IBM Parallel Environment for AIX 5L

Installation

Version 4 Release 3.0

GA22-7943-05

IBM Parallel Environment for AIX 5L

Installation

Version 4 Release 3.0

GA22-7943-05

Note
Before using this information and the product it supports, read the information in “Notices” on page 61.

Sixth Edition (October 2006)

This edition applies to version 4, release 3, modification 0 of IBM Parallel Environment for AIX 5L (product number 5765-F83) and to all subsequent releases and modifications until otherwise indicated in new editions. This edition replaces GA22-7943-04. Significant changes or additions to the text and illustrations are indicated by a vertical line (|) to the left of the change.

IBM welcomes your comments. A form for readers' comments may be provided at the back of this publication, or you may address your comments to the following address:

International Business Machines Corporation
Department 55JA, Mail Station P384
2455 South Road
Poughkeepsie, NY 12601-5400
United States of America

FAX (United States & Canada): 1+845+432-9405
FAX (Other Countries):
Your International Access Code +1+845+432-9405

IBMLink (United States customers only): IBMUSM10(MHVRCFS)
Internet e-mail: mhvrcfs@us.ibm.com

If you would like a reply, be sure to include your name, address, telephone number, or FAX number.

Make sure to include the following in your comment or note:

- Title and order number of this book
- Page number or topic related to your comment

When you send information to IBM, you grant IBM a nonexclusive right to use or distribute the information in any way it believes appropriate without incurring any obligation to you.

Contents

Tables vii

About this book ix
Who should read this book ix
How this book is organized ix
Conventions and terminology used in this book x
Abbreviated names x
Prerequisite and related information xi
Using LookAt to look up message explanations xii
How to send your comments xii
National language support (NLS) xii
Summary of changes for Parallel Environment 4.3 xiii

Chapter 1. Introducing PE 4.3 1
PE components 1

Chapter 2. Planning to install the PE software 3
PE installation requirements 3
Hardware requirements 3
Software requirements 3
Disk space requirements 6
PE Limitations 7
Information for the system administrator 7
Software compatibility within workstation clusters 7
Node resources 8
Deciding which nodes require which PE file sets or additional software 8
File systems 8
User IDs on remote nodes 9
User authorization 9
POE security method configuration 9
Cluster based security configuration 10
AIX-based security (compatibility) 10
PE Benchmarker user authorization 10
Running large POE jobs and IP buffer usage 11

Chapter 3. Installing the PE software 13
About installing PE with CSM 13
About installing PE on an IBM pSeries cluster 13
Migration installation 13
Determining which earlier file sets are installed 14
Removing earlier file sets 14
When to install the rsct.lapi.rte file set 14
When to install the rsct.lapi.nam file set 14
When to install the rsct.core.sec file set 15
When to install the loadl.so (LoadLeveler) file set 15
View the README file before installation 15
PE installation procedure summary 15
Install the PE file sets step-by-step 16
Step 1: Copy the software to a hard disk for installation over a network . . . 16
Step 2: Perform the initial installation 17
Step 3: Install PE on other nodes 20
Step 4: Verify the POE installation 23

Chapter 4. Migrating and upgrading PE 25
General overview 25
AIX compatibility 25
Coexistence 26
Migration support 26
AIX Support 26
MPI library support 27
LAPI support. 27
Online documentation 27

Chapter 5. Performing installation-related tasks 29
Removing a software component 29
Recovering from a software vital product database error. 29
Customizing the message catalog 29
Installing AFS 30
Setting up POE for AFS execution 30

Chapter 6. Understanding how installing PE alters your system 33
How installing the POE file set alters your system 33
POE installation effects 36
How installing the ppe.perf and ppe.pvt file sets alters your system 36
How installing the online documentation alters your system 37
Online pdf documentation 38

Chapter 7. Additional information for the system administrator 39
Configuring the Parallel Environment coscheduler 39
POE coscheduling parameters and limits 39
AIX dispatcher tuning 41
Using the /etc/poe.limits file 42
Entries in the /etc/poe.limits file 42
How the Partition Manager daemon handles the /etc/poe.limits file 43
Description of /etc/poe.security 43
Enabling Remote Direct Memory Access (RDMA) 44

Appendix A. Syntax of commands for running installation and deinstallation scripts 47
Installation script: PEinstall 47
Copying the installation image 47
Mounting the installation image 48
Deinstallation script: PEdeinstall 48

Appendix B. Installation verification program summary 51

Appendix C. Using additional POE sample applications 53
Bandwidth measurement test sample. 53
Verification steps 53
Broadcast test sample 54
Verification steps 54
MPI threads sample program. 55
Verification steps 56
LAPI sample programs 56

Appendix D. Parallel Environment port usage 57

Appendix E. Accessibility features for PE 59
Accessibility features. 59

Keyboard navigation 59
IBM and accessibility. 59

Notices 61
Trademarks 63
Acknowledgments 64

Glossary 65

Index 73

Tables

1. Typographic conventions x
2. PE File set requirements 4
3. Additional software requirements 5
4. Disk space requirements for installation 6
5. File sets to remove before installation 14
6. Installation procedure summary 15
7. Step 2 for installing with CSM 18
8. Method 1: Use the installp command 18
9. Filenames for different types of installations 19
10. Steps to take to determine steps remaining 20
11. Specify -copy and -mount 21
12. File names for different data types 21
13. Steps to take to determine steps remaining 22
14. Space requirements for pdbx, pmdv4, and poe components 30
15. POE directories and files installed 33
16. ppe.poe.post_i symbolic links 35
17. ppe.perf and ppe.pvt directories and files installed 36
18. Man page directories and files installed 37
19. PE port usage 57

About this book

IBM® Parallel Environment for AIX 5L™ Installation describes how to install the Parallel Environment program product on a networked cluster of IBM eServer™ pSeries® processors.

This book assumes that AIX 5L Version 5.3 Technology Level 5300-05 (AIX 5L V5.3 TL 5300-05) and the X-Windows system are already installed, if required. For information on installing AIX® and X-Windows, consult *AIX 5L Installation Guide and Reference*. Note that *AIX 5L Version 5.3 Technology Level 5300-05* (or *AIX 5L V5.3 TL 5300-05*) identify the specific maintenance level required to run PE 4.3. The name *AIX 5.3* is used in more general discussions.

To use this book, you should be familiar with the AIX operating system. Where necessary, some background information related to AIX is provided. More commonly, you are referred to the appropriate documentation.

Note: The full product name is Parallel Environment for AIX 5L Version 4 Release 3, referred to in this text as PE.

Who should read this book

This book is intended for system programmers and administrators, who plan, migrate, and install PE.

How this book is organized

This book is organized as follows:

- Chapter 1, “Introducing PE 4.3,” on page 1 is an overview of PE, describing how its various software components work together. This introduction also describes some installation considerations based on your system’s configuration.
- Chapter 2, “Planning to install the PE software,” on page 3 contains the planning information you need to consider before installing PE. Topics include the hardware and software requirements, as well as information on node resources, file systems, and user ID administration.
- Chapter 3, “Installing the PE software,” on page 13 contains the step-by-step procedure you need to follow to install PE. This chapter also lists, and describes, the product directories created and the links established by the installation process.
- Chapter 4, “Migrating and upgrading PE,” on page 25 contains specific information on some differences between earlier releases that you may need to consider before installing or using PE 4.3.
- Chapter 5, “Performing installation-related tasks,” on page 29 describes additional procedures (such as removing an installation image and customizing the message catalog) that are related to installing PE.
- Chapter 6, “Understanding how installing PE alters your system,” on page 33 describes how your system is altered when you install the various PE software file sets.
- Chapter 7, “Additional information for the system administrator,” on page 39 describes the format of PE configuration files that are created and modified by the system administrator.

- Appendix A, “Syntax of commands for running installation and deinstallation scripts,” on page 47 explains the syntax of the commands for running the installation and deinstallation scripts provided with PE.
- Appendix B, “Installation verification program summary,” on page 51 explains how the POE Installation Verification Program (IVP) works.
- Appendix C, “Using additional POE sample applications,” on page 53 describes some sample applications.

Conventions and terminology used in this book

Note that in this document, LoadLeveler®® is also referred to as *Tivoli® Workload Scheduler LoadLeveler* and *TWS LoadLeveler*.

This book uses the following typographic conventions:

Table 1. Typographic conventions

Convention	Usage
bold	**Bold** words or characters represent system elements that you must use literally, such as: command names, file names, flag names, path names, PE component names (**poe**, for example), and subroutines.
`constant width`	Examples and information that the system displays appear in `constant-width` typeface.
italic	*Italicized* words or characters represent variable values that you must supply. *Italics* are also used for book titles, for the first use of a glossary term, and for general emphasis in text.
[item]	Used to indicate optional items.
<Key>	Used to indicate keys you press.
\	The continuation character is used in coding examples in this book for formatting purposes.

In addition to the highlighting conventions, this manual uses the following conventions when describing how to perform tasks.

User actions appear in uppercase boldface type. For example, if the action is to enter the **tool** command, this manual presents the instruction as:

ENTER
tool

Abbreviated names

Some of the abbreviated names used in this book follow.

AIX Advanced Interactive Executive

CSM Clusters Systems Management

CSS communication subsystem

CTSEC cluster-based security

DPCL dynamic probe class library

dsh distributed shell

GUI	graphical user interface
HDF	Hierarchical Data Format
IP	Internet Protocol
LAPI	Low-level Application Programming Interface
MPI	Message Passing Interface
NetCDF	Network Common Data Format
PCT	Performance Collection Tool
PE	IBM® Parallel Environment for AIX®
PE MPI	IBM's implementation of the MPI standard for PE
PE MPI-IO	IBM's implementation of MPI I/O for PE
POE	parallel operating environment
pSeries	IBM eServer pSeries
PVT	Profile Visualization Tool
RISC	reduced instruction set computer
RSCT	Reliable Scalable Cluster Technology
rsh	remote shell
STDERR	standard error
STDIN	standard input
STDOUT	standard output
UTE	Unified Trace Environment
System x	IBM System x

Prerequisite and related information

The Parallel Environment for AIX library consists of:

- IBM Parallel Environment: Introduction, SA22-7947
- IBM Parallel Environment: Installation, GA22-7943
- IBM Parallel Environment: Operation and Use, Volume 1, SA22-7948
- IBM Parallel Environment: Operation and Use, Volume 2, SA22-7949
- IBM Parallel Environment: MPI Programming Guide, SA22-7945
- IBM Parallel Environment: MPI Subroutine Reference, SA22-7946
- IBM Parallel Environment: Messages, GA22-7944

To access the most recent Parallel Environment documentation in PDF and HTML format, refer to the IBM eServer Cluster Information Center on the Web at:

http://publib.boulder.ibm.com/infocenter/clresctr/vxrx/index.jsp

Both the current Parallel Environment books and earlier versions of the library are also available in PDF format from the IBM Publications Center Web site located at:

http://www.ibm.com/shop/publications/order/

It is easiest to locate a book in the IBM Publications Center by supplying the book's publication number. The publication number for each of the Parallel Environment books is listed after the book title in the preceding list.

Using LookAt to look up message explanations

LookAt is an online facility that lets you look up explanations for most of the IBM messages you encounter, as well as for some system abends and codes. You can use LookAt from the following locations to find IBM message explanations for Clusters for AIX:

- The Internet. You can access IBM message explanations directly from the LookAt Web site:

  http://www.ibm.com/eserver/zseries/zos/bkserv/lookat/
- Your wireless handheld device. You can use the LookAt Mobile Edition with a handheld device that has wireless access and an Internet browser (for example, Internet Explorer for Pocket PCs, Blazer, or Eudora for Palm OS, or Opera for Linux® handheld devices). Link to the LookAt Mobile Edition from the LookAt Web site.

How to send your comments

Your feedback is important in helping to provide the most accurate and high-quality information. If you have comments about this book or other PE documentation:

- Send your comments by e-mail to: mhvrcfs@us.ibm.com

  Be sure to include the name of the book, the part number of the book, the version of PE, and, if applicable, the specific location of the text you are commenting on (for example, a page number or table number).
- Fill out one of the forms at the back of this book and return it by mail, by fax, or by giving it to an IBM representative.

National language support (NLS)

For national language support (NLS), all PE components and tools display messages that are located in externalized message catalogs. English versions of the message catalogs are shipped with the PE licensed program, but your site may be using its own translated message catalogs. The PE components use the AIX environment variable **NLSPATH** to find the appropriate message catalog. **NLSPATH** specifies a list of directories to search for message catalogs. The directories are searched, in the order listed, to locate the message catalog. In resolving the path to the message catalog, **NLSPATH** is affected by the values of the environment variables **LC_MESSAGES** and **LANG**. If you get an error saying that a message catalog is not found and you want the default message catalog:

ENTER

```
export NLSPATH=/usr/lib/nls/msg/%L/%N

export LANG=C
```

The PE message catalogs are in English, and are located in the following directories:

/usr/lib/nls/msg/C

/usr/lib/nls/msg/En_US

/usr/lib/nls/msg/en_US

If your site is using its own translations of the message catalogs, consult your system administrator for the appropriate value of **NLSPATH** or **LANG**. For more information on NLS and message catalogs, see *AIX: General Programming Concepts: Writing and Debugging Programs*.

Summary of changes for Parallel Environment 4.3

This release of IBM Parallel Environment for AIX contains a number of functional enhancements, including:

- PE 4.3 supports only AIX 5L Version 5.3 Technology Level 5300-05, or later versions.

  AIX 5L Version 5.3 Technology Level 5300-05 is referred to as AIX 5L V5.3 TL 5300-05 or AIX 5.3.
- Support for Parallel Systems Support Programs for AIX (PSSP), the SP™ Switch2, POWER3™ servers, DCE, and DFS™ has been removed. PE 4.2 is the **last** release that supported these products.
- PE Benchmarker support for IBM System p5™ model 575 has been added.
- A new environment variable, **MP_TLP_REQUIRED** is available to detect the situation where a parallel job that should be using large memory pages is attempting to run with small pages.
- A new command, **rset_query**, for verifying that memory affinity assignments have been performed.
- Performance of MPI one-sided communication has been substantially improved.
- Performance improvements to some MPI collective communication subroutines.
- The default value for the **MP_BUFFER_MEM** environment variable, which specifies the size of the Early Arrival (EA) buffer, is now 64 MB for both IP and User Space. In some cases, 32 bit IP applications may need to be recompiled with more heap or run with **MP_BUFFER_MEM** of less than 64 MB. For more details, see the migration information in Chapter 1 of *IBM Parallel Environment: Operation and Use, Volume 1* and Appendix E of *IBM Parallel Environment: MPI Programming Guide*.

Chapter 1. Introducing PE 4.3

The Parallel Environment (PE) licensed program product is a set of software components that help you develop, debug, analyze, and run parallel FORTRAN, C, or C++ programs on a cluster of IBM eServer pSeries networked servers. Before installing PE, you should be familiar with these components and how they fit together. Refer to the following documents to learn how to use the PE components:

- *Parallel Environment: Operation and Use, Volume 1*
- *Parallel Environment: Operation and Use, Volume 2*

If you are new to PE, you will probably find *IBM Parallel Environment: Introduction* useful. For the latest information, always review the PE product README file included with the PE RPMs.

PE components

The PE components are:

Message passing and collective communication API subroutine libraries

These libraries, which contain subroutines that help application developers parallelize their code, are described in *IBM Parallel Environment: MPI Programming Guide*. For additional information about MPI, see the *IBM Parallel Environment: MPI Subroutine Reference* and the *Parallel Environment: MPI Programming Guide*.

Parallel operating environment (POE)

This software helps ease your transition from serial to parallel processing by hiding many of the differences and allowing you to continue using standard AIX tools and techniques. When you start a parallel job, the POE partition manager contacts the remote nodes, begins running your code, and oversees the job's operation.

For more information, refer to *IBM Parallel Environment: Operation and Use, Volume 1*.

pdbx A parallel line-oriented debugger based on the **dbx** debugger.

Performance collection tool

This tool enables you to collect the following types of data for one or more application processes:

- MPI and user event data
- Hardware and operating system profile data
- Communication counts (LAPI or MPI message sizes)
- OpenMP performance data.

This tool requires the installation of the Dynamic Probe Class Library (DPCL).

DPCL is no longer a part of the IBM PE for AIX licensed program, but is still shipped with PE for convenience. Instead, DPCL is now available as an open source offering that supports PE. For more information on DPCL open source project go to: **http://dpcl.sourceforge.net**.

If you have identified a problem while using DPCL , report it to the DPCL team by sending an e-mail to **dpcl-user@lists.sourceforge.net** describing the problem you are having. If the problem was discovered while making

enhancements to DPCL or developing a tool using DPCL, report the problem to the **dpcl-develop@lists.sourceforge.net** mailing list.

OpenMP profiling tool
This tool enables you to collect information for all OpenMP locking functions, OpenMP directives, and compiler-generated OpenMP functions such as function call count, wall clock time, and system time.

Communication profiling tool
The tools enables you to collect communication count (message size) information for MPI and/or LAPI applications.

Profile visualization tool
This tool helps you to analyze and process profile data files generated by the Performance Collection Tool.

PE documentation

This component is made up the following:

- **ppe.man**: man pages for MPI subroutines and PE commands and functions

PE can also be used with LookAt, which is an online facility that lets you look up explanations for most of the IBM messages you encounter, as well as for some system abends and codes. For more information, see “Using LookAt to look up message explanations” on page xii.

Chapter 2. Planning to install the PE software

When planning to install the Parallel Environment software, you need to ensure that you have met all of the necessary system requirements. You also need to think about what your programming environment will be and the strategy for using that environment.

