IBM Parallel Environment for AIX

Installation

Version 2 Release 4

GC28-1981-02

IBM Parallel Environment for AIX

Installation

Version 2 Release 4

GC28-1981-02

Note:

Before using this information and the product it supports, be sure to read the general information under “Notices” on page vii.

Third Edition (October 1998)

This edition applies to Version 2 Release 4 of IBM Parallel Environment for AIX (5765-543), and to all subsequent releases and modifications until otherwise indicated in new editions.

Order publications through your IBM representative or the IBM branch office serving your locality. Publications are not stocked at the address given below.

IBM welcomes your comments. A form for your comments appears at the back of this publication. If the form has been removed, address your comments to:

IBM Corporation, Department 55JA, Mail Station P384
522 South Road
Poughkeepsie, NY 12601-5400
United States of America

FAX (United States and Canada: 1+914+432-9405
FAX (Other Countries) Your International Access Code)+1+914+432-9405
IBMLink (United States customers only): IBMUSM10(MHVRCFS)
IBM Mail Exchange: USIB6TC9 at IBMMAIL
Internet e-mail: mhvrcfs@us.ibm.com

World Wide Web: http://www.rs6000.ibm.com (select Parallel Computing)

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

Notices

References in this publication to IBM products, programs, or services do not imply that IBM intends to make these available in all countries in which IBM operates. Any reference to an IBM product, program, or service is not intended to state or imply that only IBM's product, program, or service may be used. Any functionally equivalent product, program, or service that does not infringe any of IBM's intellectual property rights may be used instead of the IBM product, program, or service. Evaluation and verification of operation in conjunction with other products, except those expressly designated by IBM, are the user's responsibility.

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

IBM Director of Licensing
IBM Corporation
500 Columbus Avenue
Thornwood, NY 10594
USA

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
Mail Station P300
522 South Road
Poughkeepsie, NY 12601-5400
USA
Attention: Information Request

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

Trademarks

The following terms are trademarks of the International Business Machines Corporation in the United States and/or other countries:

AIX
ESCON
IBM
LoadLeveler
Micro Channel
RS/6000
RS/6000 SP
VisualAge

Adobe, Acrobat, Acrobat Reader, and PostScript are trademarks of Adobe Systems, Incorporated.

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

Motif is a trademark of Open Software Foundation.

Netscape is a registered trademark of Netscape Communications Corporation in the United States and other countries.

UNIX is a registered trademark in the United States and/or other countries licensed exclusively through X/Open Company Limited.

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

About This Book

This book is designed to help you install the IBM Parallel Environment for AIX (PE) program product on the IBM RS/6000 SP or on a networked cluster of IBM RS/6000 processors.

This book assumes that AIX and the X-Windows system are already installed, if required. For information on installing AIX and X-Windows, consult *IBM AIX for RS/6000 Installation Guide*, SC23-2341.

Who Should Use This Book

This book is intended for system programmers and administrators, but should be used by anyone responsible for installing IBM Parallel Environment for AIX. To make this book a little easier to read, the name *IBM Parallel Environment for AIX* has been abbreviated to *PE* throughout.

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

How to Use This Book

Overview of Contents

This book contains the following information:

Chapter 1, "Introduction" 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 userid administration.

Chapter 3, "PE Version 2.4 Migration Information" on page 13 contains specific information on some differences between earlier releases that you need to consider prior to installing or using PE Version 2.4.

Chapter 4, "Installing the PE Software" on page 17 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 6, "How Installation of PE Alters Your System" on page 39 describes how your system is altered when you install the various PE software filesets.

Chapter 7, "File Formats" on page 49 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 53 explains the syntax of the commands for running the installation and deinstallation scripts provided with PE.

Appendix B, “POE Installation Verification Program” on page 57 explains how the POE Installation Verification Program (IVP) works.

Appendix C, “Using Additional POE Sample Applications” on page 59 describes some sample applications.

Typographic Conventions

This book uses the following typographic conventions:

Type Style	Used For
bold	**Bold** words or characters represent system elements that you must use literally, such as command names, flag names, and path names.
italic	*Italic* words or characters represent variable values that you must supply. *Italics* are also used for book titles and for general emphasis in text.
`Constant width`	Examples and information that the system displays appear in `constant width` typeface.

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**

The symbol “●” indicates the system response to an action. So the system's response to entering the **tool** command would read:

● The Tool Main Window opens.

Related IBM Publications

Parallel Environment (PE) Publications

As an alternative to ordering the individual books, you can use SBOF-8588 to order the entire PE library.

- *IBM Parallel Environment for AIX: Hitchhiker's Guide*, GC23-3895
- *IBM Parallel Environment for AIX: Operation and Use, Volume 1*, SC28-1979
- *IBM Parallel Environment for AIX: Operation and Use, Volume 2*, SC28-1980

  Part 1: Debugging and Visualizing
  Part 2: Profiling
- *IBM Parallel Environment for AIX: MPI Programming and Subroutine Reference*, GC23-3894
- *IBM Parallel Environment for AIX: Messages*, GC28-1982

- *IBM Parallel Environment for AIX: Licensed Program Specification*, GC23-3896

AIX Publications

The following publication:

- *IBM AIX and Related Products Documentation Overview*, SC23-2456

is available on the Web at:

http://www.rs6000.ibm.com/doc_link/en_US/a_doc_lib/aixuser/aix6kdov/toc.htm

LoadLeveler Publications

- *Using and Administering LoadLeveler*, SC23-3989

PSSP Publications

The following PSSP publications are related to this book only if you run parallel programs on the IBM RS/6000 SP. These books are not related if you use an IBM RS/6000 network cluster.

PSSP 3.1 Publications

- *IBM Parallel System Support Programs for AIX: Installation and Migration Guide*, GA22-7347
- *IBM Parallel System Support Programs for AIX: Administration Guide*, SA22-7348
- *IBM Parallel System Support Programs for AIX: Managing Shared Disks*, SA22-7349
- *IBM Parallel System Support Programs for AIX: Diagnosis Guide*, GA22-7350
- *IBM Parallel System Support Programs for AIX: Command and Technical Reference*, GA22-7351
- *IBM Parallel System Support Programs for AIX: Messages Reference*, GA22-7352

PSSP 2.4 Publications

- *IBM Parallel System Support Programs for AIX: Administration Guide*, GC23-3897

RS/6000 SP Publications

- *IBM RS/6000 SP: Planning, Volume 1, Hardware and Physical Environment*, GA22-7280
- *IBM RS/6000 SP: Planning, Volume 2, Control Workstation and Software Environment*, GA22-7281

National Language Support

For National Language Support (NLS), all PE components and tools display messages located in externalized message catalogs. English versions of the message catalogs are shipped with the PE program product, but your site may be using its own translated message catalogs. The AIX environment variable **NLSPATH** is used by the various PE components 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 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 these 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 additional information on NLS and message catalogs, see *IBM Parallel Environment for AIX: Messages*, GC28-1982 and *IBM AIX Version 4 General Programming Concepts: Writing and Debugging Programs*, SC23-2533.

Accessing Online Information

In order to use the PE man pages or access the PE online (HTML) publications, the **ppe.pedocs** file set must first be installed. To view the PE online publications, you also need access to an HTML document browser such as Netscape. An index to the HTML files that are provided with the **ppe.pedocs** file set is installed in the **/usr/lpp/ppe.pedocs/html** directory.

Online Information Resources

If you have a question about the SP, PSSP, or a related product, the following online information resources make it easy to find the information:

- Access the new SP Resource Center by issuing the command: **/usr/lpp/ssp/bin/resource_center**

  Note that the **ssp.resctr** fileset must be installed before you can do this.

  If you have the Resource Center on CD-ROM, see the readme.txt file for information on how to run it.

- Access the RS/6000 Web Site at: **http://www.rs6000.ibm.com**.

Getting the Books and the Examples Online

All of the PE books are available in Portable Document Format (PDF). They are included on the product media (tape or CD-ROM), and are part of the **ppe.pedocs** file set. If you have a question about the location of the PE softcopy books, see your system administrator.

To view the PE PDF publications, you need access to the Adobe Acrobat Reader 3.0.1. The Acrobat Reader is shipped with the AIX Version 4.3 Bonus Pack and is also freely available for downloading from the Adobe web site at URL **http://www.adobe.com**.

As stated above, you can also view or download the PE books from the IBM RS/6000 Web site at **http://www.rs6000.ibm.com**. The serial and parallel programs that you find in this book are also available from the IBM RS/6000 Web site. At the time this manual was published, the full path was **http://www.rs6000.ibm.com/resource/aix_resource/sp_books.** However, note that the structure of the RS/6000 Web site can change over time.

What's New in PE 2.4?

AIX 4.3 Support

With PE 2.4, POE supports user programs developed with AIX 4.3. It also supports programs developed with AIX 4.2, intended for execution on AIX 4.3.

Parallel Checkpoint/Restart

This release of PE provides a mechanism for temporarily saving the state of a parallel program at a specific point (*checkpointing*), and then later **restarting** it from the saved state. When a program is checkpointed, the checkpointing function captures the state of the application as well as all data, and saves it in a file. When the program is restarted, the restart function retrieves the application information from the file it saved, and the program then starts running again from the place at which it was saved.

Enhanced Job Management Function

In earlier releases of PE, POE relied on the SP Resource Manager for performing job management functions. These functions included keeping track of which nodes were available or allocated and loading the switch tables for programs performing User Space communications. LoadLeveler, which had only been used for batch job submissions in the past, is now replacing the Resource Manager as the job management system for PE. One notable effect of this change is that LoadLeveler now allows you to run more than one User Space task per node.

MPI I/O

With PE 2.4, the MPI library now includes support for a subset of MPI I/O, described by Chapter 9 of the MPI-2 document: *MPI-2: Extensions to the Message-Passing Interface, Version 2.0*. MPI-I/O provides a common programming interface, improving the portability of code that involves parallel I/O.

1024 Task Support

This release of PE supports a maximum of 1024 tasks per User Space MPI/LAPI job, as opposed to the previous release, which supported a maximum of 512 tasks. For jobs using the IP version of the MPI library, PE supports a maximum of 2048 tasks.

Enhanced Compiler Support

In this release, POE now supports the following compilers:

- C
- C++
- Fortran Version 5

- xlhpf

Xprofiler Enhancements

This release includes a variety of enhancements to Xprofiler, including:

- *Save Configuration* and *Load Configuration* options for saving the names of functions, currently in the display, and reloading them later in order to reconstruct the function call tree.
- An *Undo* option that lets you undo operations that involve adding or removing nodes or arcs from the function call tree.

Message Queue Facility

The **pedb** debugger now includes a message queue facility. Part of the **pedb** debugger interface, the message queue viewing feature can help you debug Message Passing Interface (MPI) applications by showing internal message request queue information. With this feature, you can view:

- A summary of the number of active messages for each task in the application. You can select criteria for the summary information based on message type and source, destination, and tag filters.
- Message queue information for a specific task.
- Detailed information about a specific message.

Chapter 1. Introduction

The IBM Parallel Environment for AIX (PE) program product is a set of programs designed to help you develop parallel Fortran, C, or C++ programs and execute them on an IBM RS/6000 SP, a networked cluster of IBM RS/6000 systems, or a mixed SP/cluster environment.

