PowerWorks Linux Development Environment

Version 1.4.2 Release Notes

December 2001

0898000-1.4.2

CONCURRENT COMPUTER CORPORATION

Copyright

Copyright 2001 by Concurrent Computer Corporation. All rights reserved. This publication or any part thereof is intended for use with Concurrent Computer Corporation products by Concurrent Computer Corporation personnel, customers, and end–users. It may not be reproduced in any form without the written permission of the publisher.

Disclaimer

The information contained in this document is subject to change without notice. Concurrent Computer Corporation has taken efforts to remove errors from this document, however, Concurrent Computer Corporation's only liability regarding errors that may still exist is to correct said errors upon their being made known to Concurrent Computer Corporation.

License

Duplication of this manual without the written consent of Concurrent Computer Corporation is prohibited. Any copy of this manual reproduced with permission must include the Concurrent Computer Corporation copyright notice.

Trademark Acknowledgments

PowerWorks, PowerMAXION, PowerMAX OS, TurboHawk, Power Hawk, NightBench, NightSim, NIghtTrace, NightView, and MAXAda are trademarks of Concurrent Computer Corporation.

Night Hawk is a registered trademark of Concurrent Computer Corporation.

Motorola is a registered trademark of Motorola, Inc.

PowerStack is a trademark of Motorola, Inc.

Linux is a registered trademark of Linus Torvalds.

Red Hat is a registered trademark of Red Hat, Inc.

Intel is a registered trademark of Intel Corporation.

PowerPC is a registered trademark of IBM Corp.

UNIX is a registered trademark of The Open Group.

X Window System is a trademark of The Open Group.

Contents

1.0 Introduction 1

2.0 Documentation 2

3.0 Prerequisites 3

3.1 Host System 3

3.1.1 Software 3

3.1.2 Hardware 4

3.2 Target System 5

3.2.1 Software 5

3.2.2 Hardware 5

4.0 System Installation 6

4.0.1 MAXAda Activation 9

4.1 Target Installation 10

4.2 Cross-Development Libraries and Headers 13

5.0 Getting Started 15

5.1 Concurrent C/C++ 15

5.1.1 Invoking the Compiler 15

5.1.2 Include Files and Libraries 15

5.1.3 OS Versions and Target Architectures 16

5.1.4 Shared vs. Static Linking 16

5.1.5 Makefile Considerations 16

5.2 MAXAda 16

5.2.1 Invoking The Compiler 16

5.2.2 OS Versions and Target Architectures 16

6.0 Additional Considerations 17

6.1 PLDE_LD_LIBRARY_PATH and PLDE_LD_RUN_PATH 17

6.2 Installing the PLDE on Red Hat Linux 7 18

6.2.1 Install libncurses from Red Hat Linux Installation CD 18

6.2.2 Install libncurses from FTP site 19

6.2.3 Ignore dependencies 19

6.3 Red Hat 6.2 Kernel Upgrade 20

6.4 Invoking NEdit 23

6.5 Too Many Open Files 24

6.6 Debugging Programs Which Use Shared Libraries 24

7.0 Direct Software Support 26

1.0. Introduction

The PowerWorks™ Linux Development Environment (PLDE) allows users on a Linux® PC to develop applications for Concurrent real-time computer systems. The PLDE provides cross compilation, cross linking, and cross debugging and analysis tools. Editing, compilation, linking, and scheduling, as well as debug and analysis sessions, are hosted on the Linux system while the application programs execute on a system running Concurrent's PowerMAX OS™ real-time UNIX®-based operating system.

The PowerWorks Linux Development Environment consists of high-performance Ada95 and C/C++ compilers, the NightView™ symbolic debugger, NightTrace™ event analyzer, NightSim™ frequency-based scheduler, and the NightBench™ GUI program development environment.

MAXAda™ and Concurrent C/C++ are ANSI-standard compilers especially designed for the development of large-scale real-time applications. Concurrent compilers maximize run-time performance by employing multiple levels of optimized code generation. State-of-the-art optimizations include constant folding, common sub-expression elimination, constraint propagation, moving of invariant code, reduction of operator strength, peephole optimization, and efficient instruction scheduling, vital for optimal performance in PowerPC® RISC architectures.

The Concurrent compilers come with PowerMAX OS run-time libraries that allow complete compilation and linking of applications directly on the Linux system.

NightView is a graphical source-level debugging and monitoring tool specifically designed for real-time applications. NightView can monitor, debug, and patch multiple real-time processes running on multiple processors with minimal intrusion.

NightTrace is a graphical tool for analyzing the dynamic behavior of single and multiprocessor applications. NightTrace can log application data events from simultaneous processes executing on multiple CPUs or even multiple systems. NightTrace combines application events with PowerMAX OS kernel events, presenting a synchronized view of the entire system.

NightSim is a tool for scheduling and monitoring real-time applications that require predictable, repetitive process execution. With NightSim, application builders can control and dynamically adjust the periodic execution of multiple coordinated processes, their priorities, and their CPU assignments.

NightBench is a graphical user interface that provides a common work environment for the PowerWorks Linux Development Environment editor, compilers, and development tools. NightBench organizes all of the information required for the development of PowerMAX OS applications, ensures consistent, repeatable builds, and provides an efficient interface for editing, browsing, building, and debugging.

Several of the tools include a small run-time agent that executes on the PowerMAX OS target system in a non-intrusive manner, preserving the deterministic characteristics of the application.

Utilizing the PLDE utilities on a Linux system while targeting the PowerMAX OS system offloads the heavy processing associated with compilation, linking, symbolic debug translation, and GUI network traffic from the real-time target systems.

2.0. Documentation

Table 2-1 lists the PowerWorks Linux Development Environment 1.4.2 documentation available from Concurrent.