PE installation requirements

There are various system requirements for installing and running the PE software, including requirements for hardware, software, disk space, and file sets.

Hardware requirements

PE 4.3 is supported in the following environments:

- Clustered IBM pSeries POWER4™ servers, interconnected with the pSeries High Performance Switch, running the AIX 5L V5.3 TL 5300-05 64-bit kernel.
- Clustered IBM System p5 servers, interconnected with the pSeries High Performance Switch, running the AIX 5L V5.3 TL 5300-05 64-bit kernel.
- Clustered POWER3, POWER4, or IBM System p5 servers, either stand-alone or connected via a LAN supporting IP, running AIX 5L V5.3 TL 5300-05 32-bit or 64-bit kernel.

The message passing libraries support these hardware configurations:

- IBM pSeries clustered servers via IP protocol only
- IBM pSeries (models listed above) with pSeries High Performance Switch via IP and User Space

Total random access memory (RAM) and fixed disk storage requirements for the machine are based on the licensed programs and user applications you install. See “Disk space requirements” on page 6 for more information. For information on RAM and disk storage requirements for AIX and associated programs, refer to *IBM RS/6000® SP: Planning, Volume 2, Control Workstation and Software Environment* or *IBM Cluster Systems Management for AIX 5L: Planning and Installation Guide.*

Software requirements

The software required for PE includes a variety of PE components plus, in some cases, additional software. You need to decide which PE components to install on your system based on the PE features you plan to use. You may also need to install some additional products or components, based on how you plan to use PE.

PE file set requirements

Table 2 on page 4 lists the PE 4.3 file sets. Decide which of these file sets to install on the various nodes in your system, based on the PE component options you plan to use.

- For more information about nodes, see “Node resources” on page 8.
- For information about installing the following product options individually, see “PE installation procedure summary” on page 15.

Table 2. PE File set requirements

If you plan to...	...this product option is required:	File set name:	Things to consider:
...develop and execute parallel applications from a node	Parallel Operating Environment	**ppe.poe**	MPI and the **pdbx** command-line parallel debugger are part of POE. When POE is installed, it adds entries to the **/etc/services** and **/etc/inetd.conf** files. When POE is executed, a copy of the partition manager daemon is run on each remote node and is identified by these files. If you are using NIS or another master server for **/etc/services**, you need to update the individual files with the same information.
...collect LAPI, MPI, and user event data or hardware and operating system profiles.	PE Benchmarker Performance Collection Tool	**ppe.perf**	The installation of **ppe.perf** requires that you install version 3.4.1 of Dynamic Probe Class Library (DPCL). DPCL is now an open source offering available from http://dpcl.sourceforge.net (see Note 1 at the bottom of the table). DPCL is also included on the PE CD as the **ppe.dpcl** file set. You must also install the 32 bit version of Java™ Runtime Environment, version 1.4.1 or later. To collect hardware profiles requires the AIX System and Kernel Thread Performance Monitor API, file set **bos.pmapi** 5.3.0.50, or later. **ppe.perf** is required for performing byte count profiling when specifying the **MP_BYTECOUNT** option when compiling programs. See the *PE Operation and Use Volume 1* for more information on using byte count profiling and the **MP_BYTECOUNT** option.
...analyze and process profile data collected previously using the Performance Collection Tool.	PE Benchmarker Profile Visualization Tool	**ppe.pvt**	The installation of **ppe.pvt** is not dependent on the prior installation of any other PE components but in order to view the profile data collected, you must also install the 32 bit version of Java Runtime Environment, version 1.4.1 or later. To visualize MPI and user event trace information, you need to use Jumpshot, which is available from Argonne National Laboratories. (See Note 2 at bottom of table.)
...access the online documentation in man page format	PE man pages	**ppe.man**	None
...accept the eLicense agreement during installation of PE 4.3 file sets	PE license	**ppe.loc.license**	While not required to be installed, **ppe.loc.license** must be installed in the same location as the other PE install images in order to accept the license agreement.

Notes:

1. DPCL is no longer a part of the IBM PE for AIX licensed program, but is still shipped with PE for convenience. Instead, DPCL is now available as an open source offering that supports PE. For more information on the DPCL open source project go to: **http://dpcl.sourceforge.net**. If you have identified a problem while using DPCL , report it to the DPCL team by sending an e-mail to **dpcl-user@lists.sourceforge.net** describing the problem you are having. If the problem was discovered while making enhancements to DPCL or developing a tool using DPCL, report the problem to the **dpcl-develop@lists.sourceforge.net** mailing list.
2. Jumpshot is available from Argonne National Laboratories at the following FTP site:

   ftp://ftp.mcs.anl.gov/pub/mpi/misc/slog-jumpshot3_ibm24.tar.gz

Additional software requirements

Table 3 lists the additional software products or file sets that are required by PE 4.3. You need to decide which of these software products or file sets to install on your system, based on how you plan to use PE.

Table 3. Additional software requirements

If you plan to...	...this software is required:	Things to consider:
...use PE	AIX 5L Version 5.3 TL (program number 5765-G03) with Recommended Maintenance Package 5300-05, or later. The required AIX file sets include **bos.adt.base**, **bos.adt.syscalls**, **bos.rte.libc**, **bos.adt.debug**, and **bos.cpr**.	***AIX is always required.***
...run a parallel program on the IBM pSeries server cluster	**rsct.lapi.rte** 2.4.3 from AIX 5L V5.3 TL 5300-05.	Contains the communication protocol libraries for LAPI and MPI. **rsct.lapi.rte** is included on the PE product CD. For information on installing **rsct.lapi.rte**, see *RSCT for AIX 5L: LAPI Programming Guide*
...run parallel MPI or LAPI applications using multiple adapters and want support for failover and recovery	The **rsct.lapi.nam** (RSCT Network Availability Matrix) 2.4.3 file set.	If you do not need failover and recovery function, you do not need to install **rsct.lapi.nam**. In addition to **rsct.lapi.nam**, failover and recovery function also requires use of IBM's High Availability Group Services and installation of **rsct.basic.rte**. After installing **rsct.basic.rte** and **rsct.lapi.nam**, you must reboot the node. **rsct.lapi.nam** is included on the PE product CD. For information on installing **rsct.lapi.nam**, see *RSCT for AIX 5L: LAPI Programming Guide*.

Table 3. Additional software requirements (continued)

If you plan to...	...this software is required:	Things to consider:
...compile parallel executables	IBM C for AIX Version 6.0 (program number 5765-F57 *or* VisualAge® C++ Professional for AIX, Version 6.0 (program number 5765-F56) *or* IBM XL C/C++ Enterprise Edition Version 7.0 for AIX or later, (program number 5724-I11). *or* IBM XL FORTRAN for AIX Version 9.1 or later, (program number 5724-I08)	IBM C for AIX Version 6 is now part of VisualAge C++ Professional for AIX, Version 6.0, and is also available as a separate file set. VisualAge C++ Professional for AIX, Version 6.0 and IBM XL FORTRAN for AIX Version 9.1 support the latest IBM System p5 architecture.
...submit a POE job from outside a LoadLeveler cluster	**loadl.so** on the node outside the LoadLeveler cluster	See "When to install the loadl.so (LoadLeveler) file set" on page 15 for detailed information.
...use the pdbx debugger	**bos.adt.debug** file set	None
...use LoadLeveler to submit interactive POE User Space jobs or allow execution of batch jobs	LoadLeveler Version 3.4, 5765-D61 (or later)	When LoadLeveler is installed, PE 4.3 requires **LoadL.full** 3.4 to run with the latest features. See "Coexistence" on page 26 and "Migration support" on page 26 for more information.
... collect hardware profiles	AIX System and Kernel Thread Performance Monitor API, file set. For AIX 5L V5.3 TL 5300-05, **bos.pmapi** 5.3.0.50, or later, is required.	None
... view profile data collected	Java Runtime Environment Version 1.4.1 or later	To visualize MPI and user event trace information, you need to use Jumpshot, available from Argonne National Laboratories.
... use the Performance Collection Tool (PCT)	32-bit version of Java Runtime Environment Version 1.4.1 or later	None

Disk space requirements

The following table lists the amount of disk space you need in the appropriate directories for each of the separately-installable PE product options.

If you plan to install the PE software on an IBM pSeries or network cluster, each machine in the cluster must meet the disk space requirements shown in Table 4.

Table 4. Disk space requirements for installation

PE File set	Number of 512-Byte Blocks Required in Directory:		
	/usr	**/tmp**	**/etc**
ppe.man	4500	not applicable	not applicable
ppe.poe	25000	500	20
ppe.perf	45000	not applicable	10

Table 4. Disk space requirements for installation (continued)

PE File set	Number of 512-Byte Blocks Required in Directory:		
	/usr	/tmp	/etc
ppe.pvt	4500	not applicable	not applicable
ppe.dpcl	28640	not applicable	8

Note: **ppe.dpcl** is required when installing **ppe.perf**.

Note: Temp space required for installation is 128MB in **/tmp**.

PE Limitations

Some PE product options and related software are subject to certain limitations, as explained below.

MPI-IO parallel file I/O

MPI-IO in PE MPI is targeted to the IBM General Parallel File System (GPFS) for production use. File access through MPI-IO normally requires that a single GPFS file system image be available across all tasks of an MPI job. PE MPI with MPI-IO can be used for program development on any other file system that supports a POSIX interface (AFS®, DFS, JFS, or NFS) as long as all tasks run on a single node or workstation. This is not expected to be a useful model for production use of MPI-IO. PE MPI can be used without all nodes on a single file system image by using the **MP_IONODEFILE** environment variable. See *IBM Parallel Environment: Operation and Use, Volume 1* for information about **MP_IONODEFILE**.

Parallel applications and system calls

User-written parallel applications are limited in their use of system calls. See *IBM Parallel Environment: MPI Programming Guide* for a discussion of these limitations.

pdbx

When using the pdbx debugger, the application should be compiled using the parallel compiler scripts supplied with POE: **mpcc_r**, **mpCC_r**, or **mpxlf_r**. The pdbx debugger currently supports only FORTRAN 77, C, and C++.

Information for the system administrator

For system administrators, it is important to understand software compatibility for PE and how to plan out your node resources. You will also need to determine which nodes in your cluster will require which file sets. For additional information about POE system administration tasks, refer to Chapter 7, “Additional information for the system administrator,” on page 39.

Software compatibility within workstation clusters

For all processors *within a workstation cluster*, the same release level (including maintenance levels) of PE software is required. (This ensures that an individual PE application can run on any workstation in the cluster.)

About upgrading AIX without upgrading compilers

Many of the compilers link to different libraries based on the AIX OSLEVEL value when they are installed. If you migrate just AIX, you will be using libraries for a back level. Be sure to change the compiler library links or reinstall compilers.

LAPI and MPI library compatibility in PE 4.3

MPI and LAPI share a common transport layer, therefore MPI applications are dependent upon LAPI being previously installed in order to compile and execute MPI programs. You install the LAPI file set (**rsct.lapi.rte**), which is included on the PE product CD, as part of the standard PE installation procedure.

All the nodes can participate in processing parallel jobs. In doing so, all nodes must have compatible levels of the LAPI and MPI libraries installed, particularly when nodes are upgraded with new versions/releases of the libraries and when service is applied that affects the libraries. In all cases, the same version, release, and service level of the LAPI and MPI libraries must be installed on all nodes that are to participate in a parallel job.

For more information on the installation of PE, LAPI, and AIX on previously installed systems, refer to Chapter 4, “Migrating and upgrading PE,” on page 25.

Node resources

How you plan your node resources will vary according to whether you are installing PE on a pSeries cluster, with or without LoadLeveler.

On a cluster using LoadLeveler

The system administrator uses LoadLeveler to partition nodes into *pools* or *features* or both, to which he or she assigns names or numbers and other information. The workstation from which parallel jobs are started is called the *home node* and it can be any workstation on the LAN.

On a cluster without LoadLeveler

On an IBM pSeries network cluster, you assign nodes or servers to the following categories:

- *Home node* (workstation from which parallel jobs are started) for running the Partition Manager in POE
- Nodes or servers for developing and compiling applications
- Nodes or servers for executing applications in the parallel environment

  You must identify the nodes or servers running as execution nodes by name in a host list file.

Deciding which nodes require which PE file sets or additional software

An important aspect of planning your PE node resources is deciding which nodes require which PE file sets or additional software. You do not need to install all of the PE file sets on every node. Refer to “Software requirements” on page 3 for more information on the file sets and their dependencies. This information will help you decide how to install PE and additional required software on your nodes.

File systems

The PE file sets are installed in the **/usr** file system. When the **ppe.poe** file set is installed, it adds entries to the **/etc/services** and **/etc/inetd.conf** files. When **poe** is executed, a copy of the Partition Manager daemon is run on each remote node, and is identified in these files.

If you are using NIS or another master server for **/etc/services**, you need to create updates with the same information that is put into the individual files.

For more information about copying the file system and about **mcp**, see *Parallel Environment: Operation and Use, Volume 1*. For more information about **dsh** and **pcp**, see *IBM Cluster Systems Management: Command and Technical Reference*.

You can also manage files as part of Cluster System Management's (CSM) *Configuration File Manager*. With CSM, the Configuration File Manager provides a file repository for configuration files that are common across all nodes in a cluster. For more information, see *Cluster System Management: Administration Guide*.

User IDs on remote nodes

On each remote node, the system administrator must set up a user ID, other than a root ID, for each user on each remote node who will be executing serial or parallel applications or who requires POE access. See *IBM Parallel Environment: Operation and Use* for an introduction of home and remote nodes.

Each user must have an account on all nodes where a job runs. Both the user name and user ID must be the same on all nodes. Also, the user must be a member of the same named group on the home node and the remote nodes.

User authorization

There are several options for PE user authorization. You can use the POE security method, which is based on the Cluster Security Services of IBM RSCT, Cluster based security, AIX based security (the default), or PE Benchmarker user authorization, which is handled by DPCL.

POE security method configuration

PE 4.3 uses an enhanced set of security methods, based on Cluster Security Services in RSCT. POE has a security configuration option for the system administrator to determine which set of security methods are to be used in the system. There are two types of security methods supported:

- cluster based security (or CTSec)
- AIX based security (or Compatibility, which is the default)

When POE is installed, the **/etc/poe.security** file on each node will contain an entry defining the type of security method to be used on that node. For more information see the description of **/etc/poe.security** in Chapter 7, "Additional information for the system administrator," on page 39.

The use of the CTSec method will require the installation of the **rsct.core.sec** file set, along with its proper configuration. For more information, see "Cluster based security configuration" on page 10.

The use of the POE security method applies only when POE is used *without* LoadLeveler. When LoadLeveler is used (which includes all User Space jobs), LoadLeveler determines and enforces the security method, and POE will not check the security method.

Cluster based security configuration

When Cluster Based Security is the security method of choice, the system administrator will have to ensure that UNIX®® Host Based authentication is enabled and properly configured on all nodes. This entails:

- **/usr/sbin/rsct/cfg/unix.map** file exists with proper entries
- Host based authentication (HBA) is installed and configured on the nodes
- Proper public/private key set up for all of the nodes

Refer to the *RSCT Administration Guide* for specific details. From a user's point of view, when Cluster Based Security is used, users will be required to have the proper entries in the **/etc/hosts.equiv** or **.rhosts** files, in order to ensure proper access to each node, as described in "AIX-based security (compatibility)."

AIX-based security (compatibility)

When AIX-based security (compatibility) is the security method of choice, (which is also the default), POE relies on the use of AIX-based user authorization, as described below.

If AIX user authorization, or compatibility, (the default) is used as a security mechanism on the system, each node needs to be set up so that each user ID is authorized to access that node or remote link from the initiating home node. Use the **/etc/hosts.equiv** file and/or the **.rhosts** file to specify this user ID authorization, as explained below.

If the combination of the home node machine and user name:

- *is authorized* in **/etc/hosts.equiv** on the remote node, the user is authorized to run parallel tasks there.
- *is disallowed* in **/etc/hosts.equiv** on the remote node, the user is *not* able to run parallel tasks there.
- *does not appear* in **/etc/hosts.equiv**, the combination is checked in the **.rhosts** file in the user's home directory on the remote node. If the user name and the home node machine combination appears in **.rhosts**, the user is authorized to run parallel tasks on the remote node.

For more information on **.rhosts** and **/etc/host.equiv**, see the chapter on managing jobs in *IBM AIX 5L Files Reference*.

If you are using LoadLeveler to submit POE jobs, including all User Space applications, LoadLeveler is responsible for the security authentication. The security function in POE is not invoked when POE runs under LoadLeveler.

PE Benchmarker user authorization

PE Benchmarker user authentication and authorization is handled by DPCL. Details of configuration and the use of DPCL authentication and authorization is described in the DPCL Authentication and Authorization document available from the DPCL Web site at **http://dpcl.sourceforge.net/doc/index.html**.

Running large POE jobs and IP buffer usage

A POE application may require additional IP buffers (**mbufs**) under any of the following circumstances:

- PE job uses more than 128 nodes.
- Large amounts of STDIO (stdin, stdout, or stderr) are generated.
- The home node is running many POE jobs simultaneously, or there is significant additional IP traffic via mounted file system activity (or other sources), or both.
- Many large messages are passed via the UDP/IP implementation of the Message Passing Library.

The need for additional IP buffers is usually evident when repeated requests for memory are denied. Using the **netstat -m** command can tell you when such a condition exists. In such a case, it may be necessary to use the **no** command to change the network option system parameters on the home node. You can use the **no** command to initially check the values as well.