Before following the instructions in this book on how to install the PE software, you should have a basic idea of the components that make up the PE product and how they fit together. PE consists of six components that help you develop, debug, analyze, and run parallel programs. The usage of these components is discussed in:

- *IBM Parallel Environment for AIX: Operation and Use, Volume 1*, SC28-1979
- *IBM Parallel Environment for AIX: Operation and Use, Volume 2*, SC28-1980

If you are new to PE, you will probably find *IBM Parallel Environment for AIX: Hitchhiker's Guide*, GC23-3895 useful.

PE Components

The PE components are:

Message Passing and Collective Communication Application Programming Interface (API) Subroutine Libraries

These libraries, which contain subroutines that help application developers parallelize their code, are described in *IBM Parallel Environment for AIX: MPI Programming and Subroutine Reference*, GC23-3894 .

Parallel Operating Environment (POE)

This is software that 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, starts your code running, and oversees its operation.

POE also contains two X-Windows analysis tools: the Program Marker Array and the System Status Array. The Program Marker Array is a run-time analysis tool that lets you monitor a program's execution. The System Status Array lets you monitor the operational status and CPU utilization of processor nodes.

For more information, refer to *IBM Parallel Environment for AIX: Operation and Use, Volume 1*, SC28-1979.

Parallel Environment's Debuggers

PE has the following two parallel debuggers:

pdbx a line-oriented debugger based on the **dbx** debugger

pedb a Motif-based X-Windows debugger

pedb is a separately installed fileset and is described in *IBM Parallel Environment for AIX: Operation and Use, Volume 2*, SC28-1980.

Note: In PE release levels prior to Version 2 Release 2, **pedb** was known as **xpdbx**.

Visualization Tool (VT)

This tool consists of a trace generation facility and a trace display system that allow you to visualize performance characteristics of your program and system. You can use VT to play back traces recorded during a program's run (trace visualization) or as an online monitor of system activity (performance monitoring).

VT is a separately installable fileset and is described in *IBM Parallel Environment for AIX: Operation and Use, Volume 2*, SC28-1980.

Xprofiler

This is a tool that helps you analyze your parallel or serial application's performance quickly and easily. It uses procedure profiling information to construct a graphical display of the functions within your application.

Xprofiler provides quick access to the profiled data, which lets you identify the functions that are the most CPU-intensive. The graphical user interface also lets you manipulate the display in order to focus on the application's critical areas.

Xprofiler is a separately installable fileset and is described in *IBM Parallel Environment for AIX: Operation and Use, Volume 2*, SC28-1980.

PE Documentation (PEDOCS)

This is a separate fileset that contains the PE documentation. This documentation consists of:

- HTML files that allow you to read the books online (via an HTML browser such as Netscape)
- PDF files that allow you to print out copies of the books
- man pages for MPI subroutines

Chapter 2. Planning to Install the PE Software

When planning to install the IBM Parallel Environment for AIX 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. The following sections address these and other important pre-installation topics.

PE Installation Requirements

This section describes the system requirements for installing and running the PE software. It contains sections for hardware, software, and disk space requirements and provides additional information relevant to installing PE.

Hardware Requirements

The PE software runs on the following:

- IBM RS/6000 SP
- IBM RS/6000 clusters that support TCP/IP

The message passing libraries support the following hardware configurations:

- IBM RS/6000 workstation clusters via IP protocol only
- IBM RS/6000 SP System without an SP Switch via IP protocol only
- IBM RS/6000 SP System with an SP 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 Version 4.3.1 and associated programs, refer to *IBM RS/6000 SP: Planning, Volume 1, Hardware and Physical Environment*, GA22-7280 .

Software Requirements

The software required for PE includes PE filesets plus additional software, as explained in the following sections.

PE Fileset Requirements

PE Version 2.4 consists of the filesets listed in the table below. You need to decide which of these filesets to install on the various nodes in your system, based on the PE product options you plan to use.

Notes:

1. For more information about nodes, see "Node Resources" on page 8.
2. For information about installing any of the following product options individually, see "PE Installation Procedure Summary" on page 20.

Table 1. PE Fileset Requirements

If you plan to...	**...this product option is required:**	**Fileset Name:**	**Notes:**
...develop and execute parallel applications from a node	Parallel Operating Environment	**ppe.poe**	The VT trace and data collection code, as well as the **pdbx** command-line parallel debugger, is part of POE. However, if you plan to use **poestat**, you also need to install **ppe.vt**. **Installing POE with NIS** 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.
...perform trace visualization and performance monitoring or if you want to run the **poestat** command of **poe**	Visualization Tool	**ppe.vt**	You need to have POE installed on any node on which you also want to generate trace files or to do performance monitoring . You do not need POE on nodes in order to look at trace files generated from other nodes.
...use the X-Windows version of the debugger facility	X-Windows Parallel Debugger	**ppe.pedb**	The fileset **ppe.poe** is required on any node that is to install **pedb**. Install **ppe.pedb** on those nodes or system on which you want to use **pedb** to debug your code.
...use the Xprofiler X-Windows Performance Profiler tool	Xprofiler X-Windows Performance Profiler	**ppe.xprofiler**	Xprofiler extends the usability of **gprof** by providing an environment for exploring the **gmon.out** data in a variety of ways. It provides a graphical function call tree display, navigation tools, and various filters to analyze an application's performance profiling data. The installation of **ppe.xprofiler** is not dependent on the prior installation of any other PE products. **Note:** Although it is not required to install Xprofiler on every node, it is advisable to install it on at least one node in each group of nodes that have the same software library levels. (See “Limitations” on page 6.)
...access the online documentation	PE documentation	**ppe.pedocs**	Installing **ppe.pedocs** gives you the PE documentation, as described under PE Documentation (PEDOCS) on page 2.

Additional Software Requirements

PE Version 2.4 also requires some additional software products or filesets, listed in the table below. You need to decide which of these software products or filesets to install on your system, based on how you plan to use PE.

Table 2 (Page 1 of 2). Additional Software Requirements

If you plan to...	...this software is required:	Notes:
Always required	AIX Version 4.3.2 for IBM RS/6000 (5765-C34) for Servers	
...run a parallel program on the IBM RS/6000 SP	IBM Parallel System Support Programs (PSSP) for AIX (5765-D51) Version 3.1	**On the SP:** Ensure that the Communication Subsystem (CSS) libraries, Resource Manager, and System Data Repository (SDR) client libraries are available in their default locations.
...compile parallel executables	IBM C for AIX Version 4.3, 5765-AAR (part number 04L0675 with feature number 2163) *or* IBM C and C++ Compilers for AIX Version 3.6.4, 5801-AAR (part number 04L3535) *or* IBM XL Fortran for AIX Version 5.1.1 or later, 5808-AAR (part number 04L2110) **Note:** Due to limitations with threads support, XL Fortran for AIX Version 4 (5765-658) is supported in binary compatibility mode. *or* IBM XL High Performance Fortran (HPF) for AIX Version 1.3.1 or later, 5765-613 **Note:** For full debugging support within PE, use Fortran 77 and C.	
...submit a POE job that will use the SP Resource Manager from a non-SP node (for example, a standalone RS/6000 workstation run off the rack)	**ssp.clients** fileset, on the non-SP node	See “When to Install ssp.clients (SP Resource Manager)” on page 19 for detailed information.
...submit a POE job from outside a LoadLeveler cluster	**loadl.so** on the node outside the LoadLeveler cluster	See “When to Install loadl.so (LoadLeveler)” on page 20 for detailed information.
...use either of the PE debuggers	**bos.adt.debug** fileset	

Table 2 (Page 2 of 2). Additional Software Requirements

If you plan to...	...this software is required:	Notes:
...use Xprofiler	For the CDE environment: • **X11.Dt.lib 4.2.1.0** or later	
...use the VT visualization capability	IBM C and C++ Compilers for AIX Version 3.6.4, 5801-AAR (part number 04L3535)	**On the SP:** Use the Network Time Protocol (NTP) utility to synchronize the clocks on all machines. **On an IBM RS/6000 network cluster:** Use any available Internet host/server utility to synchronize the clocks on all machines. **Note:** If the SP Switch is installed, VT uses it for common time on each node, and no additional synchronization software is required.
...use LoadLeveler to allow execution of batch jobs	LoadLeveler Version 2.1	

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.

Note: If you plan to install the PE software on an IBM RS/6000 network cluster, each machine in the cluster on which you install it must meet these disk space requirements.

PE Fileset	Number of 512-Byte Blocks Required in Directory:		
	/usr	**/tmp**	**/etc**
ppe.poe	75000	15000	10
ppe.pedb	5600	N/A	N/A
ppe.vt	6000	N/A	N/A
ppe.xprofiler	6500	N/A	N/A
ppe.pedocs	15000	N/A	N/A

Limitations

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

Fortran 90 and MPI

Incompatibilities exist between Fortran 90 and MPI that may affect the ability to use such programs. For further information, refer to **/usr/lpp/ppe.poe/samples/mpif90/README.mpif90** after installing POE, and to “Enabling Fortran 90 Compiler Support” on page 36.

MPI-IO This release of PE includes a subset of the new MPI functionality defined by the MPI-IO chapter of the MPI-2 document. This MPI-2 functionality is provided in the threaded version of the MPI library, but not in the signals-based version. MPI-IO is intended to be used with the IBM Generalized Parallel File System (GPFS). MPI-IO depends on having a single file system underlying all tasks of an MPI job. Shared file systems such as NFS and AFS do not meet this requirement when they are used across multiple nodes. MPI jobs that have all tasks on a single node can use non-GPFS file systems, but this is not expected to be a useful model for production use of MPI-IO.

Parallel Applications and System Calls

User-written parallel applications are limited in their use of system calls. *IBM Parallel Environment for AIX: MPI Programming and Subroutine Reference*, GC23-3894 provides a discussion of these limitations.

Parallel Debuggers

When using the Parallel Debuggers, the application should have been compiled using the parallel compiler scripts supplied with POE (namely **mpcc**, **mpcc_r**, **mpxlf**, or **mpxlf_r**). Both debuggers currently support only Fortran 77 and C.

SP Switch Adapter-1

Customers with SP systems who have SP Switch Adapter-1 installed will not be able to use Version 2.4 of PE, because this adapter is no longer supported.

VisualAge C++

Parallel Environment does not support IBM VisualAge C++ Professional for AIX, Version 4.0 incremental compiler and C++ runtime library Version 4.0. This does not apply to the batch IBM C and C++ Version 3.6 compilers and the Version 3.6 C++ runtime libraries that are also included in VisualAge C++ Version 4.0.

VT VT will generate trace files for applications running up to 128 nodes. When the visualization portion of VT is used, some of the displays will experience degraded usability for more than 32 nodes.

Xprofiler

If users plan to collect a **gmon.out** file on one machine and then use Xprofiler to analyze the data on another machine, they should be aware that some shared (system) libraries may not be the same on the two machines, which may result in different function call tree displays for those shared libraries.

32-Bit and 64-Bit Application Support

Parallel Environment supports 32-bit applications only. 64-bit applications are not supported and will not run.

