

Lattice Diamond 3.1 Installation Notice for Linux



January 2014

Copyright

Copyright © 2014 Lattice Semiconductor Corporation.

This document may not, in whole or part, be copied, photocopied, reproduced, translated, or reduced to any electronic medium or machine-readable form without prior written consent from Lattice Semiconductor Corporation.

Trademarks

Lattice Semiconductor Corporation, L Lattice Semiconductor Corporation (logo), L (stylized), L (design), Lattice (design), LSC, CleanClock, Custom Mobile Device, DiePlus, E²CMOS, Extreme Performance, FlashBAK, FlexiClock, flexiFLASH, flexiMAC, flexiPCS, FreedomChip, GAL, GDX, Generic Array Logic, HDL Explorer, iCE Dice, iCE40, iCE65, iCEblink, iCEcable, iCEchip, iCEcube, iCEcube2, iCEman, iCEprog, iCEsab, iCEsocket, IPexpress, ISP, ispATE, ispClock, ispDOWNLOAD, ispGAL, ispGDS, ispGDX, ispGDX2, ispGDXV, ispGENERATOR, ispJTAG, ispLEVER, ispLeverCORE, ispLSI, ispMACH, ispPAC, ispTRACY, ispTURBO, ispVIRTUAL MACHINE, ispVM, ispXP, ispXPGA, ispXPLD, Lattice Diamond, LatticeCORE, LatticeEC, LatticeECP, LatticeECP-DSP, LatticeECP2, LatticeECP2M, LatticeECP3, LatticeECP4, LatticeMico, LatticeMico8, LatticeMico32, LatticeSC, LatticeSCM, LatticeXP, LatticeXP2, MACH, MachXO, MachXO2, MachXO3, MACO, mobileFPGA, ORCA, PAC, PAC-Designer, PAL, Performance Analyst, Platform Manager, ProcessorPM, PURESPEED, Reveal, SensorExtender, SiliconBlue, Silicon Forest, Speedlocked, Speed Locking, SuperBIG, SuperCOOL, SuperFAST, SuperWIDE, sysCLOCK, sysCONFIG, sysDSP, sysHSI, sysI/O, sysMEM, The Simple Machine for Complex Design, TracelD, TransFR, UltraMOS, and specific product designations are either registered trademarks or trademarks of Lattice Semiconductor Corporation or its subsidiaries in the United States and/or other countries. ISP, Bringing the Best Together, and More of the Best are service marks of Lattice Semiconductor Corporation.

Other product names used in this publication are for identification purposes only and may be trademarks of their respective companies.

Disclaimers

NO WARRANTIES: THE INFORMATION PROVIDED IN THIS DOCUMENT IS “AS IS” WITHOUT ANY EXPRESS OR IMPLIED WARRANTY OF ANY KIND INCLUDING WARRANTIES OF ACCURACY, COMPLETENESS, MERCHANTABILITY, NONINFRINGEMENT OF INTELLECTUAL PROPERTY, OR FITNESS FOR ANY PARTICULAR PURPOSE. IN NO EVENT WILL LATTICE SEMICONDUCTOR CORPORATION (LSC) OR ITS SUPPLIERS BE LIABLE FOR ANY DAMAGES WHATSOEVER (WHETHER DIRECT, INDIRECT, SPECIAL, INCIDENTAL, OR CONSEQUENTIAL, INCLUDING, WITHOUT LIMITATION, DAMAGES FOR LOSS OF PROFITS, BUSINESS INTERRUPTION, OR LOSS OF INFORMATION) ARISING OUT OF THE USE OF OR INABILITY TO USE THE INFORMATION PROVIDED IN THIS DOCUMENT, EVEN IF LSC HAS BEEN ADVISED OF THE POSSIBILITY OF SUCH DAMAGES. BECAUSE SOME JURISDICTIONS PROHIBIT THE EXCLUSION OR LIMITATION OF CERTAIN LIABILITY, SOME OF THE ABOVE LIMITATIONS MAY NOT APPLY TO YOU.

LSC may make changes to these materials, specifications, or information, or to the products described herein, at any time without notice. LSC makes no commitment to update this documentation. LSC reserves the right to discontinue any product or service without notice and assumes no obligation to correct any errors contained herein or to advise any user of this document of any correction if such be made. LSC recommends its customers obtain the latest version of the relevant information to establish, before ordering, that the information being relied upon is current.

Type Conventions Used in This Document

Convention	Meaning or Use
Bold	Items in the user interface that you select or click. Text that you type into the user interface.
<i><Italic></i>	Variables in commands, code syntax, and path names.
Ctrl+L	Press the two keys at the same time.
<code>Courier</code>	Code examples. Messages, reports, and prompts from the software.
<code>...</code>	Omitted material in a line of code.
<code>.</code> <code>.</code> <code>.</code>	Omitted lines in code and report examples.
[]	Optional items in syntax descriptions. In bus specifications, the brackets are required.
()	Grouped items in syntax descriptions.
{ }	Repeatable items in syntax descriptions.
	A choice between items in syntax descriptions.

Contents

Installing Diamond Tools	1
System Requirements	1
Memory Requirements	2
Required Support Libraries	3
Contacting Technical Support	5
Installer Contents	5
Diamond Installed Directory Structure	6
Installing Lattice Diamond on 32-Bit Platforms	7
Installing Diamond on Red Hat 64-Bit Platforms	8
Diamond 3.1 Release System Library Dependencies	10
Using an Automatic Package Installer for RPM Systems	11
Installing Diamond Tools from RPM Without Root Access	12
Red Hat 4 Users	12
Red Hat 5 and 6 Users	12
Installing Stand-Alone Reveal Analyzer	13
Installing Stand-Alone Programmer	14
Installing Stand-Alone Power Estimator	15
Installing and Configuring USB Cable and Parallel Cable	16
Licensing Lattice Diamond and Stand-Alone Power Estimator	22
License Management Versions	23
Obtaining a License	23
Editing the License File	23
Starting the License Manager	24
Stopping the License Manager	24
Installing and Running the License Manager on a Remote Server	25
Setting Up a Floating License on Linux	26
Troubleshooting a Floating License	26
Running the Lattice Diamond GUI	27
Using the Examples Directory	28
Finding the Installation History	28

Running Lattice Diamond from the Command Line	28
Running Stand-Alone Tcl Console	28
Running Using CSH or Bash Interpreters	29
Running Synplify Pro for Lattice	29
Running Stand-Alone Reveal Analyzer	29
Running Stand-Alone Programmer	30
Running Stand-Alone Deployment Tool	30
Running Stand-Alone Debugger	30
Running Stand-Alone Programming File Utility	31
Running Stand-Alone Model 300 Programmer	31
Running Stand-Alone Power Estimator	31
Running Multiple Versions of Lattice Software	31
Updating Lattice Diamond	32
Running the UPDATE Tool	32
Changing the Network Setting	32
Downloading and Installing Software Updates	33
Installing LatticeMico Development Tools	35
System and Software Requirements	35
System Requirements	35
Software Requirements	35
Additional Notes for Linux	36
Installing LatticeMico Within Diamond Software	36
Installing LatticeMico as a Stand-Alone Tool	37
LatticeMico System Library Dependencies	38
Installing 32-Bit System Library Packages Manually on Red Hat 6 64-bit Operating Systems	40
Using an Automatic Package Installer to Install LatticeMico RPM Package	42
Running LatticeMico System	42

Installing Diamond Tools

This document provides installation instructions for the Lattice Diamond[®] Linux software.

Diamond supports LatticeEC[™], LatticeECP[™], LatticeECP2[™], LatticeECP2M[™], LatticeECP3[™], LatticeSC[™], LatticeSCM[™], LatticeXP[™], LatticeXP2[™], MachXO[™], MachXO2[™], MachXO3L, Platform Manager[™], and Platform Manager 2 designs.

Note

The available devices vary depending on the type of license.

System Requirements

The following are the basic system requirements for Lattice Diamond on Linux:

- ▶ Intel Pentium or Pentium-compatible PC, or AMD Opteron system support:
For Programmer:
 - ▶ The Lattice Diamond 32-bit release contains a 32-bit driver to support the 32-bit system.
 - ▶ The Lattice Diamond 64-bit release contains a 64-bit driver to support the 64-bit system.
- ▶ Red Hat Enterprise Linux version 4.X, 5.X, 6.X, or Novell SUSE Linux Enterprise 10 SP1 or 11 operating system.

The host operating system can be either 32-bit or 64-bit.

Note

Novell SUSE Linux supports 32-bit only.

- ▶ The Lattice Diamond software has 32 bit and 64 bit specific application programs to run on Red Hat 32 / 64 platforms.
- ▶ The Lattice Diamond software is available in 32-bit and 64-bit versions. The 32-bit version can be installed on both 32-bit and 64-bit systems. If the 32-bit version is installed on a 64-bit system, the correct 32-bit libraries must be installed. See “Installer Contents” on page 5 for the list of libraries.
The 64-bit version of Lattice Diamond can only be installed on 64-bit systems.
- ▶ Approximately 5.75 GB free disk space
- ▶ Network adapter and network connectivity

Note

A floating license requires access to the license server, so both a network adapter and connectivity are required.

- ▶ 1024 x 768 graphics display
- ▶ JScript-capable Web browser
- ▶ Adobe Acrobat Reader, or equivalent PDF reader

Memory Requirements

Table 1 lists the minimum memory requirements and recommended memory for all the Lattice Semiconductor FPGA families.