The number of IP buffers allocated in the kernel is controlled by the **thewall** parameter of the **no** command. Increasing the value of the **thewall** parameter increases the number of IP buffers.

- You must have root authority to change options with the **no** command, and the setting applies to all processes running on the node on which it is executed.

You can also set the values at system boot time by adding the appropriate call to the **no** command in either **/etc/rc.net** or **/etc/rc.tcpip**.

For more information on **mbufs**, see *IBM AIX 5L Performance Management Guide*.

Chapter 3. Installing the PE software

To install PE, you first install the desired PE file sets on a single node. When that installation is complete, you can then replicate the installation image throughout the remaining nodes, using one of the suggested methods described in this chapter.

PE Version 4 is now enabled for AIX electronic licensing capability. The **ppe.loc.license** file set must be present on the same install media or in the same directory as the PE Version 4 file sets to be installed in order for the license agreement to be processed during the installation of that file set. The installer must also specify the proper option to confirm that the license has been accepted, in order for the file set to be properly installed.

About installing PE with CSM

To install the desired PE file sets on a pSeries cluster running CSM, you install the software on each node individually using SMIT or **installp**. Note that you must first install the PE file sets on at least one node of your system.

CSM cannot be installed in the same node in the cluster. For more information on CSM, refer to *IBM Cluster Systems Management for AIX 5L: Planning and Installation Guide* and *IBM Cluster Systems Management for AIX 5L: Administration Guide.*

About installing PE on an IBM pSeries cluster

Installation on an IBM pSeries network cluster without CSM will not provide system management functions. This leaves you with the following two options:

- Use the **PEinstall** script.
- Install the software on each system individually using SMIT or **installp**.

In either case, first install the PE file sets on at least one system in your cluster. When this is complete, you can replicate the installation image to your other nodes.

During the course of installing PE file sets on a cluster, you may encounter **sysck** warning messages. These messages may indicate that a particular file is also owned by another file set. If the file is also owned by one of the older PE file sets, such as PE Version 2 **ppe.poe**, this may indicate that an older version is installed.

You can ignore these warning messages and the system will function properly. However, if you later choose to remove the old file set after installing PE Version 4, you need to reinstall the new file set.

Migration installation

If you migrate from PE Version 2 or Version 3 to PE Version 4, installing the new file sets will completely replace some of the earlier release file sets, rendering them obsolete. The replaced file sets will be marked “OBSOLETE” in the object data manager (ODM) and **lslpp** by **installp**.

However, some directories and installation files will remain. Because these earlier file sets do not coexist or execute with PE Version 4, *you should uninstall your old file sets before installing the new PE file sets*, rather than installing the new file sets

on top of the old. This will conserve disk space and reduce the chance for confusion over old file set path names and executables.

CAUTION:
If you plan to uninstall the old file sets, do so *before* **installing the new file sets. If you attempt to uninstall the old file sets** *after* **installing PE Version 4, you may accidentally delete some needed files.**

Table 5 lists the old file sets that need to be removed before you install PE Version 4:

Table 5. File sets to remove before installation

PE Version	File sets to be removed
2	**ppe.pedocs** **ppe.vt** **ppe.xpdbx**
3	**ppe.html** **ppe.pdf**

Determining which earlier file sets are installed

You can use the **lslpp** command to check if any of the file sets are installed. For example, **lslpp -l poe** will tell you if the Version 1 POE file set is installed.

Removing earlier file sets

To remove file sets you can use any of the following methods:

- SMIT

  Use the Maintain Installed Software dialog found under the Software Installation and Maintenance dialog.
- **installp** command; for example:

  installp -u poe
- **PEdeinstall** script

  See “Removing a software component” on page 29.

When to install the rsct.lapi.rte file set

If you are using pSeries workstation clusters and plan to run parallel MPI or LAPI applications, you must install **rsct.lapi.rte** before or after installing PE Version 4, in order for parallel applications to execute. The **rsct.lapi.rte** file set is included on the PE product CD.

For information on installing **rsct.lapi.rte**, see *RSCT for AIX 5L: LAPI Programming Guide*.

When to install the rsct.lapi.nam file set

If you are using pSeries workstation clusters and plan to run MPI or LAPI applications using multiple High Performance Switch adapters and want support for failover and recovery, you must install **rsct.lapi.nam** before or after installing PE Version 4. Installation of **rsct.lapi.nam** is not required if failover and recovery function is not needed. Failover and recovery function also requires use of IBM’s High Availability Group Services and installation of **rsct.basic.rte**.

The **rsct.lapi.nam** file set is included on the PE product CD. The **rsct.basic.rte** file set is included with the AIX operating system.

After installing **rsct.basic.rte** and **rsct.lapi.nam**, you must reboot the node. For information on installing **rsct.lapi.nam**, see *RSCT for AIX 5L: LAPI Programming Guide*.

When to install the rsct.core.sec file set

If you plan to use the cluster based security methods based on Cluster Security Services in RSCT, you must also install the **rsct.core.sec** file set, and perform the appropriate configuration steps.

See "POE security method configuration" on page 9 for more information.

When to install the loadl.so (LoadLeveler) file set

Install this file set to submit a POE job which uses LoadLeveler from a node outside of the LoadLeveler cluster. To install, do the following:

1. Contact the system administrator of your LoadLeveler cluster to determine the path name to the exported directory containing the **loadl.so** image.
2. NFS-mount that directory on the submitting node.
3. Install the **loadl.so** file set using the following command:

   ```
   installp -aFXd device loadl.so
   ```

4. Obtain the LoadLeveler configuration file as described in *Tivoli® Workload Scheduler LoadLeveler: Using and Administering*.

View the README file before installation

Before you actually install any file set, you may want to look at its README file. The README file may contain some special or additional information about installing the file set. The PE file sets are all shipped with a copy of the README as part of the first file on the CD. This allows you to view the README using the **installp -i** command and option.

If you decide after reading the README that you would like to refer to the file later, once the file set is installed, you can find the README file in the **/usr/lpp/***fileset***/READMES** directory. The file will have a name of *fileset***.README**.

PE installation procedure summary

Table 6 summarizes the basic steps you must follow to install the PE software on a pSeries network cluster.

You can install all of the PE file sets at once, or you can install selected file sets one at a time. To determine which file sets, if any, that you want to install separately, see "PE file set requirements" on page 3.

Table 6. Installation procedure summary

If you are installing			perform these steps
ppe.poe	**ppe.man**	**ppe.perf or ppe.pvt**	

Table 6. Installation procedure summary (continued)

X	X	X	"Step 1: Copy the software to a hard disk for installation over a network" on page 16	**Standard steps**
X	X	X	"Step 2: Perform the initial installation" on page 17	
X	X	X	"Step 3: Install PE on other nodes" on page 20	
X			"Step 4: Verify the POE installation" on page 23	**Optional step**

Also, refer to Appendix C, "Using additional POE sample applications," on page 53 for more information.

Install the PE file sets step-by-step

This section provides the step-by-step procedure for installing the PE software on a pSeries network cluster. Each step includes one or more tables that guide you through choices about variables. In some cases, they refer to the use of nodes with CSM, or without.

Pay close attention to these tables as you proceed through the procedure, because they may direct you to skip certain steps.

1. Before beginning the installation procedure, be sure to do the following:
   a. Login as **root**.
   b. If you already have an earlier version of PE installed, remove the earlier version. (See "Removing a software component" on page 29.)
   c. Verify that all prerequisite software is installed.
2. A discussion of SMIT options assumes that a fast path to the install software screen is installed. Otherwise follow the SMIT path to the custom install screen.

Step 1: Copy the software to a hard disk for installation over a network

This step consists of copying the installation images off the distribution medium and exporting the installation directory, thereby making the installation images available for mounting. You must complete this step if any of the machines in your cluster do not have the proper installation device to read the distribution medium.

Note: If you already have an earlier version of PE installed, remove the earlier version before proceeding. (See "Removing a software component" on page 29.)

Substep 1: Copy the software off the distribution medium

To copy the PE software off the distribution medium, follow these instructions:

INSERT
the distribution medium in the installation device.

ENTER
smit bffcreate

This command invokes SMIT, and takes you to the window for copying software to a hard disk for future installation over the network.

PRESS
List

A window opens listing the available INPUT devices and directories for software.

SELECT
the installation device from the list of available INPUT devices.

The window listing the available INPUT devices closes and the original SMIT window indicates your selection.

PRESS
Do

The SMIT window displays the default parameters used for copying software to a hard disk.

TYPE IN
all in the **SOFTWARE name** field.

TYPE IN
/usr/sys/inst.images in the **DIRECTORY for storing software** field. This is the installation directory name.

PRESS
Do

The system copies the PE software installation images to the directory.

SELECT
Exit → Exit SMIT

The SMIT window closes.

Substep 2: Export the installation directory

To export the directory so the machines in your cluster can install the PE installation images it contains, enter **/usr/sbin/mknfsexp -d /usr/sys/inst.images**

Step 2: Perform the initial installation

This step consists of initially installing the PE installation image, using either of the following methods:

- via the **installp** command
- via the installation menus of the System Management Interface Tool (SMIT)

Either method allows you to specify whether you want to install all of the PE software file sets or just certain individual file sets.

Keep in mind that some of the PE file sets depend on others to run. Refer to “PE file set requirements” on page 3, which details these dependencies, before you do a partial installation.

Table 7. Step 2 for installing with CSM

If you are installing with CSM:	If you are installing on an IBM pSeries network cluster without CSM:
Perform this step on the initial node. You must login as **root**.	Perform this step on any machine in the cluster. You must login as **root**.

Method 1: Use the installp command

To initially install the installation image, enter the appropriate command as shown in Table 8:

Table 8. Method 1: Use the installp command

To install:	ENTER
all software file sets	**installp -a -d** *devicename* **ppe***
just the **man** file set	**installp -a -l -X -Y -d** *devicename* **ppe.man**
just the POE file set	**installp -a -l -X -Y -d** *devicename* **ppe.poe**
just the Performance Collection Tool file set and the required Dynamic Probe Class Library (DPCL)	**installp -a -l -X -Y -d** *devicename* **ppe.dpcl** and **installp -a -l -X -Y -d** *devicename* **ppe.perf**
just the Profile Visualization Tool file set	**installp -a -l -X -Y -d** *devicename* **ppe.pvt**

In the commands above:

-l (capital l)
is used to select only the specified file set.

-a applies the software products.

-X attempts to expand any file systems where there is insufficient space to do the installation.

-Y accepts the eLicense.

-d *devicename*
is the name of the installation device or directory.

ppe.perf
requires that you first install **ppe.dpcl**

The system reads and receives the installation image off the distribution medium.

Method 2: Use SMIT

To initially install the installation image using SMIT, follow these instructions:

INSERT
the distribution medium in the installation device unless you are installing over a network.

ENTER
smit install_latest

This command invokes SMIT, and takes you directly to its window for installing software.

PRESS
List

A window opens listing the available INPUT devices and directories for software.

SELECT
the installation device or directory from the list of available INPUT devices.

The window listing the available INPUT devices and directories closes and the original SMIT window indicates your selection.

PRESS
Do

The SMIT window displays the default install parameters.

TYPE The appropriate file name, as shown in Table 9:

Table 9. Filenames for different types of installations

If you want to install:	Type this in the "SOFTWARE to install" field:
All the PE software	**ppe***
Just the **man** file set	**ppe.man**
Just the POE file set	**ppe.poe**
Just the Performance Collection Tool file set and the required Dynamic Probe Class Library (DPCL)	**ppe.perf** and **ppe.dpcl**
Just the Profile Visualization Tool file set	**ppe.pvt**

After choosing the appropriate software, you may also want to change other options on the panel, as needed. For example, the panel also asks whether or not you want to expand the file systems. When you are prompted, answer **yes** to expand the file systems.

TYPE IN
yes in the **ACCEPT new license agreements?** field. If the eLicense is not accepted, none of the PE software components will be installed.

PRESS
Do

The system installs the installation image.

For more information on SMIT, see *IBM AIX 5L General Programming Concepts: Writing and Debugging Programs*.

If installation fails

If the installation is unsuccessful, a software product cleanup procedure is automatically called. The cleanup procedure removes any files that may have been restored from the distribution medium, and backs out of any post-installation procedure that may have been started.

To help determine the cause of the unsuccessful installation, refer to the installation status file. This file indicates how far installation had progressed when the errors occurred. *IBM AIX 5L General Programming Concepts: Writing and Debugging Programs* describes the status file in more detail. If you cannot determine the cause of a failed installation, contact your local IBM representative.

Determine remaining tasks

You have completed the initial installation of PE. For a description of the directories, files, and daemon processes created and the links established when the installation image was received, see Chapter 6, “Understanding how installing PE alters your system,” on page 33.

To determine which remaining steps you need to perform, refer to Table 10:

Table 10. Steps to take to determine steps remaining

If there are other nodes in your system on which you need to install PE file sets:	**If there are *not* any other nodes in your system on which you need to install PE file sets:**
Proceed to • “Step 3: Install PE on other nodes” on page 20	Skip: • “Step 3: Install PE on other nodes” on page 20 If appropriate, proceed to: • “Step 4: Verify the POE installation” on page 23

Step 3: Install PE on other nodes

This step consists of installing PE on other nodes, using either of the following methods:

- running one of the installation scripts provided with PE
- manually

Perform this step, as **root**, from a node with PE installed.

Method 1: Use the PE installation script

This method consists of:

- creating a host list file (a list of the remaining nodes on which you want to install PE)
- running the **PEinstall** installation script

Substep 1: Create a host list file

To create a host list file, follow these instructions:

1. Open a new file using any AIX text editor.

   By default, the installation script looks for a file named **host.list** in your current directory. You can, however, name the host list file anything you want. If you do choose to give your file a different name, you will have to specify that file name when you run the installation script.
2. In the file, enter one node host name on each line. For example:

```
hostname1
hostname2
hostname3
hostname4
hostname5
```

Substep 2: Run the PEinstall installation script with the -copy or -mount option

To run the installation script, enter **PEinstall** *image_name* [*host_list_file*] [**-copy | -mount**].

Notes:

1. To execute the **installp** remotely on a *mounted* image, the directory containing the image must have world-writable permissions (as created by the **chmod 777** command).

   If you do not want to create this directory with world-writable permissions, do not use the **-mount** option of **PEinstall**.
2. To have the image copied or mounted to different directories, you will need to invoke **PEinstall** for each different location or set of locations. The host list file that you specify each time you invoke **PEinstall** should reflect only those nodes that you want to use with **-copy** or **-mount**.

Table 11. Specify -copy and -mount

If you specify the -copy option, you will be prompted for:	If you specify the -mount option, you will be prompted for:
• the installation image source directory. The default is **/usr/sys/inst.images**. • the installation image destination directory which is used for all nodes in the host list. The default is **/usr/sys/inst.images**.	• the installation image source directory. The default is **/usr/sys/inst.images**. • the remote node mount point directory, which is used for all nodes in the host list. The default is **/mnt**. • whether you want the script to automatically create the remote mount directory **If your remote mount directory already exists:** Answer **no** to this prompt. **Note:** Be sure that you have issued the **chmod 777** command on this directory. **If your remote mount directory does not already exist:** Answer **yes** to this prompt. **PEinstall** issues a **mkdir** command for the directory name specified, followed by a **chmod 777**.

Substep 3: Specify the file set(s) to be installed

When you are prompted for the name of the file set you want to install, enter the appropriate file name, as shown in Table 12:

Table 12. File names for different data types

If you want to install:	Type this when prompted:
all the PE software	**all**
just the **man** file set	**ppe.man**
just the POE file set	**ppe.poe**
just the Dynamic Probe Class Library (DPCL)	**ppe.dpcl** **Note:** This component must be installed in order to use the Performance Collection Tool.

Table 12. File names for different data types (continued)

If you want to install:	Type this when prompted:
just the Performance Collection Tool file set	**ppe.perf** **Note:** In order to use the Performance Collection Tool, DPCL (the **ppe.dpcl** file set) must be currently installed.
just the Profile Visualization Tool file set	**ppe.pvt**

For each node in the host list, **PEinstall** executes the following **installp** command:

installp -aYFX -d/*image_directory*/*image_name fileset*

This command installs both the **usr** and **root** portion of the file set in the image specified.

Errors that may occur during installation: The following severe installation errors will cause the installation process to terminate completely:

- The host list file cannot be found.
- No installation image name was specified.

For other errors, a message may appear describing the error, and then processing will continue. The same message will be logged in a file named **PEnode.log** in the current working directory. If you see error messages, look in this file, as the node on which the error occurred is always displayed and logged. This helps you identify any nodes on which the file set(s) did not get successfully installed. When you correct the errors, you can then rerun the **PEinstall** script just for those nodes.

Method 2: Installing PE manually

As a system administrator, you may want to have more control over the installation of PE, and install it manually to other nodes, using SMIT or **installp**.

During "Step 1: Copy the software to a hard disk for installation over a network" on page 16, you created an installation image that you can use to replicate the installation of PE file sets on the other nodes of your system. By making this image available to the other nodes, either by copying or mounting the image file, you can use SMIT or **installp** to install the image.

The installation image of PE file sets does not require any special consideration. You may use SMIT or **installp** as described in "Method 1: Use the installp command" on page 18. You can also set up a host list file, and run **installp** via **rsh**, and install the PE file sets on multiple nodes.

Determine remaining tasks

You have completed installing PE on the other nodes in your system.

To determine which remaining steps you need to perform, refer to Table 13:

Table 13. Steps to take to determine steps remaining

If you installed POE:	If you did not install POE:
Proceed to: • "Step 4: Verify the POE installation" on page 23	Skip: • "Step 4: Verify the POE installation" on page 23

Step 4: Verify the POE installation

This step consists of testing the installation of POE, using the POE installation verification program (IVP). You can find this program in **/usr/lpp/ppe.poe/samples/ivp**.