Information for the System Administrator

Software Compatibility Within Workstation Clusters and Within Partitions

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

When you use *partitioning* (available in PSSP Version 2 or later) on an IBM RS/6000 SP, you may have partitions at different levels of PE software; however, *within a partition,* all the nodes must be at the same level of PE software. (This ensures that an individual PE application can run on any node in the partition.)

Table 3 on page 14 lists the versions of PSSP and AIX required on a particular workstation cluster or partition, depending on the version of PE installed on that cluster or partition, and the possible migration paths.

Note: PSSP cannot be put on a workstation.

Note 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 and you will be using libraries for a back level, be sure to change the compiler library links or reinstall compilers.

Node Resources

How you plan your node resources will vary according to whether you are installing PE on an IBM RS/6000 SP or an IBM RS/6000 network cluster.

On an IBM RS/6000 SP...

...Using Resource Manager: On an SP system, you partition nodes into *pools* and assign numbers and other information to these pools.

Pools and the Resource Manager: Pools are managed by the Resource Manager. You tell the POE Partition Manager which pools to use; the Partition Manager in turn requests the Resource Manager for nodes in the specified pools.

For more information on the Resource Manager and setting up pools, see *IBM Parallel System Support Programs for AIX: Administration Guide*, GC23-3897.

...Within a LoadLeveler Cluster: On an SP system within a LoadLeveler cluster, the system administrator uses LoadLeveler to partition nodes into *pools* and/or *features*, to which he or she assigns numbers and other information.

Home Node: The workstation from which parallel jobs are started. The *home node* can be any workstation on the LAN.

On an IBM RS/6000 Network Cluster

On an IBM RS/6000 network cluster, you assign workstations to the following categories:

- *Home node* (workstation from which parallel jobs are started) to run the Partition Manager in POE
- Workstations for *developing and compiling* applications
- Workstations for *executing applications* in the parallel environment

  You need to identify these workstations running as execution nodes by name in a host list file.

Deciding Which Nodes Require Which PE Filesets or Additional Software

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

File Systems

The PE filesets (**poe**, **vt**, **pedb**, **pedocs**, and **xprofiler**)are installed in the **/usr** file system. When the **poe** fileset 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.

If you do not use a shared file system, you need to copy the user's executable files to the other nodes. To copy them, use the scripts provided by PE: **mprcp** and **mpmkdir**. You can also use **mcp**, the message passing file copy command. For more information on copying the file system and these scripts, see *IBM Parallel Environment for AIX: Operation and Use, Volume 1*, SC28-1979.

Also, you can declare these files part of a file collection. A file collection is a set of files and directories that are duplicated on multiple machines in a network and managed by tools that simplify their control and maintenance. For more information about file collections, see *IBM Parallel System Support Programs for AIX: Administration Guide*, SA22-7348 .

User IDs on Remote Nodes

The system administrator must set up a user ID, other than a root ID, for each user on each remote node that requires POE access.

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

With PE Version 2 Release 4, interactive and batch parallel jobs can be submitted under LoadLeveler. When LoadLeveler is used, LoadLeveler is completely responsible for the user authorization. Any user authorization under POE is bypassed.

When LoadLeveler is *not* used, POE handles the user authorization. The following sections on POE user authorization apply when POE is used without LoadLeveler.

POE supports two methods of user authorization for submitting a parallel job:

AIX authorization (default)

via **/etc/hosts.equiv** or **.rhosts** entries

This is the default mechanism.

DFS/DCE authorization

where POE checks for a valid set of DCE credentials for the user

The user authorization mechanism is controlled by the **MP_AUTH** POE environment variable. This variable can be defined by the system administrator in the **/etc/poe.limits** file, as described in “Using the /etc/poe.limits File” on page 50, so that users do not need to decide which mechanism to use.

The two types of authorization cannot be mixed in the same parallel job. All tasks and nodes defined for a POE job must use the same type of authorization.

Using AIX User Authorization

If AIX user authorization (the default) is used as a security mechanism on the system, each node needs to be set up so that each userid is authorized to access that node or remote link from the initiating home node. The **/etc/hosts.equiv** file and/or the **.rhosts** file are used 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**, refer to the chapter on managing jobs in *IBM AIX Version 4 Files Reference for AIX*, SC23-2512.

Using DFS/DCE User Authorization

If DFS/DCE user authorization is used as a security mechanism on the system, POE accepts a valid set of DCE user credentials as user authorization for executing parallel jobs.

In order to use DFS/DCE with POE, the following are required:

- a valid set of DCE credentials, established using a **dce_login** procedure
- the DCE credentials copied to each node, using the **poeauth** command
- the **MP_AUTH** POE environment variable set to **DFS**, either in the **/etc/poe.limits** file or by the user

Note: When DFS/DCE authorization is selected, there is no need for entries in either the **/etc/hosts.equiv** file or the **.rhosts** file, as these are not checked by POE.

For more information about running POE in a DFS/DCE environment, and about the **poeauth** command, see *IBM Parallel Environment for AIX: Operation and Use, Volume 1*, SC28-1979 .

Port Numbers

When POE is installed, it modifies entries in **/etc/services** and in **/etc/inetd.conf** to install the Partition Manager daemon. In doing so, it requires an available port number which must be the same number on all nodes on which POE is to be installed and running. You need to ensure such a port number is available.

Running Large POE Jobs and IP Buffer Usage

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

- Partition size is larger than 128 nodes.
- Large amounts of STDIO (stdin, stdout, or stderr) are generated.
- The home node is running many POE jobs simultaneously, and/or there is significant additional IP traffic via mounted file system activity or other sources.
- Many large messages are passed via the UDP 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 and option 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 or on the SP nodes being used in the partition. (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.

Notes:

1. 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.
2. In AIX Version 4.3, the **thewall** default value is 16384.

On SP nodes, you can use the **dsh** command to execute the **no** command on each node of an SP. See the section on tuning in *IBM Parallel System Support Programs for AIX: Administration Guide*, SA22-7348 for more information on **dsh**).

For non-SP nodes, 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 in general on mbufs, see *IBM AIX Versions 3.2 and 4 Performance Monitoring and Tuning*, SC23-2365 .

Running Multiple Versions of POE

POE Version 1 and POE Version 2 are not compatible. All of your tasks must run with *either* POE Version 2 or with POE Version 1, not a combination of the two. . The POE home node and all remote nodes must run with the same version of code. You must be at the same level of AIX and PSSP within a partition to submit PE jobs. See Chapter 3, “PE Version 2.4 Migration Information” on page 13 for more information.

Partition Manager Daemon Services and Installation

As part of the Version 2 installation the Partition Manager daemon (**pmd**) and POE executables have different names than their Version 1 counterparts. Also, different TCP/IP port numbers and daemon service names are utilized. Furthermore, Version 2 and Version 1 files use different directory path names.

The following table summarizes the differences and can be used to tell which version of POE you have if you are not sure.

Type of Name or Number	POE Version 1	POE Version 2
Service name in /etc/services	pm2	pmv2
Daemon name in /etc/inetd.conf	pmd2	pmdv2
Default port number	6124	6125
pmd executable name	pmd2	pmdv2
File path name	/usr/lpp/poe	/usr/lpp/ppe.poe

Chapter 3. PE Version 2.4 Migration Information

This chapter is intended for customers migrating from earlier releases of PE to PE Version 2.4. It contains specific information on some differences between earlier releases that you need to consider prior to installing or using PE Version 2.4. To find out which release of PE you currently have installed, use **lslpp**.

General Overview

If you have an earlier release of PE already installed, installing PE Version 2.4 filesets involves a migration installation on top of the earlier filesets. The earlier filesets will be replaced, unnecessary files and directories will be removed and rendered obsolete, and disk space conserved.

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 Fileset Alters Your System” on page 39.

To the ODM and **lslpp**, however, the earlier filesets will show as installed but marked “OBSOLETE.” Additionally, some older directories and installation-related files may remain. A potential problem that may result from this is that if you later attempt to remove an older fileset, files from the *newer* fileset may be removed instead.

To avoid this potential side effect, completely remove older releases of the PE filesets *before* you begin installation. For more details, see “Migration Installation” on page 18.

AIX Compatibility

PE Version 2.4 commands and applications are compatible with AIX Version 4.3.2 or later only, not with earlier versions of AIX.

Coexistence

PE on the SP requires that all nodes involved in a parallel job be running the same level of PE.

When LoadLeveler and PE coexist on a node, they must be at one of these levels:

1. LoadLeveler 2.1 with PE 2.4
2. LoadLeveler 1.3 with PE 2.3
3. LoadLeveler 1.3 with PE 2.2

The PE libraries used to run a job on a node must be compatible with the PSSP libraries on that node. The following combinations are compatible:

1. PE 2.3 and PSSP 2.3
2. PE 2.3 and PSSP 2.4
3. PE 2.3 and PSSP 3.1 (with restrictions)

4. PE 2.4 and PSSP 2.3 (with restrictions)
5. PE 2.4 and PSSP 2.4 (with restrictions)
6. PE 2.4 with PSSP 3.1

The restrictions are the same in all cases:

- No use of Parallel ESSL
- No explicit use of non-blocking collective communications to the MPI standard
- No functions that are new in PE 2.4 (for example: MPI-IO, 1024 task support, and multiple user space support).

Migration Support

PE does not support node-by-node migration. All of the nodes in a system partition must be migrated to a new level of PE at the same time. PE supports running a particular release of PE on plus or minus one level of AIX or PSSP, which lets customers migrate to a new level of AIX or PSSP without having to change to a new level of PE.

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

1. AIX 4.2.1, PSSP 2.3, and PE 2.3 to either AIX 4.3.2, PSSP 2.3 (plus a required PTF) and PE 2.3 (plus a required PTF), or AIX 4.2.1, PSSP 2.4 and PE 2.4 or PE 2.3 (plus a required PTF)
2. AIX 4.2.1, PSSP 2.3 (plus a required PTF) and PE 2.3 (plus a required PTF), or AIX 4.2.1, PSSP 2.4 and PE 2.3 (plus a required PTF) to AIX 4.3.2, PSSP 2.4 and PE 2.3 (plus a required PTF)
3. AIX 4.2.1 and PSSP 2.4 can migrate to AIX 4.3.2, PSSP 3.1, and PE 2.4
4. AIX 4.2.1 and PSSP 2.4 can migrate to AIX 4.3.2 and PSSP 3.1 without having to migrate to PE 2.4

For example, a customer running PE 2.3, PSSP 2.3, and AIX 4.2.1 will be able to migrate to AIX 4.3.2 without migrating to a new level of PE (or PSSP).

Note: A PTF is required for PE 2.3 to run in binary compatibility mode on AIX 4.3.2, but the PTF can be applied to PE 2.3 before migrating from AIX 4.2.1 to 4.3.2.

Table 3. PE, PSSP, and AIX Versions for a Workstation Cluster or Partition

Version of PE Installed on Workstation Cluster or Partition	Required Version of PSSP	Required Version of AIX
2.4	3.1	4.3.2 or later
2.3	2.3 or 2.4	4.2.1 or 4.3.1
2.2	2.2.0	4.1.4
2.1	2.1	4.1.3
1	1	3.2.5

MPI and MPL Applications

- The MPI function became available in Version 2.
- The MPL message-passing applications are source-compatible between PE 1.2 and PE 2.4, but must be recompiled.