Table 1: Recommended Memory for Linux

Device	Size	32-Bit Operating Systems		64-Bit Operating Systems	
		Minimum	Recommended	Minimum	Recommended
LatticeEC, LatticeECP	Up to 20K LUT	512 MB	768 MB	1 GB	1.5 GB
	Up to 50K LUT	768 MB	1 GB	1.5 GB	2 GB
LatticeECP2/M	Up to 20K LUT	768 MB	1 GB	1.5 GB	2 GB
	Up to 50K LUT	1 GB	1.5 GB	2 GB	3 GB
	Up to 100K LUT	1 GB	2 GB	2 GB	4 GB
LatticeECP3	Up to 95K LUT	2 GB	3 GB	4 GB	6 GB
	Up to 150K LUT	3 GB	4 GB	6 GB	8 GB
LatticeSC/M	Up to 40K LUT	768 MB	1 GB	1.5 GB	2 GB
	Up to 115K LUT	1 GB	2.5 GB	2 GB	5 GB

Table 1: Recommended Memory for Linux (Continued)

Device	Size	32-Bit Operating Systems		64-Bit Operating Systems	
		Minimum	Recommended	Minimum	Recommended
LatticeXP, LatticeXP2	Up to 20K LUT	512 MB	768 MB	1 GB	1.5 GB
	Up to 50K LUT	768 MB	1 GB	1.5 GB	2 GB
MachXO, MachXO2, MachXO3L	All	256 MB	512 MB	512 MB	1 GB
Platform Manager, Platform Manager 2	All	256 MB	512 MB	512 MB	1 GB

Required Support Libraries

Diamond is packaged in Red Hat Package Manager format (RPM). Users must download and install specific operating system PERL RPMs. The exact file names are dependent upon the user's operating system. In order to install Diamond, several support libraries must be acquired from the operating system vendor's repository. See the following tables.

Note

It is the responsibility of the person installing the software to ensure that all of the required libraries are present.

Table 2: Diamond Base

lld-linux.so.2	libICE.so.6	libutil.so.1
libasound.so.2	libjpeg.so.62	libusb-0.1.so.4
libatk-1.0.so.0	libm.so.6	libX11.so.6
libc.so.6	libnsl.so.1	libXcursor.so.1
libdl.so.2	libpango-1.0.so.0	libXext.so.6
libexpat.so.0	libpangox-1.0.so.0	libXft.so.2
libfontconfig.so.1	libpangoxft-1.0.so.0	libXi.so.6
libfreetype.so.6	libpthread.so.0	libXinerama.so.1
libgdk_pixbuf-2.0.so.0	librt.so.1	libXmu.so.6
libgdk-x11-2.0.so.0	libSM.so.6	libXrender.so.1
libglib-2.0.so.0	libstdc++.so.5	libXtst.so.6
libgthread-2.0.so.0	libthread_db.so.1	

Table 3: LatticeMico System

libasound.so.2	libjpeg.so.62	libsoftokn3.so
libatk-1.0.so.0	libm.so.6	libssl3.so
libc.so.6	libncurses.so.5	libstdc++.so.5
libdl.so.2	libnsl.so.1	libthread_db.so.1
libexpat.so.0	libnspr4.so	libusb-0.1.so.4
libfontconfig.so.1	libnss3.so	libX11.so.6
libfreetype.so.6	libpango-1.0.so.0	libXext.so.6
libgdk_pixbuf-2.0.so.0	libpangox-1.0.so.0	libXft.so.2
libgdk-x11-2.0.so.0	libpangoxft-1.0.so.0	libXi.so.6
libglib-2.0.so.0	libplc4.so	libXrender.so.1
libgmodule-2.0.so.0	libplds4.so	libXss.so.1
libgmp.so.3	libpng12.so.0	libXt.so.6
libgobject-2.0.so.0	libpthread.so.0	libXtst.so.6
libgthread-2.0.so.0	librt.so.1	libz.so.1
libgtk-x11-2.0.so.0	libsmime3.so	
libICE.so.6	libSM.so.6	

Table 4: Reveal Stand-Alone

libc.so.6
libdl.so.2
libpthread.so.0
librt.so.1
libusb-0.1.so.4

Table 5: Programmer Stand-Alone

libc.so.6
libdl.so.2
libpthread.so.0
librt.so.1
libusb-0.1.so.4

Table 6: Power Estimator Stand-Alone

linux.rpm	libdl.so.2	librt.so.1
libatk-1.0.so.0	libexpat.so.0	libtiff.so.3
libc.so.6	libpthread.so.0	libusb-0.1.so.4

Contacting Technical Support

FAQs The first place to look. The [Answer Database](#) provides solutions to questions that many of our customers have already asked. Lattice Applications Engineers are continuously adding to the Database.

Telephone Support Hotline Receive direct technical support for all Lattice products by calling Lattice Applications from 5:30 a.m. to 6 p.m. Pacific Time.

- ▶ For USA & Canada: 1-800-LATTICE (528-8423)
- ▶ For other locations: +1 503 268 8001

In Asia, call Lattice Applications from 8:30 a.m. to 5:30 p.m. Beijing Time (CST), +0800 UTC. Chinese and English language only.

- ▶ For Asia: +86 21 52989090

E-mail Support

- ▶ techsupport@latticesemi.com
- ▶ techsupport-asia@latticesemi.com

For Local Support Contact your nearest [Lattice Sales Office](#).

Installer Contents

The Lattice Diamond RPM installation packages are available for download from <http://www.latticesemi.com/latticediamond>. Click the **Downloads** tab. Some documents and downloads are not visible to anonymous visitors. To view all items, please log in to your Lattice account. Follow the product download instructions and uncompress the software.

The following describes the contents of the Lattice Diamond installer.

- ▶ The Lattice Diamond, Reveal, Programmer, and Power Estimator Linux installer rpm files have the following convention for 32-bit or 64-bit platforms.
 - ▶ `[product]_[version]-i386-linux.rpm` - 32-bit platform
 - ▶ `[product]_[version]-x86_64-linux.rpm` - 64-bit platform
- ▶ The Lattice Diamond, Reveal, Programmer, and Power Estimator rpm file names are as follows:
 - ▶ `diamond_3_1-base_x64-96-x86_64-linux.rpm`
 - ▶ `diamond_3_1-base-96-i386-linux.rpm`
 - ▶ `powerestimator_3_1_x64-96-x86_64-linux.rpm`
 - ▶ `powerestimator_3_1-96-i386-linux.rpm`
 - ▶ `programmer_3_1_x64-96-x86_64-linux.rpm`
 - ▶ `programmer_3_1-96-i386-linux.rpm`
 - ▶ `reveal_3_1_x64-96-x86_64-linux.rpm`

- ▶ reveal_3_1-96-i386-linux.rpm
- ▶ The Lattice Diamond “base” installation file (.rpm) installs Lattice Diamond and Synplify Pro for Lattice. See “Installing Lattice Diamond on 32-Bit Platforms” on page 7 or “Installing Diamond on Red Hat 64-Bit Platforms” on page 8 for more information.
- ▶ The Reveal installation file (.rpm) installs the stand-alone Reveal Analyzer. It is not necessary to install this package if the Diamond Base RPM is installed. See “Installing Stand-Alone Reveal Analyzer” on page 13 for more information.
- ▶ The Programmer installation file (.rpm) installs the stand-alone Diamond Programmer. It is not necessary to install this package if the Diamond Base RPM is installed. See “Installing Stand-Alone Programmer” on page 14 for more information.
- ▶ The Power Estimator (.rpm) file installs the stand-alone Diamond Power Estimator. It is not necessary to install this package if the Diamond Base RPM is installed. See “Installing Stand-Alone Power Estimator” on page 15 for more information.
- ▶ The vmpar64 (.rpm) file installs the addition files needed to allow 64-bit Linux systems to enable the parallel port to be able to use the Lattice parallel programming cable. See the “Lattice USB2A USB Download Cable Configuration” section of “Parallel Programming Cable Configuration” on page 16.
- ▶ The LatticeMico installation file (.rpm) installs LatticeMico Development Tools and GNU-based Compiler Tools. See “Installing LatticeMico Development Tools” on page 35 for more information.
- ▶ The Installation Notice contains installation and licensing information.

Diamond Installed Directory Structure

Table 7 shows the Lattice Diamond software directory structure after installation:

Table 7: The Diamond Software Directory Structure

File or Directory	Description
bin/lin/	This directory contains files for running the Lattice Diamond 32-bit GUI.
bin/lin64/	This directory contains files for running the Lattice Diamond 64-bit GUI.
cae_library/	This directory contains synthesis header libraries for Synplify Pro in both Verilog and VHDL formats. It also contains the libraries for interface kits (ispLSI [®] design and simulation libraries).
data/	This directory contains files for the Lattice Diamond software.
docs/	This directory contains Lattice Diamond documentation, including manuals, tutorials, and the online Help.

Table 7: The Diamond Software Directory Structure (Continued)

File or Directory	Description
embedded_source/	This directory contains source code files for the JTAG Full VME, JTAG Slim VME, Slave SPI, I2C, and sysCONFIG embedded systems for Lattice FPGA devices.
examples/	This directory contains Lattice design examples.
ispfpga/	This directory contains files for the Lattice Diamond software.
license/	This directory contains the license agreement.
module/	This directory contains Lattice module files.
synpbase/	This directory contains files for the Synplify Pro for Lattice software.
tcltk/	This directory contains tcl/tk related files.