Note: In order to successfully run the IVP, you will need to have **rsct.lapi.rte** already installed.

To run the POE IVP, at the control workstation (or other home node):

LOGIN
as a user other than **root**, and start **ksh**.

ENTER
export LANG=C

ENTER
cd /usr/lpp/ppe.poe/samples/ivp

ENTER
./ivp.script

This runs an installation verification test that checks if the message-passing program successfully executed using two tasks on this node. The output should resemble the following:

```
Verifying the existence of the Binaries
Partition Manager daemon /etc/pmdv4 is executable
POE files seem to be in order
Compiling the ivp sample program
Output files will be stored in directory /tmp/ivp495786
Creating host.list file for this node
Setting the required environment variables
Executing the parallel program with 2 tasks

Threaded 32bit library built on: Apr 21 2003 12:51:46 level(CS2A_Pre-build).
POE IVP: running as task 0 on node c284f2ih01
POE IVP: there are 2 tasks running
POE IVP: running as task 1 on node c284f2ih01
POE IVP: all messages sent
POE IVP: task 1 received <POE IVP Message Passing Text>

Parallel program ivp.out return code was 0

Executing the parallel program with 2 tasks, threaded library

Threaded 32bit library built on: Apr 21 2003 12:51:46 level(CS2A_Pre-build).
POE IVP_r: running as task 0 on node c284f2ih01
POE IVP_r: there are 2 tasks running
POE IVP_r: all messages sent
POE IVP_r: running as task 1 on node c284f2ih01
POE IVP_r: task 1 received <POE IVP Message Passing Text -
           Threaded Library>

Parallel program ivp_r.out return code was 0

If both tests return a return code of 0, POE IVP
 is successful.  To test system message passing,
 run the tests in /usr/lpp/ppe.poe/samples/poetest.bw and poetest.cast
 To test threaded message passing,
 run the tests in /usr/lpp/ppe.poe/samples/threads
End of IVP test
```

If errors are encountered, your output contains messages that describe these errors. You can correct the errors and run the **ivp.script** again, if desired.

Additional POE sample applications – POE also has sample applications for doing the following:

- Point-to-point bandwidth measurement tests
- Broadcast from task 0 to all of the rest of the nodes in the partition
- MPI Threads sample programs

See Appendix C, “Using additional POE sample applications,” on page 53 for more information.

View the README file after installation –

Once you have installed the PE file sets, refer to the README file provided with each file set for any additional installation or usage information. You can find the README file in **/usr/lpp/***fileset***/READMES** as *fileset***.README**.

For information about other procedures related to PE installation, see Chapter 5, “Performing installation-related tasks,” on page 29.

Chapter 4. Migrating and upgrading PE

These instructions explain how to migrate from earlier releases of PE to PE 4.3. There are differences between earlier releases that you need to consider before installing or using PE 4.3. When we refer to PE Version 4 or PE 4.3, we mean the latest version of PE, which is PE PE 4.3, unless otherwise specified.

PE Version PE 4.3 is the latest available supported level of PE Version 4. Customers running with PE Version 4.2.2, 4.2.1 or 4.2.0 are encouraged to install PE 4.3 to obtain the latest available service levels for PE Version 4.3. To find out which release of PE you currently have installed, issue the **lslpp** command.

You may need to reference PE documentation during migration. If so, see "Prerequisite and related information" on page xi for more information.

General overview

If you have an earlier release of PE already installed, installing the PE Version 4 file sets involves a migration installation on top of the earlier file sets. The earlier file sets will be *completely* replaced, unnecessary files and directories will be removed and rendered obsolete, and disk space conserved.

Because some existing files, for example the compiler utility scripts, may have been modified, these files are saved before they are replaced. The files are saved in the **/usr/lpp/save.config/usr/lpp/ppe.poe/bin** directory.

There are several files saved as part of the migration installation, in case those files were previously modified. For specific details, refer to "How installing the POE file set alters your system" on page 33.

To the Object Data Manager (ODM) and **lslpp**, however, the earlier file sets will show as installed but marked *OBSOLETE*. Also, some older directories and installation-related files may remain.

Note that if you later attempt to remove an older file set, files from the *newer* file set may be removed instead. To avoid this potential side effect, completely remove older releases of the PE file sets *before* you begin installation. If your installation currently has **ppe.vt** or **ppe.pedb** file sets installed, you should remove them because PE Version 4 does not support them. These file sets are not automatically removed or marked obsolete by newer installations of PE, although they should no longer be used with current versions of PE. For more details, see "Migration installation" on page 13.

AIX compatibility

PE Version 4 commands and applications are compatible with AIX 5.3 only; not with earlier versions of AIX.

Coexistence

All nodes in a parallel job must be running the same versions of PE and LoadLeveler, at the same maintenance levels.

When LoadLeveler and PE coexist on a node, they must be one of the following:

- LoadLeveler 3.4 with PE Version 4.3 or later
- LoadLeveler 3.4 with PE Version 4.2.2
- LoadLeveler 3.3.1 or later with Parallel Environment 4.2.2.

It is recommended that both PE 4.3 and LoadLeveler 3.4 be installed at their latest support levels to provide the latest functional support. Some important functions are not available in earlier versions. For example, in order to use the LoadLeveler scheduling affinity function, you must have LoadLeveler 3.3.1 (or later) installed.

In order to use the support for the user RDMA context (rCxt) blocks, you must be running AIX 5L V5.3 TL 5300-05 and you must have LoadLeveler 3.4 or 3.3.1 and LAPI 2.4.2 installed.

As of PE 4.3, PE no longer supports PSSP. PE 4.2 was the last release to support PSSP 3.5. Also, PE 4.3 only supports AIX 5L V5.3 TL 5300-05 (there is no support for AIX 5.2).

Beginning with PE 4.2, the SP Switch is no longer supported.

Migration support

PE does not support node-by-node migration. You must migrate all of the nodes in a system partition or parallel cluster to a new level of PE at the same time.

In general, the preferred upgrade path for PE is to upgrade the AIX level and then the PE level. There are a number of migration paths available:

1. AIX 5.1 PSSP 3.4, and PE 3.2 to AIX 5L V5.3 TL 5300-05 and PE 4.2
2. AIX 5.1 PSSP 3.4, and PE 3.2 to AIX 5L V5.3 TL 5300-05 and PE 4.3
3. AIX 5.2 PSSP 3.5, and PE 4.1.0 or 4.1.1 to AIX 5L V5.3 TL 5300-05 and PE 4.2
4. AIX 5.2 PSSP 3.5 and PE 4.2 to AIX 5L V5.3 TL 5300-05 and PE 4.2
5. AIX 5.2 PSSP 3.5 and PE 4.2 to AIX 5L V5.3 TL 5300-05 and PE 4.3
6. AIX 5.2 (no PSSP), and PE 4.1.0 or 4.1.1 to AIX 5L V5.3 TL 5300-05 and PE 4.2
7. AIX 5.2 (no PSSP) and PE 4.1.0 or 4.1.1 to AIX 5L V5.3 TL 5300-05 and PE 4.3
8. AIX 5.2 (no PSSP) and PE 4.2 to AIX 5L V5.3 TL 5300-05 and PE 4.2
9. AIX 5.2 (no PSSP) and PE 4.2 to AIX 5L V5.3 TL 5300-05 and PE 4.3
10. AIX 5.3 (no PSSP) and PE 4.2 to AIX 5L V5.3 TL 5300-05 and PE 4.3

AIX Support

PE Version 4.3 supports AIX 5L Version 5.3 Technology Level 5300-05 (AIX 5L V5.3 TL 5300-05). The pSeries High Performance Switch (HPS) is now supported on AIX 5L V5.3 TL 5300-05.

PE Version 4.3 also provides support for AIX 5L V5.3 TL 5300-05 threaded profiling support. See *IBM Parallel Environment for AIX 5L: Operation and Use, Volume 2* for more information.

Note that under AIX 5.3, PE Version 4.3 requires LAPI (**rsct.lapi.rte**) Version 2.4.3.

MPI library support

PE Version 4 provides support for its threaded version of the MPI library only. A non-threaded (or signal based) library is also shipped, and its symbols are exported from the threaded library, **libmpi_r.a**, for binary compatibility.

Binary compatibility is supported for existing applications that have been dynamically linked or created with the non-threaded compiler scripts from previous versions of POE. There is no binary compatibility for statically bound executables.

Existing applications built as non-threaded applications will execute as single threaded applications in the PE Version 4 environment. Users and application developers should understand the implications of their programs running as threaded applications, as described in the appropriate sections of the *MPI Programming Guide*.

LAPI support

Beginning with PE 4.3, LAPI is shipped as a file set on the PE product CD. As in the previous release, this file set is called **rsct.lapi**. and contains three install images; **rsct.lapi.rte**, **rsct.lapi.nam**, and **rsct.lapi.samp**.

MPI uses LAPI as a message transport protocol. MPI users will require LAPI to be previously installed. Users and application developers may need to understand this relationship, as described in the appropriate sections of the PE *MPI Programming Guide*, and the *RSCT for AIX 5L: LAPI Programming Guide*.

Online documentation

The location of the online documentation has changed. Previously, it was available via the Web in PDF format at **http://www-03.ibm.com/servers/eserver/pseries/library/sp_books/**, or from the IBM eServer Cluster Information Center at: **http://publib.boulder.ibm.com/clresctr/windows**. As of PE 4.3, the location of the IBM eServer Cluster Information Center has changed to **http://publib.boulder.ibm.com/infocenter/clresctr/vxrx/index.jsp**.

PE Version 4 continues to ship man pages, in the **ppe.man** file set, which completely replaces earlier versions of the man pages already installed. For more information, see *Migration Installation* and *How Installing the Online Documentation Alters Your System*.

Chapter 5. Performing installation-related tasks

After you have finished installing PE, there are a number of tasks that you may need to perform from time to time that are related to the original installation procedure provided in Chapter 3, “Installing the PE software,” on page 13. These tasks include removing a software component and customizing the message catalog.

Removing a software component

During the installation process, you may decide to remove a PE software component from the system. If you have already installed it on a number of nodes, you can use the **PEdeinstall** script provided with PE, to do the removals.

For detailed information about this script and instructions describing how to run it, see “Deinstallation script: PEdeinstall” on page 48.

Recovering from a software vital product database error

If you install PE frequently, you may encounter an error such as:

```
0503-283 :  Error in the Software Vital Product Data.  The "usr"
        part of a product does not have the same requisite file
        as the "root" part.  The product is: ppe.poe 4.3
```

This usually means that there is an incompatibility in the Object Data Manager (ODM). This could be as a result of installing a version of a product where prerequisites may have changed.

You need to remove the entries for a product from ODM. The following set of commands removes the entries for POE (the ppe.poe file set). To remove entries for a different file set, replace ppe.poe in the following commands with the appropriate file set name.

```
ODMDIR=/usr/lib/objrepos odmdelete -o product -qlpp_name=ppe.poe
ODMDIR=/usr/lib/objrepos odmdelete -o lpp -qname=ppe.poe
ODMDIR=/etc/objrepos odmdelete -o product -qlpp_name=ppe.poe
ODMDIR=/etc/objrepos odmdelete -o lpp -qname=ppe.poe
```

Customizing the message catalog

All PE file sets use message cataloging so that messages can appear in languages other than English. Each file set has message catalogs installed in a directory located by the **NLSPATH** environment variable. The message catalogs are installed in three common English language paths and are in the format of *component***.cat**. The paths are:

/usr/lib/nls/msg/C
/usr/lib/nls/msg/En_US
/usr/lib/nls/msg/en_US

1. Before verifying the installation for POE, you should set the **LANG** environment variable to **C**.
2. If the message catalogs are installed in a directory other than **C**, modify **/etc/environment** to set the **NLSPATH** to the appropriate directory. You also need to set the user’s **LANG** environment variable.

Installing AFS

These are the instructions for tailoring the parallel operating environment for execution with the AFS file system. The source files **settokens.c** and **gettokens.c** are intended to be used with Transarc's Kerberos Authentication program, but should be usable as a guide for other environments.

The files needed for setting up the AFS execution are in the **/usr/lpp/ppe.poe/samples/afs** directory. They are:

README.afs
README file that contains much of the same information contained here.

gettokens.c
Subroutine to get an AFS token on the node where the user is logged on (or already authenticated)

settokens.c
Subroutine to put an AFS token on the remote node that is running the user's executable

makefile
Makefile for creating object modules from **settokens.c** and **gettokens.c**

buildAFS
Sample shell script for replacing the routines **settoken** and **gettokens** distributed with POE by the routines built by the makefile

Setting up POE for AFS execution

Perform the following procedure as **root** for setting up POE for AFS execution:

ENTER
cd /usr/lpp/ppe.poe/samples/afs to switch to the appropriate directory or copy the contents of the directory to a convenient location.

ENTER
the **make** command to create the files **settokens.o** and **gettokens.o** from **gettokens.c** and **settokens.c**. If you are not using the Transarc system, you may need to alter these routines to provide the desired token access. The calling sequence of the parameters cannot be changed.

VERIFY
that the partition manager daemon, **pmdv4**, the home node partition manager, **poe**, and the parallel debugger, **pdbx**, are in **/usr/lpp/ppe.poe/bin**. If not, modify the **buildAFS** script.

Before completing the following step, ensure that you have the following amounts of available space in the current directory, as shown in Table 14:

Table 14. Space requirements for pdbx, pmdv4, and poe components

Component(s) being built	Total available space required (in megabytes)
pdbx, **pmdv4**, **poe**	2

ENTER
buildAFS to create new versions of **pdbx**, **pmdv4**, and **poe** in the current directory. If the linking step fails, locate the libraries containing the modules that were not found, and alter the library search list in **buildAFS** to include them.

MOVE **pdbx**, **pmdv4**, and **poe** to their usual location in **/usr/lpp/ppe.poe/bin** on each node. You can rename the old versions in case they need to be restored. Make sure that they are made executable.

You should not have to modify your program executables. You can now pass AFS authorization across the partition.

The **.rhosts** file in the user's home directory must include the nodes that are intended for Parallel Operating Environment use. This ensures that the proper access is permitted.

Chapter 6. Understanding how installing PE alters your system

Your system is altered when you install the various PE software file sets. Directories and files are created, the daemon processes are created, and links are established by the installation process.

How installing the POE file set alters your system

The **ppe.poe** file set includes all of the components of the parallel operating environment (POE), and consists of:

- API subroutine libraries (message passing and collective communication)
- Parallel compilation scripts
- Parallel profiling capability
- The parallel utility library
- Partition manager
- The **pdbx** debugger

Installing this file set, as described in "Step 2: Perform the initial installation" on page 17, does the following:

1. Creates the directories and files detailed in Table 15:

Table 15. POE directories and files installed

Directory or file	Description
/usr/lib/nls/msg/en_US/pempl.cat /usr/lib/nls/msg/En_US/pempl.cat /usr/lib/nls/msg/C/pempl.cat	Message Catalog for Message Passing Library
/usr/lib/nls/msg/en_US/pepdbx.cat /usr/lib/nls/msg/En_US/pepdbx.cat /usr/lib/nls/msg/C/pepdbx.cat	Message Catalog for pdbx
/usr/lib/nls/msg/en_US/pepoe.cat /usr/lib/nls/msg/En_US/pepoe.cat /usr/lib/nls/msg/C/pepoe.cat	Message catalog for POE
/usr/lpp/ppe.poe/bin/mpamddir	Shell script for echoing an AMD mountable directory name
/usr/lpp/ppe.poe/bin/mcp	Executable for multiple file copy utility
/usr/lpp/ppe.poe/bin/mcpgath	Executable for parallel file copy gather utility
/usr/lpp/ppe.poe/bin/mcpscat	Executable for parallel file copy scatter utility
/usr/lpp/ppe.poe/bin/mpcc_r	Shell script for compiling threaded parallel C programs
/usr/lpp/ppe.poe/bin/mpCC_r	Shell script for compiling threaded parallel C++ programs
/usr/lpp/ppe.poe/bin/mpiexec	Portable MPI startup script
/usr/lpp/ppe.poe/bin/mpxlf_r	Shell script for compiling threaded parallel FORTRAN programs
/usr/lpp/ppe.poe/bin/mpxlf90_r	Shell script for compiling threaded parallel FORTRAN 90 programs
/usr/lpp/ppe.poe/bin/mpxlf95_r	Shell script for compiling threaded parallel FORTRAN 95 programs

Table 15. POE directories and files installed (continued)

