PowerWorks Linux Development Environment

Version 2.1 Release Notes

December 2002

0898000-2.1

CONCURRENT COMPUTER CORPORATION

Copyright

Copyright 2002 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, NightProbe, 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 4
- 3.1 Host System 4
  - 3.1.1 Software 4
  - 3.1.2 Hardware 5
- 3.2 Target System 6
  - 3.2.1 Software 6
  - 3.2.2 Hardware 6

4.0 System Installation 7
- 4.1 PLDE Updates 11
- 4.2 License Activation 12
  - 4.2.1 NightStar Tools 12
  - 4.2.2 MAXAda 12
- 4.3 Target Installation 14
  - 4.3.1 Target Installation Instructions 15
    - 4.3.1.1 Mounting the CD-ROM device 15
    - 4.3.1.2 Installing the target portions of the NightStar tools . . . 15
    - 4.3.1.3 Installing the target portions of the compilers 16
    - 4.3.1.4 Unmounting the CD-ROM device 16
- 4.4 Installation of Software from Previous Releases 17
- 4.5 Cross-Development Libraries and Headers 19

5.0 Getting Started 21
- 5.1 Concurrent C/C++ 21
  - 5.1.1 Invoking the Compiler 21
  - 5.1.2 Include Files and Libraries 21
  - 5.1.3 OS Versions and Target Architectures 22
  - 5.1.4 Shared vs. Static Linking 22
  - 5.1.5 Makefile Considerations 22
- 5.2 Concurrent Fortran 77 22
  - 5.2.1 Invoking the Compiler 22
  - 5.2.2 Libraries 22
  - 5.2.3 OS Versions and Target Architectures 23
  - 5.2.4 Shared vs. Static Linking 23
- 5.3 MAXAda 23
  - 5.3.1 Invoking the Compiler 23
  - 5.3.2 OS Versions and Target Architectures 23

6.0 Additional Considerations 25
- 6.1 Known Issues with Red Hat 8.0 25
  - 6.1.1 rpm command 25
  - 6.1.2 Localization 25
- 6.2 PLDE_LD_LIBRARY_PATH and PLDE_LD_RUN_PATH 25
- 6.3 Invoking NEdit 26
- 6.4 Too Many Open Files 27

6.5 Debugging Programs Which Use Shared Libraries 27
7.0 Direct Software Support 29

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, Fortran 77, and C/C++ compilers; the NightView™ symbolic debugger; the NightTrace™ event analyzer; the NightSim™ frequency-based scheduler; the NightBench™ GUI program development environment; and the NightProbe™ data monitoring tool.

MAXAda™, Concurrent Fortran 77, 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.

NightProbe is a real-time graphical tool for monitoring, recording, and altering program data within one or more executing programs using non-intrusive techniques. It can be used in a development environment as a tool for debugging, or in a production environment to create a “control panel” for program input and output.

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 2.1 documentation available from Concurrent.

Table 2-1. PowerWorks Linux Development Environment Version 2.1 Documentation

Manual Name	Pub. Number
PowerWorks Linux Development Environment Version 2.1 Release Notes	0898000-2.1
PowerWorks Linux Development Environment Tutorial	0898100-010
AXI for MAXAda Reference Manual	0890518-000
AXI for MAXAda Version 3.3.2 Release Notes (Linux)	0898518-3.3.2
Concurrent C/C++ Reference Manual	0890497-030
Concurrent C/C++ Version 5.4 Release Notes (Linux)	0898497-5.4
Data Monitoring Reference Manual	0890493-000
Data Monitoring Version 3.3.1 Release Notes (Linux)	0898493-3.3.1
Fortran 77 Cross-Compiler Version 5.4 Release Notes (Linux)	0898536-5.4
hf77 Fortran Reference Manual	0890240-090
MAXAda Reference Manual	0890516-100
MAXAda Version 3.3.2 Release Notes (Linux)	0898516-3.3.2
NightBench User's Guide	0890514-070
NightBench Version 2.3 Release Notes (Linux)	0898514-2.3
NightProbe User's Guide	0890465-040
NightProbe Version 2.5.1 Release Notes (Linux)	0898465-2.5.1
NightSim User's Guide	0890480-030
NightSim Version 3.3.1 Release Notes (Linux)	0898480-3.3.1
NightTrace Manual	0890398-090
NightTrace Version 5.1.2 Release Notes (Linux)	0898398-5.1.2
NightView User's Guide	0890395-240
NightView Version 5.5 Release Notes (Linux)	0898395-5.5
NightView Pocket Reference	0890475-050
Quick Reference for shmdefine	0898010-000