Installing Lattice Diamond on 32-Bit Platforms

Before installing the Lattice Diamond software, download the installation files from the Lattice Semiconductor Web site. For more information on how to download the Diamond software, go to <http://www.latticesemi.com/latticediamond>. Click the **Downloads** tab. Some documents and downloads are not visible to anonymous visitors. To view all items, please log in to your Lattice account. Follow the product download instructions and uncompress the software.

Note

Platform Manager 2 devices require the Diamond Platform Designer tool. The LatticeMico Development Tools must be installed along with Diamond in order to use Platform Designer. Refer to “Installing LatticeMico Development Tools” on page 27.

RPM installation files are provided to simplify the installation process. The RPM file has a companion MD5 checksum file. The MD5 file permits you to validate the integrity of the RPM file.

Note

- ▶ You need root privilege to install an RPM package.
- ▶ In the Diamond installation, blank space characters are not allowed in the installation directory path name.

Lattice Diamond for 32-bit systems is installed with the **diamond_3_1-base-96-i386-linux.rpm** installation file. The file installs all Lattice Semiconductor FPGA devices on your 32-bit system. It also installs the Synplify Pro for Lattice software. The following section guides you through the installation procedure step by step.

To install the Diamond software on 32-bit systems:

1. Verify the integrity of the first RPM file. Make sure the companion MD5 file is located in the same directory as the RPM file.

```
cd <directory_with_RPM>
rpm -K diamond_3_1-base-96-i386-linux.rpm
```

2. Install the Diamond software. The default installation directory for Diamond is /usr/local/diamond. You have the option of changing the installation directory.

- ▶ To install to the default location (/usr/local/diamond):

```
sudo rpm -Uvh <path_to_package>/diamond_3_1-base-96-i386-
linux.rpm
```

- ▶ To override the default location:

```
sudo rpm -Uvh --prefix <install_path> <path_to_package>/
diamond_3_1-base-96-i386-linux.rpm
```

Note

The `--prefix` can not be used to install directly to Network Attached Storage. The `--prefix` must point to a location writable by the `sudo` command.

If networked installation is desired, install Diamond locally, and then transfer the file structure to your Network Attached Storage system.

Note

The RPM installation process for Diamond verifies the Linux operating system has all of the required support libraries installed. If library files required for running Diamond are not installed on the system Diamond will not be installed.

If you want to force the installation to continue, you can use “`--nodeps`” on the `rpm` command to ignore the dependency information. Diamond may not function correctly if the “`--nodeps`” switch is used to force installation of Diamond to proceed.

Installing Diamond on Red Hat 64-Bit Platforms

Before installing the Lattice Diamond software, download the installation files from the Lattice Semiconductor Web site. For more information on how to download the Diamond software, go to <http://www.latticesemi.com/latticediamond>. Click the **Downloads** tab. Some documents and downloads are not visible to anonymous visitors. To view all items, please log in to your Lattice account. Follow the product download instructions and uncompress the software.

RPM installation files are provided to simplify the installation process. The RPM file has a companion MD5 checksum file. The MD5 file permits you to validate the integrity of the RPM file.

Note

- ▶ You need root privilege to install an RPM package.
 - ▶ In the Diamond installation, blank space characters are not allowed in the installation directory path name.
-

Lattice Diamond for 64-bit systems is installed with the **diamond_3_1-base-96-x86_64-linux.rpm** installation file. The file installs all Lattice Semiconductor FPGA devices on your 64-bit system. It also installs the Synplify Pro for Lattice software.

To install the Diamond software on 64-bit systems:

1. Verify the integrity of the first RPM file. Make sure the companion MD5 file is located in the same directory as the RPM file.

```
cd <directory_with_RPM>
rpm -K diamond_3_1-base-96-x86_64-linux.rpm
```

2. Install the Diamond software. The default installation directory for Diamond is `/usr/local/diamond`. You have the option of changing the installation directory.

- ▶ To install to the default location (`/usr/local/diamond`):

```
sudo rpm -Uvh <path_to_package>/diamond_3_1-base-96-x86_64-linux.rpm
```

- ▶ To override the default location:

```
sudo rpm -Uvh --prefix <install_path> <path_to_package>/diamond_3_1-base-96-x86_64-linux.rpm
```

Note

The `--prefix` can not be used to install directly to Network Attached Storage. The `--prefix` must point to a location writable by the `sudo` command.

If networked installation is desired, install Diamond locally, and then transfer the file structure to your Network Attached Storage system.

Note

The RPM installation process for Diamond verifies the Linux operating system has all of the required support libraries installed. If library files required for running Diamond are not installed on the system Diamond will not be installed.

If you want to force the installation to continue, you can use `--nodeps` on the `rpm` command to ignore the dependency information. Diamond may not function correctly if the `--nodeps` switch is used to force installation of Diamond to proceed.

Diamond 3.1 Release System Library Dependencies

The Diamond RPM package depends on a set of packages that provide 32-bit system libraries. Some of these packages may depend on lower level packages, the indirect dependencies. To fulfill this dependency, these packages must be installed before Diamond 3.1 can be installed.

The following packages are the packages `diamond_3_1-base-96-i386-linux.rpm` depends on. Indirect dependencies are listed with indented levels.

- ▶ glibc
 - ▶ nss-softokn-freebl
- ▶ libjpeg
- ▶ libtiff
 - ▶ zlib
 - ▶ libstdc++
 - ▶ libgcc
- ▶ glib2
 - ▶ libselinux
 - ▶ gamin
- ▶ libusb
- ▶ freetype
- ▶ fontconfig
 - ▶ expat
- ▶ libX11
 - ▶ libxcb
 - ▶ libXau
- ▶ libICE
 - ▶ libSM
- ▶ libuuid
- ▶ libXt
- ▶ libXext
- ▶ libXrender
- ▶ libXi
- ▶ libXft

Installing the packages one by one requires the parent package to be installed last. The package deepest in the tree must be installed first. For example, `libICE` package depends on `libSM`. The `libSM` package must be installed prior to `libICE`.

Two cyclic dependencies are found with Diamond, glibc and nss-softokn-freebl, and glib2 and gamin. Installing these packages requires the two RPM's to be specified on the same rpm command line.

Using an Automatic Package Installer for RPM Systems

If you have an officially licensed Red Hat release and register at Red Hat support site with an RHN (Red Hat Network) serial number, you can use the YUM installation mechanism.

YUM is an automatic updater and package installer/remover for rpm systems. It automatically computes dependencies and determines the order of installation for the dependent packages. It makes it easier to install necessary support libraries.

If you register with an RHN, Red Hat automatically sets up the YUM repository information on your machine. You only need to run the following command, assuming root privileges, to install diamond_3_1-base without going through the manual rpm installation process described in section 3.

```
# yum install diamond_3_1-base-96-i386-linux.rpm
```

If you do not register your operating system installation with Red Hat RHN, you still can setup a locally mounted DVD as a YUM repository. You must perform two steps.

1. Create a YUM repository file at /etc/yum.repos.d/iso.repo. Note that the baseurl must reflect your actual DVD mounted path.

```
[My-ISO-Repository]
baseurl="file:///media/RHEL_6.2_x86_64_Disc"
enabled=1
```

2. Import the GPG key files associated with the release package files.

```
# cd /media/RHEL_6.2_x86_64_Disc
# rpm --import *GPG*
```

You can run the following command to verify if the local YUM repository is setup successfully, as follows:

```
# yum repolist
...
Repository 'My-ISO-Repository' is missing name in
configuration, using id
My-ISO-Repository                | 4.0 kB      00:00 ...
My-ISO-Repository/primary_db     | 3.0 MB      00:00 ...
repo id                           repo name      status
My-ISO-Repository                My-ISO-Repository  3,480
repolist: 3,480
```

You can consult the following two web sites for further information about YUM.

<http://yum.baseurl.org/>

http://docs.redhat.com/docs/en-US/Red_Hat_Enterprise_Linux/6/html/Deployment_Guide/ch-yum.html

Installing Diamond Tools from RPM Without Root Access

The Diamond tools can be installed from RPM without requiring root access. One application of this is installing directly to a remote filesystem, allowing one installation to serve multiple Linux client machines. In order to install directly to the network the RPM installation must be performed without elevated permissions. Use the following procedures to install all of the Diamond tools to a remote filesystem.

The commands shown should be run from a system that is similar/identical to the one to be used by the Diamond users.

Red Hat 4 Users

```
cd ~
mkdir -p myrpmdb/var/lock/rpm
```

Create a temporary RPM database

```
/usr/bin/rpmbd --initdb --root $HOME/myrpmdb --dbpath $HOME/myrpmdb
```

Check to see if the Linux installation is missing any support libraries.

```
/bin/rpm --test -Uvh <diamond_package>.rpm
```

Install the software

```
/bin/rpm --root $HOME/myrpmdb --initdb $HOME/myrpmdb --nodeps --prefix <path_to_remote_filesystem> -Uvh <diamond_package>.rpm
```

Red Hat 5 and 6 Users

```
cd ~
mkdir -p myrpmdb/var/lock/rpm
```

Create a temporary RPM database

```
/usr/bin/rpmbd --initdb --root $HOME/myrpmdb --dbpath $HOME/myrpmdb
```

Check to see if the Linux installation is missing any support libraries.

```
/bin/rpm --test -Uvh <diamond_package>.rpm
```

Install the software

```
/bin/rpm --initdb $HOME/myrpmdb --nodeps --prefix <path_to_remote_filesystem> -Uvh <diamond_package>.rpm
```

Once all packages are installed the temporary RPM database can be removed.