Table 2-1. PowerWorks Linux Development Environment Version 1.4.2 Documentation

Manual Name	Pub. Number
PowerWorks Linux Development Environment Version 1.4.2 Release Notes	0898000-1.4.2
PowerWorks Linux Development Environment Tutorial	0898100-000
AXI for MAXAda Reference Manual	0890518-000
AXI for MAXAda Version 3.3.1 Release Notes (Linux)	0898518-3.3.1
C/C++ Reference Manual	0890497-030
C/C++ Version 5.3 Release Notes (Linux)	0898497-5.3
Data Monitoring Reference Manual	0890493-000
Data Monitoring Version 3.3.1 Release Notes (Linux)	0898493-3.3.1
MAXAda Reference Manual	0890516-100
MAXAda Version 3.3.1 Release Notes (Linux)	0898516-3.3.1-003
NightBench User's Guide	0890514-070
NightBench Version 2.2 Release Notes (Linux)	0898514-2.2
NightSim User's Guide	0890480-020
NightSim Version 3.2.2 Release Notes (Linux)	0898480-3.2.2
NightTrace Manual	0890398-070
NightTrace Version 4.2 Release Notes (Linux)	0898398-4.2
NightView User's Guide	0890395-220
NightView Version 5.3 Release Notes (Linux)	0898395-5.3
NightView Pocket Reference	0890475-040

Copies of the Concurrent documentation can be ordered by contacting the Concurrent Software Support Center. The toll-free number for calls within the continental United States is 1-800-245-6453. For calls outside the continental United States, the number is 1-954-283-1822 or 1-305-931-2408.

Additionally, the manuals listed above are available:

- online using the PowerWorks Linux Development Environment utility, **`nhelp`**
- in PDF format in the **`documentation`** directory of the PLDE Installation CD
- on the Concurrent Computer Corporation web site at www.ccur.com

3.0. Prerequisites

Prerequisites for PowerWorks Linux Development Environment Version 1.4.2 for both the host system and target system are as follows:

3.1. Host System

3.1.1. Software

- Red Hat® Linux*

IMPORTANT

There are special considerations that must be taken into account when installing the PowerWorks Linux Development Environment on a system running Red Hat Linux 7. See "Installing the PLDE on Red Hat Linux 7" on page 18 for more information.

- Required capabilities

NOTE

The following capabilities are normally installed as part of the standard installation of Red Hat Linux. During installation of the PLDE, the user will be notified if required capabilities do not exist on the Linux system.

* This product has been extensively tested on Red Hat Linux 6.2, 7.0, and 7.1. However, this product has not been tested with versions of Linux supplied by other vendors.

- Red Hat Linux

Capabilities	RPMs providing these capabilities
/bin/ksh **/bin/sh** **XFree86-devel** **ld-linux.so.2** **libICE.so.6** **libSM.so.6** **libX11.so.6** **libXext.so.6** **libXmu.so.6** **libXp.so.6** **libXt.so.6** **libc.so.6** **libc.so.6(GLIBC_2.0)** **libc.so.6(GLIBC_2.1)** **libdl.so.2** **libdl.so.2(GLIBC_2.0)** **libdl.so.2(GLIBC_2.1)** **libm.so.6** **libm.so.6(GLIBC_2.0)** **libm.so.6(GLIBC_2.1)** **libnbcommon.so** **libncurses.so.4** **libnsl.so.1** **libpthread.so.0** **libpthread.so.0(GLIBC_2.0)** **libpthread.so.0(GLIBC_2.1)** **libresolv.so.2** **libstdc++-libc6.1-1.so.2** **libutil.so.1** **libutil.so.1(GLIBC_2.0)** **rpm >= 3.0.3**	*Red Hat 6.2:* **XFree86-3.3.6-20** **XFree86-libs-3.3.6-20** **bash-1.14.7-22** **egcs-c++-1.1.2-30** **glibc-2.1.3-15** **libstdc++-2.9.0-30** **ncurses-5.0-11** **pdksh-5.2.14-2** **rpm-3.0.4-0.48** *Red Hat 7.0:* **XFree86-devel-4.0.1-1** **XFree86-libs-4.0.1-1** **bash-2.04-11** **compat-glibc-6.2-2.1.3.2** **compat-libstdc++-6.2-2.9.0.9** **glibc-2.1.92-14** **ncurses4-5.0-2** **pdksh-5.2.14-8** **rpm-4.0-4** *Red Hat 7.1:* **XFree86-devel-4.0.3-5** **XFree86-libs-4.0.3-5** **bash-2.04-21** **compat-glibc-6.2-2.1.3.2** **compat-libstdc++-6.2-2.9.0.14** **glibc-2.2.2-10** **ncurses4-5.0-2** **pdksh-5.2.14-12** **rpm-4.0.2-8**

3.1.2. Hardware

- an Intel®-based PC - 300 MHz or higher (recommended minimum configuration)
- 64MB physical memory (recommended minimum configuration)

3.2. Target System

3.2.1. Software

- PowerMAX OS 4.3 or later
- Required PowerMAX OS software packages (see “Target Installation” on page 10 for more information):

Package	Version
`Nviewp`	5.3
`nsimserver`	3.2.2
`ntracelog`	4.2
`c++53runtime`	5.3

3.2.2. Hardware

- Computer Systems:

  Power Hawk™ 620 and 640

  Power Hawk 710, 720 and 740

  PowerStack™ II and III

  Night Hawk® Series 6000

  TurboHawk™

  PowerMAXION™

- Board-Level Products:

  Motorola® MVME2604

  Motorola MVME4604

4.0. System Installation

A single command installs (or uninstalls) all software components of the PowerWorks Linux Development Environment.

NOTE

As mentioned in "Documentation" on page 2, release notes are provided for each of the individual components of the PLDE. These release notes include individual installation instructions for each respective component; however, separate installation of components is rarely required. The preferred method is to utilize the installation scripts described below which install ALL components of the PLDE.

NOTE

The user must be root in order to install the PowerWorks Linux Development Environment on the Linux system.