Directory or file	Description
/usr/lpp/ppe.poe/bin/pdbx	Executable to run the command-line interface of the PE debugging facility
/usr/lpp/ppe.poe/bin/PEdeinstall	Shell script to remove an installation of PE on IBM pSeries nodes
/usr/lpp/ppe.poe/bin/PEinstall	Shell script to complete the installation process on IBM pSeries nodes
/usr/lpp/ppe.poe/bin/pmadjpri	Dispatching priority adjustment coscheduler daemon
/usr/lpp/ppe.poe/bin/pmdv4	A daemon process that runs on each of your processor nodes
/usr/lpp/ppe.poe/bin/poe	Partition manager executable
/usr/lpp/ppe.poe/bin/poeckpt	Executable for checkpointing interactive POE applications
/usr/lpp/ppe.poe/bin/poerestart	Executable for restarting POE applications
/usr/lpp/ppe.poe/bin/poekill	Shell script for terminating all POE started tasks
/usr/lpp/ppe.poe/bin/pm_set_affinity	Executable for task affinity assignment
/usr/lpp/ppe.poe/bin/rset_query	Executable for displaying affinity resources
/usr/lpp/ppe.poe/include	Directory of header files containing declarations used by other installed files
/usr/lpp/ppe.poe/include/pm_ckpt.h	Header for compiling programs with Checkpoint and Restart capability
/usr/lpp/ppe.poe/include/thread/mpi.mod	MPI Fortran module support (use MPI)
/usr/lpp/ppe.poe/include/thread64/mpi.mod	MPI Fortran 64–bit module support (use MPI)
/usr/lpp/ppe.poe/include/thread64/mpif.h	Header for compiling 64–bit threaded MPI Fortran applications
/usr/lpp/ppe.poe/lib/libmpi.a	Archive library containing subroutines for parallel message-passing programs
/usr/lpp/ppe.poe/lib/libmpi_r.a	Archive library containing subroutines for parallel message-passing programs in a threads environment
/usr/lpp/ppe.poe/lib/libppe.a /usr/lpp/ppe.poe/lib/libppe_r.a	Archive library containing subroutines for POE
/usr/lpp/ppe.poe/lib	Directory containing shared libraries and objects used by POE and MPI programming interfaces.
/usr/lpp/ppe.poe/lib/libpoeapi.a	Archive library containing subroutines for the POE API
/usr/lpp/ppe.poe/README S/poe.README	Memo to users relating to this release
/usr/lpp/ppe.poe/samples	Directory containing sample programs for the program marker array and other samples
/usr/lpp/ppe.poe/include/poeapi.h	Header file for the POE API
/usr/lpp/ppe.poe/include/thread/mpif.h	Header file for compiling threaded MPI FORTRAN applications
/usr/lpp/ppe.poe/samples/scripts/poewhere	Script for displaying the stack trace for each thread of a program
/usr/lpp/ppe.poe/samples/swtbl	Directory containing sample code for running User Space POE jobs without LoadLeveler
/usr/lpp/ppe.poe/samples/ntbl	Directory containing sample code for running user space jobs without LoadLeveler, using the network table API

Table 15. POE directories and files installed (continued)

Directory or file	Description
/etc/poe.security	Security method configuration file

2. When the **installp** command successfully restores POE's files from the distribution medium, the command looks at the **ppe.poe.post_i** file for post–installation steps. As part of these post–installation steps, **ppe.poe.post_i** sets up the symbolic links, as shown in Table 16:

Table 16. ppe.poe.post_i symbolic links

This link:	To:
/etc/pmdv4	/usr/etc/pmdv4
/usr/bin/mpcc	/usr/lpp/ppe.poe/bin/mpcc_r
/usr/bin/mpcc_r	/usr/lpp/ppe.poe/bin/mpcc_r
/usr/bin/mpCC	/usr/lpp/ppe.poe/bin/mpCC_r
/usr/bin/mpCC_r	/usr/lpp/ppe.poe/bin/mpCC_r
/usr/bin/mpamddir	/usr/lpp/ppe.poe/bin/mpamddir
/usr/bin/mpxlf	/usr/lpp/ppe.poe/bin/mpxlf_r
/usr/bin/mpxlf_r	/usr/lpp/ppe.poe/bin/mpxlf_r
/usr/bin/mpxlf90	/usr/lpp/ppe.poe/bin/mpxlf90_r
/usr/bin/mpxlf90_r	/usr/lpp/ppe.poe/bin/mpxlf90_r
/usr/bin/mpxlf95	/usr/lpp/ppe.poe/bin/mpxlf95_r
/usr/bin/mpxlf95_r	/usr/lpp/ppe.poe/bin/mpxlf95_r
/usr/bin/mcp	/usr/lpp/ppe.poe/bin/mcp
/usr/bin/mcpgath	/usr/lpp/ppe.poe/bin/mcpgath
/usr/bin/mcpscat	/usr/lpp/ppe.poe/bin/mcpscat
/usr/bin/mpiexec	/usr/lpp/ppe.poe/bin/mpiexec
/usr/bin/pdbx	/usr/lpp/ppe.poe/bin/pdbx
/usr/bin/pmdadjpri	/usr/lpp/ppe.poe/bin/pmdadjpri
/usr/bin/poe	/usr/lpp/ppe.poe/bin/poe
/usr/bin/poeckpt	/usr/lpp/ppe.poe/bin/poeckpt
/usr/bin/poekill	/usr/lpp/ppe.poe/bin/poekill
/usr/bin/poerestart	/usr/lpp/ppe.poe/bin/poerestart
/usr/bin/rset_query	/usr/lpp/ppe.poe/bin/rset_query
/usr/etc/pmdv4	/usr/lpp/ppe.poe/bin/pmdv4
/etc/pm_set_affinity	/usr/lpp/ppe.poe/bin/pm_set_affinity
/usr/sbin/PEdeinstall	/usr/lpp/ppe.poe/bin/PEdeinstall
/usr/sbin/PEinstall	/usr/lpp/ppe.poe/bin/PEinstall

3. During installation, if an existing version of **ppe.poe** is installed, the following files are saved during installation of the new version in the **/usr/lpp/save.config** directory:

```
/etc/poe.security
/usr/lpp/ppe.poe/bin/mpamddir
/usr/lpp/ppe.poe/bin/mpcc
/usr/lpp/ppe.poe/bin/mpCC
/usr/lpp/ppe.poe/bin/mpcc_r
```

```
/usr/lpp/ppe.poe/bin/mpCC_r
/usr/lpp/ppe.poe/bin/mpxlf
/usr/lpp/ppe.poe/bin/mpxlf90
/usr/lpp/ppe.poe/bin/mpxlf95
/usr/lpp/ppe.poe/bin/mpxlf_r
/usr/lpp/ppe.poe/bin/mpxlf90_r
/usr/lpp/ppe.poe/bin/mpxlf95_r
/usr/lpp/ppe.poe/lib/poe.cfg
/usr/lpp/ppe.poe/bin/makelibc
```

If these files were previously modified, the older versions are preserved in the **/usr/lpp/save.config** directory and the new versions will need to be updated.

POE installation effects

Also, as part of the post-installation steps, the following changes occur:

1. The file **/etc/services** is modified in the following manner:
   - If no entry for the pmv4 service is found, an entry is added using port 6127/tcp.
   - If an entry exists for the pmv4 service that uses port 6127/tcp, no change is made to the **/etc/services** file.
   - If one of the following is true:
     a. A pmv4 entry exists for a port other than 6127/tcp
     b. A 6127/tcp entry exists for a service other than pmv4

     the user receives a warning and is instructed to correct the problem before running POE. When the user receives this warning, he must manually update the **/etc/services** file to ensure that the port number for the pmv4 service is the same on all machines that could run POE Version 4.
2. The file **/etc/inetd.conf** is modified.

   An entry for the pmv4 service that spawns the **/etc/pmdv4** daemon is created if no pmv4 entry exists.
3. **inetd** is refreshed.
4. If a symbolic link for **/usr/etc/digd** to **/usr/lpp/ppe.vt/bin/digd** exists, but **/usr/lpp/ppe.vt/bin/digd** itself does not exist, the link is removed.
5. If **/usr/lpp/x11/lib/x11/app-defaults/PMarray** exists, it is removed, along with the subdirectory if it is empty.
6. Executable versions of **mcp**, **mcpgath**, and **mcpscat** are created.

How installing the ppe.perf and ppe.pvt file sets alters your system

PE Benchmarker consists of the Performance Collection Tool **ppe.perf** file set and the Profile Visualization Tool **ppe.pvt** file set. In addition, if **ppe.perf** is installed, **ppe.dpcl** must also be installed. Installing PE Benchmarker creates the directories and files detailed in Table 17:

Table 17. ppe.perf and ppe.pvt directories and files installed

Directory or file	Description
/usr/lpp/ppe.perf/README$	Directory containing necessary files to be read by the customer.
/usr/lpp/ppe.perf/bin	Directory containing executables.

Table 17. ppe.perf and ppe.pvt directories and files installed (continued)

Directory or file	Description
/usr/lpp/ppe.perf/config	Directory containing configuration files PCT uses to determine hardware counter group. Also contains a dictionary file that describes the OpenMP runtime functions.
/usr/lpp/ppe.perf/deinstl	Directory containing packaging information for uninstalling the product.
/usr/lpp/ppe.perf/help	Directory containing help images and files.
/usr/lpp/ppe.perf/lib	Directory containing libraries used by PCT.
/usr/lpp/ppe.perf/resources	Directory containing JAVA resource bundles used for labelled text in graphical user interface.
/usr/lpp/ppe.perf/samples	Directory containing sample programs and scripts.
/usr/lpp/ppe.perf	Directory containing the necessary Performance Collection Tool files.
/usr/lpp/ppe.pvt	Directory containing the necessary Profile Visualization Tool files.

In addition, symbolic links set up for PE Benchmarker files and libraries are placed in the following directories.

- **/usr/include**
- **/usr/lib**
- **/etc**

How installing the online documentation alters your system

The online documentation is composed of the following file sets:

- **ppe.man**: contains the PE man pages

These file sets completely replace the contents of the **ppe.pedocs** file set that existed in earlier versions of PE.

Installing these file sets “Step 2: Perform the initial installation” on page 17 creates the directories and files detailed in Table 18.

When you migrate from earlier versions of the **ppe.pedocs** file set, the files previously installed are removed. The file set is changed to an OBSOLETE state in the SWVPD and ODM.

The **ppe.man** file set includes files that contain the PE man pages, as described in Table 18. Once you have installed the **ppe.man** file set, you can find the man pages in the appropriate path: **/usr/man/cat1** or **/usr/man/cat3**.

Table 18. Man page directories and files installed

Directory or file	Description
/usr/man/cat1	Directory containing man page files for PE commands

Table 18. Man page directories and files installed (continued)

Directory or file	Description
/usr/man/cat3	Directory containing man page files for API message-passing subroutines
/usr/lpp/ppe.man/README S/ppe.man.README	Installation README file

Online pdf documentation

You can view, search, and print documentation in PDF format on the World Wide Web at:

http://www.ibm.com/servers/eserver/pseries/library/sp_books

or from the IBM eServer Cluster Information Center at:

http://publib.boulder.ibm.com/clresctr/windows

Chapter 7. Additional information for the system administrator

System administrators should familiarize themselves with the formats of the PE files that they will create and edit (in **/etc**). These files are used for configuring the coscheduler, overriding default environment variable values, choosing a security method, and enabling RDMA.

Configuring the Parallel Environment coscheduler

The PE coscheduler works by alternately, and synchronously, raising and lowering the AIX dispatch priority of the tasks in an MPI job. The objective is for all the tasks to have the same priority across all processors, and to force other system activity into periodic and aligned time slots during which the MPI tasks do not actively compete for CPU resources.

The synchronous operation of the coscheduler is effective only if the global switch clock capability of the High Performance Switch is operational. In other environments, the coscheduler uses the local AIX time on the node to determine when to change priorities.

There are two components of the PE coscheduler support: the POE coscheduling parameters and limits, and the AIX dispatcher tuning parameters.

POE coscheduling parameters and limits

A coscheduler activation is specified by the following:

- The high (favored) priority value.
- The low (unfavored) priority value.
- The percentage of time that the MPI tasks will be set to their favored priority value.
- The period of alternation.

The values above can be specified on a per-user or class basis, with limits set by the system administrator in either case. The limits and classes are defined in the **/etc/poe.priority** file. It is assumed that this file is the same on each node in the cluster.

The range of parameters permitted in the adjustment record is purposely set to be as unrestricted as possible. The user and system administrator (who owns the configuration file) must evaluate the effect of various parameter settings in their own operating environment. Carefully read the notes accompanying the file format description. The following are descriptions of the parameters.

username
: name of the user. Wildcards are allowable for the user name, in the form of an asterisk (*). For wild card values, these will allow defaults to be set for a user that does not have an explicit entry defined. When the file contains an entry for a specific user, that entry constrains the values for that user, regardless of the wild card values.

classname
: name assigned to the class, to which the **MP_PRIORITY** value can be set. Additionally, there can be additional constraints defined using the MAXIMUM and MINIMUM class entries, on a *first match* basis, where the

first match for that user takes precedence. Also, a MAXIMUM or MINIMUM can be defined for a particular user, meaning that user cannot exceed those values.

hipriority
the dispatching priority assigned to the favored portion of the cycle.

lopriority
the dispatching priority assigned to the rest of the cycle.

percenthi
the percentage of the cycle at which the job is at hipriority (percent).

period
length of adjustment cycle, in seconds.

Records can be in the following format:

```
# user     class     hipri     lopri     percentage      period
# ----     -------   -----     -----     ----------     ------
  pfc      special     40        100          90.5          5
  *        ten50       50        100          90            5
  *        MAXIMUM    100        100          97           10
  *        MINIMUM     40         60           0            1
  trj      ten40       40        100          90            5
```

Furthermore, the *first match* policy also applies to the case where there are multiple entries for a user - the first entry found for that user will take precedence.

For example, considering the following entries:

```
# user     class     hipri     lopri     percentage     period
# ----     -------   -----     -----     ----------     ------
  trj      MAXIMUM     90        100          97           10
  *        MAXIMUM    100        100          97           10
  ibm      MAXIMUM     80        100          97           10
```

- user trj cannot go above 90 for the hipri value
- everyone else (including user ibm) can go up to 100
- user ibm's entry, because it follows the wildcarded MAXIMUM, is ignored

The **MP_PRIORITY** environment variable may be specified in one of two forms:

- a class name as the only value, or
- a colon separated list of values specified by the user, for the key parameters, in the following format:

  ```
  hipriority:lopriority:percentage:period
  ```

The values specified or implied by the **MP_PRIORITY** variable will be evaluated against the MAXIMUM and MINIMUM settings in the **/etc/poe.priority** file, and they will only take effect under the following conditions:

- when a MAXIMUM setting is specified in the file, and each value in the environment variable is less than or equal to the corresponding value in the file.
- when a MINIMUM setting is specified in the file, and each value in the environment variable is greater than or equal to the corresponding value in the file.

Comments are allowed in the file, when preceded by the # sign, such that everything following the # will be ignored.

1. The normal AIX dispatching priority is 60. If both **hipriority** and **lopriority** are set to values less than 60, a compute bound job will prevent other users from being dispatched.

2. The **hipriority** value must be equal to or greater than 12. If the value is between 12 and 20, the job competes with system processes for cycles, and may disrupt normal system activity.
3. If **hipriority** value is less than 30, keystroke capture will be inhibited during the **hipriority** portion of the dispatch cycle.
4. If **hipriority** is less than 16, the job will not be subject to the AIX scheduler during the high priority portion of the cycle.
5. The **lopriority** value must be less than or equal to 254.
6. If the **hipriority** value is less than (more favored than) the priority of the high performance switch fault-service daemon, and if the low priority portion of the adjustment cycle is less than two seconds, then switch fault recovery will be unsuccessful, and the node will be disconnected from the switch.
7. The coscheduling process allows programs using the User Space library to maximize their effectiveness in interchanging data. The process may also be used for programs using IP, either over the switch or over another supported device. However, if the high priority phase of the user's program is more favored than the network processes (typically priorities 36-39), the required IP message passing traffic may be blocked and cause the program to hang.
8. Consult the include file **/usr/include/sys/pri.h** for definitions of the priorities used for normal AIX functions.
9. The parameter file **/etc/poe.priority** defines the scheduling parameters for tasks running on that node.
10. The **MP_PRIORITY_NTP** environment variable determines whether the POE priority adjustment coscheduler will turn NTP off during the priority adjustment period, or leave it running. The value of **no** (which is the default) instructs the POE coscheduler to turn the NTP daemon off (if it was running) and restart NTP later, after the coscheduler completes. Specify a value of **yes** to inform the coscheduler to keep NTP running during the priority adjustment cycles (if NTP was not running, NTP will not be started). The value of this environment variable can be overridden using the **-priority_ntp** flag.

AIX dispatcher tuning

The AIX dispatcher is tuned by setting parameters of the **schedo** command. The two parameters of particular interest to the coscheduler are:

big_tick_size
: Sets the scheduling time slice interval, in units of 10 milliseconds. The default is 1 (corresponding to the normal AIX 10 millisecond time slice). The value can be as large as 100, which would make the interval between dispatcher activations 1000 milliseconds (one second). The value must also divide evenly into 100.

  Between activations, tasks running on a processor are not examined for replacement unless they do I/O or voluntarily yield the processor. Because running the dispatcher itself takes some time, increasing the value of the **big_tick_size** parameter reduces the overhead for dispatching, but may not provide CPU cycles to some system activities as often as they would desire.

force_grq
: If enabled, assigns all processes, that are not part of a PE/MPI job, to the Global Run Queue. The intention is to allow all non-MPI activity to compete equally for the block of CPU resource that becomes available periodically.

Without setting this option, non-MPI processes may queue up for the processor they used previously, even if that processor is busy and another processor is idle.

Notes:

1. These options are only fully effective if the AIX kernel is running with the *real time* option, which is enabled by:
   - **bosdebug -R on**
   - **bosboot -a** (assuming that the existing kernel is to be used)
   - **shutdown -Fr** (to reboot the node).