Note

This process can also be used for local filesystem installations by a non-privileged user.

Installing Stand-Alone Reveal Analyzer

Reveal Analyzer is included in the Diamond installation. If you want to use the tool without installing Lattice Diamond, use one of the following files to install the stand-alone Reveal Analyzer:

- ▶ **reveal_3_1-96-i386-linux.rpm** for 32-bit systems.
- ▶ **reveal_3_1_x64-96-x86_64-linux.rpm** for 64-bit systems.

To install the stand-alone Reveal Analyzer:

1. Verify the integrity of the RPM file. Make sure the companion MD5 file is located in the same directory as the RPM file.

For 32-bit systems:

```
cd <directory_with_RPM>
rpm -K reveal_3_1-96-i386-linux.rpm
```

For 64-bit systems:

```
cd <directory_with_RPM>
rpm -K reveal_3_1_x64-96-x86_64-linux.rpm
```

2. Install the Reveal Analyzer software. The default installation directory for Reveal Analyzer is **/usr/local/reveal**. You have the option of changing the installation directory.

- ▶ To install to the default location (/usr/local/reveal):

For 32-bit systems:

```
sudo rpm -Uvh <path_to_package>/reveal_3_1-96-i386-
linux.rpm
```

For 64-bit systems:

```
sudo rpm -Uvh <path_to_package>/reveal_3_1_x64-96-x86_64-
linux.rpm
```

- ▶ To override the default location:

```
sudo rpm -Uvh --prefix <install_path> <path_to_package>/
reveal_3_1-96-i386-linux.rpm
```

Note

The `--prefix` can not be used to install directly to Network Attached Storage. The `--prefix` must point to a location writable by the `sudo` command.

If networked installation is desired, install Diamond locally, and then transfer the file structure to your Network Attached Storage system.

Installing Stand-Alone Programmer

Programmer is included in the Diamond installation and consists of five tools:

- ▶ Programmer
- ▶ Deployment Tool
- ▶ Download Debugger
- ▶ Programming File Utility
- ▶ Model 300 Programmer

If you want to use Programmer, Deployment Tool, Download Debugger, Programming File Utility, or Model 300 Programmer without installing Lattice Diamond, use one of the following files to install the stand-alone Programmer:

- ▶ **programmer_3_1-96-i386-linux.rpm** for 32-bit systems.
- ▶ **programmer_3_1_x64-96-x86_64-linux.rpm** for 64-bit systems.

To install the stand-alone Programmer:

1. Verify the integrity of the RPM file.

For 32-bit systems:

```
cd <directory_with_RPM>
rpm -K programmer_3_1-96-i386-linux.rpm
```

For 64-bit systems:

```
cd <directory_with_RPM>
rpm -K programmer_3_1_x64-96-x86_64-linux.rpm
```

2. Install the Programmer software. The default installation directory for Programmer is **/usr/local/programmer**. You have the option of changing the installation directory.

- ▶ To install to the default location (/usr/local/programmer):

For 32-bit systems:

```
sudo rpm -Uvh <path_to_package>/programmer_3_1-96-i386-
linux.rpm
```

For 64-bit systems:

```
sudo rpm -Uvh <path_to_package>/programmer_3_1_x64-96-
x86_64-linux.rpm
```

- ▶ To override the default location:

For 32-bit systems:

```
sudo rpm -Uvh --prefix <install_path> <path_to_package>/
programmer_3_1-96-i386-linux.rpm
```

For 64-bit systems:

```
sudo rpm -Uvh --prefix <install_path> <path_to_package>/
programmer_3_1_x64-96-x86_64-linux.rpm
```

Note

The `--prefix` can not be used to install directly to Network Attached Storage. The `--prefix` must point to a location writable by the `sudo` command.

If networked installation is desired, install Diamond locally, and then transfer the file structure to your Network Attached Storage system.

Installing Stand-Alone Power Estimator

Power Estimator is included in the Diamond installation. If you want to use the tool without installing Lattice Diamond, use one of the following files to install the stand-alone Power Estimator:

- ▶ **powerestimator_3_1-96-i386-linux.rpm** for 32-bit systems.
- ▶ **powerestimator_3_1_x64-96-x86_64-linux.rpm** for 64-bit systems.

Stand-alone Power Estimator requires a license. Refer to “Licensing Lattice Diamond and Stand-Alone Power Estimator” on page 22.

To install the stand-alone Power Estimator:

1. Verify the integrity of the RPM file.

For 32-bit systems:

```
cd <directory_with_RPM>
rpm -K powerestimator_3_1-96-i386-linux.rpm
```

For 64-bit systems:

```
cd <directory_with_RPM>
rpm -K powerestimator_3_1_x64-96-x86_64-linux.rpm
```

2. Install the Power Estimator software. The default installation directory for Power Estimator is **/usr/local/powerestimator**. You have the option of changing the installation directory.

- ▶ To install to the default location (`/usr/local/powerestimator`):

For 32-bit systems:

```
sudo rpm -Uvh <path_to_package>/powerestimator_3_1-96-i386-linux.rpm
```

For 64-bit systems:

```
sudo rpm -Uvh <path_to_package>/powerestimator_3_1_x64-96-x86_64-linux.rpm
```

- ▶ To override the default location:

For 32-bit systems:

```
sudo rpm -Uvh --prefix <install_path> <path_to_package>/powerestimator_3_1-96-i386-linux.rpm
```

For 64-bit systems:

```
sudo rpm -Uvh --prefix <install_path> <path_to_package>/  
powerestimator_3_1_x64-96-x86_64-linux.rpm
```

Note

The `--prefix` can not be used to install directly to Network Attached Storage. The `--prefix` must point to a location writable by the `sudo` command.

If networked installation is desired, install Diamond locally, and then transfer the file structure to your Network Attached Storage system.

Installing and Configuring USB Cable and Parallel Cable

This section provides information on USB cable and parallel cable installation and configuration. These instructions are applicable for both 32-bit and 64-bit Linux systems.

Parallel Programming Cable Configuration

Before using the Lattice parallel programming cable, the computer's parallel port must be enabled and have the correct read/write permission. Contact the system administrator if the parallel port is not enabled or does not have the correct read/write permission.

You can check the current permissions of the parallel port by navigating to the `/dev` directory and doing a lookup on the port, as follows

```
cd /dev ls -l parport0
```

There are two main ways to give read/write permission to the parallel port. The first is to give read/write permission to the parallel port to anyone that has access to that machine. The second is more secure, where only selected individuals will have read/write permission to the parallel port that has access to that machine. Below are the descriptions on how to setup parallel port zero to have read/write permission.

Giving Parallel Port Zero Read/Write Permission to All Users

1. Log in as root.
2. Change directory to the `/dev` directory as follows:

```
cd /dev <return>
```
3. Give read/write permission to parallel port zero (parport0) as follows:

```
chmod 666 parport0 <return>
```

Giving Parallel Port Zero Read/Write Permission to Selected Individuals

There are two files to change on the machine to give selected individuals read/write permission to parallel port zero. Both files are located in the `/etc` directory. The two files are "group" and "nsswitch.conf".

1. Log in as root.
2. Change directory to the `/dev` directory as follows:

```
cd /dev <return>
```
3. Give read/write permission to parallel port zero (`parport0`) to owner and group as follows:

```
chmod 666 parport0 <return>
```
4. Change directory to the `/etc` directory as follows:

```
cd /etc <return>
```
5. Using your favorite text editor, edit the file “group” as follows:
Add the login name for all the users you want to give read/write permission for parallel port zero by adding the login names to the “`lp:x:7:`” line. This line may vary from system to system. Two examples are shown below.
Example 1: Change:

```
lp:x:7:  
to:  
lp:x:7:user_name1,user_name2,etc
```


Example 2: Change:

```
lp:x:7:daemon,lp  
to:  
lp:x:7:daemon,lp,user_name1,user_name2,etc
```


where `user_name1`, `etc.`, are the login names for all the users you want to give read/write permission for parallel port zero. Save the changes and close the “group” file.
6. Using your favorite text editor, edit the file “`nsswitch.conf`” as follows:
Look at the group line. Ensure “`files`” is there.
For example:

```
group: files nis
```
7. Reboot for the changes to take effect.

Lattice USB2A USB Download Cable Configuration

Programmer software can directly access USB devices using `libusb`, which accesses the USB file system. To verify that you have the USB library installed, do the following:

1. Plug in the USB cable.
2. On the command line, type:

```
/sbin/lssusb
```


You should see an entry similar to the following:

```
> Bus 002 Device 002: ID 1134:8001
```

3. If you receive an error when you type `/sbin/lshw`, you probably do not have `lshw` installed. To install the `lshw`, go to:

<http://libusb.sourceforge.net/>

4. Unplug the cable.

Note

The USB cable must be disconnected from the PC when performing the installations. You must have root access and be able to log in as a super user.

Driver Setup for Lattice USB2A and FTDI USB Host chip Download Cable on Red Hat Linux Enterprise 6 and Newer

Automated UDEV Method Configuration

1. Create an entry called “plugdev” in the `/etc/group` file, and then add anyone logging into the system into that group. Make sure `/usr/bin/id` shows you as a member of the `plugdev` group. Log out if required.