To install the PowerWorks Linux Development Environment, issue the following commands on your Linux system:

1. Insert the PowerWorks Linux Development Environment Installation CD in the CD-ROM drive
2. Mount the CD-ROM drive (assuming the standard mount entry for the CD-ROM device exists in **`/etc/fstab`**)

   ```
   mount /mnt/cdrom
   ```

3. Change the current working directory to the directory containing the PowerWorks Linux Development Environment installation scripts

   ```
   cd /mnt/cdrom/linux-i386
   ```

4. Invoke the appropriate PowerWorks Linux Development Environment installation script

NOTE

The **`PLDE.install`** script should be used for systems that do NOT already have a version of the PLDE installed.

The **`PLDE.upgrade`** script should be used on systems that already have a previous version of the PLDE installed.

Run the appropriate script:

`./PLDE.install`

or

`./PLDE.upgrade`

NOTE

The **`PLDE.upgrade`** script will update the default release of the Concurrent C/C++ compiler to version 5.3 for the system on which it is invoked. Older releases of Concurrent C/C++ previously installed on that system will still be available for use. See the section titled "Multiple Release Support" in the "Compilation" chapter of the *C/C++ Reference Manual* (0890497-030) for more information.

By default, the PLDE is mainly installed in **`/usr/opt`** (~570MB). To install in a different directory, add

`--relocate /usr/opt=`*directory*

to the **`PLDE.install`** (or **`PLDE.upgrade`**) command where *directory* is the desired directory.

IMPORTANT

Due to a bug in the standard Red Hat Linux 7.1 product installation mechanism, **`rpm`** (see **`rpm(8)`**), the **`--relocate`** option does not function properly. However, the **`--relocate`** option may be used when installing the PowerWorks Linux Development Environment on Red Hat Linux versions prior to version 7.1.

NOTE

You *must* issue the **`--relocate`** option with the **`PLDE.upgrade`** command if the **`--relocate`** option was issued with the original **`PLDE.install`** command. In this case, the *directory* specified with the **`PLDE.upgrade`** command must match the *directory* previously specified to **`PLDE.install`**.

If the **`--relocate`** option was *not* issued with the original **`PLDE.install`** command, you *cannot* use **`--relocate`** with the **`PLDE.upgrade`** command.

You may see messages similar to the following during an install (or uninstall):

```
failed to stat /nfsfilesystem: Stale NFS file handle
```

where *nfsfilesystem* may be any NFS filesystem. These messages may be ignored.

If there is not enough disk space to complete the installation, the PowerWorks Linux Development Environment installation script will issue warnings similar to the following:

```
rpm needs n MB on the filesystem filesystem
```

If this occurs, run the **PLDE.uninstall** script (ignoring any diagnostic messages that result) and then issue the **PLDE.install** command with the **--relocate** option as described above, specifying a *directory* on a filesystem with enough disk space for the complete installation.

NOTE

On some versions of Linux, the **PLDE.install** (or **PLDE.upgrade**) script may abort, claiming insufficient space in the target directory. If the target directory does, in fact, have adequate space, the user may add the **--ignore-size** option to **PLDE.install** (or **PLDE.upgrade**) in order to proceed with the installation.

Due to a bug in the previous version of the PLDE, messages similar to the following may be generated during the execution of the **PLDE.upgrade** script. These messages are harmless and may be ignored.

```
rmdir: /usr/opt/plde-pmax-crossdev-4.3/nh/usr/X11R6: Not a directory
rmdir: /usr/opt/plde-pmax-crossdev-4.3/nh/usr/X11R5: Not a directory
rmdir: /usr/opt/plde-pmax-crossdev-4.3/moto/usr/X11R6: Not a directory
rmdir: /usr/opt/plde-pmax-crossdev-4.3/moto/usr/X11R5: Not a directory
rmdir: /usr/opt/plde-pmax-crossdev-5.0/synergy/usr/X11R6: Not a directory
rmdir: /usr/opt/plde-pmax-crossdev-5.0/synergy/usr/X11R5: Not a directory
```

Due to a known bug in the standard Red Hat Linux product installation mechanism, **rpm** (see **rpm(8)**), messages similar to the following may also be generated during the execution of the **PLDE.upgrade** script. These messages are also harmless and may be ignored:

```
package plde-x11-pmax4.3 not found in conflict index
package plde-x11-pmax5.0 not found in conflict index
package plde-datamon-pmax4.3 not found in conflict index
package plde-datamon-pmax5.0 not found in conflict index
package plde-hf77libs-pmax4.3 not found in conflict index
package plde-hf77libs-pmax5.0 not found in conflict index
package plde-ntracelibs-pmax4.3 not found in conflict index
package plde-ntracelibs-pmax5.0 not found in conflict index
```

5. Change the current working directory outside the **/mnt/cdrom** hierarchy

   ```
   cd /
   ```

6. Unmount the CD-ROM drive (otherwise, you will be unable to remove the PowerWorks Linux Development Environment Installation CD from the CD-ROM drive)

   ```
   umount /mnt/cdrom
   ```

NOTE

If any of the required capabilities listed in "Prerequisites" on page 3 are not installed, the PowerWorks Linux Development Environment installation script will issue warnings. Run the **PLDE.uninstall** script (ignoring any diagnostic messages that result) and then install the necessary capabilities before issuing the **PLDE.install** script.

The PowerWorks Linux Development Environment requires that certain packages be installed on the PowerMAX OS target systems. See "Target Installation" on page 10 for more information.

To uninstall the PowerWorks Linux Development Environment, execute the following script:

PLDE.uninstall

found in the **linux-i386** directory on the PowerWorks Linux Development Environment Installation CD. (Follow the installation instructions above for mounting the CD-ROM drive, maneuvering to the correct working directory, unmounting the CD-ROM drive, etc.)