Parallel Client Function No Longer Available

The Parallel Client Function feature available in Version 1 is no longer available or supported in Version 2. You need to install the entire POE fileset on any IBM RS/6000 workstation running parallel jobs.

Parallel Desktop Removed

The Parallel Desktop function of PE Version 1 has been removed. It is no longer supported.

POE Compile Script Configuration File

POE no longer uses its own compiler configuration file (**/usr/lpp/ppe.poe/lib/poe.cfg** is no longer shipped). If you had modified **poe.cfg** in previous versions, you may need to make similar changes in the default compiler configuration files or to your compiler scripts. The **poe.cfg** file is saved as part of a migration installation, so any previous changes are preserved. See “How Installing the poe Fileset Alters Your System” on page 39 for the files that are saved.

SP_NAME Environment Variable

Use of the **SP_NAME** environment variable, which determines the job management system to use (LoadLeveler or the SP system Resource Manager), is changed for this release. If all nodes to be used for the parallel job exist in a PSSP 2.4 partition, the **SP_NAME** environment variable should be set to the name of the control workstation of the SP system on which these nodes exist. This is the only case that results in POE contacting the Resource Manager rather than LoadLeveler for node allocation requests. See “When to Install ssp.clients (SP Resource Manager)” on page 19 for more information.

Chapter 4. Installing the PE Software

This chapter explains how to install the five PE filesets.

PE Installation Overview

Installation on an IBM RS/6000 SP is different from installation on an IBM RS/6000 network cluster. The main difference is that the SP installation allows you to use the SP system management functions to manage and maintain the system software, whereas these functions are not available on an IBM RS/6000 network cluster.

In both environments, you first install the desired PE filesets 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 .

About Installing PE on an SP System

You can install the desired PE filesets on an SP in one of three ways:

- Use the SP system management software; this allows you to control and automate much of the software maintenance throughout your system.

  To use this method, refer to the chapter on “Performing Software Maintenance” in *IBM Parallel System Support Programs for AIX: Installation and Migration Guide*, GA22-7347 .

- Use the **PEinstallSP** script.
- Install the software on each node individually using **SMIT** or **installp**.

In any event, you must first install the PE filesets on at least one node of your system. Preferably, install them on the control workstation to allow use of the system management software functions.

When You Install PE on the SP Control Workstation

If you install PE on the SP Control Workstation (using one of the methods explained later in this chapter), you will need to do the following in order to use the User Space libraries:

1. Login as root and create the link to the user space library for your switch adapter type.

   - For TB2 switch adapters, the links are:

     ln -s /usr/lpp/ssp/css/libtb2/libmpci.a /etc/ssp/css/libus/libmpci.a

     ln -s /usr/lpp/ssp/css/libtb2/libmpci_r.a /etc/ssp/css/libus/libmpci_r.a

   - For TB3 switch adapters, the links are:

     ln -s /usr/lpp/ssp/css/libtb3/libmpci.a /etc/ssp/css/libus/libmpci.a

     ln -s /usr/lpp/ssp/css/libtb3/libmpci_r.a /etc/ssp/css/libus/libmpci_r.a

   Note:

   Establish this link before installing POE. The installation steps depend on the correct adapter libraries being properly linked. It is is normally created by /usr/lpp/ssp/css/rc.switch when called from /etc/inittab running on an SP node. The link does not get created on the SP control workstation since it cannot determine what type of switch adapter should be used. Failure to set up this link may prevent you from using POE.

2. Verify the POE installation by running the Installation Verification Program (IVP).

If You Receive sysck Warning Messages

During the course of installing PE filesets on an SP, you may encounter **sysck** warning messages that a particular file is also owned by another fileset. If the file is also owned by one of the PSSP filesets, such as **ssp.css**, then these messages can be ignored. However, if the warning messages are for older PE related filesets, such as **poe** or **poe_pcf**, then this may indicate an older version is installed.

These warning messages can be ignored, as the system will function properly. However, if you later choose to remove the old fileset after installing PE Version 2.4 , you will need to repeat the installation of the new fileset.

About Installing PE on an IBM RS/6000 Cluster

Installation on an IBM RS/6000 network cluster is similar to that on the SP, with the exception that there are no system management functions, leaving 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 filesets 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 filesets on a cluster, you may encounter sysck warning messages that a particular file is also owned by another fileset. If the file is also owned by one of the older PE filesets, such as **poe** or **poe_pcf**, then this may indicate an older version is installed.

These warning messages can be ignored, as the system will function properly. However, if you later choose to remove the old fileset after installing PE Version 2.4, you will need to repeat the installation of the new fileset.

Migration Installation

If you migrate from PE Version 1 or Version 2.1 to PE Version 2 Release 4 , installing the new filesets will completely replace some of the earlier release filesets, rendering them obsolete. The replaced filesets will be marked “OBSOLETE” in the ODM and **lslpp** by **installp**.

However, some directories and install files will remain. Since these earlier filesets do not coexist or execute with PE Version 2.4 , *you should uninstall your old filesets before installing the new PE filesets*, rather than installing the new filesets on top of the old. This will conserve disk space and reduce the chance for confusion over old fileset path names, executables, etc.

CAUTION:
If you plan to uninstall the old filesets, do so *before* installing the new filesets. If you attempt to uninstall the old filesets *after* installing PE Version 2.4 , you may accidentally delete some needed files that may affect your system.

The following table lists the old filesets that need to be removed before installing PE Version 2.4 :

PE Version	Filesets to be Removed
1	**poe** **poe_pcf** **xpdbx** **vt** **pedocs**
2.1	**ppe.xpdbx** **ppe.pedocs**

Determining Which Earlier Filesets Are Installed

You can use the **lslpp** command to check if any of the above filesets are installed. For example, **lslpp -l poe** will tell you if the Version 1 poe fileset is installed.

Removing Earlier Filesets

To remove filesets 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** or **PEdeinstallSP** scripts

  See “Removing an Installation Image” on page 35.

When to Install ssp.clients (SP Resource Manager)

Install this fileset to submit a POE job which uses the SP Resource Manager from a non-SP node. To install, do the following:

1. Contact the system administrator of your SP to determine the pathname to the exported directory containing the pssp.installp image.
2. NFS mount that directory on the submitting node.
3. Install the **ssp.clients** fileset using the following command:

   installp -aFXd /<mounted_image_directory>/pssp.installp ssp.clients

When to Install loadl.so (LoadLeveler)

Install this fileset 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** fileset using the following command:

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

4. Obtain the LoadLeveler configuration file. (See *Using and Administering LoadLeveler*, SC23-3989 .)

Viewing the README File Prior to Installation

Prior to actually installing any fileset, you may want to look at its README file. The README file may contain some special or additional information about installing the fileset. The PE filesets are all shipped with a copy of the README as part of the first file on the tape. 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 fileset is installed you can find the README file in the **/usr/lpp/**<*fileset*>**/README** directory, with a name of <*fileset*>**.README**.

PE Installation Procedure Summary

Summarized below are the basic steps you must follow to install the PE software on the SP or an IBM RS/6000 network cluster.

You can install all of the PE filesets at once, or you can install selected filesets one at a time. To determine which filesets, if any, that you want to install separately, see “PE Fileset Requirements” on page 3.

If you are installing:					**Perform these steps:**	
ppe.poe	**ppe.vt**	**ppe.pedb**	**ppe.xprofiler**	**ppe.pedocs**		
X	X	X	X	X	“Step 1: Copying the Software to a Hard Disk for Installation Over a Network” on page 21	**Standard Steps**
					“Step 2: Performing the Initial Installation” on page 23	
					“Step 3: Installing PE on Other Nodes” on page 26	

If you are installing:					Perform these steps:	
ppe.poe	ppe.vt	ppe.pedb	ppe.xprofiler	ppe.pedocs		
X					“Step 4: Verifying the POE Installation” on page 30	**Optional Steps**
	X				“Step 5: Verifying the VT Installation” on page 32	
X					Appendix C, “Using Additional POE Sample Applications” on page 59	

Installing the PE Filesets Step-by-Step

This section provides the step-by-step procedure for installing the PE software on the SP or on the IBM RS/6000 network cluster. Each step includes one or more tables that guide you through choices about such variables as:

- the type of system (SP or IBM RS/6000 network cluster) on which you are installing PE
- the specific PE filesets you are installing

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

Notes:

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 an Installation Image” on page 35.)
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. For information on the correct SMIT path, see *IBM Parallel System Support Programs for AIX: Installation and Migration Guide*, GA22-7347 .

Step 1: Copying 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.

If you are installing on the SP:	If you are installing on an IBM RS/6000 network cluster:
You must always complete this step.	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 an Installation Image" on page 35.)

Substep 1: Copying the Software off the Distribution Medium

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

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 for copying software to a hard disk.

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

TYPE IN the appropriate installation directory name in the **DIRECTORY for storing software** field, as shown in the table below:

If you are installing on the SP:	If you are installing on an IBM RS/6000 network cluster:
/spdata/sys1/install/pssplpp	**/usr/sys/inst.images**

PRESS **Do**

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

SELECT **Exit → Exit SMIT**

- The SMIT window closes.

Substep 2: Exporting the Installation Directory

To export the directory so the machines in your cluster can install the PE installation images it contains, enter the appropriate command, as shown in the following table:

If you are installing on the SP:	If you are installing on an IBM RS/6000 network cluster:
/usr/sbin/mknfsexp -d /spdata/sys1/install/pssplpp	**/usr/sbin/mknfsexp -d /usr/sys/inst.images**

Step 2: Performing 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 filesets or just certain individual filesets.

Note: Keep in mind that some of the PE filesets depend on others to run. "PE Fileset Requirements" on page 3 details these dependencies. Refer to this section before you do a partial installation.

If you are installing on the SP:	If you are installing on an IBM RS/6000 network cluster:
Perform this step on the initial SPnode or on the control workstation. You must login as **root**.	Perform this step on any machine in the cluster. You must login as **root**.

Method 1: Using the installp Command

To initially install the installation image, enter the appropriate command as shown in the following table:

To install:	ENTER
all software filesets	**installp -a -d** *devicename* **ppe***
just the poe fileset	**installp -a -I -X -d** *devicename* **ppe.poe**
just the pedb fileset	**installp -a -I -X -d** *devicename* **ppe.pedb**
just the VT fileset	**installp -a -I -X -d** *devicename* **ppe.vt**
just the Xprofiler fileset	**installp -a -I -X -d** *devicename* **ppe.xprofiler**
just the pedocs fileset	**installp -a -I -X -d***devicename* **ppe.pedocs**
Note: In the commands above: **-I** (capital i) is used to select only the specified fileset. **-a** applies the software products. *devicename* is the name of the installation device or directory. For more information on the **installp** command, refer to *IBM AIX Version 4.1 Commands Reference*, SBOF-1851 and *IBM AIX Version 4 General Programming Concepts: Writing and Debugging Programs*, SC23-2533.	

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

Method 2: Using SMIT

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

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 the following table:

If you want to install:	Type this in the “SOFTWARE to install” field:
all the PE software	**ppe***
just the poe fileset	**ppe.poe**
just the pedb fileset	**ppe.pedb**
just the VT fileset	**ppe.vt**
just the Xprofiler fileset	**ppe.xprofiler**
just the pedocs fileset	**ppe.pedocs**

Note: 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.