Example:

```
plugdev:x:501:username
```

2. Type the following command to execute the script:

```
sudo <install_path>/data/vmdata/udevsetup_rh6
```

The script should terminate with a “Setup successful” output.

Manual UDEV Method Configuration

1. Create an entry called “plugdev” in the `/etc/group` file, and then add anyone logging into the system into that group. Make sure `/usr/bin/id` shows you as a member of the `plugdev` group. Log out if required. For example:

```
plugdev:x:501:username
```

2. Create a working file called `10-local.rules`.
3. Add the following information to the `10-local.rules` file:

```
#Lattice
BUS=="usb",ACTION=="add",SYSFS{idVendor}=="1134",SYSFS{idProduct}=="8001",MODE=="0660",
,GROUP=="plugdev",SYMLINK+="lattice-%n"
#FTDI
BUS=="usb",ACTION=="add",SYSFS{idVendor}=="0403",SYSFS{idProduct}=="6010",MODE=="0660",
,GROUP=="plugdev",SYMLINK+="ftdi-%n"
SUBSYSTEM=="usb",DRIVER=="ftdi_sio",ATTRS{idVendor}=="0403",SYSFS{idProduct}=="6010",A
TTR{bInterfaceNumber}=="00",RUN+="/bin/sh -c 'basename %p >/sys/bus/usb/drivers/
ftdi_sio/unbind' "
```


Note

Each BUS and SUBSYSTEM entry must be a single line, or split using the “\” line continuation character.

4. Copy the file into the /etc/udev/rules.d/10-local.rules directory. If a file by this name already exists, simply append the working file to it. For example:

```
sudo cp 10-local.rules /etc/udev/rules.d/.
```

Or, if the file already exists:

```
sudo cat 10-local.rules >>/etc/udev/rules.d/10-local.rules
```

5. Make the new UDEV settings active, as follows:

```
sudo /usr/bin/skill -HUP udevd
```

6. Plug in the USB cable.

Driver Setup for Lattice USB2A and FTDI USB Host chip Download Cable on Red Hat Linux Enterprise 5

Automated UDEV Method Configuration

1. Execute the script as follows:

```
sudo <install_path>/data/vmdata/udevsetup
```

The script should terminate with a “Setup successful” output.

Manual UDEV Method Configuration

1. Create a working file called 10-local.rules.
2. Add the following information to the 10-local.rules file:

```
#Lattice
BUS=="usb",ACTION=="add",SYSFS{idVendor}=="1134",SYSFS{idProduct}=="8001",MODE=="0660",OWNER=="root",SYMLINK+="lattice-%n"
#FTDI
BUS=="usb",ACTION=="add",SYSFS{idVendor}=="0403",SYSFS{idProduct}=="6010",MODE=="0660",OWNER=="root",SYMLINK+="ftdi-%n"
BUS=="usb",DRIVER=="ftdi_sio",SYSFS{bInterfaceNumber}=="00",RUN+="/bin/sh -c 'basename `dirname %p` >/sys/bus/usb/drivers/ftdi_sio/unbind'"
```

Note

Each BUS and SUBSYSTEM entry must be a single line, or split using the “\” line continuation character.

3. Copy the file into the /etc/udev/rules.d/10-local.rules directory. If a file by this name already exists, simply append the working file to it. For example:

```
sudo cp 10-local.rules /etc/udev/rules.d/.
```

Or, if the file already exists:

```
sudo cat 10-local.rules >>/etc/udev/rules.d/10-local.rules
sudo chmod 755 /etc/udev/rules.d/10-local.rules
```

4. Create a working file called "lattice.perms".
5. Add the following information to the lattice.perms file:

```
<lattice>=/dev/lattice*
<console>0660 <lattice>0660 root
<ftdi>=/dev/ftdi*
<console>0660 <ftdi>0660 root
```

6. Copy the file into /etc/security/console.perms.d directory as follows:

```
sudo cp lattice.perms /etc/security/console.perms.d/.
```

Or, if the file already exists:

```
sudo cat lattice.perms >>/etc/security/console.perms.d/
lattice.perms
sudo chmod 755 /etc/security/console.perms.d/lattice.perms
```

7. Make the new UDEV settings active as follows:

```
sudo /usr/bin/skill -HUP udevd
```

Driver Setup for Lattice USB2A Download Cable on Red Hat Linux Enterprise 4 and Older

Automated Hot Plug Method Configuration:

The automated configuration script adds a line in the file /etc/hotplug/usb.usermap and creates the file /etc/hotplug/usb/usbccable.

1. Navigate to the directory where you installed Diamond. The default directory is `<install_path>/lsc/diamond/3.1`. Change directory to `<install_path>/lsc/diamond/3.1/data/vmdata`. Inside the directory is the file `usbsetup`.
2. Grant execute permissions to the `usbsetup` script by typing the following command:

```
sudo chmod 755 usbsetup
```

3. Make sure that you are in the directory where the `usbsetup` file is located.
4. Type the following command to execute the script:

```
sudo ./usbsetup
```

The script should terminate with a "Setup successful" output.

Manual Hot Plug Method Configuration

Use the following manual configuration instructions. You can also use this method for a customized setup.

1. Edit the /etc/hotplug/usb.usermap file as follows:

```
sudo vi /etc/hotplug/usb.usermap
```

- Append the following line to the end of the file:

```
usbcable 0x0003 0x1134 0x8001 0x0000 0x0000 0x00 0x00 0x00
0x00 0x00 0x00 0x00000000
```

- Use one of the following methods to create the script that will be executed each time the USB device is inserted:

- Manually create the `/etc/hotplug/usb/usbcable` file.

In the following example, the device is owned by “root,” the group is assigned to “usbusers,” and the device is readable and writable by both “root” and “usbusers.”

```
#!/bin/bash if [ "${ACTION}" = "add" ] && [ -f
"${DEVICE}" ] then
chown root "${DEVICE}" chgrp usbusers "${DEVICE}" chmod
660 "${DEVICE}"
fi
```

When you create the `usbcable` file, pay attention to the spacing since some statements are sensitive to the omission of spaces.

- Copy the Linux `usbcam` file and save it as `usbcable`.

On most versions of Linux, a `usbcam` script is available that changes the device ownership to the user who inserts the cable and logs into the console. The `usbcam` script is located in the same `/etc/hotplug/` directory.

If you prefer to use this script, copy the `usbcam` file and save it as `usbcable` as follows:

```
sudo cp /etc/hotplug/usbcam /etc/hotplug/usbcable
```

- Grant the script execute permission as follows:

```
sudo chmod 755 /etc/hotplug/usbcable
```

Note

The USB cable must be removed and reinserted whenever one user logs out and another user logs in. Otherwise, the system will still recognize the previous user as the owner of the cable, and an error message will be generated.

FTDI USB Host Chip Configuration on Red Hat Linux Enterprise 4 and Older

For Linux Enterprise 4 and older, only manual installation is available.

Manual Installation

- Add the following line to `/etc/fstab`:

```
none /proc/bus/usb usbdevfs defaults,devmode=0666 0 0
```

There have been reports that you may need to use the following command for some distributors:

```
none /proc/bus/usb usbdevfs defaults,mode=0666 0 0 (use
usbfs in 2.6 kernels)
```

2. Remount all in the fstab file

```
mount -a
```

3. Plug in the cable.

4. Unload the ftdi_sio driver if it is attached to your device.

```
sudo /sbin/rmmod ftdi_sio
```

5. Unload the usbserial driver if it is attached to your device.

```
sudo /sbin/rmmod usbserial
```

If you have problems with these procedures, check with `usbview` (search on the internet for application or it can be sent to you by FTDI) to check the USB file system is mounted properly.

Licensing Lattice Diamond and Stand-Alone Power Estimator

The Lattice Diamond development tool and stand-alone Power Estimator are licensed software. In order for you to launch the tools you have to configure a FLEXlm license. The license can be either node-locked to the local machine, or acquired from a license server accessible from a LAN connection.

Note

Diamond permits the creation of configuration bitstreams for all of Lattice Semiconductor's FPGAs and CPLDs. However, a license is required for some devices.

The default location of the license file is

`<install_path>/diamond/3.1/license/license.dat`. If this location is changed, you must set the `LM_LICENSE_FILE` environment variable to include the new path name.

Note

In order to change the `LM_LICENSE_FILE` variable, you may need to edit the `.cshrc` or `.bashrc` configuration file located in your home directory.

License Management Versions

Table 8 lists the version numbers of the FLEXIm Imutil utility, the FLEXIm Imgrd utility, and the Lattice daemon used for license management for both 32-bit and 64-bit systems.

Table 8: License Management Utility Versions

Filename	Version	Description
Imutil	11.10	FLEXIm license server utility
Imgrd	11.10	FLEXIm end-user utility
lattice	11.4	The Lattice Semiconductor licensing daemon

Obtaining a License

To register and license your Diamond software:

1. Obtain the host ID of your license server with the following command:

For 32-bit systems:

```
<install_path>/diamond/3.1/ispfpga/bin/lin/Imutil lmhostid
```

For 64-bit systems:

```
<install_path>/diamond/3.1/ispfpga/bin/lin64/Imutil lmhostid
```

2. Go to the Licensing section of the Lattice Semiconductor Web site (www.latticesemi.com/license) and follow the on-screen instructions.

Note

Lattice Semiconductor supports licensing for a single server or three redundant servers. If you are using three redundant servers, enter all three server host IDs on the License File/Registration Form.