During the uninstall process, messages similar to the following may be issued:

```
package plde-x11-pmax4.3 not found in conflict index
package plde-x11-pmax5.0 not found in conflict index
package plde-datamon-pmax4.3 not found in conflict index
package plde-datamon-pmax5.0 not found in conflict index
package plde-hf77libs-pmax4.3 not found in conflict index
package plde-hf77libs-pmax5.0 not found in conflict index
package plde-ntracelibs-pmax4.3 not found in conflict index
package plde-ntracelibs-pmax5.0 not found in conflict index
```

These messages appear due to a known bug in the standard Red Hat Linux product installation mechanism, **rpm** (see **rpm(8)**), which is used by the **PLDE.uninstall** script. These messages may be ignored.

4.0.1. MAXAda Activation

The MAXAda product requires activation before it can be utilized. Installation of MAXAda automatically activates the product for a 30-day evaluation period. During that time period, you can utilize the product for purposes of evaluation only. Permanent activation requires an activation code. Contact Concurrent Software Distribution by calling 1-800-666-5405 or by faxing 1-800-666-5404. Customers outside the continental United States can contact Concurrent Software Distribution by calling 1-954-283-1836 or by faxing 1-954-283-1835. Concurrent Software Distribution may also be reached by email at softdist@ccur.com. If you purchased MAXAda, the activation code is provided on the PowerWorks Linux Development Environment Installation CD case.

Issue the following command as the root user to permanently activate the product on your system:

/usr/ada/bin/a.install -rel phase3.3.1 -activate *code*

where *code* is the code provided on the PLDE Installation CD case.

4.1. Target Installation

The PowerWorks Linux Development Environment requires that the following software packages are installed on the PowerMAX OS target systems:

NightStar Tool	Package	Version
NightView	**`Nviewp`**	5.3
NightSim	**`nsimserver`**	3.2.2
NightTrace	**`ntracelog`**	4.2

Compiler	Package	Version
Concurrent C/C++	**`c++53runtime`**	5.3

These packages are normally installed during the installation of the corresponding NightStar tool or compiler on the PowerMAX OS system. However, the packages are also included on the PowerWorks Linux Development Environment Installation CD and may be installed by following the procedures below if any of the above packages have not been installed on the target system, or if the versions installed on the target system do not conform to the above table.

IMPORTANT

The versions of the required software packages installed on the PowerMAX OS target systems *must* conform to the versions specified in the above table. If this is not the case, follow the procedures below.

NOTE

Installation of these packages will not cause incompatibilities with the current NightStar tools or compilers installed on your PowerMAX system. These new versions are preferred even when the host and targets are both running PowerMAX OS.

NOTE

If your PowerMAX system is missing any of these packages, it may first require installation of additional prerequisite packages. For more information consult the release notes for the PowerMAX OS versions of the packages listed above: NightView Version 5.3 Release Notes (PowerMAX) (0890395-5.3); NightTrace Version 4.2 Release Notes (0890398-4.2); NighSim Version 3.2.2 Release Notes (PowerMAX) (0890480-3.2.2); and Concurrent C/C++ Version 5.3 Release Notes (PowerMAX) (0890497-5.3).

If your PowerMAX OS system has a CD-ROM device:

```
mkdir /mnt/tmp
mount -F cdfs /dev/dev_name /mnt/tmp
cd /mnt/tmp/powermax-ppc604
```

If your PowerMAX OS system does not have a CD-ROM device:

On the Linux system, insert the PowerWorks Linux Development Environment Installation CD in the CD-ROM drive, mount the CD-ROM drive, and use NFS to export it to the PowerMAX OS system:

```
mount /mnt/cdrom
exportfs powermax_os_system:/mnt/cdrom
```

On the PowerMAX OS system, mount the CD-ROM as an NFS filesystem:

```
mkdir /tmp/cdrom
mount -F nfs linux_system:/mnt/cdrom /tmp/cdrom
cd /tmp/cdrom/powermax-ppc604
```

On the PowerMAX OS system, install the target portions of the NightStar tools via the following commands:

```
cp ./nightstar.Z /tmp/nightstar.Z
zcat /tmp/nightstar.Z | pkgadd -d -
```

NOTE

If the **ntracelog** or **Nviewp** 5.1 (or earlier) packages already exist on the PowerMAX OS system and their versions are different from those specified in the table above, you must first remove them using the PowerMAX OS utility **pkgrm** before installing the new versions.

However, if the **nsimserver** or **Nviewp** 5.2 (or later) packages already exist on the PowerMAX OS system and their versions are different from those specified in the table above, do NOT remove them, but instead supply the **-a check** arguments to the above **pkgadd** command. This will allow multiple versions of these packages to coexist on the system. Use of the **-a check** arguments also causes the **pkgadd** command to become extra verbose.

On the PowerMAX OS system, install the target portions of the compilers via the following commands:

```
cp ./compiler.Z /tmp/compiler.Z
zcat /tmp/compiler.Z | pkgadd -d -
```

NOTE

If the **c++53runtime** package already exists on the PowerMAX OS system and its version is different from that specified in the table above, supply the **-a check** arguments to the above **pkgadd** command. This will allow multiple versions of these packages to coexist on the system. Use of the **-a check** arguments also causes the **pkgadd** command to become extra verbose.

If your PowerMAX OS system has a CD-ROM device:

```
cd /
umount /mnt/tmp
```

If your PowerMAX OS system does not have a CD-ROM device:

On the PowerMAX OS system, unmount the NFS filesystem:

```
cd /
umount /tmp/cdrom
```

On the Linux system, unexport the NFS filesystem, and then unmount the CD-ROM:

```
cd /
exportfs -u powermax_os_system:/mnt/cdrom
umount /mnt/cdrom
```

4.2. Cross-Development Libraries and Headers

This section describes the PowerMAX OS libraries that must be installed on the Linux system in order to cross-link. These libraries are automatically installed when the **`PLDE.install`** script is used for installation (see “System Installation” on page 6 for more details).

NOTE

In rare cases, it may be necessary to install (or uninstall) these libraries independent of the installation of the PowerWorks Linux Development Environment software suite. This may be done using the standard Linux product installation mechanism, **`rpm`** (see **`rpm(8)`**). The RPMs corresponding to these libraries may be found in the **`/linux`** directory on the PowerWorks Linux Development Environment Installation CD.