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, manuals listed in Table 2-1 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 2.1 for both the host system and target system are as follows:

3.1. Host System

3.1.1. Software

- Red Hat® Linux*
- 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 7.1, 7.2, and 7.3. 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 7.3:* **bash-2.05a-13** **compat-glibc-6.2-2.1.3.2** **compat-libstdc++-6.2-2.9.0.16** **glibc-2.2.5-39** **glibc-debug-2.2.5-39** **ncurses4-5.0-5** **pdksh-5.2.14-16** **plde-nbench-2.3-001** **rpm-4.0.4-7x.18** **XFree86-devel-4.2.0-8** **XFree86-libs-4.2.0-8**

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 w/ patch 11 (P11)

  or

  PowerMAX OS 5.1 w/ service release 5 (SR5)

- Required PowerMAX OS software packages (see “Target Installation” on page 14 for more information)

NOTE

Support for selected older patch and service levels is provided as well. See “Installation of Software from Previous Releases” on page 17 for details.

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.

The following tables show the RPMs that will be installed.

Table 1. Compilers

Item	RPM
Concurrent C/C++	**plde-c++-5.4-001-2.i386.rpm** **plde-c++help-5.4-001-1.i386.rpm** **plde-c++invoker-5.4-001.i386.rpm**
Concurrent Fortran	**plde-f77-5.4-000-1.i386.rpm**
MAXAda	**plde-MAXAda-3.3.2-002-3.i386.rpm** **plde-MAXAda-invoker-3.3.2-000.i386.rpm** **plde-MAXAda-rm-3.3.2-1.i386.rpm** **plde-MAXaxi-3.3.2-000-2.i386.rpm**

Table 2. Tools

Item	RPM
NightBench	**plde-nbench-2.3-002.i386.rpm** **plde-nbench-ada-2.3-002.i386.rpm** **plde-nbench-c-2.3-002.i386.rpm**
NightProbe	**ccur-nprobe-2.5.1-001.i386.rpm** **ccur-nprobeserv-2.5.1-001.i386.rpm**
NightSim	**ccur-nsim-3.3.1-005.i386.rpm** **ccur-nsimserver-3.3.1-005.i386.rpm**
NightTrace	**ccur-ntrace-5.1.2-005.i386.rpm** **ccur-ntracelog-5.1.2-005.i386.rpm**
NightView	**ccur-NightView-5.5-000.i386.rpm** **ccur-Nviewp-5.5-000.i386.rpm**

Table 3. Utilities and Support

Item	RPM
Élan License Manager	**ccur-elanlm-5.0-6.i386.rpm**
ktalk	**ccur-ktdev-6.4.2-006.i386.rpm**
NightStar	**ccur-nstar-1.2-000.i386.rpm**
PLDE tutorial	**plde-tutorial-2.1-000.i386.rpm**

Table 3. Utilities and Support

Scripts	**ccur-HyperHelp-scripts-6.4.2-002.i386.rpm**
shmdefine	**ccur-shmdefine-1.0-000.i386.rpm**
Utilities	**ccur-x11progs-6.4.2-006.i386.rpm**

