01/01 00:00:00
    Nb Seats:               15
    Nb Checkouts:           3
    Last Checkout Expiration: 2013/02/10 14:12:13

    Product:                MedusaLabs Test Tools version 1 for VM
    Expiration:             2014/01/01 00:00:00
    Nb Seats:               50
    Nb Checkouts:           2
    Last Checkout Expiration: 2013/02/03 10:41:19

____CHECKOUTS
    -------------------------------------------------------------------------
    Checked-out Product:  MedusaLabs Test Tools version 1 for VM
    Checkout Start:       2013/01/24 10:33:09
    Checkout Expiration:  2013/02/03 10:33:09
    Checkout Type:        Network
    Client:               WIN7:5034
    IP:                   10.23.1.51
    User:                 win7\joe

    Checked-out Product:  MedusaLabs Test Tools version 1
    Checkout Start:       2013/01/18 13:40:15
    Checkout Expiration:  2013/02/10 14:12:13
    Checkout Type:        Network
    Client:               WIN8:5034
    IP:                   10.23.1.44
    User:                 win8\joe

    Checked-out Product:  MedusaLabs Test Tools version 1
    Checkout Start:       2013/01/23 15:48:26
    Checkout Expiration:  2013/01/28 15:48:26
    Checkout Type:        Network
    Client:               CORP777:5034
    IP:                   10.23.1.73
    User:                 CORP\employee123
```

```
Checked-out Product:  MedusaLabs Test Tools version 1
Checkout Start:       2013/01/24 10:33:08
Checkout Expiration:  2013/02/03 10:33:08
Checkout Type:        Network
Client:               redhat6_1:5034
IP:                   192.168.113.159
User:                 root

Checked-out Product:  MedusaLabs Test Tools version 1 for VM
Checkout Start:       2013/01/24 10:41:19
Checkout Expiration:  2013/02/03 10:41:19
Checkout Type:        Network
Client:               redhat6_2:5034
IP:                   10.23.1.42
User:                 root
```

A couple of things to note regarding the "`mlmadmin -reportserverstate`" command output:

- Under "PRODUCTS", "MedusaLabs Test Tools version 1" (this is the "regular" license) and "MedusaLabs Test Tools version 1 for VM" (this is the VM license) with total seat counts and current number of checkouts for each.

- Under "CHECKOUTS", "WIN7" and "redhat6_2" are two VMs with VM licenses checked out.

Similarly, on a client system, running "`mlmadmin -reportclientstate`" will display the same information only for that client.

The normal license check-in procedure, such as executing "pain -Z" from the command line or from the GUI's host **Properties** dialog box will check in both licenses at once and clear redundant checkouts.

Chapter 4
Using Medusa Labs Test Tools

In this chapter:

Launch Medusa Labs Test Tools

To launch Medusa Labs Test Tools in a command line environment:

1   Open a new window or shell (a command window in Windows or terminal window in UNIX).

2   Type of the name of the tool with the command line switch you want to use and press *Enter*.

You can type `-h` to view a list of the available switches, for example:

```
pain -h
```

Refer to the *Medusa Labs Test Tools Suite User's Guide* "Using the Command Line Switches" chapter for more information.

The tools will automatically attempt to check out a license if one is not found on the system.

To launch the Medusa Labs Test Tools Graphical User Interface (GUI) application by:

• Double-clicking the **Medusa Labs Test Tools** shortcut icon on the desktop.

-or-

• Clicking the Windows Start menu and then choosing **Programs\Medusa Labs Test Tools\Medusa Labs Test Tools**.

Checking Out a License

It is best to perform a checkout from a single instance of one of the I/O tools on the command line before beginning any scripted testing. If any errors occur during the checkout process, the error messages are displayed on the console. If the checkout is successful, the license duration and expiration date/time is displayed on the console.

Network Checkout

If Medusa Labs Test Tools does not find a valid key on the client system, the tools automatically attempt to contact the license server and checkout a license key. The default checkout time is five days, unless you changed this during installation. You can override the default checkout time by using the `–Z` (case sensitive) switch on the Medusa Labs Test Tools command line. Use this switch with a numeric value to indicate the length of the checkout in days as shown in the following examples.

Examples:

Checkout for 10 days: `pain –Z10`

Checkout for 7 days: `maim –Z7`

If you finish using the tools on a system before the checkout time limit is reached, you can check in the license early by using the `–Z` switch alone as shown in the following example.

Example:

```
pain –Z
```

The license key is returned to the license server and becomes immediately available for another system to use.

For more information about using switches, refer to the *Medusa Labs Test Tools Suite User's Guide*, Chapter 3, "Using the Command Line Switches".

> ⓘ **Important:** Hardware changes or system date/time changes may invalidate a checked out license. To avoid invalidating a checked out license, check in your license prior to making these system changes, then check out the license once your changes are complete.

Remote Checkout

When a client system is unable to contact a license server and perform a checkout directly, you can use the remote checkout approach. You can also use remote checkouts to temporarily share Medusa Labs Test Tools with a third party for reproducing test scenarios. Any Windows system with network connectivity to the license server can perform a remote checkout with the GetKey utility.

1   Run one of the I/O Test Tools (Pain or Maim) on the client system.

When a checkout attempt fails, the tools automatically generate a system fingerprint file in the config directory. The file is named after the system host name, with a `.dat` extension, for example: `myhost.dat`.

You can also generate a license .dat file using the menu bar in the Medusa Labs Test Tools GUI. In the **File** menu, select the **Generate License .dat File...** selection.

2   Transport the `.dat` file from the client system to the networked system running the GetKey utility, with access to the license server.

You can do this using any available method, including: floppy, USB flash drive, peer to peer network, etc. If you are an off-site user, you can also e-mail this file for checkout from another location and the license key can be e-mailed back.

3   Run the GetKey utility on the networked system to perform a checkout for the client system. The path to the configuration file is passed to GetKey with the -f switch. You specify the number of days for the checkout with the `-z` switch (for example: getkey -fmyhost.dat -z3).

4   GetKey will contact the license server and request a license checkout. If successful, it will create a file in the current directory with the same name as the fingerprint file, with a ".lic" extension (for example, myhost.lic.)

5   You must take this file back to the client system to install the license code.

6   On the client system, run one of the I/O tools (Pain or Maim) with the license switch used to install the authorization code. The syntax of this switch is: `-Z#file_name`, where `file_name` is the location and name of the authorization code file created with GetKey.

Example:

Pain -Z#c:\temp\myhost.lic

7   The tool will install the license and display the checkout time available. You can now use any I/O tool for the checkout duration.

As an alternative to steps 6 and 7, you can install a license using the menu bar in the Medusa Labs Test Tools GUI. To do so, in the **File** menu, select the **Install License From File...** selection.

Checking In a Remote License

You wish to check in a remote license (typically a license for an isolated system that does not have network access to the license server) rather than waiting the license to expire. To check in a license for another system:

1 From the remote (isolated) system that has the checked-out license, enter `getkey -r` to create a license return file named 'hostname.ret' to check in a license to the MLM license server by a proxy machine.

2 Physically move the 'hostname.ret' file from the remote system to your proxy system.

3 On the proxy system, enter `getkey -r<return_file>` to check-in the license for the other system.

If `<return_file>` is a directory, such as `-r.` for the current directory, then `getkey` processes every file in that directory with the '.ret' extension.

Updating Remote Systems

Medusa Labs Test Tools give you the ability to update remote systems from Medusa Labs Test Tools Graphical User Interface.

To update a remote system:

1   Open Medusa Labs Test Tools.

2   Go to **File > Update Remote Systems**. This will open the **Update Remote Systems** dialog box.

Figure 10: Install Updates Dialog Box
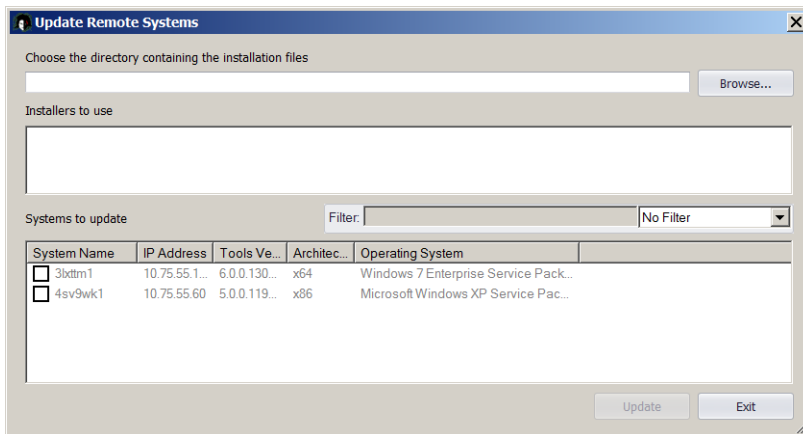
3   Click **Browse** to search for the installation files to update the remote system. Available installers will be displayed in the **Installers to use** panel.

4   Select the installer to use. The **Systems to update** panel will be automatically updated with compatible remote systems.

> **Note:** Please note that remote systems that are incompatible with the installers will be excluded automatically. For example, Windows systems will not be listed if you have selected Solaris installers only. Also, those systems already having the same or newer Medusa Labs Test Tools version installed will be grayed out.

5   Click *Update* to begin the updating process.

If you want to update a remote system residing on a different subnet from your local system, you need to connect to the remote subnet first. The remote system on the other subnet must also have a compatible installation of Medusa Labs Test Tools.

To connect to a different subnet:

1   Close the **Install Updates** dialog box.

2   Right-click anywhere on the Targets area and select **Connect to Remote Subnet** from the context menu. The **Specify Remote System** dialog box appears.

3   Enter the hostname or IP address of a system on the remote subnet.

4   Click *Connect* to connect to the remote system.

Troubleshooting

Windows Data Execution Protection

The Windows Data Execution Protection feature may prohibit Medusa Labs Test Tools services from running. The tools need a service called maui.exe to run on Windows systems. You may be prompted by Windows to allow the service to run after initial installation.

Software Firewalls

Windows or other firewalls may prevent the remote testing features or licensing from running. We recommend disabling software firewalls on your test systems.

Ping to Check Server

Make sure that the server is online and responds to a ping. If the server answers a ping, but checkouts still fail, there might be an issue with a firewall or proxy server on the network that is blocking the request. Medusa Labs Test Tools and the license server communicate over TCP/IP, using port 5033 for the Medusa License Manager.

Ensure that your network configuration allows this traffic to pass.

Check License Server Service

Make sure that the following service is running on your license server system:

```
Medusa License Manager Service
```

If the server responds to a ping, but checkouts still fail, it is possible that the service is stopped.

Check Server and Port Values

Medusa Labs Test Tools read license server information from the configuration file `MedusaTools.cfg` in the application's installation directory.

- Using the Medusa License Manager, make sure that MLM_SERVER and MLM_PORT values are correct. The server name should be a host name that resolves to the system running the license server and the default port is 5033 as shown below:

```
MLM_SERVER=10.23.13.2;
MLM_PORT=5033;
```

Check Server Dongle

The USB security dongle provided with the license server software must be attached for the license server to function. Make sure the provided driver for the USB security dongle is correctly installed. Run `mlmadmin -reportserverstate` to see that the USB security dongle is attached and visible.

Check System Time

Ensure that system time and date is accurate on both the client and server systems. Changing the system time while a license checkout is in progress on a system may cause the license to be invalidated on that system.

Check System Name

Each server running Medusa Labs Test Tools must have a unique name. Licensing failures may be the result of duplicate system names on the network.

Running a Test

When you enter a test tool name and command line switch, then press **Enter**, you run a test. If no command line switches are specified, the tools will begin running with default settings to one or more files that will be created in the current working directory. By default, Medusa Labs Test Tools run until you manually stop them or an unrecoverable error occurs. For more information, refer to the *Medusa Labs Test Tools Suite User's Guide*, Chapter 3, "Using the Command Line Switches."

Stopping the Application

Use one of the following methods to stop running an instance of Medusa Labs Test Tools:

- Using the keyboard, press **Ctrl+C** simultaneously.
- In the GUI, use the mouse to click **Stop** or **Stop All**.

Getting Help

Help is available online after you launch the application in the command line. To view a list of the available Medusa Labs Test Tools switches, type the name of the Test Tool and `-?` or `-h`, then press **Enter**. For example:

```
pain -? -or- pain -h
```

The *Medusa Labs Test Tools Suite User's Guide* is available using the GUI's Help menu

The Help is also available on the installation disk.

Index

T

test tools, stopping, 36
test, running, 36
troubleshooting, 34
troubleshooting virtual machine licensing, 26

U

updating remote systems, 33
URL web page, 9
USB key, HASP, 11
utility
   FindLBA, 4
   GetKey, 4

V

virtual machine licensing, 24

W

web page URL, 9
Windows installation, 14 to 16
Windows operating systems, 10

JDSU