PRESS **Do**

- The system installs the installation image.

Note: The POE installation process checks to see if the **digd** daemon is running. If it is not running, it will start the daemon. If it is running, it will not start it, and will tell you the daemon's process number, so you can kill it manually.

Note: For more information on SMIT, refer to *IBM AIX Version 4 General Programming Concepts: Writing and Debugging Programs*, SC23-2533.

If Installation Fails

If installation fails, 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 ascertain the cause of a failed installation, refer to the installation status file. This file indicates how far installation had progressed when the errors occurred. The status file is described in more detail in *IBM AIX Version 4 General Programming Concepts: Writing and Debugging Programs*, SC23-2533. If you cannot ascertain the cause of a failed installation, contact your local IBM representative.

Proceeding to the Next Step

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, “How Installation of PE Alters Your System” on page 39.

To determine which remaining steps you need to perform, refer to the following table:

If there are other nodes in your system on which you need to install PE filesets:	**If there are *not* any other nodes in your system on which you need to install PE filesets:**
Proceed to • “Step 3: Installing PE on Other Nodes” on page 26	Skip: • “Step 3: Installing PE on Other Nodes” on page 26 If appropriate, proceed to: • “Step 4: Verifying the POE Installation” on page 30 *and/or* • “Step 5: Verifying the VT Installation” on page 32

Step 3: Installing 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

If you are installing on the SP:	**If you are installing on an IBM RS/6000 network cluster:**
Perform this step from the initial SP node or from the control workstation as **root**. Ensure that you have executed the **k4init** command to obtain Kerberos authentication for accessing your nodes. **Note:** On an SP, an alternate installation method is to use the SP system management software, as described in the chapter on “Performing Software Maintenance” in *IBM Parallel System Support Programs for AIX: Installation and Migration Guide*, GA22-7347 . However, a disadvantage of that method is that it causes each node to reboot after installation. If you do not want each node to reboot after installation, use one of the two installation methods explained in this step.	Perform this step from a node with PE installed as **root**.

Method 1: Using an 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 installation script that is appropriate for your system

Substep 1: Creating a Host List File

To create a host list file, follow the instructions below:

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

   Note: By default, the installation scripts look for a file named **host.list** in your current directory; however, you can 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 per line. For example:

```
hostname1
hostname2
hostname3
hostname4
hostname5
```

Substep 2: Running the Installation Script with the -copy or -mount option

To run the installation script, enter the command that is appropriate for your system, as listed in the table below. (For a detailed explanation of the syntax of each of these commands, see Appendix A, “Syntax of Commands for Running Installation and Deinstallation Scripts” on page 53.)

<table>
<tr><th>If you are installing on the SP:</th><th>If you are installing on an IBM RS/6000 network cluster:</th></tr>
<tr><td><b>PEinstallSP</b> <i>image_name</i> [<i>host_list_file</i>] [<b>-f</b> <i>fanout_value</i>] [<b>-copy | -mount</b>]</td><td><b>PEinstall</b> <i>image_name</i> [<i>host_list_file</i>] [<b>-copy | -mount</b>]</td></tr>
<tr><td colspan="2"><b>Notes:</b>
1. To execute the <b>installp</b> remotely on a <i>mounted</i> image, the directory containing the image must have world-writable permissions (as created by the <b>chmod 777</b> command).
If you do not want to create this directory with world-writable permissions, do not use the <b>-mount</b> option of <b>PEinstallSP</b> or <b>PEinstall</b>.
2. To have the image copied or mounted to different directories, you will need to invoke <b>PEinstallSP</b> or <b>PEinstall</b> for each different location or set of locations. The host list file that you specify each time you invoke <b>PEinstallSP</b> or <b>PEinstall</b> should reflect only those nodes that you want to use with <b>-copy</b> or <b>-mount</b>.</td></tr>
</table>

If you specify the -copy option, you will be prompted for:

- the installation image source directory

  The default is:

  For PEinstallSP:

  /spdata/sys1/install/pssplpp

For PEinstall:

/usr/sys/inst.images

- the installation image destination directory which is used for all nodes in the host list

  The default is:

  For PEinstallSP:

  /spdata/sys1/install/pssplpp

  For PEinstall:

  /usr/sys/inst.images

If you specify the -mount option, you will be prompted for:

- the installation image source directory

  The default is:

  For PEinstallSP:

  /spdata/sys1/install/pssplpp

  For PEinstall:

  /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 or **PEinstallSP** issues a **mkdir** command for the directory name specified, followed by a **chmod 777**.

Substep 3: Specifying the Fileset(s) to be Installed

When you are prompted for the name of the fileset you want to install, enter the appropriate file name as shown in the following table:

If you want to install:	Type this when prompted:
all the PE software	**all**
just the poe fileset	**ppe.poe**
just the pedb fileset	**ppe.pedb**
just the VT fileset	**ppe.vt**
just the Xprofiler fileset	**ppe.xprofiler**
just the pedocs fileset	**ppe.pedocs**

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

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

  This command installs both the usr and root portion of the fileset 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 be displayed 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 fileset(s) did not get successfully installed. When you correct the error(s), you can then go back and rerun the **PEinstallSP** or **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: Copying the Software to a Hard Disk for Installation Over a Network” on page 21, you created an installation image that you can use to replicate the installation of PE filesets 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 filesets does not require any special consideration. You may use SMIT or **installp** as described in “Method 1: Using the installp Command” on page 23. You can also set up a host list file, and run **installp** via **dsh** (for SP systems only), or **rsh**, and install the PE filesets on multiple nodes.

Proceeding to the Next Step

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

To determine which remaining steps you need to perform, refer to the following table:

If you installed POE:	**If you did not install POE:**
Proceed to: • “Step 4: Verifying the POE Installation” on page 30	Skip: • “Step 4: Verifying the POE Installation” on page 30 Proceed to: • “Step 5: Verifying the VT Installation” on page 32

Step 4: Verifying the POE Installation

Note: This step applies only if you have POE installed.

This step consists of testing the installation of POE, using the POE Installation Verification Program (IVP). This program is provided in **/usr/lpp/ppe.poe/samples/ivp**. For details about how the POE IVP works, see Appendix B, “POE Installation Verification Program” on page 57.

To run the POE IVP, follow the instructions below.

At the control workstation (or other home node):

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

ENTER **export LANG=C**

ENTER the following:

1. **cd /usr/lpp/ppe.poe/samples/ivp**
2. **./ivp.script**

- This runs an installation verification test that checks for successful execution of a message-passing program using two tasks on this node. The output should resemble the following:

```
Verifying the location of the Libraries
Verifying the existence of the Binaries
Partition Manager daemon /etc/pmdv2 is executable
POE files seem to be in order
Compiling the ivp sample program
Output files will be stored in directory /tmp/ivp15480
Creating host.list file for this node
Setting the required environment variables
Executing the parallel program with 2 tasks

POE IVP: running as task 0 on node pe03
POE IVP: running as task 1 on node pe03
POE IVP: there are 2 tasks running
POE IVP: task   1 received <POE IVP Message Passing Text>
POE IVP: all messages sent

Parallel program ivp.out return code was 0

Executing the parallel program with 2 tasks, threaded library

POE IVP_r: running as task 1 on node pe03
POE IVP_r: running as task 0 on node pe03
POE IVP_r: there are 2 tasks running
POE IVP_r: task   1 received <POE IVP Message Passing Text -
           Threaded Library>
POE IVP_r: all messages sent

Parallel program ivp_r.out return code was 0

If both tests return a return code of 0, POE IVP
is successful.  To test POWERparallel system message passing,
run the tests in ../samples/poetest.bw and poetest.cast
To test threaded message passing,
run the tests in ../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 59 for more information.

Proceeding to the Next Step

You have completed verifying the POE installation.

To determine whether you need to proceed to the next step, refer to the following table:

If you installed VT:	**If you did not install VT:**
Proceed to: • "Step 5: Verifying the VT Installation" on page 32	Do not proceed to: • "Step 5: Verifying the VT Installation" on page 32

Step 5: Verifying the VT Installation

Note: This step applies only if you have VT installed.

This step consists of:

- verifying that VT was installed successfully
- verifying that the VT trace generation mechanism is operating correctly

Substep 1: Verifying that VT Was Installed Successfully

To verify that VT was installed correctly, follow the instructions below.

At the control workstation (or other home node):

ENTER **export LANG=C**

ENTER **vt -tracefile /usr/lpp/ppe.vt/samples/vtsample.trc**

- This command starts VT.

PRESS **User Load Balance** under **Computation** on the VT View Selector Panel.

PRESS **Play** on the main control panel.

If the trace file plays to the end while updating the display, VT was installed successfully.

Substep 2: Verifying that the VT Trace Generation Mechanism is Operating Correctly

To verify that the VT trace generation mechanism is operating correctly, follow the instructions below *after installing POE*.

At the control workstation (or other home node):

COPY either **/usr/lpp/ppe.vt/samples/vtsample.f** or **/usr/lpp/ppe.vt/samples/vtsample.c** into another directory.

COMPILE the file you choose under POE.

- For C, enter **mpcc -g -o vtsample vtsample.c**
- For Fortran, enter **mpxlf -g -o vtsample vtsample.f**

RUN the file with the option **-tracelevel 9**.

- This results in a file called **vtsample.trc** being generated in the directory from which you ran it.

If you are successful in examining this trace file with VT as described previously, VT is operating correctly.

Using VT

Once you have completed verifying the installation of the VT fileset, you can use VT as described in *IBM Parallel Environment for AIX: Operation and Use, Volume 2*, SC28-1980.

Viewing the README File After Installation

Once you have installed the PE filesets (**ppe.poe**, **ppe.vt**, **ppe.pedb**, **ppe.xprofiler**, and **ppe.pedocs**), refer to the README file provided with each fileset for any additional installation or usage information. The README file is located in **/usr/lpp/**<*fileset*>**/README** as <*fileset*>**.README**.

Related Procedures

For information about other procedures related to PE installation, see Chapter 5, “Installation-Related Procedures” on page 35.

Chapter 5. Installation-Related Procedures

This chapter contains various procedures that are related to the PE installation procedure provided in Chapter 4, “Installing the PE Software” on page 17.

Removing an Installation Image

During the installation process, you may decide to remove a fileset from the system. If you have already installed it on a number of nodes, you may wish to use one of the “deinstallation” scripts provided with PE:

- **PEdeinstallSP**
- **PEdeinstall**

For detailed information about these scripts and how to run them, see “Deinstallation Scripts” on page 55.

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     2.4.0.0
```