The names of the RPMs containing the PowerMAX OS libraries minimally required for cross-linking are:

```
plde-pmax-crossdev-4.3
plde-pmax-crossdev-5.0
plde-pmax-crossdev-5.1
```

and are used when linking for a PowerMAX OS 4.3, PowerMAX OS 5.0, and PowerMAX OS 5.1 target system, respectively. The files associated with these RPMs are:

```
plde-pmax-crossdev-4.3-P8-1.i386.rpm
plde-pmax-crossdev-5.0-SR1-4.i386.rpm
plde-pmax-crossdev-5.1-SR2-1.i386.rpm
```

NOTE

The version number is part of the name of the RPM. Because of that, it is possible to install both RPMs on the Linux system at the same time. This allows the user to generate executables for multiple PowerMAX OS versions from the same Linux system.

The following RPMs add additional libraries, which may be needed to cross-link certain programs:

plde-pmax-crypt-4.3 **plde-pmax-crypt-5.0** **plde-pmax-crypt-5.1**	contains domestic encryption libraries (not present in international shipments) for the version of PowerMAX OS referenced in the RPM name
plde-datamon-3.3.1	contains libraries and headers for Data Monitoring 3.3.1
plde-datamon-pmax4.3 **plde-datamon-pmax5.0** **plde-datamon-pmax5.1**	allows Data Monitoring 3.3.1 to be cross-linked in programs for the version of PowerMAX OS referenced in the RPM name
plde-hf77libs-4.1	contains Fortran 77 libraries 4.1
plde-hf77libs-pmax4.3 **plde-hf77libs-pmax5.0** **plde-hf77libs-pmax5.1**	allows Fortran 77 libraries 4.1 to be cross-linked in programs for the version of PowerMAX OS referenced in the RPM name
plde-ntracelibs-4.2	contains NightTrace 4.2 libraries
plde-ntracelibs-pmax4.3 **plde-ntracelibs-pmax5.0** **plde-ntracelibs-pmax5.1**	allows NightTrace 4.2 libraries to be cross-linked in programs for the version of PowerMAX OS referenced in the RPM name
plde-x11-6.4.2	contains X11 6.4.2 libraries
plde-x11-pmax4.3 **plde-x11-pmax5.0** **plde-x11-pmax5.1**	allows X11 6.4.2 libraries to be cross-linked in programs for the version of PowerMAX OS referenced in the RPM name

5.0. Getting Started

The PowerWorks Linux Development Environment Tutorial provides stepwise instructions introducing several of the PLDE tools.

Additional considerations for getting started with MAXAda and Concurrent C/C++ compilations are covered in the *MAXAda Version 3.3.1 Release Notes* (0898516-3.3.1) and the *Concurrent C/C++ Version 5.3 Release Notes* (0898497-5.3) and are summarized in the following sections.

5.1. Concurrent C/C++

The following should be taken into consideration in order to use Concurrent C/C++ with the PowerWorks Linux Development Environment.

5.1.1. Invoking the Compiler

The Concurrent C/C++ compiler can be used via the NightBench Program Development Environment (**nbench**) or via command line invocation using **ec** or **ec++**.

You must have **/usr/ccs/bin** in your PATH environment variable to use the compiler.

Do not use **cc** or **c++**; these are native Linux compilers which are unrelated (and incompatible at the object level) with the Concurrent C/C++ cross-compiler.

See the section titled “Invoking the Compiler” in the *Concurrent C/C++ Version 5.3 Release Notes* (0898497-5.3) for more information.

NOTE

The processors **as**, **ld**, and **ar** refer to the Linux native processors. To invoke these processors independent of the C/C++ drivers, e.g. in scripts or due to Makefile activities, reference them with the appropriate **/pmax/os/***osversion***/***arch***/usr/ccs/bin** path (where *osversion* is the version of PowerMAX OS running on the target system and *arch* is the architecture of the target system - see the section titled “OS Versions and Target Architectures” in the *Concurrent C/C++ Version 5.3 Release Notes* (0898497-5.3) for more information.).

5.1.2. Include Files and Libraries

The Concurrent C/C++ compiler automatically searches for include files and libraries in the directory trees rooted at **/pmax/os**. Do not reference **/usr/include** or **/usr/lib** when building with the Concurrent C/C++ compiler; files under the **/usr/include** and **/usr/lib** directories are native Linux files and are incompatible with the corresponding PowerMAX OS files.

See the section titled “Include Files and Libraries” in the *Concurrent C/C++ Version 5.3 Release Notes* (0898497-5.3) for more information.

5.1.3. OS Versions and Target Architectures

The Concurrent C/C++ compilation system supports building programs for various PowerMAX OS versions and target systems. There are several methods available for selecting the OS version and target architecture of choice.

See the section titled "OS Versions and Target Architectures" in the *Concurrent C/C++ Version 5.3 Release Notes* (0898497-5.3) for more information.

5.1.4. Shared vs. Static Linking

By default, the Concurrent C/C++ compiler links using shared objects. Resultant PowerMAX OS C++ programs will therefore require, at a minimum, the C++ runtime shared library (**`libCruntime.so`**). If that file is not installed on your PowerMAX OS system, you can install the full PowerMAX OS version of the Concurrent C/C++ compiler, install just the **`c++53runtime`** package (as described in "Target Installation" on page 10), or relink your program using static libraries.

See the section titled "Shared vs. Static Linking" in the *Concurrent C/C++ Version 5.3 Release Notes* (0898497-5.3) for more information.

5.1.5. Makefile Considerations

When using Makefiles from a PowerMAX OS system with the PLDE, you must be careful to ensure that the appropriate compiler is invoked. Specifically, invocations of **`cc`** or **`c++`**, which may occur by default, will incorrectly invoke the native Linux compilers.

See the section titled "Makefile Considerations" in the *Concurrent C/C++ Version 5.3 Release Notes* (0898497-5.3) for solutions to this problem.