   After the reboot is complete, the presence of the *real time* option may be verified by displaying the value of the symbol **rt_kernel** from the kdb debugger. If it is nonzero, the *real time* option has been successfully enabled.
2. Setting the **big_tick_size** option to a value other than 1, in combination with the real time option, has the side effect of synchronizing the dispatcher activations on a node, so that all processor time slices end at the same time. This is in contrast to the normal operation of the AIX dispatcher, in which the time slice ends are deliberately offset within the 10 millisecond period to minimize contention for locks on AIX control structures. Also, the time slice ends are synchronized to the AIX system clock, so that one of the time slices ends at an even number of seconds. In other words, the fractional seconds must be zero (HH:MM:SS:00). The time-of-day synchronization of the time slices only occurs if **big_tick_size** is greater than 1.
3. Returning **big_tick_size** to 1 does not reset this time slice offset, which persists for the life of the kernel session.
4. Changes to **big_tick_size** and **force_grq** can only be made by a root user, and take effect immediately without a reboot. If **force_grq** is set to zero, the normal AIX mechanism of trying to reassign a process to its previous processor is resumed.

Using the /etc/poe.limits file

An optional file named **poe.limits** can be created in the **/etc** directory, enabling the system administrator to override the default values for certain POE environment variables, and to limit the value set by a user. This is useful in cases where the environment variable default values might cause problems on a particular node. For example, if a node had only 64M of real memory, the default value of 64M for **MP_BUFFER_MEM** would be too high. To correct this problem, the system administrator would specify a lower value for **MP_BUFFER_MEM** in the **/etc/poe.limits** file on that node.

Entries in the /etc/poe.limits file

Entries in the **/etc/poe.limits** file must be in the form:

supported_object **=** *value*

where *supported_object* is currently limited to the following:

- **MP_BUFFER_MEM**
- **MP_USE_LL**

For **MP_BUFFER_MEM**, you can provide a value in one of two ways:

- Specify a single value to indicate the pool size for memory to be allocated at MPI initialization time and dedicated to buffering of early arrivals.

- Specify two values. The first value indicates the pool size for memory to be allocated at MPI initialization time (pre_allocated_size). The second value indicates an upper bound of memory to be used if the pre-allocated pool is not sufficient (maximum_size). Note that when you specify two values, you must delineate them with a comma. Spaces before or after the comma are not allowed. If you omit the first value (start the value string with a comma), the pre_allocated_size will be set to the default (64 MB).

Note also:

- If the value of **MP_BUFFER_MEM** is set in **/etc/poe.limits** on one node, the same value must be specified as an entry in **/etc/poe.limits** on all other nodes. If the nodes are set to different values, some jobs may fail.
- If the preallocated size of **MP_BUFFER_MEM** is set to less than 64 MB, this smaller value becomes the default. If the preallocated value of **MP_BUFFER_MEM** is set to larger than 64 MB, this larger value becomes the limit to which the MPI library sets the preallocated size (but the default remains 64 MB).

For more information about specifying values for **MP_BUFFER_MEM**, see *IBM Parallel Environment: Operation and Use, Volume 1*.

Note: Any line in the **/etc/poe.limits** file with the character **#** or **!** in the first column is treated as a comment.

How the Partition Manager daemon handles the /etc/poe.limits file

If the **/etc/poe.limits** file has been set up on a particular node, the Partition Manager daemon (**pmdv4**) on that node performs the following:

1. Compares the values specified in the **/etc/poe.limits** file against the environment variables received from the home node
2. If necessary, resets the environment variables as follows:

   MP_BUFFER_MEM

   If the value in the environment *exceeds* the value specified in **/etc/poe.limits**, **pmdv4** resets the value to that specified in **/etc/poe.limits**.

   MP_USE_LL

   If the value in the file is set to **yes** and POE determines that the job is not being run under LoadLeveler, the job is terminated. Setting the value to **no** has no effect.
3. If a *supported_object* is specified in **/etc/poe.limits** but is *not set* in the environment, sets the value to that specified in **/etc/poe.limits**

Note: If the **/etc/poe.limits** file contains entries with either unsupported objects to the left of the equal sign or with invalid (nonnumeric for **MP_BUFFER_MEM**) values to the right, the Partition Manager daemon flags these entries in the **pmdlog** for that node. The Partition Manager daemon also uses the **pmdlog** to indicate when a *supported_object* has been set or reset in the environment.

Description of /etc/poe.security

The **/etc/poe.security** file contains a simple ASCII text entry of one of the following:

1. COMPAT - where the previously defined AIX or DCE based security will be used for compatibility (this is the default).
2. CTSEC - where the clusters based CTSec security methods are to be used.

The contents of the file are case insensitive, and will allow for leading and trailing spaces, and blank lines. Only one value is expected. If multiple values or invalid values are specified, a terminating error and message will occur.

This file is owned and writable only by root, so only systems administrators with root level access can update it. The method specified must be the same throughout all of the nodes in a parallel job - they cannot be mixed. The lack of an entry in **/etc/poe.security** (or the lack of the file altogether) is an error.

POE will also check if the appropriate method specified in **/etc/poe.security** is configured on each node. For instance, when CTSec is enabled, it will ensure the CTSec libraries are installed.

Enabling Remote Direct Memory Access (RDMA)

Remote Direct Memory Access (RDMA) is a mechanism which allows large contiguous messages to be transferred while reducing the message transfer overhead. It is used with data striping and bulk data transfer.

RDMA may be used either implicitly or explicitly. To use implicit RDMA, **MP_USE_BULK_XFER** must be set to **YES**, which causes all MPI or LAPI messages that are larger than some threshold to use the bulk transfer or implicit RDMA path. If necessary, **MP_USE_BULK_XFER** can be overridden with the command line option, **-use_bulk_xfer**.

Explicit RDMA is only available to LAPI programs that use the rCxt resources requested by LoadLeveler. **MP_ RDMA_COUNT** is used to specify the number of user rCxt blocks. This number represents the total number of rCxt blocks required by the application program, and is determined by the number of remote handles that the program requires, divided by 128 and adding 2. **MP_ RDMA_COUNT** supports the specification of multiple values when multiple protocols are involved.

Note that the **MP_RDMA_COUNT**/**–rdma_count** option signifies the number of rCxt blocks the user has requested for the job, and LoadLeveler determines the actual number of rCxt blocks that will be allocated for the job. POE will use the value of **MP_RDMA_COUNT** to specify the number of rCxt blocks requested on the LoadLeveler MPI and/or LAPI network information when the job is submitted. The number of rCxt blocks will be the same for every window of the same protocol.

See the section on using RDMA in *Parallel Environment for AIX: Operation & Use Volume 1* for more detailed information.

Note that the values of **MP_RDMA_COUNT** and **MP_USE_BULK_XFER** are only significant for interactive jobs. For jobs that are submitted directly to LoadLeveler, the LoadLeveler keywords take precedence.

Before RDMA can be used with POE, the administrator and the end user need to perform the following tasks:

- Set the **SCHEDULE_BY_RESOURCES = RDMA** keyword, in the LoadLeveler configuration file. **SCHEDULE_BY_RESOURCES** specifies which consumable

resources are considered by the LoadLeveler schedulers. For more information, see *IBM LoadLeveler: Using and Administering*.

Note that you can confirm which nodes have been enabled by using the LoadLeveler command **llstatus -R**. In the following example output for the **llstatus -R** command, the f4rp02 node is not enabled for RDMA:

```
a [f4rp02]kgoin>llstatus -R

Machine                       Consumable Resource(Available, Total)
----------------------------- ---------------------------------------------
f3rp01.ppd.pok.ibm.com  RDMA(4,4)+<
f3rp02.ppd.pok.ibm.com
f4rp03.ppd.pok.ibm.com  suiteshare(16,16) RDMA(4,4)+<
f4rp04.ppd.pok.ibm.com  RDMA(4,4)+<

Resources with "+" appended to their names have the Total value reported from St
artd.
Resources with "<" appended to their names were created automatically.
a [f4rp02]kgoin>
```

In addition to the system administration tasks, the user must also enable bulk transfer, as follows:

If you are an interactive user, set the **MP_USE_BULK_XFER** environment variable to **yes**:

```
MP_USE_BULK_XFER=yes
```

The default setting for **MP_USE_BULK_XFER** is **no**. See *IBM Parallel Environment for AIX: Operation and Use, Volume 1* for more information about setting **MP_USE_BULK_XFER**.

If you are a batch JCF user, specify:

```
# @ bulkxfer = true
```

Appendix A. Syntax of commands for running installation and deinstallation scripts

PE provides two scripts for installing and deinstalling Parallel Environment. You can use **PEinstall** to install the PE file sets on IBM pSeries nodes. You can also use **PEdeinstall** to automatically remove all of the PE file sets that were previously installed.

Installation script: PEinstall

You can use the **PEinstall** script to install the PE file sets on IBM pSeries nodes using the Remote Shell **(rsh)**.

To run the **PEinstall** script, first set up a host list file of all nodes on which you want to install a particular file set. You must have **/usr** resident. The **PEinstall** script either mounts or copies the installation image to each node in the list, and then executes the proper **installp** command to install the product, including automatically accepting the product license.

The **PEinstall** script has one required parameter and two optional parameters. The syntax is:

PEinstall *image_name* [*host_list_file*] [**-copy | -mount**]

Where:

image_name
: is *required*. It specifies the name of the file that contains the **installp** image, of which the PE file set is a part.

host_list_file
: is *optional* and specifies the name of file containing the list of nodes on which you want to install the file set. The default file name is **host.list** in the current working directory. If a host list file cannot be found, the script exits with an error message.

You can specify either **-copy** or **-mount** to tell **PEinstall** to copy or mount the installation image to each node. The default is **-copy**.

Copying the installation image

Using the **-copy** option (or allowing it as the default) informs **PEinstall** to copy the named image to each node using **rcp**. You are prompted for the following information when you specify **-copy** (or defaulted):

- The installation image source directory. The default is **/usr/sys/inst.images**.
- The installation image destination directory which is used for all nodes in the node list. The default is **/usr/sys/inst.images**.

  Note: To have the image copied to different directories, invoke **PEinstall** for each different location or set of locations. Your **host.list** file should reflect only those nodes that you want to use with **-copy**.

The image is copied to the destination directory with the name specified as the *image_name* parameter. Be sure there is enough space in the destination directory file system for the image. Each image occupies approximately three megabytes.

Mounting the installation image

Specifying the **-mount** option informs **PEinstall** to mount the named image to each node using **rsh**. You are prompted for the following information when you specify **-mount**:

- The installation image source directory. The default is **/usr/sys/inst.images**.
- The remote node mount point directory. This is used for all nodes in the node list. The default is **/mnt**.

  To have the image mounted to different directories, invoke **PEinstall** for each different location or set of locations. Your **host.list** file should reflect only those nodes that you want to use with **-mount**.
- When mounting the image, **PEinstall** also asks if you want to create the remote mount directory. If your remote mount directory already exists, answer no to this prompt.

PEinstall issues a **mkdir** command for the directory name specified, followed by a **chmod 777**. To execute the **installp** remotely on a mounted image, the directory containing the image needs to have this permission.

To avoid creating the directory with world-writable permissions, do not use the **-mount** option of **PEinstall**.

Deinstallation script: PEdeinstall

When you install a PE file set, you do so first on a single node (or the control workstation). Then, you either copy or mount the installation image to the additional nodes in your system. When you remove a file set completely from your system, you do the opposite:

- First you remove the file set from the other nodes in your system, using the **PEdeinstall** script.
- Then you remove the file set from the initial installation node (or control workstation).

Removing an installation of a file set removes all files already installed for that file set. As a result, the **PEdeinstall** script will be removed from each node the **installp -u** command is run against. For this reason, you may want to consider copying **PEdeinstall** from **/usr/lpp/ppe.poe/bin** to another location before rejecting the installation of the file set. However, if you follow the previously mentioned sequence of removing a file set from the other nodes first, and then removing it from the initial node last, these scripts will remain available until the file set is removed from the initial node.

PEdeinstall issues the proper **installp** command using the Remote Shell (**rsh**).

The **PEdeinstall** script has the following syntax:

PEdeinstall *image_name* [*host_list_file*]

Where:

image_name
: is *required*, and specifies the file name of the **installp** image you want removed.

host_list_file
: is *optional* and specifies the name of the file containing the list of nodes

from which you want the image removed. The default file name is **host.list** in the current working directory. If this file cannot be found, the script exits with an error message.

For each node, **PEdeinstall** issues the following **installp** command:

installp -ugX *image_name*

This command removes both the user and root portions of all the products in the image specified.

If there is a problem removing an installed product on a node, an error message is listed and logged in a file named **PEnode.log** in the current working directory. The product removals continue for the remaining nodes.

Appendix B. Installation verification program summary

The POE Installation Verification Program (IVP) is an ideal way to determine if you have set up your system correctly before running your applications. It is located in the **/usr/lpp/ppe.poe/samples/ivp** directory, and is invoked by the **ivp.script** shell script. The IVP checks for the needed files and libraries and makes sure that everything is in order. It also issues messages when it finds something wrong.

You need the following in order to run the **ivp.script**:

- A nonroot userid that is properly authorized in **/etc/hosts.equiv** or the local **.rhosts** file.
- Access to a C compiler.

If the previous conditions are true, the IVP does the following:

1. Verifies that:
   - **poe**, **pmdv4**, **mpcc**, and **mpcc_r** are there, and are executable.
   - The **mpcc** and **mpcc_r** scripts are in the path.
   - The **/etc/services** file contains an entry for **pmv4** (the Partition Manager daemon).
   - The **/etc/inetd.conf** file contains an entry for **pmv4**, and that the daemon it points to is executable.
2. Creates a working directory in **/tmp/ivp***pid* to compile and run sample programs. Note that *pid* is the process id.
   - Compiles sample programs.
   - Creates a **host.list** file with local host names listed twice.
   - Runs sample programs using Internet Protocol (IP) on two tasks, using both threaded and non-threaded libraries.
   - Removes all files from **/tmp**, as well as the temporary working directory.
   - Checks for the dbx **bos.adt.debug** file set for **pdbx**.

For specific steps on verifying the installation, refer to “Step 4: Verify the POE installation” on page 23.

Appendix C. Using additional POE sample applications

PE provides POE sample applications for measuring the MPI point-to-point communication bandwidth between two tasks, broadcasting from task 0 to the all of the other nodes in the partition, and for using the MPI message passing library with user-created threads.

In order to be able to run these samples, POE must be fully installed and **rsct.lapi.rte** is also required.

Bandwidth measurement test sample

The purpose of this sample is to measure the MPI point-to-point communication bandwidth between two tasks. The sample code is in the directory called **/usr/lpp/ppe.poe/samples/poetest.bw**. This directory contains a test application called **bw.f**, which does a point-to-point bandwidth measurement test. The code needs only two nodes to run.

You should have the following files:

README.bw
: README file containing instructions on running the sample application, which is the same information presented here.

bw.f Sample application FORTRAN source file.

bw.run
: Script for compiling and executing the sample application.

makefile
: Makefile for creating the sample application.

The C and FORTRAN compilers must be available.

Verification steps

Follow these steps to verify your system:

1. Create the **bw** executable. Log in as a nonroot user and perform the following steps:

   ENTER
   : **cd /usr/lpp/ppe.poe/samples/poetest.bw** to switch to the appropriate directory. If you do not have write access to this directory, copy the needed files from here to a directory that is writable.

   ENTER
   : **make** to invoke the makefile, which compiles **bw.f** and creates the **bw** executable.

2. Create a file that lists the names of the nodes to be used for program execution.

   CREATE
   : a file named **host.list** and edit the file to add two entries, one per line. The entries should list the two nodes on which the executable is to run.

3. Run the **bw** executable. The **bw.run** script compiles **bw.f**, if not already compiled, and runs the **bw** executable from the current working directory.

   ENTER
   : **./bw.run** [*css_library*]

where:

css_library
is **us** for User Space message passing or **ip** for IP message passing.

4. Check your output.

VERIFY
your output by comparing it to the following output. The output should finish in about one minute, using the User Space message passing library. The execution time for IP is five minutes or longer. The actual response time depends on your LAN traffic.

```
Input: none

Output to terminal by this program: (Note that the order is
unpredictable.)
Hello from node 0
Hello from node 1
MEASURED BANDWIDTH = ....... MB/sec
```

Broadcast test sample

The purpose of this sample is to perform a broadcast from task 0 to the rest of the nodes running this program. You can find this sample in the directory called **/usr/lpp/ppe.poe/samples/poetest.cast**. This sample test code touches all nodes in the partition.

You should have the following files:

README.cast
README file containing instructions on running the sample application, which is the same information presented here.

bcast.f
Sample application FORTRAN source.

makefile
Makefile for compiling the sample application.

bcast.run.
Script for compiling and executing the sample application.

The FORTRAN compiler must be available.

Verification steps

Follow these steps to verify your system:

1. Create the **bcast** executable. Log in as a nonroot user and follow these steps:

ENTER
cd /usr/lpp/ppe.poe/samples/poetest.cast to switch to the appropriate directory. If you do not have write access to this directory, copy the needed files from here to a directory that is writable.

ENTER
make to invoke the makefile, which compiles **bcast.f**, to create the executable.

2. Create a file that lists the names of nodes to be used for program execution.

CREATE

a file named **host.list** and edit it by adding the names of the nodes on which to execute this program, with one entry per line.

3. Run the **bcast** executable. The **bcast.run** script compiles **bcast.f**, if not already compiled, and runs the **bcast** executable from the current working directory.

   ENTER

   ./bcast.run *ntasks* [*css_library*]

   where the required parameter is the following:

   ntasks the number of tasks (nodes) in the partition.

   Make sure that there are at least *ntasks* entries in the **host.list** file.

   and the optional parameter is:

   css_library

   us for User Space message passing (default) or **ip** for IP message passing.