Lattice Semiconductor will send your Diamond license file (license.dat) to you by e-mail within one working day. After you receive the license file from Lattice Semiconductor, copy the license.dat file to the Diamond license directory as follows:

```
<install_path>/diamond/3.1/license/license.dat
```

Editing the License File

You must edit a floating license file to specify the server name and the path to the Lattice daemon.

The following example shows part of a floating license file:

```
SERVER nodename 1234abcd 1710
DAEMON lattice daemon_path
FEATURE LSC_DIAMOND_A lattice 10.0 01-jan-9999 100 \
85B686493C86
```

```
VENDOR_STRING=LSC_DIAMOND_A
```

Edit the SERVER line by replacing the *nodename* with the host name and the port ID (1710). The port ID, 1710 in this example, must be assigned a TCP/IP port number that is not already in use on the server, so you might need to change it.

Edit the Lattice DAEMON line, replacing *daemon_path* with the path to Lattice Diamond. For lmgrd V11, the path is:

For 32-bit systems:

```
<install_path>/diamond/3.1/ispfpga/bin/lin
```

For 64-bit systems:

```
<install_path>/diamond/3.1/ispfpga/bin64/lin
```

When editing these lines, type them exactly as you received them. All entries are case-sensitive.

Note

The encryption codes are in hexadecimal format (digits 0-9, and lower-case letters a-f or upper-case letters A-F).

Starting the License Manager

Type the following command on one line to start the license manager daemon:

For 32-bit systems:

```
<install_path>/diamond/3.1/ispfpga/bin/lin/lmgrd  
-l <install_path>/diamond/3.1/license/license.log  
-c <install_path>/diamond/3.1/license/license.dat
```

For 64-bit systems:

```
<install_path>/diamond/3.1/ispfpga/bin/lin64/lmgrd  
-l <install_path>/diamond/3.1/license/license.log  
-c <install_path>/diamond/3.1/license/license.dat
```

Redirecting output to a log file is helpful when you debug licensing problems. The *-l* switch tells the license manager to send its output to a log file (license.log), and *-c* tells it which license to serve. The log file contains information on the status of the server and the daemon and TCP port in use. It also shows which users have checked out the license and the checkout time.

Stopping the License Manager

If it is necessary to stop the FLEXlm license manager, follow this procedure:

1. Confirm that the daemon is running by typing the following command:

```
ps -ef | grep lmgrd
```

2. If `lmgrd.exe` is running, type the following command on one line to stop the daemon:

For 32-bit systems:

```
<install_path>/diamond/3.1/ispfpga/bin/lin/lmutil lmdown -c  
<install_path>/diamond/3.1/license/license.dat
```

For 64-bit systems:

```
<install_path>/diamond/3.1/ispfpga/bin/lin64/lmutil lmdown  
-c <install_path>/diamond/3.1/license/license.dat
```

The following prompt appears:

```
Shutting down FLEXlm on nodes: <hostname>  
Are you sure? [y/n]:
```

3. Type **Y** and press **Enter** to shut down the license daemon.

Installing and Running the License Manager on a Remote Server

You can install and run the License Manager from a location other than the default directory.

To install the License Manager to a different location, copy the files `lattice`, `lmgrd`, and `lmutil` from the following directory:

```
<install_path>/diamond/3.1/ispfpga/bin/lin/
```

to the desired location. For example:

```
<remote_server_install_path>/my_machine/lattice_license/
```

Starting the License Manager from a Remote Server

Type the following command on one line to start the license manager daemon:

```
<remote_server_install_path>/lmgrd  
-l <install_path>/diamond/3.1/license/license.log  
-c <install_path>/diamond/3.1/license/license.dat
```

Stopping the License Manager on a Remote Server

If it is necessary to stop the FLEXlm license manager running on a remote server, follow this procedure:

1. Confirm that the daemon is running by typing the following command:

```
ps -ef | grep lmgrd
```

2. If `lmgrd.exe` is running, type the following command on one line to stop the daemon:

```
<remote_server_install_path>/lmutil lmdown -c  
<install_path>/diamond/3.1/license/license.dat
```

The following prompt appears:

```
Shutting down FLEXlm on nodes: <hostname>  
Are you sure? [y/n]:
```

3. Type **Y** and press **Enter** to shut down the license daemon.

Setting Up a Floating License on Linux

The licensing steps just outlined can be performed on a centralized license server. Then each client points to the license file on that machine. To gain access to the licenses on the remote license server, you need to set the environment variable `LM_LICENSE_FILE` value to `license_port_number@linux_host_name`.

Troubleshooting a Floating License

The following procedures can be useful in troubleshooting common floating license issues.

Check the Network

Use the ping command to check network status. For example, type:

```
ping <license_server>
```

If the license server is running, you can expect a return as follows:

```
64 bytes from <license_server> <IP_address>
```

To end, type CTRL+C.

Check License File with Imutil

Use the `lmutil` tool to troubleshoot the status of your license file. The `lmutil` tool is located in the following location:

For 32-bit systems:

```
<install_path>/diamond/3.1/ispfpga/bin/lin
```

For 64-bit systems:

```
<install_path>/diamond/3.1/ispfpga/bin/lin64
```

The `lmstat` command determines the features of your license file.

- ▶ The `-a` argument displays all information.
- ▶ The `-c` argument uses the specified license files.

The following is an example of using the `lmutil lmstat` command to check the license file status:

```
./lmutil lmstat -a -c <license_port>@<license_server>
```

The license status is returned, including feature lines, number of licenses issued, and licenses in use.

Running the Lattice Diamond GUI

The Lattice Diamond software has a graphical user interface (GUI). When you run the `diamond` executable file, the software automatically performs the environment setup.

To start the Lattice Diamond GUI:

1. Set up the license as follows:

If your license file (`license.dat`) is not under `<install_path>/3.1/license`, you must set the `LM_LICENSE_FILE` variable to the location of your license file. For example:

For CSH users:

```
setenv LM_LICENSE_FILE ( /<license_directory>/license.dat
$LM_LICENSE_FILE)
```

For BASH users:

```
export LM_LICENSE_FILE=$LM_LICENSE_FILE:/
<license_directory>/license.dat
```

2. Run the `diamond` executable file in the command line as follows:

For 32-bit systems:

```
<install_path>/diamond/3.1/bin/lin/diamond &
```

For 64bit systems:

```
<install_path>/diamond/3.1/bin/lin64/diamond &
```

With the `diamond` script, you can also run the following tools in stand-alone mode.

To invoke stand-alone Power Calculator, run:

```
diamond -pwc
```

To invoke stand-alone Reveal Analyzer, run:

```
diamond -rva
```

To invoke stand-alone IPexpress, run:

```
diamond -ipx
```

To invoke stand-alone Epic, run:

```
diamond -epic
```

Refer to the online Help for more information about the Diamond GUI.

Using the Examples Directory

In order to use the design examples directory, you must copy the files from the server to your local system and change the write permissions. Copy the files from the server examples directory (*<install_path>/diamond/3.1/examples*) to the equivalent path and directory on your local system.

Finding the Installation History

The Diamond software records a log of your installation history. You can find the history in the Diamond main window.

To view the installation history:

1. Open the Diamond GUI.
2. Choose **Help > About Lattice Diamond**.

Running Lattice Diamond from the Command Line

There are two ways to run from the command line: through Diamond's Tcl Console or by running executable files directly.

Running Stand-Alone Tcl Console

The Lattice Diamond development environment includes Tcl Console, which allows you to run scripts for automating common tasks. Tcl Console is also available outside of the user interface in order to run custom scripts. To launch the stand-alone Tcl Console, enter the following on a command line:

For 32-bit systems:

```
<install_path>/diamond/3.1/bin/lin/pnmainc
```

or

```
<install_path>/diamond/3.1/bin/lin/diamondc
```

For 64-bit systems:

```
<install_path>/diamond/3.1/bin/lin64/pnmainc
```

or

```
<install_path>/diamond/3.1/bin/lin64/diamondc
```

These commands configure the environment allowing all of the underlying design tools to be run. Refer to the online Help for more information about the command line.

Running Using CSH or Bash Interpreters

Before running any other commands, you need to run the following command:

For 32-bit systems:

```
cd <install_path>/diamond/3.1/bin/lin source setupenv
```

For 64-bit systems:

```
cd <install_path>/diamond/3.1/bin/lin64 source setupenv
```

BASH users must run the following commands:

For 32-bit systems:

```
export bindir=<install_path>/diamond/3.1/bin/lin  
source $bindir/diamond_env
```

For 64-bit systems:

```
export bindir=<install_path>/diamond/3.1/bin/lin64  
source $bindir/diamond_env
```

After that, you can run the executable files directly. For example, you can invoke the Diamond GUI by:

```
pnmain &
```

Or, you can run Power Calculator by running:

```
pwcmmain &
```

Running Synplify Pro for Lattice

To run Synplify Pro for Lattice, do the following:

1. Open the Diamond GUI.
2. Open any project to enable the commands on the Tools menu.
3. Choose **Tools > Synplify Pro for Lattice**.