Table 4. Cross-Development Libraries and Headers

Item	RPM
Cross-Development Libraries and Headers	**plde-pmax-crossdev-4.3-P11-2.i386.rpm** **plde-pmax-crossdev-5.1-SR5-1.i386.rpm** **plde-pmax-crypt-4.3-P11-1.i386.rpm** **plde-pmax-crypt-5.1-SR5-1.i386.rpm** **plde-datamon-3.3.1-002-2.i386.rpm** **plde-datamon-pmax4.3-3.3.1-5.i386.rpm** **plde-datamon-pmax5.1-3.3.1-3.i386.rpm** **plde-hf77libs-4.1-001-5.i386.rpm** **plde-hf77libs-pmax4.3-4.1-5.i386.rpm** **plde-hf77libs-pmax5.1-4.1-3.i386.rpm** **plde-ntracelibs-5.1.2-006-1.i386.rpm** **plde-ntracelibs-pmax4.3-5.1.2-2.i386.rpm** **plde-ntracelibs-pmax5.1-5.1.2-2.i386.rpm** **plde-x11-6.4.2-004-1.i386.rpm** **plde-x11-pmax4.3-6.4.2-4.i386.rpm** **plde-x11-pmax5.1-6.4.2-3.i386.rpm**

NOTE

As mentioned in “Documentation” on page 2, release notes are provided for each of the individual major 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 script described below which installs 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 Power-Works Linux Development Environment installation scripts

   `cd /mnt/cdrom/linux-i386`
4. Invoke the PowerWorks Linux Development Environment installation script

   `./PLDE.install`

   This script automaticaly installs or upgrades appropriately based on the RPMs already installed on your system. To avoid the automatic upgrade, completely remove all previous PLDE RPMs using the **`PLDE.uninstall`** script before running **`PLDE.install`**.

NOTE

The **`PLDE.install`** script will set the default release of the Concurrent C/C++ compiler to version 5.4 for the system on which it is invoked. Previous releases of C/C++ on the system will remained installed and will still be available for use. See the section titled “Multiple Release Support” in the “Compilation” chapter of the *C/C++ Reference Manual* (0890497) for more information.

The **`PLDE.install`** script will set the default release of MAXAda to 3.3.2 for the system on which it is invoked. If a previous release was installed, it will remain installed and will still be available for use. See the section titled “a.mkenv” in the “MAXAda Utilities” chapter of the *MAXAda Reference Manual* (0890516) for more information.

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.

NOTE

On some versions of Linux, the **PLDE.install** 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 **--ignoresize** option to **PLDE.install** in order to proceed with the installation.

Due to a bug in an earlier version of the PLDE, messages similar to the following may be generated during the execution of the **PLDE.install** 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
```

Due to a bug in the standard Red Hat Linux product installation mechanism, **rpm** (see **rpm(8)**), the final percentage shown in the installation statistics may indicate a total of only ~ 75% when in reality all packages have been installed. The final lines in the output of **PLDE.install** script may look similar to the following:

```
26:plde-pmax-crypt-5.1    ########################################### [ 72%]
27:plde-f77-5.4           ########################################### [ 75%]
PLDE 2.1 Installation Complete.
```

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

The PowerWorks Linux Development Environment requires that certain packages be installed on the PowerMAX OS target systems. See “Target Installation” on page 14 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.)

4.1. PLDE Updates

In between full releases of the PLDE, individual packages may require updating due to maintenance, enhancements, or to remain in sync with the corresponding software on PowerMAX OS systems.

All updates to the PLDE release are available on the Concurrent web server at:

```
http://redhawk.ccur.com
```

In order to retrieve updates, you will need a user ID and password which is available from Concurrent Software Distribution. Contact 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`.

We recommend checking the web server occassionally for PLDE bug fixes, enhancements, and packages supporting new PowerMAX OS software versions and patches.

4.2. License Activation

4.2.1. NightStar Tools

Use of the NightStar tools requires a license key.

The license key authorizes use of the tools on a specific machine with a specific maximum number of simultaneous users.

At the end of the installation process, a server code will be displayed. You must contact Concurrent Computer Corporation via email at `softdist@ccur.com` and include your server code, site-id, product name and model number to obtain your permanent license key. A response will be emailed to you containing your permanent key.

Your site-id, product name and model number are supplied on a License Manager Information sheet accompanying the installation CD.

If you forget your server code, you may obtain it by executing the following command:

```
/usr/bin/elmcode
```

During the installation, you will be prompted to enter the license key. You may respond with any of the following:

1. Your permanent license key,
2. The temporary license key (printed on the installation CD case) which allows a single user to use the tools for a short period of time until the permanent license key is received, or
3. <Enter> which will bypass installation of the license key. Note, however, that bypassing license activation will render portions of the PowerWorks Linux Development Environment ineffective until either the temporary or permanent license key is installed.

To install the permanent license key (or the temporary key if license activation was bypassed), enter the following command on the system where the PowerWorks Linux Development Environment is installed:

```
/usr/bin/elmadmin -c -r1/"key"
```

where *key* is either the permanent license key obtained from Concurrent or the temporary license key printed on the installation CD case. Note that it is necessary to place double-quotes around the *key*.

4.2.2. MAXAda

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 phase3.3.2 -activate` *code*

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

4.3. Target Installation

The revision levels of the software installed on the PowerMAX OS target systems should match the revision levels of the corresponding software installed on the Linux host system.

Therefore, for proper operation with the PowerWorks Linux Development Environment Version 2.1, it is recommended that the software running on the PowerMAX OS target systems are at the following revision levels:

OS Version	Patch Level
PowerMAX OS 4.3	P11
PowerMAX OS 5.1	SR5

NightStar Tool	Package	Version	Patches
NightProbe	**nprobeserv**	2.5.1	
NightSim	**nsimserver**	3.3.1	
NightTrace	**ntracelog**	5.1.2	**ntracelog-006**
NightView	**Nviewp**	5.5	

Compiler	Package	Version	Patches
Concurrent C/C++	**c++54runtime**	5.4	**c++-001**
Concurrent Fortran 77	**hf77libs**	4	**hf77libs-001**

These packages are normally installed during the installation of the corresponding operating system, NightStar tool, or compiler on the PowerMAX OS system. However, it may be necessary to manually install these packages on the target systems if they have not been installed or if the versions installed on the target systems do not conform to the above tables. The packages corresponding to the above NightStar tools and compilers are included on the PowerWorks Linux Development Environment Installation CD and may be installed by following the steps in “Target Installation Instructions” on page 15.

If you wish to target a PowerMAX OS system running software with revision levels older than those listed above, ensure that the corresponding RPMs are installed on the Linux host system. Selected versions of RPMs from previous releases and patch levels are available in the **linux-i386/obsolescent** directory on the PowerWorks Linux Development Environment Installation CD. See “Installation of Software from Previous Releases” on page 17 for more information.

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 5.5 Release Notes (PowerMAX)* (0890395-5.5); *NightTrace Version 5.1.2 Release Notes (PowerMAX)* (0890398-5.1.2); *NightSim Version 3.3.1 Release Notes (PowerMAX)* (0890480-3.3.1); and *Concurrent C/C++ Version 5.4 Release Notes (PowerMAX)* (0890497-5.4).

4.3.1. Target Installation Instructions

4.3.1.1. Mounting the CD-ROM device

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
```

4.3.1.2. Installing the target portions of the NightStar tools

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.

4.3.1.3. Installing the target portions of the compilers

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++54runtime** 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.