5.2. MAXAda

The following should be taken into consideration in order to use MAXAda with the PowerWorks Linux Development Environment.

5.2.1. Invoking The Compiler

The MAXAda compiler can be used via the NightBench Program Development Environment (**`nbench`**) or via command line invocation.

You must have **`/usr/ada/bin`** in your `PATH` environment variable to use the compilation system.

5.2.2. OS Versions and Target Architectures

The MAXAda compilation system supports building programs for various PowerMAX OS versions and target architectures. There are several methods available for selecting the OS version and target architecture of choice.

You *must* select an OS version and target architecture to be able to link any PowerMAX OS Ada program using the PLDE.

See the section titled "Operational Differences between PowerMAX OS and Linux" in the *MAXAda Version 3.3.1 Release Notes* (0898516-3.3.1) for more information.

6.0. Additional Considerations

6.1. PLDE_LD_LIBRARY_PATH and PLDE_LD_RUN_PATH

The PowerWorks Linux Development Environment linker, **/usr/ccs/bin/ld.pmax**, no longer recognizes the environment variables LD_LIBRARY_PATH and LD_RUN_PATH. These environment variables conflict with variables of the same name which are used by the native Linux system linker, **/bin/ld**. In fact, when using the PLDE environment under the KDE desktop, this conflict would cause a fatal error in the PLDE linker.

These environment variables have been replaced with similarly named variables which should be used when compiling under the PowerWorks Linux Development Environment.

The following text is revised from the PowerMAX OS man page for **ld(1)**, describing the behavior of the PLDE linker:

> On the host system, the environment variable PLDE_LD_LIBRARY_PATH may be used to specify library search directories. In the most general case, it will contain two directory lists separated by a semicolon:
>
> *dirlist1*;*dirlist2*
>
> Thus, if **ld.pmax** is called with the following occurrences of **-L**:
>
> **ld.pmax . . . -L*path1* . . . -L*pathn* . . . -l*x***
>
> then the search path ordering for the library *x* (**lib*x*.so** or **lib*x*.a**) is:
>
> ***dirlist1 path1* . . . *pathn dirlist2* LIBPATH**
>
> where LIBPATH is the default library search path for the PLDE cross-development libraries.
>
> When executing the resultant object file on the target system, LD_LIBRARY_PATH is still used to specify library search directories to the dynamic linker at run time. That is, if LD_LIBRARY_PATH exists in the target environment, the dynamic linker will search the directories it names before its default directory for shared objects to be linked with the program at execution.
>
> Additionally, on the host system, the environment variable PLDE_LD_RUN_PATH (which also contains a directory list) may be used to specify library search directories to the dynamic linker. If present and not empty, it is passed to the dynamic linker by **ld.pmax** via data stored in the output object file.

On the host system, LD_LIBRARY_PATH and LD_RUN_PATH are assumed to be relevant to the native Linux system linker and are ignored by the PLDE linker. You must instead use PLDE_LD_LIBRARY_PATH and/or PLDE_LD_RUN_PATH when building with the PLDE.

If you already have Makefiles or scripts that set LD_LIBRARY_PATH or LD_RUN_PATH when building from a Linux system, you must modify them in order to preserve existing functionality when building with PLDE 1.4.2 and later.

In addition, the PLDE linker accepts an option:

'-QLD_RUN_PATH=*filename*'

which may be used to provide the name of a file which contains a path for use by the target dynamic linker at runtime in locating shared objects. This pathname is embedded in the resulting object file. In PLDE 1.4.2 and later versions, an alternate form,

'-QPLDE_LD_RUN_PATH=*filename*'

is also accepted as equivalent.

Normally, users don't invoke the PLDE linker, **ld.pmax**, directly. The PLDE C/C++ compilation tools and the MAXAda **a.build** and **a.link** tools invoke **ld.pmax** appropriately when linking programs and shared libraries. The comments made above regarding the use of PLDE_LD_LIBRARY_PATH and PLDE_LD_RUN_PATH with **ld.pmax** apply equally to the C/C++ and MAXAda linking tools.

6.2. Installing the PLDE on Red Hat Linux 7

During installation of the PowerWorks Linux Development Environment on a system running Red Hat Linux 7, the NightView RPM may fail a dependency check related to **libncurses.so.4** because Red Hat 7 installs the **libncurses.so.5** library instead of **libncurses.so.4**. This causes the installation of the NightView RPM to fail, thereby causing the **PLDE.install** script to fail to completely install the PLDE.

You may alleviate this problem using one of the following solutions.

6.2.1. Install libncurses from Red Hat Linux Installation CD

The Red Hat 7 CD should contain an RPM to provide backward compatibility with **libncurses.so.4**. It isn't installed by default, at least in some configurations. It should be installed manually, following this procedure (as root):

Insert the Red Hat 7 Disc 1 and issue the following commands:

```
mount /mnt/cdrom
cd /mnt/cdrom/RedHat/RPMS
rpm -i ncurses4-5.0-2.i386.rpm
cd /
umount /mnt/cdrom
```

Afterward, completely remove any PLDE RPMs from your system by executing the **PLDE.uninstall** command, then reinstall the PLDE by executing the **PLDE.install** command.

Insert the PowerWorks Linux Development Environment CD

```
mount /mnt/cdrom
cd /mnt/cdrom/linux-i386
./PLDE.uninstall   # ignore diags about uninstalled RPMs
./PLDE.install
cd /
umount /mnt/cdrom
```

6.2.2. Install libncurses from FTP site

If you don't have a Red Hat Linux 7.0 Installation CD set available, you can download the backward-compatibility **curses** library with a web browser from this URL:

```
ftp://ibiblio.org/pub/Linux/distributions/redhat/redhat-7.0/i386/en/RedHat/RPMS
```

by selecting:

```
ncurses4-5.0-2.i386.rpm
```

Or, you can use **ftp** directly as follows:

```
# ftp ibiblio.org
...
Name (ibiblio.org): anonymous
...
Password: [your email address]
...
ftp> cd
/pub/Linux/distributions/redhat/redhat-7.0/i386/en/RedHat/RPMS
...
ftp> get ncurses4-5.0-2.i386.rpm
200 PORT command successful.
150 Opening BINARY mode data connection for ncurses4-5.0-2.i386.rpm
(161135 bytes).
226 Transfer complete.
...
ftp> ^D
```

After obtaining it either way, install it with this command (as root):

```
rpm -i ncurses4-5.0-2.i386.rpm
```

Afterward, re-install the PLDE as indicated in “Install libncurses from Red Hat Linux Installation CD” on page 18.