Running Stand-Alone Reveal Analyzer

After the installation, you can invoke the stand-alone Reveal Analyzer in the command line:

For 32-bit systems:

```
% <install_path>/reveal/3.1/bin/lin/revealrva &
```

For 64-bit systems:

```
% <install_path>/reveal/3.1/bin/lin64/revealrva &
```

Running Stand-Alone Programmer

After the installation, you can invoke the stand-alone Programmer in the command line:

Running Programmer on 32-bit systems:

```
% <install_path>/programmer/3.1/bin/lin/programmer &
```

Running Programmer on 64-bit systems:

```
% <install_path>/programmer/3.1/bin/lin64/programmer &
```

Running Stand-Alone Deployment Tool

After the installation, you can invoke the stand-alone Deployment Tool in the command line:

Running Deployment Tool on 32-bit systems:

```
% <install_path>/programmer/3.1/bin/lin/deployment &
```

Running Deployment Tool on 64-bit systems:

```
% <install_path>/programmer/3.1/bin/lin64/deployment &
```

Running Stand-Alone Debugger

After the installation, you can invoke the stand-alone Debugger in the command line:

Running Download Debugger on 32-bit systems:

```
% <install_path>/programmer/3.1/bin/lin/debugger &
```

Running Download Debugger on 64-bit systems:

```
% <install_path>/programmer/3.1/bin/lin64/debugger &
```

Running Stand-Alone Programming File Utility

After the installation, you can invoke the stand-alone Programming File Utility in the command line:

Running Programming File Utility on 32-bit systems:

```
% <install_path>/programmer/3.1/bin/lin/fileutility &
```

Running Programming File Utility on 64-bit systems:

```
% <install_path>/programmer/3.1/bin/lin64/fileutility &
```

Running Stand-Alone Model 300 Programmer

After the installation, you can invoke the stand-alone Model 300 Programmer in the command line:

Running Model 300 Programmer on 32-bit systems:

```
% <install_path>/programmer/3.1/bin/lin/model300 &
```

Running Model 300 Programmer on 64-bit systems:

```
% <install_path>/programmer/3.1/bin/lin64/model300 &
```

Running Stand-Alone Power Estimator

After the installation, you can invoke the stand-alone Power Estimator in the command line:

For 32-bit systems:

```
% <install_path>/powerestimator/3.1/bin/lin/powerestimator &
```

For 64-bit systems:

```
% <install_path>/powerestimator/3.1/bin/lin64/powerestimator &
```

Note

The stand-alone Power Estimator requires a license. See “Licensing Lattice Diamond and Stand-Alone Power Estimator” on page 22.

Running Multiple Versions of Lattice Software

You can run multiple versions of ispLEVER and Lattice Diamond at the same time from either the command line or the graphical user interface. You can run current and previous versions.

Updating Lattice Diamond

After you have registered and licensed your installation, check the Lattice Semiconductor Web site for new software updates, device support, and enhancements. Make sure that you have the latest software by checking for updates regularly.

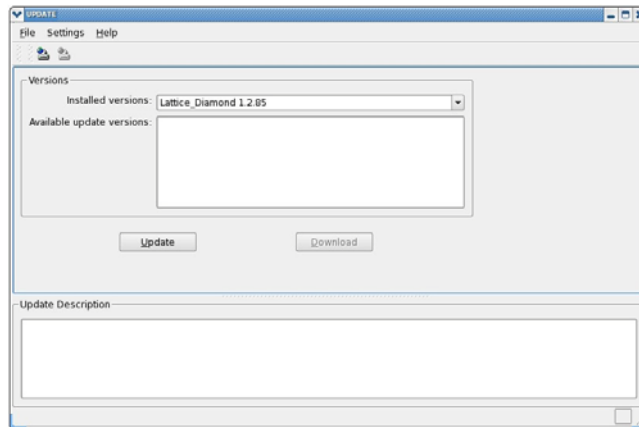
Running the UPDATE Tool

To run the UPDATE tool:

- ▶ Run the update executable file in the command line:

```
update
```

The UPDATE window appears, as shown below.

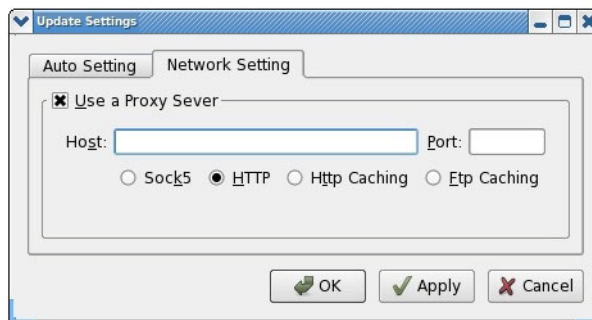


Changing the Network Setting

To enable online checking for software updates, you must indicate how your computer accesses the Internet.

To change the Internet connection settings:

1. In the UPDATE window, choose **Settings > Update Settings**.
2. In the Update Settings dialog box, select the **Network Setting** tab.



3. In the Network Setting tab, do either of the following:
 - ▶ If you must go through a proxy server before connecting to the Internet, select **Use a Proxy Server** (it is selected by default). The proxy server prevents outsiders from breaking into your organization's private network. Ask your system administrator for the URL address and port assignment.
 - ▶ If you use direct Internet access, clear the **Use a Proxy Server** option.
4. Click **OK**.

Downloading and Installing Software Updates

When you use the automatic checking feature, the Diamond software notifies you whenever an update version becomes available.

You can also manually check for software updates.

To manually check for and download a Diamond update version:

1. In the UPDATE window, click **Update**.

The software goes online to check for updates, and lists the available Diamond update versions (if any) in the Available update versions box.
2. Select the update version that you want to download, and click **Download**.

The software starts downloading the installer of the update version to your computer. You may install it later.

Note

You can also use the **Help > Check for Updates** command in the Diamond window to check for and download software update.

To install a downloaded Diamond update version:

1. Close all Diamond tools.
2. Uncompress the downloaded installer.
3. Go to the location where you put the uncompressed installer.
4. From the command line, execute the RPM installation file.

Note

You need the root privilege to install an RPM package.

Installing LatticeMico Development Tools

This document explains how to install LatticeMico Development Tools on the Linux operating system.

System and Software Requirements

Your system must meet the following minimum system and software requirements to run LatticeMico Development Tools on Linux.

System Requirements

- ▶ Intel Pentium or Pentium-compatible PC with USB port
- ▶ 32-bit Red Hat Enterprise Linux version 4.X or 5.X operating system
- ▶ 2 GB memory recommended for FPGAs
- ▶ Approximately 436 megabytes of free disk space
- ▶ 1024 x 768 graphics display

Software Requirements

- ▶ Perl version 5.8.0 or later. The following Perl modules are required:
 - ▶ XML::DOM
 - ▶ XML::Parser
 - ▶ XML::RegExp
 - ▶ Getop::Std

- ▶ Scalar::Util
- ▶ FindBin::Bin

These modules are normally found in the native version of Perl that comes with Red Hat Linux.

- ▶ PDF browser, such as Adobe Acrobat
- ▶ Mozilla or Netscape browser (optional)

Additional Notes for Linux

You may need the following information to run LatticeMico on Linux:

- ▶ If you want to configure your system to use an external browser instead of the Eclipse help browser, you can change this in the LatticeMico main window by choosing the **Window > Preferences** command.
- ▶ Eclipse requires shared libraries from the Mozilla (also known as Seamonkey) Web browser. The Eclipse software expects to find a set of IA32 shared libraries. It may be necessary for you to manually configure the MOZILLA_FIVE_HOME environment variable to point to a valid Mozilla browser library installation.
- ▶ If you cannot access the PDF files referenced in the LatticeMico online Help, you can configure LatticeMico to use an external browser by following these steps:
 - a. In the LatticeMico software, choose **Window > Preferences**.
 - b. Click on **Help**.
 - c. Select **Use external browser**.
 - d. Click on **Web Browser**.
 - e. Select **Use external Web browser**.
 - f. Select **Firefox** or **Netscape**.
 - g. Click **OK**.

Installing LatticeMico Within Diamond Software

To take advantage of the full features and functionality of the LatticeMico Development Tools, Lattice Semiconductor recommends that you install the 3.1 version of Diamond before installing the LatticeMico Development Tools.

Users designing with Platform Manager 2 devices must install LatticeMico System with Diamond in order to make the Diamond Platform Designer software functional.

The LatticeMico software is available for download from the Lattice Semiconductor Web site. For more information on how to download the LatticeMico software, go to <http://www.latticesemi.com/latticediamond> and click the **Downloads** tab. Some documents and downloads are not visible to anonymous visitors. To view all items, please log in to your Lattice account.

This section describes how to install the LatticeMico Development Tools on top of the Diamond software. These procedures assume that you have already installed Diamond 3.1.

Note

You need the root privilege to install an RPM package.

To install LatticeMico Development Tools:

1. Install Lattice Diamond 3.1 first, for example, in `<install_path>/diamond`.
2. Verify the integrity of the RPM file.

```
cd <directory_with_RPM>
rpm -K diamond_3_1-lm-96-i386-linux.rpm
```

3. Execute the RPM installation file as follows:

To install to the default location (`/usr/local/diamond`), use:

```
sudo rpm -Uvh <path_to_package>/diamond_3_1-lm-96-i386-
linux.rpm
```

To override the default location use:

```
sudo rpm -Uvh --prefix=<install_path> <path_to_package>/
diamond/3.1/micosystem
```

4. After the installation is completed, verify that the `/micosystem` directory is created in the `<install_path>/diamond/3.1` directory, for example, `/usr/local/diamond/3.1/micosystem`.
5. You can then run LatticeMico System as follows:

```
<install_path>/diamond/3.1/micosystem/LatticeMicoLauncher &
```

Installing LatticeMico as a Stand-Alone Tool

If you do not have the current version of Diamond installed, you can still install the LatticeMico Development Tools, but their functionality will be limited. Mico System Builder (MSB) will not be fully functional. You can create platforms, but the platform generator will not be fully functional because of missing Diamond executables and functions. The design-rule checker will also not be fully functional. However, the graphical user interfaces of the C/C++ Software Project Environment (C/C++ SPE) and the debug environment will be fully functional.