4.3.1.4. Unmounting the CD-ROM device

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
```

4.4. Installation of Software from Previous Releases

Selected versions of RPMs from previous releases and patch levels are available on the PowerWorks Linux Development Environment Installation CD.

Users may wish to install these RPMs in favor of the most recent versions (which are installed by default).

The following software packages are included in the **`linux-i386/obsolescent`** directory on the PowerWorks Linux Development Environment Version 2.1 Installation CD:

`ccur-NightView-5.4-000.i386.rpm`
`plde-c++-5.2-004-1.i386.rpm`
`plde-c++-5.3-003-1.i386.rpm`
`plde-c++help-5.2-004-1.i386.rpm`
`plde-c++help-5.3-001-2.i386.rpm`
`plde-c++invoker-5.3-001.i386.rpm`
`plde-f77-5.3-000-2.i386.rpm`
`plde-MAXAda-3.3.1-004-3.i386.rpm`
`plde-MAXAda-invoker-3.3.1-004.i386.rpm`
`plde-MAXaxi-3.3.1-001-2.i386.rpm`
`plde-nbench-2.2-000.i386.rpm`
`plde-nbench-ada-2.2-000.i386.rpm`
`plde-nbench-c-2.2-000.i386.rpm`
`plde-ntrace-4.2-004.i386.rpm`
`plde-ntracelibs-4.2-004-1.i386.rpm`
`plde-ntracelibs-pmax4.3-4.2-3.i386.rpm`
`plde-ntracelibs-pmax5.1-4.2-2.i386.rpm`
`plde-pmax-crossdev-4.3-P6-1.i386.rpm`
`plde-pmax-crossdev-4.3-P7-1.i386.rpm`
`plde-pmax-crossdev-4.3-P8-1.i386.rpm`
`plde-pmax-crossdev-4.3-P9-1.i386.rpm`
`plde-pmax-crossdev-4.3-P10-1.i386.rpm`
`plde-pmax-crossdev-5.1-SR0-1.i386.rpm`
`plde-pmax-crossdev-5.1-SR1-1.i386.rpm`
`plde-pmax-crossdev-5.1-SR2-1.i386.rpm`
`plde-pmax-crossdev-5.1-SR4-1.i386.rpm`
`plde-pmax-crypt-4.3-P6-1.i386.rpm`
`plde-pmax-crypt-4.3-P7-1.i386.rpm`
`plde-pmax-crypt-4.3-P8-1.i386.rpm`
`plde-pmax-crypt-4.3-P9-1.i386.rpm`
`plde-pmax-crypt-4.3-P10-1.i386.rpm`
`plde-pmax-crypt-5.1-SR0-1.i386.rpm`
`plde-pmax-crypt-5.1-SR1-1.i386.rpm`
`plde-pmax-crypt-5.1-SR2-1.i386.rpm`
`plde-pmax-crypt-5.1-SR4-1.i386.rpm`

To install the older versions, remove the newer versions of the RPMs you wish to replace after installing the complete PLDE (see “System Installation” on page 7), and then install the older versions using the **`rpm`** command (see **`rpm(8)`**).

NOTE

Given the interdependencies between PLDE RPMs, replacing RPMs with older versions can be somewhat complicated. It is recommended that such activities are completed by an experienced Linux system adminstrator or that the user request specific aid from the Concurrent Software Support Center (see “Direct Software Support” on page 29).

4.5. 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 7 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-5.1`

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

`plde-pmax-crossdev-4.3-P11-2.i386.rpm`
`plde-pmax-crossdev-5.1-SR5-1.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-5.1`**	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-pmax5.1`**	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 4.1 libraries
plde-hf77libs-pmax4.3 **plde-hf77libs-pmax5.1**	allows Fortran 77 4.1 libraries to be cross-linked in programs for the version of PowerMAX OS referenced in the RPM name
plde-ntracelibs-5.1.2	contains NightTrace 5.1.2 libraries
plde-ntracelibs-pmax4.3 **plde-ntracelibs-pmax5.1**	allows NightTrace 5.1.2 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-pmax5.1**	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 (0898100) provides stepwise instructions introducing several of the PLDE tools.

In addition, the following sections contain considerations for getting started with the various PowerWorks Linux Development Environment compilers.

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.

See the *Concurrent C/C++ Reference Manual* (0890497) for additional information.

5.1.1. Invoking the Compiler

The Concurrent C/C++ compiler can be used 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++ Reference Manual* (0890497) 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++ Reference Manual* (0890497) for more information.).

In addtion, the Concurrent C/C++ compiler can be used via the NightBench Program Development Environment (**`nbench`**). See the *NightBench User's Guide* (0890514) 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++ Reference Manual* (0890497) 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++ Reference Manual* (0890497) 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++54runtime`** package (as described in "Target Installation" on page 14), or relink your program using static libraries.

See the section titled "Shared vs. Static Linking" in the *Concurrent C/C++ Reference Manual* (0890497) 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++ Reference Manual* (0890497) for solutions to this problem.

5.2. Concurrent Fortran 77

The following should be taken into consideration in order to use the Concurrent Fortran 77 Cross-Compiler with the PowerWorks Linux Development Environment.

See the *Concurrent Fortran 77 Cross-Compiler Version 5.4 Release Notes* (0898536-5.4) for more information.

5.2.1. Invoking the Compiler

To use the Concurrent Fortran 77 Cross-Compiler, specify the following in your PATH environment variable:

```
PATH=$PATH:/usr/ccs/bin
```

The compiler should then be invoked with **`xf77`**.

5.2.2. Libraries

By default, the Concurrent Fortran 77 Cross-Compiler automatically looks for PowerMAX OS libraries in the tree rooted at:

`/pmax/os/`*version*`/`*arch*

where *version* and *arch* indicate the PowerMAX OS version and target architecture of your choice (see "OS Versions and Target Architectures" on page 23 for more details).

Libraries located under **/usr/lib** are native Linux files and are unrelated and incompatible with the corresponding files for PowerMAX OS. Do not attempt to utilize files from those directories when building PowerMAX OS programs.

5.2.3. OS Versions and Target Architectures

The Concurrent Fortran 77 Cross-Compiler supports building PowerMAX OS programs for various versions of PowerMAX OS and various 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 Fortran 77 Cross-Compiler Version 5.4 Release Notes* (0898536-5.4) for more information.

5.2.4. Shared vs. Static Linking

By default, the Concurrent Fortran 77 Cross-Compiler links with shared libraries.

Thus, if you attempt to execute your Fortran 77 program on a PowerMAX OS system it will require, at a minimum, the shared library **libhU77.so**, **F77.s0**, **I77.s0**, **lhead.so**, and **libc.so**.

If your PowerMAX OS system doesn't have either the Concurrent Fortran 77 product installed, your program will fail to execute.

To link your program using static libraries, append the **-Zlink=static** option to your command line:

```
xf77 -Zlink=static -o main main.f
```

5.3. MAXAda

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

See the *MAXAda Version 3.3.2 Release Notes* (0898516-3.3.2) for more information.

5.3.1. Invoking the Compiler

MAXAda provides a number of command-line utilities as the interface to its compilation system. See the "MAXAda Utilities" chapter of the *MAXAda Reference Manual* (0890516) for detailed information on each of these utilities.

You must have **/usr/ada/bin** in your PATH environment variable to use the MAXAda command-line utilities.

In addtion, the MAXAda compiler can be used via the NightBench Program Development Environment (**nbench**). See the *NightBench User's Guide* (0890514) for more information.

5.3.2. 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 3.3.2 Release Notes* (0898516-3.3.2) for more information.

6.0. Additional Considerations

6.1. Known Issues with Red Hat 8.0

6.1.1. rpm command

Due to a known bug in Red Hat 8.0, if you manually install RPMs using the **rpm** command (see **rpm(8)**) (as opposed to using the **PLDE.install** script - see "System Installation" on page 7), you must first set the following environment variable before executing the command:

```
RPM_INSTALL_PREFIX=/usr/opt
export RPM_INSTALL_PREFIX
```

6.1.2. Localization

If you fail to see text displayed in the NightBench text areas or see messages similar to the following:

```
Warning:
    Name: field
    Class: XmTextField
    Character '\61' not supported in font.  Discarded.

Warning:
    Name: field
    Class: XmTextField
    Character '\62' not supported in font.  Discarded.

Warning:
    Name: field
    Class: XmTextField
    Character '\62' not supported in font.  Discarded.
```

it may be necessary to set the following environment variable:

```
LC_ALL=POSIX
export LC_ALL
```

as a workaround for a known bug with localization in Red Hat 8.0.

6.2. 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 2.1 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 2.1 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.

6.3. 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.

6.4. 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
```

6.5. 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`**
`5.1`	**`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.