NOTE

Any other Red Hat mirror site can be used, as well. See the following URL for a listing:

```
http://www.redhat.com/download/mirror.html
```

6.2.3. Ignore dependencies

If you cannot obtain the **ncurses4-5.0** compatibility library, then you can install PLDE manually using the **--nodeps** option to **PLDE.install**. This causes the installation to ignore *any* missing RPMs.

Insert the PowerWorks Linux Development Environment CD and issue the following commands:

```
mount /mnt/cdrom
cd /mnt/cdrom/linux-i386
./PLDE.uninstall   # ignore diags about uninstalled RPMs
./PLDE.install --nodeps
cd /
umount /mnt/cdrom
```

NOTE

NightView's non-GUI interfaces will not function if it is installed this way, but the GUI still will.

6.3. Red Hat 6.2 Kernel Upgrade

Red Hat 6.2 ships with the 2.2.14-5.0 kernel. That version of the kernel contains bugs in the memory management subsystem that cause processes to fail in a variety of ways when system memory and swap space are heavily utilized. These bugs do not exist in the 2.2.12 kernel that ships with Red Hat 6.1.

Red Hat Support has made available kernel upgrades for Red Hat 6.2. The RPMs can be downloaded from the following URL:

http://www.redhat.com/support/errata/RHSA-2001-013.html

NOTE

The URL will discuss a security advisory which was the reason that Red Hat Support provided the RPMs. Nonetheless, these RPMs do correct the memory management bugs.

Instructions for selecting the appropriate RPMs are provided below.

A kernel upgrade is a little more complicated than an ordinary RPM upgrade. For complete information, Red Hat Support makes available instructions at the following URL:

http://www.redhat.com/support/docs/howto/kernel-upgrade/kernel-upgrade.html

Following is a brief set of instructions that should work on stock Red Hat 6.2 systems, which have undergone no kernel configuration changes by the user:

1. Determine the architecture of the system on which the kernel upgrade will be installed. Possible values are `i386`, `athlon`, `i486`, `i586`, or `i686`. Use the following command:

   ```
   $ uname -m
   i686          (example result)
   ```

   Occasionally, a prefix such as `osfmach3_` also is present. If so, just use the suffix such as `i386` in place of `osfmach3_i386`.

2. Determine the predefined kernel configuration. Use the following command:

```
$ grep image /etc/lilo.conf
image=/boot/vmlinuz-2.2.14-5.0smp    (example result)
```

   The following relates the configuration to the output from grep:

   none

```
image=/boot/vmlinuz-2.2.14-5.0
```

   smp

```
image=/boot/vmlinuz-2.2.14-5.0smp
```

   enterprise

```
image=/boot/vmlinuz-2.2.14-5.0enterprise
```

3. Download the RPMs from the Red Hat ftp site. For each architecture and kernel configuration, the appropriate set of RPMs is listed:

   i386, athlon, i486:

   none:

```
kernel-2.2.17-14.i386.rpm
kernel-doc-2.2.17-14.i386.rpm
kernel-ibcs-2.2.17-14.i386.rpm
kernel-pcmcia-cs-2.2.17-14.i386.rpm
kernel-source-2.2.17-14.i386.rpm
kernel-utils-2.2.17-14.i386.rpm
```

   smp:

```
kernel-doc-2.2.17-14.i386.rpm
kernel-ibcs-2.2.17-14.i386.rpm
kernel-pcmcia-cs-2.2.17-14.i386.rpm
kernel-smp-2.2.17-14.i386.rpm
kernel-source-2.2.17-14.i386.rpm
kernel-utils-2.2.17-14.i386.rpm
```

i586:

none:

```
kernel-2.2.17-14.i586.rpm
kernel-doc-2.2.17-14.i386.rpm
kernel-ibcs-2.2.17-14.i386.rpm
kernel-pcmcia-cs-2.2.17-14.i386.rpm
kernel-source-2.2.17-14.i386.rpm
kernel-utils-2.2.17-14.i386.rpm
```

smp:

```
kernel-doc-2.2.17-14.i386.rpm
kernel-ibcs-2.2.17-14.i386.rpm
kernel-pcmcia-cs-2.2.17-14.i386.rpm
kernel-smp-2.2.17-14.i586.rpm
kernel-source-2.2.17-14.i386.rpm
kernel-utils-2.2.17-14.i386.rpm
```

i686:

none:

```
kernel-2.2.17-14.i686.rpm
kernel-doc-2.2.17-14.i386.rpm
kernel-ibcs-2.2.17-14.i386.rpm
kernel-pcmcia-cs-2.2.17-14.i386.rpm
kernel-source-2.2.17-14.i386.rpm
kernel-utils-2.2.17-14.i386.rpm
```

smp:

```
kernel-doc-2.2.17-14.i386.rpm
kernel-ibcs-2.2.17-14.i386.rpm
kernel-pcmcia-cs-2.2.17-14.i386.rpm
kernel-smp-2.2.17-14.i686.rpm
kernel-source-2.2.17-14.i386.rpm
kernel-utils-2.2.17-14.i386.rpm
```

enterprise:

```
kernel-doc-2.2.17-14.i386.rpm
kernel-enterprise-2.2.17-14.i686.rpm
kernel-ibcs-2.2.17-14.i386.rpm
kernel-pcmcia-cs-2.2.17-14.i386.rpm
kernel-source-2.2.17-14.i386.rpm
kernel-utils-2.2.17-14.i386.rpm
```

4. Change your directory to the one in which you downloaded the appropriate set of RPMs.

5. Upgrade the kernel RPMs:

   ```
   $ rpm -Uvh *.rpm
   ```

   Diagnostics of this form are produced, but are harmless:

   ```
   package Makefile not listed in file index
   ```

6. Rebuild the RAM disk used by the kernel while booting. Use *one* of the following commands which mentions the appropriate configuration:

   ```
   $ mkinitrd /boot/initrd-2.2.17-14.img 2.2.17-14
   ```

   or

   ```
   $ mkinitrd /boot/initrd-2.2.17-14smp.img 2.2.17-14smp
   ```

   or

   ```
   $ mkinitrd /boot/initrd-2.2.17-14enterprise.img 2.2.17-14enterprise
   ```

   On some systems, the above command isn't absolutely necessary, but executing it won't harm anything.