Note

You need the root privilege to install an RPM package.

To install LatticeMico Development Tools:

1. Verify the integrity of the RPM file.

```
cd <directory_with_RPM>
```

2. Execute the RPM installation file as follows:

To install to the default location, use:

```
sudo rpm -Uvh <path_to_package>/diamond_3_1-lm-96-i386-  
linux.rpm
```

To override the default location use:

```
sudo rpm -Uvh --prefix=<install_path> <path_to_package>/  
diamond/3.1/micosystem
```

3. After the installation is completed, verify that the /micosystem directory is created in the <install_path>/latticemicosystem/3.1 directory, for example, /usr/local/latticemicosystem/3.1/micosystem.

LatticeMico System Library Dependencies

LatticeMico System for Linux depends on a set of 32-bit system libraries that are different from the Diamond base release package.

The following packages are the packages on which diamond_3_1-lm-96-i386-linux.rpm are dependent. Indirect dependencies are listed with indented levels.

- ▶ glibc
 - ▶ nss-softokn-freebl
- ▶ libjpeg
- ▶ glib2
 - ▶ libselinux
 - ▶ gamin
- ▶ libusb
- ▶ freetype
- ▶ fontconfig
 - ▶ expat
- ▶ libX11
 - ▶ libxcb
 - ▶ libXau
- ▶ libICE
 - ▶ libSM
- ▶ libuuid
- ▶ libXt

- ▶ libXext
- ▶ libXrender
- ▶ libXi
- ▶ libXft
- ▶ libXtst
- ▶ alsa-lib
- ▶ atk
- ▶ nss
 - ▶ nss-util
 - ▶ nspr-devel
 - ▶ nspr
 - ▶ nss-softokn
 - ▶ sqlite
 - ▶ readline
 - ▶ ncurses-libs
- ▶ gmp
- ▶ pango
 - ▶ cairo
 - ▶ libthai
 - ▶ libpng
 - ▶ pixman
 - ▶ libXcomposite
 - ▶ libXfixes
 - ▶ libXdamage
 - ▶ libXinerama
 - ▶ libXrandr
- ▶ compat-expat1
- ▶ compat-libstdc++-33
- ▶ cups-libs
 - ▶ avahi
 - ▶ audit-libs
 - ▶ avahi-libs
 - ▶ dbus-libs
 - ▶ libcap
 - ▶ libattr
 - ▶ libdaemon

- ▶ libtiff
- ▶ gnutls
- ▶ keyutils-libs
- ▶ krb5-libs
- ▶ libgcrypt
- ▶ libgpg-error
- ▶ libtasn1
- ▶ gtk2
 - ▶ jasper-libs
- ▶ perl-XML-DOM
- ▶ perl-XML-RegExp

Installing 32-Bit System Library Packages Manually on Red Hat 6 64-bit Operating Systems

The following sequence shows how the 32-bit system libraries must be installed before the diamond_3_1-lm LMS package can be installed. The example assumes the Red Hat Enterprises 6 64-bit installation DVD is mounted in the following path:

```
/media/RHEL_6.2\ x86_64\ Disc
```

There are total 64 packages that must be installed manually before installing the LatticeMico package. Some of the commands below install more than one package. These packages have cyclic dependencies.

```
# cd /media/RHEL_6.2\ x86_64\ Disc/Packages/
# rpm -i glibc-2.12-1.47.el6.i686.rpm \
      nss-softoken-freebl-3.12.9-11.el6.i686.rpm
# rpm -i libjpeg-6b-46.el6.i686.rpm
# rpm -i zlib-1.2.3-27.el6.i686.rpm
# rpm -i glib2-2.22.5-6.el6.i686.rpm \
      libselinux-2.1.94-5.2.el6.i686.rpm \
      gamin-0.1.10-9.el6.i686.rpm
# rpm -i libgcc-4.4.6-3.el6.i686.rpm
# rpm -i libstdc++-4.4.6-3.el6.i686.rpm
# rpm -i libusb-0.1.12-23.el6.i686.rpm
# rpm -i freetype-2.3.11-6.el6_1.7.i686.rpm
# rpm -I expat-2.1.1-9.1.el6.i686.rpm
# rpm -i fontconfig-2.8.0-3.el6.i686.rpm
# rpm -i libSM-1.1.0-7.1.el6.i686.rpm \
      libuuid-2.17.2-12.4.el6.i686.rpm
# rpm -i libXt-1.0.7-1.el6.i686.rpm
# rpm -i libXext-1.1-3.el6.i686.rpm
# rpm -i libXrender-0.9.5-1.el6.i686.rpm
# rpm -i libXi-1.3-3.el6.i686.rpm
```

```
# rpm -i libXft-2.1.13-4.1.el6.i686.rpm
# rpm -i libXtst-1.0.99.2-3.el6.i686.rpm
# rpm -i alsa-lib-1.0.22-3.el6.i686.rpm
# rpm -i atk-1.28.0-2.el6.i686.rpm
# rpm -i nspr-4.8.8-3.el6.i686.rpm
# rpm -i nspr-devel-4.8.8-3.el6.i686.rpm
# rpm -i nss-util-3.12.10-2.el6.i686.rpm
# rpm -i ncurses-libs-5.7-3.20090208.el6.i686.rpm
# rpm -i readline-6.0-3.el6.i686.rpm
# rpm -i sqlite-3.6.20-1.el6.i686.rpm
# rpm -i nss-softokn-3.12.9-11.el6.i686.rpm
# rpm -i nss-3.12.10-16.el6.i686.rpm
# rpm -i gmp-4.3.1-7.el6.i686.rpm
# rpm -i pixman-0.18.4-1.el6_0.1.i686.rpm
# rpm -i libpng-1.2.46-1.el6_1.i686.rpm
# rpm -i cairo-1.8.8-3.1.el6.i686.rpm
# rpm -i libthai-0.1.12-3.el6.i686.rpm
# rpm -i pango-1.28.1-3.el6_0.5.i686.rpm
# rpm -i libXfixes-4.0.4-1.el6.i686.rpm
# rpm -i libXcomposite-0.4.1-2.el6.i686.rpm
# rpm -i libXdamage-1.1.2-1.el6.i686.rpm
# rpm -i libXinerama-1.1-1.el6.i686.rpm
# rpm -i libXrandr-1.3.0-4.el6.i686.rpm
# rpm -i audit-libs-2.1.3-3.el6.i686.rpm
# rpm -i dbus-libs-1.2.24-5.el6_1.i686.rpm
# rpm -i avahi-libs-0.6.25-11.el6.i686.rpm
# rpm -i libattr-2.4.44-7.el6.i686.rpm
# rpm -i libcap-2.16-5.5.el6.i686.rpm
# rpm -i libdaemon-0.14-1.el6.i686.rpm
# rpm -i avahi-0.6.25-11.el6.i686.rpm
# rpm -i libtiff-3.9.4-1.el6_0.3.i686.rpm
# rpm -i compat-expat1-1.95.8-8.el6.i686.rpm
# rpm -i compat-libstdc++-33-3.2.3-69.el6.i686.rpm
# rpm -i gnutls-2.8.5-4.el6.i686.rpm
# rpm -i keyutils-libs-1.4-3.el6.i686.rpm
# rpm -i krb5-libs-1.9-22.el6.i686.rpm
# rpm -i libgcrypt-1.4.5-9.el6.i686.rpm
# rpm -i libgpg-error-1.7-4.el6.i686.rpm
# rpm -i libtasn1-2.3-3.el6.i686.rpm
# rpm -i cups-libs-1.4.2-44.el6.i686.rpm
# rpm -i jasper-libs-1.900.1-15.el6.i686.rpm
# rpm -i gtk2-2.18.9-6.el6.i686.rpm
# rpm -i perl-XML-RegExp-0.03-7.el6.noarch.rpm
# rpm -i perl-XML-DOM-1.44-7.el6.noarch.rpm
# rpm -i diamond_3_1-lm-96-i386-linux.rpm
```

After you install the above 32-bit system library packages, the following LatticeMico package can be installed.

```
# rpm -i diamond_3_1-lm-96-i386-linux.rpm
```

Note that this example is based on Red Hat Enterprises 6 64 Update 2 release. If you have a different Red Hat Enterprises 6 64 update release, the version number may be different on your installation DVD image.

Using an Automatic Package Installer to Install LatticeMico RPM Package

If you have a YUM repository setup as described in “Using an Automatic Package Installer for RPM Systems” on page 11 you only need to run the following command to install `diamond_3_1-lm` without going through the manual RPM installation process described in “Installing 32-Bit System Library Packages Manually on Red Hat 6 64-bit Operating Systems” on page 40.

```
# yum install diamond_3_1-lm-96-i386-linux.rpm
```

Running LatticeMico System

After LatticeMico System is installed, enter the following command to run the software:

```
<install_path>/latticemicosystem/3.1/LatticeMicoLauncher &
```