4. Check your output.

   VERIFY

   your output by comparing it with the following output. The output should finish in about one minute if your system does not have more than 64 nodes. The actual response time depends on your LAN traffic. Note that the order of these lines is unpredictable.

```
Input: none

Output to terminal by this program:
Hello from node 0
Hello from node 1
...
Hello from node (p-1)
BROADCAST TEST COMPLETED SUCCESSFULLY
```

   If the test did not succeed, you should see the following message on the terminal:

```
BROADCAST TEST FAILED on node x (where x is some integer)
```

   For every node that did not pass the test, a line similar to the previous line appears.

MPI threads sample program

The purpose of this sample program is to illustrate the use of the MPI message passing library with user-created threads. You can find the sample program in the **/usr/lpp/ppe.poe/samples/threads** directory.

You should have the following files:

README.threads
: README file containing instructions on running the sample program.

threaded_ring.c
: Sample program source file for testing threaded MPI library with user threads.

makefile
: Makefile for compiling the threaded sample program.

threads.run
Script for compiling and executing the user threads sample program, **threaded_ring**.

The C compiler must be available.

Verification steps

Follow these steps to run the sample threads application on your system:

1. Create the executables by logging in as a nonroot user, and doing the following:

   ENTER
   cd /usr/lpp/ppe.poe/samples/threads to switch to the appropriate directory. If you do not have write access to this directory, copy the needed files from here to a directory that is writable.

   ENTER
   make invoke the makefile, which compiles both source programs to create the executable, **threaded_ring**.

2. Create a file that lists the names of nodes to be used for program execution.

   CREATE
   a file named **host.list** and edit it by adding the names of the nodes on which to execute this program, with one entry per line.

3. Run the **threaded_ring** executable. The **threads.run** script compiles **threaded_ring.c**,if not already compiled, and runs the **threaded_ring** executable from the current working directory.

   ENTER
   threads.run [*css_library*]

   where:

   css_library
   specifies the library to use. Type **ip** to use the UDP/IP library. Type **us** to use the User Space library. These names are case-sensitive. User Space is the default.

   The program should issue only the message ″TEST COMPLETE″ from task 0.

   ```
   0:TEST COMPLETE
   ```

LAPI sample programs

Several sample programs exist that illustrate the use of the Low-level Applications Programming Interface (LAPI). Refer to the *LAPI Programming Guide* for specific details.

Appendix D. Parallel Environment port usage

Table 19 provides port information for the Parallel Environment.

Table 19. PE port usage

Service name	Port number	Protocol	Source port range	Required or optional
pmv4	6127	TCP	n/a	Required
dpcl	7895	TCP	n/a	Required

The service names in Table 19 are defined as follows:

pmv4 Partition Manager daemon inetd service

dpcl Dynamic Probe Class Library (DPCL). DPCL is no longer a part of the IBM PE for AIX licensed program, but is still shipped with PE for convenience. Instead, DPCL is now available as an open source offering that supports PE. For more information on DPCL open source project go to: **http://dpcl.sourceforge.net**.

When POE is installed, an entry is added in **/etc/services** and in **/etc/inetd.conf** to describe the partition manager daemon. The entry that is added to **/etc/services** defines a port number used by pmdv4 to communicate with the POE process on the home node. PE attempts to use port number 6127. However if this port is already in use then PE will try to use port 6128 and so forth. As a result, the port number selected may not be the same for all nodes of a cluster. In the event that some of the nodes cannot communicate with other nodes, check the **/etc/services** file to make sure that all nodes use the same port number.

Appendix E. Accessibility features for PE

Accessibility features help a user who has a physical disability, such as restricted mobility or limited vision, to use information technology products successfully.

Accessibility features

The following list includes the major accessibility features in IBM Parallel Environment. These features support:

- Keyboard-only operation.
- Interfaces that are commonly used by screen readers.
- Keys that are tactilely discernible and do not activate just by touching them.
- Industry-standard devices for ports and connectors.
- The attachment of alternative input and output devices.

Note: The IBM eServer Cluster Information Center and its related publications are accessibility-enabled for the IBM Home Page Reader. You can operate all features using the keyboard instead of the mouse.

Keyboard navigation

This product uses standard Microsoft® Windows® navigation keys.

IBM and accessibility

See the *IBM Accessibility Center* at **http://www.ibm.com/able** for more information about the commitment that IBM has to accessibility.

Notices

This information was developed for products and services offered in the U.S.A.

IBM may not offer the products, services, or features discussed in this document in other countries. Consult your local IBM representative for information on the products and services currently available in your area. Any reference to an IBM product, program, or service is not intended to state or imply that only that IBM product, program, or service may be used. Any functionally equivalent product, program, or service that does not infringe any IBM intellectual property right may be used instead. However, it is the user's responsibility to evaluate and verify the operation of any non-IBM product, program, or service.

IBM may have patents or pending patent applications covering subject matter described in this document. The furnishing of this document does not grant you any license to these patents. You can send license inquiries, in writing, to:

IBM Director of Licensing
IBM Corporation
North Castle Drive
Armonk, NY 10504-1785
U.S.A.

For license inquiries regarding double-byte (DBCS) information, contact the IBM Intellectual Property Department in your country or send inquiries, in writing, to:

IBM World Trade Asia Corporation
Licensing
2-31 Roppongi 3-chome, Minato-ku
Tokyo 106-0032, Japan

The following paragraph does not apply to the United Kingdom or any other country where such provisions are inconsistent with local law:

INTERNATIONAL BUSINESS MACHINES CORPORATION PROVIDES THIS PUBLICATION "AS IS" WITHOUT WARRANTY OF ANY KIND, EITHER EXPRESS OR IMPLIED, INCLUDING, BUT NOT LIMITED TO, THE IMPLIED WARRANTIES OF NON-INFRINGEMENT, MERCHANTABILITY OR FITNESS FOR A PARTICULAR PURPOSE. Some states do not allow disclaimer of express or implied warranties in certain transactions, therefore, this statement may not apply to you.

This information could include technical inaccuracies or typographical errors. Changes are periodically made to the information herein; these changes will be incorporated in new editions of the publication. IBM may make improvements and/or changes in the product(s) and/or the program(s) described in this publication at any time without notice.

Any references in this information to non-IBM Web sites are provided for convenience only and do not in any manner serve as an endorsement of those Web sites. The materials at those Web sites are not part of the materials for this IBM product and use of those Web sites is at your own risk.

IBM may use or distribute any of the information you supply in any way it believes appropriate without incurring any obligation to you.

Licensees of this program who wish to have information about it for the purpose of enabling: (i) the exchange of information between independently created programs and other programs (including this one) and (ii) the mutual use of the information which has been exchanged, should contact:

IBM Corporation
Department LJEB/P905
2455 South Road
Poughkeepsie, NY 12601-5400
U.S.A

Such information may be available, subject to appropriate terms and conditions, including in some cases, payment of a fee.

The licensed program described in this document and all licensed material available for it are provided by IBM under terms of the IBM Customer Agreement, IBM International Program License Agreement or any equivalent agreement between us.

Any performance data contained herein was determined in a controlled environment. Therefore, the results obtained in other operating environments may vary significantly. Some measurements may have been made on development-level systems and there is no guarantee that these measurements will be the same on generally available systems. Furthermore, some measurement may have been estimated through extrapolation. Actual results may vary. Users of this document should verify the applicable data for their specific environment.

Information concerning non-IBM products was obtained from the suppliers of those products, their published announcements or other publicly available sources. IBM has not tested those products and cannot confirm the accuracy of performance, compatibility or any other claims related to non-IBM products. Questions on the capabilities of non-IBM products should be addressed to the suppliers of those products.

All statements regarding IBM's future direction or intent are subject to change or withdrawal without notice, and represent goals and objectives only.

This information contains examples of data and reports used in daily business operations. To illustrate them as completely as possible, the examples include the names of individuals, companies, brands, and products. All of these names are fictitious and any similarity to the names and addresses used by an actual business enterprise is entirely coincidental.

COPYRIGHT LICENSE:

This information contains sample application programs in source language, which illustrates programming techniques on various operating platforms. You may copy, modify, and distribute these sample programs in any form without payment to IBM, for the purposes of developing, using, marketing or distributing application programs conforming to the application programming interface for the operating platform for which the sample programs are written. These examples have not been thoroughly tested under all conditions. IBM, therefore, cannot guarantee or imply reliability, serviceability, or function of these programs.

Each copy or any portion of these sample programs or any derivative work, must include a copyright notice as follows:

© (your company name) (year). Portions of this code are derived from IBM Corp. Sample Programs. © Copyright IBM Corp. _enter the year or years_. All rights reserved.

All implemented function in the PE MPI product is designed to comply with the requirements of the Message Passing Interface Forum, MPI: A Message-Passing Interface Standard. The standard is documented in two volumes, Version 1.1, University of Tennessee, Knoxville, Tennessee, June 6, 1995 and *MPI-2: Extensions to the Message-Passing Interface*, University of Tennessee, Knoxville, Tennessee, July 18, 1997. The second volume includes a section identified as MPI 1.2 with clarifications and limited enhancements to MPI 1.1. It also contains the extensions identified as MPI 2.0. The three sections, MPI 1.1, MPI 1.2 and MPI 2.0 taken together constitute the current standard for MPI.

PE MPI provides support for all of MPI 1.1 and MPI 1.2. PE MPI also provides support for all of the MPI 2.0 Enhancements, except the contents of the chapter titled *Process Creation and Management*.

If you believe that PE MPI does not comply with the MPI standard for the portions that are implemented, contact IBM Service.

Trademarks

The following are trademarks of International Business Machines Corporation in the United States, other countries, or both:

AFS
AIX
AIX 5L
BladeCenter®
DFS
eServer
IBM
IBMLink™
LoadLeveler
OpenPower™
POWER™
POWER3
POWER4
pSeries
RS/6000
SP
System p5
System x
Tivoli
VisualAge

UNIX is a registered trademark of The Open Group in the United States and other countries.

Linux is a trademark of Linus Torvalds in the United States, other countries, or both.

Intel®, Intel logo, Intel Inside®, Intel Inside logo, Intel Centrino™, Intel Centrino logo, Celeron®, Intel Xeon™, Intel SpeedStep®, Itanium®, and Pentium® are trademarks or registered trademarks of Intel Corporation or its subsidiaries in the United States and other countries.

Microsoft, Windows, Windows NT, and the Windows logo are trademarks of Microsoft Corporation in the United States, other countries, or both.

Java and all Java-based trademarks are trademarks of Sun Microsystems, Inc. in the United States, other countries, or both.

Other company, product, and service names may be trademarks or service marks of others.

Acknowledgments

The PE Benchmarker product includes software developed by the Apache Software Foundation, *http://www.apache.org*.

Glossary

A

AFS. Andrew File System.

address. A value, possibly a character or group of characters that identifies a register, a device, a particular part of storage, or some other data source or destination.

AIX. Abbreviation for Advanced Interactive Executive, IBM's licensed version of the UNIX operating system. AIX is particularly suited to support technical computing applications, including high-function graphics and floating-point computations.

API. Application programming interface.

application. The use to which a data processing system is put; for example, a payroll application, an airline reservation application.

argument. A parameter passed between a calling program and a called program or subprogram.

attribute. A named property of an entity.

Authentication. The process of validating the identity of a user or server.

Authorization. The process of obtaining permission to perform specific actions.

B

bandwidth. For a specific amount of time, the amount of data that can be transmitted. Bandwidth is expressed in bits or bytes per second (bps) for digital devices, and in cycles per second (Hz) for analog devices.

blocking operation. An operation that does not complete until the operation either succeeds or fails. For example, a blocking receive will not return until a message is received or until the channel is closed and no further messages can be received.

breakpoint. A place in a program, specified by a command or a condition, where the system halts execution and gives control to the workstation user or to a specified program.

broadcast operation. A communication operation where one processor sends (or broadcasts) a message to all other processors.

buffer. A portion of storage used to hold input or output data temporarily.

C

C. A general-purpose programming language. It was formalized by Uniforum in 1983 and the ANSI standards committee for the C language in 1984.

C++. A general-purpose programming language that is based on the C language. C++ includes extensions that support an object-oriented programming paradigm. Extensions include:
- strong typing
- data abstraction and encapsulation
- polymorphism through function overloading and templates
- class inheritance.

chaotic relaxation. An iterative relaxation method that uses a combination of the Gauss-Seidel and Jacobi-Seidel methods. The array of discrete values is divided into subregions that can be operated on in parallel. The subregion boundaries are calculated using the Jacobi-Seidel method, while the subregion interiors are calculated using the Gauss-Seidel method. See also *Gauss-Seidel*.

client. A function that requests services from a server and makes them available to the user.

cluster. A group of processors interconnected through a high-speed network that can be used for high-performance computing.

Cluster 1600. See IBM eServer Cluster 1600.

collective communication. A communication operation that involves more than two processes or tasks. Broadcasts, reductions, and the **MPI_Allreduce** subroutine are all examples of collective communication operations. All tasks in a communicator must participate.

command alias. When using the PE command-line debugger **pdbx**, you can create abbreviations for existing commands using the **pdbx alias** command. These abbreviations are known as *command aliases.*

communicator. An MPI object that describes the communication context and an associated group of processes.

compile. To translate a source program into an executable program.

condition. One of a set of specified values that a data item can assume.

core dump. A process by which the current state of a program is preserved in a file. Core dumps are usually associated with programs that have encountered an unexpected, system-detected fault, such as a

Segmentation Fault or a severe user error. The current program state is needed for the programmer to diagnose and correct the problem.

core file. A file that preserves the state of a program, usually just before a program is terminated for an unexpected error. See also *core dump.*

current context. When using the **pdbx** debugger, control of the parallel program and the display of its data can be limited to a subset of the tasks belonging to that program. This subset of tasks is called the *current context.* You can set the current context to be a single task, multiple tasks, or all the tasks in the program.

D

data decomposition. A method of breaking up (or decomposing) a program into smaller parts to exploit parallelism. One divides the program by dividing the data (usually arrays) into smaller parts and operating on each part independently.

data parallelism. Refers to situations where parallel tasks perform the same computation on different sets of data.

dbx. A symbolic command-line debugger that is often provided with UNIX systems. The PE command-line debugger **pdbx** is based on the **dbx** debugger.

debugger. A debugger provides an environment in which you can manually control the execution of a program. It also provides the ability to display the program's data and operation.

distributed shell (dsh). An Parallel System Support Programs command that lets you issue commands to a group of hosts in parallel. See *IBM Parallel System Support Programs for AIX: Command and Technical Reference* for details.

domain name. The hierarchical identification of a host system (in a network), consisting of human-readable labels, separated by decimal points.

DPCL target application. The executable program that is instrumented by a Dynamic Probe Class Library (DPCL) analysis tool. It is the process (or processes) into which the DPCL analysis tool inserts probes. A target application could be a serial or parallel program. Furthermore, if the target application is a parallel program, it could follow either the SPMD or the MPMD model, and may be designed for either a message-passing or a shared-memory system.

E

environment variable. (1) A variable that describes the operating environment of the process. Common environment variables describe the home directory, command search path, and the current time zone. (2) A variable that is included in the current software environment and is therefore available to any called program that requests it.

Ethernet. A baseband local area network (LAN) that allows multiple stations to access the transmission medium at will without prior coordination, avoids contention by using carrier sense and deference, and resolves contention by using collision detection and delayed retransmission. Ethernet uses carrier sense multiple access with collision detection (CSMA/CD).

event. An occurrence of significance to a task — the completion of an asynchronous operation such as an input/output operation, for example.

executable. A program that has been link-edited and therefore can be run in a processor.

execution. To perform the actions specified by a program or a portion of a program.

expression. In programming languages, a language construct for computing a value from one or more operands.

F

fairness. A policy in which tasks, threads, or processes must be allowed eventual access to a resource for which they are competing. For example, if multiple threads are simultaneously seeking a lock, no set of circumstances can cause any thread to wait indefinitely for access to the lock.

Fiber Distributed Data Interface (FDDI). An American National Standards Institute (ANSI) standard for a local area network (LAN) using optical fiber cables. An FDDI LAN can be up to 100 kilometers (62 miles) long, and can include up to 500 system units. There can be up to 2 kilometers (1.24 miles) between system units and concentrators.

file system. The collection of files and file management structures on a physical or logical mass storage device, such as a diskette or minidisk.

fileset. (1) An individually-installable option or update. Options provide specific functions. Updates correct an error in, or enhance, a previously installed program. (2) One or more separately-installable, logically-grouped units in an installation package. See also *licensed program* and *package.*

foreign host. See *remote host.*

FORTRAN. One of the oldest of the modern programming languages, and the most popular language for scientific and engineering computations. Its name is a contraction of *FORmula TRANslation.* The two most common FORTRAN versions are FORTRAN

77, originally standardized in 1978, and FORTRAN 90. FORTRAN 77 is a proper subset of FORTRAN 90.

function cycle. A chain of calls in which the first caller is also the last to be called. A function that calls itself recursively is not considered a function cycle.

functional decomposition. A method of dividing the work in a program to exploit parallelism. The program is divided into independent pieces of functionality, which are distributed to independent processors. This method is in contrast to data decomposition, which distributes the same work over different data to independent processors.

functional parallelism. Refers to situations where parallel tasks specialize in particular work.