7. Remove the old RAM disk.

   ```
   $ rm -f /boot/initrd-2.2.12-20.img
   $ rm -f /boot/initrd-2.2.12-20smp.img
   $ rm -f /boot/initrd-2.2.12-20enterprise.img
   ```

8. Edit the **`/etc/lilo.conf`** file. Within it, *one* of the following lines should be present:

   ```
   image=/boot/vmlinuz-2.2.14-5.0
   image=/boot/vmlinuz-2.2.14-5.0smp
   image=/boot/vmlinuz-2.2.14-5.0enterprise
   ```

   Change it to the corresponding line from the following list:

   ```
   image=/boot/vmlinuz-2.2.17-14
   image=/boot/vmlinuz-2.2.17-14smp
   image=/boot/vmlinuz-2.2.17-14enterprise
   ```

9. Reinstall the boot record with the new information from **`lilo.conf`**:

   ```
   $ lilo
   Added linux *
   ```
   (example result)

10. Reboot the system:

    ```
    $ shutdown -r
    ```

6.4. Invoking NEdit

The PowerWorks Linux Development Environment includes a version of the popular editor NEdit. The PLDE version has been enhanced with additional editing capabilities and has been integrated with the PLDE toolset. However, some Linux distributions also provide NEdit. These versions are typically

named **nedit**. In order to avoid any conflicts and to insure certain functionality, the PLDE version of the tool has been named **neditor** and is installed in **/usr/bin/X11**. If **/usr/bin/X11/nedit** is not already installed, the PLDE version is copied to **/usr/bin/X11/nedit** as well.

6.5. Too Many Open Files

If failures occur with the following error message:

```
errno = 23 (Too many open files in system)
```

you may edit the **/etc/sysctl.conf** file, adding lines similar to the following:

```
# Increase the maximum number of files on the system
fs.file-max = 16384
```

The default value for the `fs.file-max` kernel configurable is 4096, so a fourfold increase should be sufficient in most cases. However, you may choose a value that is appropriate for your particular system.

With the entry in **/etc/sysctl.conf**, the configurable will take effect on all subsequent reboots. To have it take effect immediately, issue the following command:

```
sysctl -p /etc/sysctl.conf
```

6.6. Debugging Programs Which Use Shared Libraries

When using NightView to debug a program, the portion of NightView running on the host needs to be able to locate all executable files required for the program's execution, including any shared libraries required by the program. If a program is linked in shared mode (which is the default mode for C/C++), it will require, at a minimum, **/usr/lib/libc.so.1**.

The host side of NightView detects this requirement and attempts to open **/usr/lib/libc.so.1**. However, when the host side of NightView is running on Linux, that pathname refers to a shared library that is completely unrelated to the actual shared library used by the program being debugged.

NightView automatically detects this situation and looks for the corresponding system shared library used when the program was linked with the PLDE. For example:

```
/pmax/os/4.3/moto/usr/lib/libc.so.1
```

NightView checks the contents of the host and target files and, if they match, will automatically provide the appropriate **translate-object-file** settings.

NightView also examines the **/etc/vfstab** file on the target system and attempts to locate otherwise inaccessible files using NFS mount specifications in that file.

If NightView cannot find a required file from the host system, it will issue a diagnostic and may fail to properly debug portions of your program that utilize shared libraries.

To correct this, use the **translate-object-file** command (the abbreviated name is **xl**), before invoking the program inside NightView. You will need to specify the appropriate path to the actual shared library that was used when the program was linked.

For example:

```
xl /usr/lib /pmax/os/4.3/moto/usr/lib
```

tells NightView to translate any target pathnames starting with **/usr/lib** to **/pmax/os/4.3/moto/usr/lib** when accessing them from the host side of NightView.

In our example, the host side of NightView would then attempt to open **/pmax/os/4.3/moto/usr/lib/libc.so.1**, which is the actual shared object used when the program was linked and corresponds to the shared library in use by the program on the target system.

The OS version number and target architecture in the **/pmax** path must be specified appropriately for your program, depending on the **-osversion** and **-arch** settings you used to link your program. The currently supported OS version/target architecture settings are:

OS Version	Target Architecture
4.3	**nh**, **moto**
5.0	**synergy**
5.1	**nh**, **moto**, **synergy**

Additional information relating to this topic is available in the *NightView User's Guide* (0890395) in the sections entitled "translate-object-file" and "Remote Dialogues".

Alternatively, you could link your programs statically, by supplying the **-Zlink=static** link option to the C/C++ driver.

7.0. Direct Software Support

Software support is available from a central source. If you need assistance or information about your system, please contact the Concurrent Software Support Center at 1-800-245-6453. Our customers outside the continental United States can contact us directly at 1-954-283-1822 or 1-305-931-2408. The Software Support Center operates Monday through Friday from 8 a.m. to 7 p.m., Eastern Standard time.

Calling the Software Support Center gives you immediate access to a broad range of skilled personnel and guarantees you a prompt response from the person most qualified to assist you. If you have a question requiring on-site assistance or consultation, the Software Support Center staff will arrange for a field analyst to return your call and schedule a visit.