G

Gauss-Seidel. An iterative relaxation method for solving Laplace's equation. It calculates the general solution by finding particular solutions to a set of discrete points distributed throughout the area in question. The values of the individual points are obtained by averaging the values of nearby points. Gauss-Seidel differs from Jacobi-Seidel in that, for the *i+1*st iteration, Jacobi-Seidel uses only values calculated in the *i*th iteration. Gauss-Seidel uses a mixture of values calculated in the *i*th and *i+1*st iterations.

global max. The maximum value across all processors for a given variable. It is global in the sense that it is global to the available processors.

global variable. A variable defined in one portion of a computer program and used in at least one other portion of the computer program.

gprof. A UNIX command that produces an execution profile of C, COBOL, FORTRAN, or Pascal programs. The execution profile is in a textual and tabular format. It is useful for identifying which routines use the most CPU time. See the man page on **gprof**.

graphical user interface (GUI). A type of computer interface consisting of a visual metaphor of a real-world scene, often of a desktop. Within that scene are icons, which represent actual objects, that the user can access and manipulate with a pointing device.

GUI. Graphical user interface.

H

high performance switch. The high-performance message-passing network that connects all processor nodes together.

hook. A **pdbx** command that lets you re-establish control over all tasks in the current context that were previously unhooked with this command.

home node. The node from which an application developer compiles and runs his program. The home node can be any workstation on the LAN.

host. A computer connected to a network that provides an access method to that network. A host provides end-user services.

host list file. A file that contains a list of host names, and possibly other information, that was defined by the application that reads it.

host name. The name used to uniquely identify any computer on a network.

hot spot. A memory location or synchronization resource for which multiple processors compete excessively. This competition can cause a disproportionately large performance degradation when one processor that seeks the resource blocks, preventing many other processors from having it, thereby forcing them to become idle.

I

IBM eServer Cluster 1600. An IBM eServer Cluster 1600 is any CSM-managed cluster comprised of POWER microprocessor based systems (including RS/6000 SMPs, RS/6000 SP nodes, and pSeries SMPs).

IBM Parallel Environment (PE) for AIX. A licensed program that provides an execution and development environment for parallel C, C++, and FORTRAN programs. It also includes tools for debugging, profiling, and tuning parallel programs.

installation image. A file or collection of files that are required in order to install a software product on system nodes. These files are in a form that allows them to be installed or removed with the AIX **installp** command. See also *fileset*, *licensed program*, and *package.*

Internet. The collection of worldwide networks and gateways that function as a single, cooperative virtual network.

Internet Protocol (IP). The IP protocol lies beneath the UDP protocol, which provides packet delivery between user processes and the TCP protocol, which provides reliable message delivery between user processes.

IP. Internet Protocol.

J

Jacobi-Seidel. See *Gauss-Seidel.*

K

Kerberos. A publicly available security and authentication product that works with the Parallel System Support Programs software to authenticate the execution of remote commands.

kernel. The core portion of the UNIX operating system that controls the resources of the CPU and allocates them to the users. The kernel is memory-resident, is said to run in *kernel mode* (in other words, at higher execution priority level than *user mode*), and is protected from user tampering by the hardware.

L

Laplace's equation. A homogeneous partial differential equation used to describe heat transfer, electric fields, and many other applications.

latency. The time interval between the initiation of a send by an origin task and the completion of the matching receive by the target task. More generally, latency is the time between a task initiating data transfer and the time that transfer is recognized as complete at the data destination.

licensed program. A collection of software packages sold as a product that customers pay for to license. A licensed program can consist of packages and file sets a customer would install. These packages and file sets bear a copyright and are offered under the terms and conditions of a licensing agreement. See also *fileset* and *package.*

lightweight corefiles. An alternative to standard AIX corefiles. Corefiles produced in the *Standardized Lightweight Corefile Format* provide simple process stack traces (listings of function calls that led to the error) and consume fewer system resources than traditional corefiles.

LoadLeveler. A job management system that works with POE to let users run jobs and match processing needs with system resources, in order to make better use of the system.

local variable. A variable that is defined and used only in one specified portion of a computer program.

loop unrolling. A program transformation that makes multiple copies of the body of a loop, also placing the copies within the body of the loop. The loop trip count and index are adjusted appropriately so the new loop computes the same values as the original. This transformation makes it possible for a compiler to take additional advantage of instruction pipelining, data cache effects, and software pipelining.

See also *optimization.*

M

management domain . A set of nodes configured for manageability by the Clusters Systems Management (CSM) product. Such a domain has a management server that is used to administer a number of managed nodes. Only management servers have knowledge of the whole domain. Managed nodes only know about the servers managing them; they know nothing of each other. Contrast with *peer domain*.

menu. A list of options displayed to the user by a data processing system, from which the user can select an action to be initiated.

message catalog. A file created from a message source file that contains application error and other messages, which can later be translated into other languages without having to recompile the application source code.

message passing. Refers to the process by which parallel tasks explicitly exchange program data.

Message Passing Interface (MPI). A standardized API for implementing the message-passing model.

MIMD. Multiple instruction stream, multiple data stream.

Multiple instruction stream, multiple data stream (MIMD). A parallel programming model in which different processors perform different instructions on different sets of data.

MPMD. Multiple program, multiple data.

Multiple program, multiple data (MPMD). A parallel programming model in which different, but related, programs are run on different sets of data.

MPI. Message Passing Interface.

N

network. An interconnected group of nodes, lines, and terminals. A network provides the ability to transmit data to and receive data from other systems and users.

Network Information Services. A set of network services (for example, a distributed service for retrieving information about the users, groups, network addresses, and gateways in a network) that resolve naming and addressing differences among computers in a network.

NIS. See *Network Information Services*.

node. (1) In a network, the point where one or more functional units interconnect transmission lines. A computer location defined in a network. (2) A single location or workstation in a network. Usually a physical entity, such as a processor.

node ID. A string of unique characters that identifies the node on a network.

nonblocking operation. An operation, such as sending or receiving a message, that returns immediately whether or not the operation was completed. For example, a nonblocking receive will not wait until a message arrives. By contrast, a blocking receive will wait. A nonblocking receive must be completed by a later test or wait.

O

object code. The result of translating a computer program to a relocatable, low-level form. Object code contains machine instructions, but symbol names (such as array, scalar, and procedure names), are not yet given a location in memory. Contrast with *source code.*

optimization. A widely-used (though not strictly accurate) term for program performance improvement, especially for performance improvement done by a compiler or other program translation software. An optimizing compiler is one that performs extensive code transformations in order to obtain an executable that runs faster but gives the same answer as the original. Such code transformations, however, can make code debugging and performance analysis very difficult because complex code transformations obscure the correspondence between compiled and original source code.

option flag. Arguments or any other additional information that a user specifies with a program name. Also referred to as *parameters* or *command-line options.*

P

package. A number of file sets that have been collected into a single installable image of licensed programs. Multiple file sets can be bundled together for installing groups of software together. See also *fileset* and *licensed program.*

parallelism. The degree to which parts of a program may be concurrently executed.

parallelize. To convert a serial program for parallel execution.

parallel operating environment (POE). An execution environment that smooths the differences between serial and parallel execution. It lets you submit and manage parallel jobs. It is abbreviated and commonly known as POE.

parameter. (1) In FORTRAN, a symbol that is given a constant value for a specified application. (2) An item in a menu for which the operator specifies a value or for which the system provides a value when the menu is interpreted. (3) A name in a procedure that is used to refer to an argument that is passed to the procedure. (4) A particular piece of information that a system or application program needs to process a request.

partition. (1) A fixed-size division of storage. (2) A logical collection of nodes to be viewed as one system or domain. System partitioning is a method of organizing the system into groups of nodes for testing or running different levels of software of product environments.

Partition Manager. The component of the parallel operating environment (POE) that allocates nodes, sets up the execution environment for remote tasks, and manages distribution or collection of standard input (STDIN), standard output (STDOUT), and standard error (STDERR).

pdbx. The parallel, symbolic command-line debugging facility of PE. **pdbx** is based on the **dbx** debugger and has a similar interface.

PE. The Parallel Environment for AIX licensed program.

peer domain. A set of nodes configured for high availability by the RSCT configuration manager. Such a domain has no distinguished or master node. All nodes are aware of all other nodes, and administrative commands can be issued from any node in the domain. All nodes also have a consistent view of the domain membership. Contrast with *management domain*.

performance monitor. A utility that displays how effectively a system is being used by programs.

PID. Process identifier.

POE. parallel operating environment.

pool. Groups of nodes on a system that are known to LoadLeveler, and are identified by a pool name or number.

point-to-point communication. A communication operation that involves exactly two processes or tasks. One process initiates the communication through a *send* operation. The partner process issues a *receive* operation to accept the data being sent.

procedure. (1) In a programming language, a block, with or without formal parameters, whose execution is invoked by means of a procedure call. (2) A set of related control statements that cause one or more programs to be performed.

process. A program or command that is actually running the computer. It consists of a loaded version of the executable file, its data, its stack, and its kernel data structures that represent the process's state within a multitasking environment. The executable file contains the machine instructions (and any calls to shared

objects) that will be executed by the hardware. A process can contain multiple threads of execution.

The process is created with a **fork()** system call and ends using an **exit()** system call. Between **fork** and **exit**, the process is known to the system by a unique process identifier (PID).

Each process has its own virtual memory space and cannot access another process's memory directly. Communication methods across processes include pipes, sockets, shared memory, and message passing.

prof. A utility that produces an execution profile of an application or program. It is useful to identify which routines use the most CPU time. See the man page for **prof**.

profiling. The act of determining how much CPU time is used by each function or subroutine in a program. The histogram or table produced is called the execution profile.

pthread. A thread that conforms to the POSIX Threads Programming Model.

R

reduced instruction-set computer. A computer that uses a small, simplified set of frequently-used instructions for rapid execution.

reduction operation. An operation, usually mathematical, that reduces a collection of data by one or more dimensions. For example, the arithmetic SUM operation is a reduction operation that reduces an array to a scalar value. Other reduction operations include MAXVAL and MINVAL.

Reliable Scalable Cluster Technology. A set of software components that together provide a comprehensive clustering environment for AIX. RSCT is the infrastructure used by a variety of IBM products to provide clusters with improved system availability, scalability, and ease of use.

remote host. Any host on a network except the one where a particular operator is working.

remote shell (rsh). A command that lets you issue commands on a remote host.

RISC. See *reduced instruction-set computer*.

RSCT. See *Reliable Scalable Cluster Technology*.

RSCT peer domain. See *peer domain*.

S

shell script. A sequence of commands that are to be executed by a shell interpreter such as the Bourne shell (**sh**), the C shell (**csh**), or the Korn shell (**ksh**). Script commands are stored in a file in the same format as if they were typed at a terminal.

segmentation fault. A system-detected error, usually caused by referencing an non-valid memory address.

server. A functional unit that provides shared services to workstations over a network — a file server, a print server, or a mail server, for example.

signal handling. In the context of a message passing library (such as MPI), there is a need for asynchronous operations to manage packet flow and data delivery while the application is doing computation. This asynchronous activity can be carried out either by a signal handler or by a service thread. The early IBM message passing libraries used a signal handler and the more recent libraries use service threads. The older libraries are often referred to as the *signal handling* versions.

Single program, multiple data (SPMD). A parallel programming model in which different processors execute the same program on different sets of data.

source code. The input to a compiler or assembler, written in a source language. Contrast with *object code.*

source line. A line of source code.

SPMD. Single program, multiple data.

standard error (STDERR). An output file intended to be used for error messages for C programs.

standard input (STDIN). The primary source of data entered into a command. Standard input comes from the keyboard unless redirection or piping is used, in which case standard input can be from a file or the output from another command.

standard output (STDOUT). The primary destination of data produced by a command. Standard output goes to the display unless redirection or piping is used, in which case standard output can go to a file or to another command.

STDERR. Standard error.

STDIN. Standard input.

STDOUT. Standard output.

stencil. A pattern of memory references used for averaging. A 4-point stencil in two dimensions for a given array cell, x(i,j), uses the four adjacent cells, x(i-1,j), x(i+1,j), x(i,j-1), and x(i,j+1).

subroutine. (1) A sequence of instructions whose execution is invoked by a call. (2) A sequenced set of instructions or statements that can be used in one or more computer programs and at one or more points in a

computer program. (3) A group of instructions that can be part of another routine or can be called by another program or routine.

synchronization. The action of forcing certain points in the execution sequences of two or more asynchronous procedures to coincide in time.

system administrator. (1) The person at a computer installation who designs, controls, and manages the use of the computer system. (2) The person who is responsible for setting up, modifying, and maintaining the Parallel Environment.

T

target application. See *DPCL target application*.

task. A unit of computation analogous to a process. In a parallel job, there are two or more concurrent tasks working together through message passing. Though it is common to allocate one task per processor, the terms *task* and *processor* are not interchangeable.

thread. A single, separately dispatchable, unit of execution. There can be one or more threads in a process, and each thread is executed by the operating system concurrently.

TPD. Third party debugger.

tracing. In PE, the collection of information about the execution of the program. This information is accumulated into a trace file that can later be examined.

tracepoint. Tracepoints are places in the program that, when reached during execution, cause the debugger to print information about the state of the program.

trace record. In PE, a collection of information about a specific event that occurred during the execution of your program. For example, a trace record is created for each send and receive operation that occurs in your program (this is optional and might not be appropriate). These records are then accumulated into a trace file that can later be examined.

U

unrolling loops. See *loop unrolling.*

user. (1) A person who requires the services of a computing system. (2) Any person or any thing that can issue or receive commands and message to or from the information processing system.

User Space. A version of the message passing library that is optimized for direct access to the high performance switch. User Space maximizes performance by passing up all kernel involvement in sending or receiving a message.

utility program. A computer program in general support of computer processes; for example, a diagnostic program, a trace program, a sort program.

utility routine. A routine in general support of the processes of a computer; for example, an input routine.

V

variable. (1) In programming languages, a named object that may take different values, one at a time. The values of a variable are usually restricted to one data type. (2) A quantity that can assume any of a given set of values. (3) A name used to represent a data item whose value can be changed while the program is running. (4) A name used to represent data whose value can be changed, while the program is running, by referring to the name of the variable.

X

X Window System. The UNIX industry's graphics windowing standard that provides simultaneous views of several executing programs or processes on high resolution graphics displays.

Index

A

abbreviated names x
accessibility 59
  keyboard 59
  shortcut keys 59
acknowledgments 64
acronyms for product names x
AFS installation 30
API subroutine libraries, described 1

C

conventions x

D

disability 59
disk space requirements
  PE feature 6
  pedb product option 6
  pedocs product option 6
  poe product option 6
  ppe.perf product option 6
  ppe.pvts product option 6
  Xprofiler product option 6

F

file formats
  /etc/poe.priority dispatching adjustment parameters 39

H

hardware requirements
  MPI/MPL libraries 3
  PE feature 3

I

installation image 13, 47
  mounting 47
installation procedure
  PE feature 15, 29
installation verification program (IVP) 23
Installation Verification Program (IVP) 51

L

limitations
  PE component 7
  PE feature 7
  related software 7
LookAt message retrieval tool xii

M

message catalog customization 29
message retrieval tool, LookAt xii
migrating 25
mounting the installation image 47

P

parallel operating environment (POE), described 1
pdbx debugger, described 1
pdbx Performance Collection Tool, described 1
pdbx Profile Visualization Tool, described 2
PE component
  limitations 7
PE documentation, described 2
PE feature
  disk space requirements 6
  hardware requirements 3
  how installation alters system 33
  installation planning 3
  installation procedure 15, 29
  installation requirements 3
  limitations 7
  software requirements 3
pedb option
  components described 3
pedocs file set
  how installation alters system 37
pedocs option
  components described 3
pedocs product option
  disk space requirements 6
planning to install PE 3
Planning to install PE
  file systems 8
poe option
  components described 3
poe product option
  disk space requirements 6
port numbers
  POE 57
ppe.perf file set
  how installation alters system 36
ppe.poe file set
  how installation alters system 33
ppe.pvt file set
  how installation alters system 36

R

README file 15
requirements for installation 3

S

safe coding practices 51

shortcut keys
  keyboard 59
software requirements
  PE feature 3
system administrator 7
system partitioning 7

T

trademarks 63

U

user authorization 9
user IDs 9

X

Xprofiler product option
  disk space requirements 6

Reader's Comments– We'd like to hear from you

IBM Parallel Environment for AIX 5L
Installation
Version 4 Release 3.0

Publication No. GA22-7943-05

We appreciate your comments about this publication. Please comment on specific errors or omissions, accuracy, organization, subject matter, or completeness of this book. The comments you send should pertain to only the information in this manual or product and the way in which the information is presented.

For technical questions and information about products and prices, please contact your IBM branch office, your IBM business partner, or your authorized remarketer.

When you send comments to IBM, you grant IBM a nonexclusive right to use or distribute your comments in any way it believes appropriate without incurring any obligation to you. IBM or any other organizations will only use the personal information that you supply to contact you about the issues that you state on this form.

Comments:

Thank you for your support.

Submit your comments using one of these channels:

- Send your comments to the address on the reverse side of this form.
- Send your comments via e-mail to: mhvrcfs@us.ibm.com

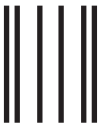
If you would like a response from IBM, please fill in the following information:

Name	Address
Company or Organization	
Phone No.	E-mail address

Readers' Comments — We'd Like to Hear from You
GA22-7943-05

Cut or Fold
Along Line

Fold and Tape | **Please do not staple** | Fold and Tape



NO POSTAGE
NECESSARY
IF MAILED IN THE
UNITED STATES

BUSINESS REPLY MAIL

FIRST-CLASS MAIL PERMIT NO. 40 ARMONK, NEW YORK

POSTAGE WILL BE PAID BY ADDRESSEE

IBM Corporation
Department 55JA, Mail Station P384
2455 South Road
Poughkeepsie NY
12601-5400

Fold and Tape | **Please do not staple** | Fold and Tape

Cut or Fold
Along Line

Program Number: 5765-F83

GA22-7943-05