This usually means there is an incompatibility in the Object Data Manager (ODM) 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 is one way to solve this problem (using POE as an example):

```
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

Message cataloging is used by all PE filesets so that messages can be displayed in languages other than English. Each fileset 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/en_US
/usr/lib/nls/msg/En_US
/usr/lib/nls/msg/C

Notes:

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

Enabling Fortran 90 Compiler Support

There are some restrictions and limitations in the MPI language bindings for Fortran 90.

For more information on the restrictions and implications of using MPI and the Fortran 90 compiler scripts, refer to the **/usr/lpp/ppe.poe/samples/mpif90/README.mpif90** file after POE has been installed.

Header Files

An MPI Fortran 90 language bindings header file (**mpif90.h**) is shipped as a sample file; if you have users with *existing programs* that depend on this file, you need to copy it to the usual location so that these users can continue to compile such programs with it.

With PE Version 2 Release 3, the **mpif.h** header file is formatted to work for both **mpxlf** and **mpxlf90** compilation. Users writing *new code* should use **mpif.h**.

Installing Communication Subsystem (CSS) Component of IBM Parallel System Support Programs for AIX After POE has been Installed

The CSS installation processing automatically establishes links to their message passing libraries in place of those supplied by POE. For your information, the following links are made when CSS is installed after POE:

This link:	**To:**
/usr/lpp/ppe.poe/lib/ip/libmpci.a	/usr/lpp/ssp/css/libip/libmpci.a
/usr/lpp/ppe.poe/lib/ip/libmpci_r.a	/usr/lpp/ssp/css/libip/libmpci_r.a
/usr/lpp/ppe.poe/lib/us/libmpci.a	/usr/lpp/ssp/css/libus/libmpci.a
/usr/lpp/ppe.poe/lib/us/libmpci_r.a	/usr/lpp/ssp/css/libus/libmpci_r.a

Servicing the POE Version of libc.a and libc_r.a

POE maintains copies of **libc.a** and **libc_r.a**, which are used to create the entry and exit points when a user's application is compiled with POE. In doing so, it creates a version of **libc.a** and **libc_r.a** for POE's use based on the system's copies.

As a result, when applying service that changes **/usr/lib/libc.a** and **/usr/lib/libc_r.a**, it is necessary to rebuild the POE version of **libc.a**. (In AIX Version 4.3, **libc_r.a** is a link to **libc.a**.) For this reason, POE ships a script called **makelibc** that automatically rebuilds its copies of **libc.a** and **libc_r.a**. Copies of **libc.a** and **libc_r.a** are also created in the **/usr/lpp/ppe.poe/lib/profiled** directory, if AIX Base Profiling Support is installed.

Any time you apply service that modifies **/usr/lib/libc.a** , you should run the **makelibc** script.

The **makelibc** script is in the **/usr/bin** path. You must run as **root** in order to run the script and rebuild the libraries.

AFS Installation Instructions

This section contains the instructions for tailoring the Parallel Operating Environment and pedb 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 in this section

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 which 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

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

1. Type **cd /usr/lpp/ppe.poe/samples/afs** to switch to the appropriate directory or copy the contents of the directory to a convenient location.
2. Issue 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.
3. Verify that the partition manager daemon, **pmdv2**, the home node partition manager, **poe**, and the parallel debugger, **pdbx**, are in **/usr/lpp/ppe.poe/bin**. Verify that the X-Windows parallel debugger, **pedb**, is in **/usr/lpp/ppe.pedb/bin**. If not, modify the **buildAFS** script.

   Note: Before completing the following step, ensure that you have the following amounts of available space in the current directory:

Component(s) Being Built	Total Available Space Required (in Megabytes)
poe, pmdv2, and pdbx	2
pedb	1.5

4. Type **buildAFS** to create new versions of poe, pdbx, pedb, and pmdv2 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.

   Note: If pedb is not currently installed, the steps to modify it for AFS execution are skipped.

5. Move poe, pdbx, and pmdv2 to their normal location in **/usr/lpp/ppe.poe/bin** on each node. Move pedb to its normal location in **/usr/lpp/ppe.pedb/bin**. You can rename the old versions in case they need to be restored. Make sure that they are made executable.

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

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

Chapter 6. How Installation of PE Alters Your System

This chapter describes how your system is altered when you install the various PE software filesets. It describes the directories and files created, the daemon processes created, and the links established by the installation process.

How Installing the poe Fileset Alters Your System

The poe fileset includes all the components of the Parallel Operating Environment, and consists of the:

- Partition Manager
- API subroutine libraries (message passing and collective communication)
- Program Marker Array
- System Status Array
- **pdbx** debugger
- Parallel profiling capability
- Parallel Utility Library
- Parallel compilation scripts
- VT trace file records

Installing this fileset as described in “Step 2: Performing the Initial Installation” on page 23, does the following:

1. If migrating from a previous version or release of POE, the following occurs:
   - Any installed files from Version 1 **poe** and **poe_pcf** filesets are removed. The filesets are changed to an OBSOLETE state in the SWVPD and ODM.
   - Man pages for commands that were shipped in **/usr/man/cat1** in Version 2 Release 1 **ppe.pedocs** fileset are removed and replaced with the copies shipped with **ppe.poe**.
2. Creates the directories and files detailed in the following table.

Table 4 (Page 1 of 4). poe Directories/Files Installed

Directory/File	Description
/usr/lib/nls/msg/en_US/mpci_err.cat /usr/lib/nls/msg/En_US/mpci_err.cat /usr/lib/nls/msg/C/mpci_err.cat	Message catalog for Message Passing Client Interface
/usr/lib/nls/msg/En_US/pedig.cat /usr/lib/nls/msg/en_US/pedig.cat /usr/lib/nls/msg/C/pedig.cat	Message catalog for VT trace functions
/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

Table 4 (Page 2 of 4). poe Directories/Files Installed

Directory/File	**Description**
/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/lib/nls/msg/En_US/poestat.cat /usr/lib/nls/msg/en_US/poestat.cat /usr/lib/nls/msg/C/poestat.cat	Message catalog for POESTAT
/usr/lpp/ppe.poe/bin/dbe	Parallel Debugger engine
/usr/lpp/ppe.poe/bin/digd	VT dig daemon
/usr/lpp/ppe.poe/bin/dfsnum.exp	Import file for building the **poeauth** command for DFS support
/usr/lpp/ppe.poe/bin/makedfs	Shell script for building the **poeauth** command for DFS support
/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	Shell script for compiling parallel C programs
/usr/lpp/ppe.poe/bin/mpcc_chkpt	Shell script for compiling C programs for checkpoint/restart capability
/usr/lpp/ppe.poe/bin/mpcc_r	Shell script for compiling threaded parallel C programs
/usr/lpp/ppe.poe/bin/mpCC	Shell script for compiling parallel C++ programs
/usr/lpp/ppe.poe/bin/mpCC_chkpt	Shell script for compiling C++ programs for checkpoint/restart capability
/usr/lpp/ppe.poe/bin/mpCC_r	Shell script for compiling threaded parallel C++ programs
/usr/lpp/ppe.poe/bin/mprcp	Shell script for distributing files
/usr/lpp/ppe.poe/bin/mpmkdir	Shell script for making directories
/usr/lpp/ppe.poe/bin/mpxlf	Shell script for compiling parallel Fortran programs
/usr/lpp/ppe.poe/bin/mpxlf90	Shell script for compiling parallel Fortran 90 programs
/usr/lpp/ppe.poe/bin/mpxlf_chkpt	Shell script for compiling Fortran programs for Checkpoint/Restart capability
/usr/lpp/ppe.poe/bin/mpxlf90_chkpt	Shell script for compiling Fortran 90 programs for checkpoint/restart capability
/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/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 RS/6000 nodes

Table 4 (Page 3 of 4). poe Directories/Files Installed

Directory/File	**Description**
/usr/lpp/ppe.poe/bin/PEdeinstallSP	Shell script to remove an installation of PE on SP nodes
/usr/lpp/ppe.poe/bin/PEinstall	Shell script to complete the installation process on IBM RS/6000 nodes
/usr/lpp/ppe.poe/bin/PEinstallSP	Shell script for installing PE on SP nodes
/usr/lpp/ppe.poe/bin/pmadjpri	Dispatching priority adjustment daemon
/usr/lpp/ppe.poe/bin/pmarray	The Program Marker Array program
/usr/lpp/ppe.poe/bin/pmdv2	An Internet daemon process that runs on each of your processor nodes
/usr/lpp/ppe.poe/bin/poe	Partition Manager executable
/usr/lpp/ppe.poe/bin/poe_dce_shr.o	Shared object used at run time to resolve DFS/DCE functions
/usr/lpp/ppe.poe/bin/poeauth.o	Object version of **poeauth** command for DFS support
/usr/lpp/ppe.poe/bin/poekill	Shell script for terminating all POE started tasks
/usr/lpp/ppe.poe/bin/poestat	The System Status Array program
/usr/lpp/ppe.poe/include	Directory of header files containing declarations used by other installed files
/usr/lpp/ppe.poe/lib/ip/libmpci.a	Message Passing Client Interface library (IBM RS/6000 network cluster only)
/usr/lpp/ppe.poe/lib/ip/libmpci_r.a	Message Passing Client Interface library (IBM RS/6000 network cluster only) in a threaded environment
/usr/lpp/ppe.poe/lib/libppe.a	Archive library containing subroutines for POE and VT
/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 threaded environment
/usr/lpp/ppe.poe/lib/libvtd.a	VT tracing library
/usr/lpp/ppe.poe/lib/libvtd_r.a	VT tracing library in a threaded environment
/usr/lpp/ppe.poe/lib/HPSOclk_chkpt.o	Shared object used at run time to resolve checkpoint/restart clock functions
/usr/lpp/ppe.poe/lib/pm_replaced.o	POE exit object code
/usr/lpp/ppe.poe/README/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/samples/mpif90/mpxlf90	Shell script for compiling parallel Fortran 90 programs
/usr/lpp/ppe.poe/samples/mpif90/mpxlf90_r	Shell script for compiling threaded parallel Fortran 90 programs
/usr/lpp/ppe.vt/bin/dig	VT Statistics Collector daemon
/usr/lpp/ppe.vt/bin/rc.vtdig	Shell script that starts the daemon
/usr/man/cat1	Directory containing man page files for PE commands
/usr/lpp/ppe.poe/include/thread/mpif.h	Header file for compiling threaded MPI Fortran applications

Table 4 (Page 4 of 4). poe Directories/Files Installed

Directory/File	**Description**
/usr/lpp/ppe.poe/samples/swtbl	Directory containing sample code for running User Space POE jobs without LoadLeveler

3. When the **installp** command successfully restores the 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 following symbolic links:

This link:	**To:**
/etc/digd	/usr/etc/digd
/etc/pmdv2	/usr/etc/pmdv2
/etc/rc.vtdig	/usr/lpp/ppe.vt/bin/rc.vtdig
/usr/bin/mpcc	/usr/lpp/ppe.poe/bin/mpcc
/usr/bin/mpcc_chkpt	/usr/lpp/ppe.poe/bin/mpcc_chkpt
/usr/bin/mpcc_r	/usr/lpp/ppe.poe/bin/mpcc_r
/usr/bin/mpCC	/usr/lpp/ppe.poe/bin/mpCC
/usr/bin/mpCC_chkpt	/usr/lpp/ppe.poe/bin/mpCC_chkpt
/usr/bin/mpCC_r	/usr/lpp/ppe.poe/bin/mpCC_r
/usr/bin/makedfs	/usr/lpp/ppe.poe/bin/makedfs
/usr/bin/mpamddir	/usr/lpp/ppe.poe/bin/mpamddir
/usr/bin/mpmkdir	/usr/lpp/ppe.poe/bin/mpmkdir
/usr/bin/mprcp	/usr/lpp/ppe.poe/bin/mprcp
/usr/bin/mpxlf	/usr/lpp/ppe.poe/bin/mpxlf
/usr/bin/mpxlf_chkpt	/usr/lpp/ppe.poe/bin/mpxlf_chkpt
/usr/bin/mpxlf_r	/usr/lpp/ppe.poe/bin/mpxlf_r
/usr/bin/mpxlf90	/usr/lpp/ppe.poe/bin/mpxlf90
/usr/bin/mpxlf90_chkpt	/usr/lpp/ppe.poe/bin/mpxlf90_chkpt
/usr/bin/mpxlf90_r	/usr/lpp/ppe.poe/bin/mpxlf90_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/pdbx	/usr/lpp/pdbx/bin/pdbx
/usr/bin/pmarray	/usr/lpp/ppe.poe/bin/pmarray
/usr/bin/pmdadjpri	/usr/lib/ppe.poe/bin/pmdadjpri
/usr/bin/poe	/usr/lpp/poe/bin/poe
/usr/bin/poekill	/usr/lpp/ppe.poe/bin/poekill
/usr/bin/poestat	/usr/lpp/ppe.poe/bin/poestat
/usr/etc/dbe	/usr/lpp/ppe.poe/bin/dbe
/usr/etc/digd	/usr/lpp/ppe.vt/bin/digd
/usr/etc/pmdv2	/usr/lpp/ppe.poe/bin/pmdv2

This link:	To:
/usr/lib/X11/app-defaults/PMarray	/usr/lpp/ppe.poe/samples/marker/PMarray.ad
/usr/lpp/poe/lib/ip/libmpci.a	/usr/lpp/ssp/css/libip/libmpci.a (SP only)
/usr/lpp/poe/lib/us/libmpci.a	/usr/lpp/ssp/css/libus/libmpci.a (SP only)
/usr/sbin/PEdeinstall	/usr/lpp/poe/bin/PEdeinstall
/usr/sbin/PEdeinstallSP	/usr/lpp/poe/bin/PEdeinstallSP
/usr/sbin/PEinstall	/usr/lpp/ppe.poe/bin/PEinstall
/usr/sbin/PEinstallSP	/usr/lpp/ppe.poe/bin/PEinstallSP

4. 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:

   ```
   /usr/lpp/ppe.poe/bin/mpcc
   /usr/lpp/ppe.poe/bin/mpCC
   /usr/lpp/ppe.poe/bin/mpxlf
   /usr/lpp/ppe.poe/bin/mpxlf90
   /usr/lpp/ppe.poe/bin/mpcc_r
   /usr/lpp/ppe.poe/bin/mpCC_r
   /usr/lpp/ppe.poe/bin/mpxlf_r
   /usr/lpp/ppe.poe/bin/mpxlf90_r
   /usr/lpp/ppe.poe/bin/mpcc_chkpt
   /usr/lpp/ppe.poe/bin/mpCC_chkpt
   /usr/lpp/ppe.poe/bin/mpxlf_chkpt
   /usr/lpp/ppe.poe/bin/mpxlf90_chkpt
   /usr/lpp/ppe.poe/bin/mpamddir
   /usr/lpp/ppe.poe/lib/poe.cfg
   ```

   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 occurs:

1. The file **/etc/services** is modified in the following manner:
   - If no entry for the pmv2 service is found, an entry is added using port 6125/tcp.
   - If an entry exists for the pmv2 service which uses port 6125/tcp, no change is made to the **/etc/services** file.
   - | If one of the following is true:
     a. A pmv2 entry exists for a port other than 6125/tcp
     b. A 6125/tcp entry exists for a service other than pmv2

     the user receives a warning and is instructed to correct the problem before running poe. When this warning is received, the user must manually update the **/etc/services** file in such a way as to ensure that the port number for the pmv2 service is the same on all machines which could run POE Version 2.
2. The file **/etc/inetd.conf** is modified.

An entry for the pmv2 service which spawns the /etc/pmdv2 daemon is created if no pmv2 entry exists.

3. **inetd** is refreshed.
4. POE's version of **libc.a** is created by extracting and replacing **shr.o**. The POE version of **libc_r.a** is a symbolic link to **libc.a**.

   A version of **libc.a** (and a link for **libc_r.a**) are also created in the **/usr/lpp/ppe.poe/lib/profiled** directory, if AIX Base Profiling Support is installed.
5. The file **/etc/inittab** is modified using **mkitab** unless an entry already exists so that the **/etc/rc.vtdig** shell script runs when the system is restarted to start the VT tracing daemon called **/etc/digd**.
6. The dig daemon is started.
7. The existence of IP and US versions of **libmpci.a** and **libmpci_r.a** from CSS is verified.
8. Executable versions of **mcp**, **mcpgath**, and **mcpscat** are created.
9. For DFS systems, the executable version of **poeauth** is created.

How Installing the pedb Fileset Alters Your System

Note: In PE, Version 2 Release 2, **xpdbx** was renamed to **pedb**.

The **pedb** fileset is the X-Windows interface of the PE Debugging Facility. Installing it, as described in "Step 2: Performing the Initial Installation" on page 23, does the following:

1. If migrating from a previous version or release of **xpdbx**, the files installed from the Version 1 **xpdbx** fileset or Version 2 **ppe.xpdbx** fileset are removed. These filesets are changed to an OBSOLETE state in the SWVPD and ODM.
2. Creates the files described in the following table.

Table 5. pedb Directories/Files Installed

Directories/File	**Description**
/usr/lib/nls/msg/En_US/pedb.cat /usr/lib/nls/msg/en_US/pedb.cat /usr/lib/nls/msg/C/pedb.cat	Message Catalog for pedb
/usr/lpp/ppe.pedb/bin	Directory containing scripts called by debugger array visualization feature options
/usr/lpp/ppe.pedb/bin/pedb	X-Windows debugger executable
/usr/lpp/ppe.pedb/bitmaps	Directory containing dialog bitmaps
/usr/lpp/ppe.pedb/defaults/Pedb.ad	Defaults file for X-Windows and Motif resource variables
/usr/lpp/ppe.pedb/README/pedb.README	Installation README file
/usr/lpp/ppe.pedb/samples	Directory containing optional function, such as Data Explorer Visualization Tools. See *IBM Parallel Environment for AIX: Operation and Use, Volume 2.*

3. The following symbolic links are made during the ppe.pedb fileset installation process:

This link:	To:
/usr/bin/pedb	/usr/lpp/ppe.pedb/bin/pedb
/usr/bin/xpdbx	/usr/lpp/ppe.pedb/bin/pedb
/usr/lib/X11/app-defaults/Pedb	/usr/lpp/ppe.pedb/defaults/Pedb.ad

How Installing the VT Fileset Alters Your System

The VT fileset is the Visualization Tool. Installing it as described in “Step 2: Performing the Initial Installation” on page 23 does the following:

1. If migrating from Version 1 of VT, the files installed from the VT fileset are removed. The fileset is changed to an OBSOLETE state in the SWVPD and ODM.
2. Creates the directories and files detailed in the following table:

Table 6 (Page 1 of 2). VT Directories/Files Installed

Directory/File	Description
/usr/lib/nls/msg/En_US/pevt.cat /usr/lib/nls/msg/en_US/pevt.cat /usr/lib/nls/msg/C/pevt.cat	Message Catalog for VT
/usr/lpp/ppe.vt/Bitmaps	Directory containing icon Bitmaps for each VT View. These icons appear in the View Selector window.
/usr/lpp/ppe.vt/bin/digq	Child process run when the Resource Manager is not used. Determines what nodes are available to monitor.
/usr/lpp/ppe.vt/bin/dug	Program used by VT during performance monitoring to communicate with the dig Statistics Collector daemon
/usr/lpp/ppe.vt/bin/vt	The VT program
/usr/lpp/ppe.vt/Config	Directory of configuration files used by the VT control code to initialize the Views for trace visualization
/usr/lpp/ppe.vt/ConfigPM	Configuration files used by the VT control code to initialize the Views for performance monitoring
/usr/lpp/ppe.vt/CPBitmaps	Directory of user interface bitmaps used to create the Trace Visualization window
/usr/lpp/ppe.vt/include/pevt.cat	Message catalog for VT
/usr/lpp/ppe.vt/include/VT_mpi.h	VT MPI header file
/usr/lpp/ppe.vt/include/VT_trc.h	VT trace header file
/usr/lpp/ppe.vt/README/vt.README	Installation README file
/usr/lpp/ppe.vt/Resource/VT.ad	Defaults file for X-Windows and Motif resource variables

Table 6 (Page 2 of 2). VT Directories/Files Installed

Directory/File	Description
/usr/lpp/ppe.vt/samples	Directory containing sample programs
/usr/lpp/ppe.vt/ViewSelect/System.PMView	Layout definition of View Selector Window for performance monitoring
/usr/lpp/ppe.vt/ViewSelect/System.PView	Layout definition of View Selector Window for trace visualization
/usr/lpp/ppe.vt/util	Trace file view utility files

3. The following changes are made as part of the VT fileset installation process:

   - Creates the **vt**, **dug**, **dig**, and **digq** executable files by running the bind process against the object files and the library for VT.
   - Creates the following symbolic links:

This Link:	To:
/usr/bin/digq	/usr/lpp/ppe.vt/bin/digq
/usr/bin/dug	/usr/lpp/ppe.vt/bin/dug
/usr/bin/vt	/usr/lpp/ppe.vt/bin/vt
/usr/include/VT_trc.h	/usr/lpp/ppe.vt/include/VT_trc.h
/usr/include/VT_mpi.h	/usr/lpp/ppe.vt/include/VT_mpi.h
/usr/lib/X11/app-defaults/Vt	/usr/lpp/ppe.vt/Resource/VT.ad

How Installing the Xprofiler Fileset Alters Your System

The Xprofiler fileset is the Xprofiler X-Windows Performance Profiler. Installing it as described in “Step 2: Performing the Initial Installation” on page 23 creates the directories and files detailed in the following table:

Table 7. Xprofiler Directories and Files Installed

Directory or File	Description
/usr/lib/nls/msg/En_US/xprofiler.cat /usr/lib/nls/msg/en_US/xprofiler.cat /usr/lib/nls/msg/C/xprofiler.cat	Message catalog for Xprofiler
/usr/lpp/ppe.xprofiler/defaults/Xprofiler.ad	Defaults file for X-Windows and Motif resource variables
/usr/lpp/ppe.xprofiler/bin/.startup_script	Startup script for Xprofiler
/usr/lpp/ppe.xprofiler/bin/xprofiler	Xprofiler executable
/usr/lpp/ppe.xprofiler/help/en_US/xprofiler.sdl /usr/lpp/ppe.xprofiler/help/en_US/xprofiler_msg.sdl /usr/lpp/ppe.xprofiler/help/en_US/graphics	Online help
/usr/lpp/ppe.xprofiler/README/xprofiler.README	Installation README file
/usr/lpp/ppe.xprofiler/samples	Directory containing sample programs

The following symbolic link is made during the ppe.xprofiler fileset installation process:

This link:	To:
/usr/lib/X11/app-defaults/Xprofiler	/usr/lpp/ppe.xprofiler/defaults/Xprofiler.ad
/usr/bin/xprofiler	/usr/lpp/ppe.xprofiler/bin.startup_script

How Installing the pedocs Fileset Alters Your System

The **pedocs** fileset consists of man page files and the PE documentation in HTML and PDF format. Installing it as described in “Step 2: Performing the Initial Installation” on page 23 creates the directories and files detailed in the following table.

Please note that when you migrate from Version 1, the files installed from the **pedocs** fileset are removed. The fileset is changed to an OBSOLETE state in the SWVPD and ODM.

Table 8. pedocs Directories/Files Installed

Directory/File	Description
/usr/lpp/ppe.pedocs/graphics	Directory containing HTML graphics files
/usr/lpp/ppe.pedocs/html	Directory containing HTML versions of the online books
/usr/lpp/ppe.pedocs/html/pebooks.html	Sample index file
/usr/lpp/ppe.pedocs/pdf	Directory containing PDF versions of the online books
/usr/lpp/ppe.pedocs/README/pedocs.README	File describing available PE documentation and how to access it
/usr/man/cat3	Directory containing man page files for API message-passing subroutines

Installing the ppe.pedocs HTML Files

The **ppe.pedocs** fileset includes HTML files that contain online versions of the PE publications. Once you have installed the **ppe.pedocs** fileset, the PE HTML publications will be located at **/usr/lpp/ppe.pedocs/html**. Because other parts of PE link to the HTML publications, these files should not be moved from the **/usr/lpp/ppe.pedocs/html** directory.

A sample index file, **/usr/lpp/ppe.pedocs/html/pebooks.html** has also been provided. It shows you how to set up a single launching point from which users can access all of the online books.

Chapter 7. File Formats

This chapter describes formats for PE files created and edited by system administrators (in **/etc**).

/etc/poe.priority Dispatching Adjustment Parameters

Certain applications can benefit from enhanced dispatching priority during execution. POE provides a service for periodically adjusting the dispatching limits of a user's task between set limits. The service is specified by entries in the file **/etc/poe.priority**, as described below. Each record in the file associates a set of dispatching parameters (high and low priorities, duty factor, and adjustment period) with each user authorized to use this service. The user specifies which service entry to use by setting the MP_PRIORITY environment variable prior to invoking POE. There is no command-line equivalent.

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 user

classname name assigned to class - user selects class with POE environment variable MP_PRIORITY

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 portion of the cycle at which the job is at hipriority (percent)

period length of adjustment cycle, in seconds

Notes:

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 127.
6. If the **hipriority** value is less than (more favored than) the priority of the SP 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 priority adjustment process allows programs using the User Space access to the SP Switch 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. Each node may have a different priority file.
10. The primary performance enhancement is achieved when the user's application can run with minimal interference from the standard AIX daemons running on each node. This is achieved when the user's application is scheduled with a fixed priority more favored than the daemons, which typically run with a priority setting of 60.
11. Status and error messages generated during the priority adjustment process are written to the file **/tmp/pmadjpri.log**. Also, any error or diagnostic information from POE's invocation of the priority adjustment function will be recorded in the partition manager log (controlled by the **POE MP_PMDLOG** environment variable and **-pmdlog flag**.)

Examples:

```
userfoo   super     12    64    95    10
userfoo   sponge    64    120   50    30
edgar     default   20    64    95    10
edgar     normal    60    60    50    20
```

Using the /etc/poe.limits File

The optional **/etc/poe.limits** file enables the system administrator to override the default values for certain POE environment variables.

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 **MP_BUFFER_MEM**, **MP_AUTH**, and **MP_USE_LL**.

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

Handling of /etc/poe.limits by the Partition Manager Daemon (pmdv2)

If the **/etc/poe.limits** file has been set up on a particular node, the Partition Manager daemon (**pmdv2**) on that node:

1. compares the values specified in the **/etc/poe.limits** file against the environmment 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**, then **pmdv2** resets the value to that in **/etc/poe.limits**.

   MP_AUTH

   If the value in the environment is *different* from the value specified in **/etc/poe.limits**, then **pmdv2** resets the value to that in **/etc/poe.limits**.

   | **MP_USE_LL**

   | If the value in the file is **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 any entries either with unsupported objects to the left of the equal sign or with invalid (non-numeric for **MP_BUFFER_MEM**, or not **AIX** or **DFS** for **MP_AUTH**) values to the right, **pmdv2** flags these entries in the **pmdlog** for that node. **pmdv2** also uses the **pmdlog** to indicate when a *supported_object* has been set or reset in the environment.

Appendix A. Syntax of Commands for Running Installation and Deinstallation Scripts

This appendix explains the syntax of the commands for running the installation and deinstallation scripts provided with PE.

Installation Scripts

PE provides the following two installation scripts, explained in the sections below:

- For the SP: **PEinstallSP**
- For an IBM RS/6000: **PEinstall**

PEinstallSP

You can use the **PEinstallSP** script to install any of the filesets on SP nodes through the use of the Distributed Shell (**dsh**). In order to run the **PEinstallSP** script, you must first set up a host list file of all nodes on which you want a particular fileset installed. The **PEinstallSP** 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.

Since **PEinstallSP** makes extensive use of **dsh**, it can only be run on SP nodes. Also, you must ensure that you have executed the **k4init** command to obtain Kerberos authentication for accessing your nodes.

The **PEinstallSP** script has one required parameter and three optional parameters. The syntax is:

PEinstallSP *image_name* [*host_list_file*] [**-f** *fanout_value*] [**-copy | -mount**]

Where:

image_name is *required* and represents the name of the file containing the installp image of which the fileset is a part.

host_list_file is *optional*, as the file containing the list of nodes on which you want to install the fileset. 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.

fanout_value is *optional*, as the **dsh** fanout value. This is the number of nodes for concurrent command execution. The default is 16. See the **dsh** command in *IBM Parallel System Support Programs for AIX: Command and Technical Reference*, GA22-7351 for more information.

You can specify either **-copy** or **-mount** to instruct **PEinstallSP** to copy or mount the installation image to each node. The default is **-copy**. For more information on copying or mounting installation images, refer to “Copying the Installation Image” on page 54 or “Mounting the Installation Image” on page 54.

PEinstall

You can use the **PEinstall** script to install any of the PE filesets on IBM RS/6000 nodes using the Remote Shell **(rsh)**.

To run the **PEinstall** script, you must first set up a host list file of all nodes on which you want to install a particular fileset, and 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.

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*, and represents the name of the file containing the installp image of which the PE fileset is a part.

host_list_file is *optional*, as the file containing the list of nodes on which you want to install the fileset. 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 **PEinstallSP** or **PEinstall** to copy the named image to each node using **rcp**. You are prompted for the following information when **-copy** is specified (or defaulted):

- The installation image source directory. The default is **/spdata/sys1/install/pssplpp** for **PEinstallSP** and **/usr/sys/inst.images** for **PEinstall**.
- The installation image destination directory which is used for all nodes in the node list. The default is **/spdata/sys1/install/pssplpp** for **PEinstallSP** and **/usr/sys/inst.images** for **PEinstall**.

  Note: To have the image copied to different directories, invoke **PEinstallSP** or **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 **PEinstallSP** or **PEinstall** to mount the named image to each node using **dsh** (for **PEinstallSP**) or **rsh** (for **PEinstall**). You are prompted for the following information when **-mount** is specified:

- The installation image source directory. The default is **/spdata/sys1/install/pssplpp** for **PEinstallSP** and **/usr/sys/inst.images** for **PEinstall**.

- The remote node mount point directory. This is used for all nodes in the node list. The default is **/mnt**.

  Note: To have the image mounted to different directories, invoke **PEinstallSP** or **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, **PEinstallSP** or **PEinstall** also asks if you want it to create the remote mount directory. If your remote mount directory already exists, answer no to this prompt.

PEinstallSP or **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 **PEinstallSP** or **PEinstall**.

Deinstallation Scripts

When you install a PE fileset, you do so first on a single node (or the control workstation), and then propagate the installation image to the additional nodes in your system. When you remove a fileset completely from your system, you do the opposite:

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

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

PEdeinstallSP issues the proper **installp** command using the Distributed Shell (**dsh**).

Note:

> Before you execute the **PEdeinstallSP**, ensure that you have previously obtained Kerberos authentication (via **k4init**) for accessing your nodes.

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

The scripts have the following syntax:

PEdeinstallSP *image_name* [*host_list_file*] [**-f** *fanout_value*]

PEdeinstall *image_name* [*host_list_file*]

Where:

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

host_list_file is *optional*, as the file containing the list of nodes that you want the image removed from. 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.

fanout_value is *optional*, as the **dsh** fanout value. This is the number of nodes for concurrent command execution. The default is 16. See the **dsh** command in *IBM Parallel System Support Programs for AIX: Command and Technical Reference*, GA22-7351 for more information.

For each node, **PEdeinstallSP** or **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 install on a node, an error message is issued and logged in a file named **PEnode.log** in the current working directory, and the install removals continue for the remaining nodes.

Appendix B. POE Installation Verification Program

To enable you to test the installation of POE, an Installation Verification Program (IVP) is provided in **/usr/lpp/ppe.poe/samples/ivp**. The IVP tests to see if these conditions are true:

- A non-root user ID is properly authorized in **/etc/hosts.equiv** or **.rhosts**.
- There is access to a C compiler. (The C compiler must be installed.)

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

- Ensures that:
  - The following MPI libraries are there or linked for IBM RS/6000 SP systems:

    /usr/lpp/ppe.poe/lib/ip/libmpci.a

    /usr/lpp/ppe.poe/lib/ip/libmpci_r.a

    /usr/lpp/ppe.poe/lib/us/libmpci.a

    /usr/lpp/ppe.poe/lib/us/libmpci_r.a
  - **poe**, **pmdv2**, **mpcc**, and **mpcc_r** are there and are executable.
  - **mpcc** and **mpcc_r** scripts are in the path.
  - The file **/etc/services** contains an entry for **pmv2**, the Partition Manager daemon.
  - The file **/etc/inetd.conf** contains an entry for **pmv2**, and the daemon it points to is executable.
- Creates a work directory in **/tmp/ivp** <*pid*> to compile and run sample programs.

  Note: *pid* is the process id.
- Compiles sample programs.
- Creates a **host.list** file with local host names listed twice.
- Runs sample programs using IP protocol to two tasks.
- Removes all files from **/tmp** as well as the temporary directory.
- Checks for the dbx **bos.adt.debug** fileset, for parallel debuggers.

For instructions on running the POE IVP, see “Step 4: Verifying the POE Installation” on page 30.

Appendix C. Using Additional POE Sample Applications

This appendix explains the POE sample applications for doing the following:

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

Bandwidth Measurement Test Sample

The purpose of this sample is to measure the point-to-point bandwidth between two tasks . The sample code is in the directory called **/usr/lpp/ppe.poe/samples/poetest.bw**. This directory contains a test code called **bw.f**, which does a point-to-point bandwidth measurement test. The code needs only 2 nodes to run.

You should have the following files:

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

bw.f Sample application Fortran source file.

bw.run Script for compiling application.

rtc.s Source for creating **rtc.o** file needed for building and running sample application **makefile** to create **rtc.o**.

makertc **makefile** to create **rtc.o**

makefile **makefile** for creating sample application.

Note: 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 non-root user and perform the following steps:
   a. Type **cd /usr/lpp/ppe.poe/samples/poetest.bw** to switch to the appropriate directory. If this directory is not writable to the user, the user must copy the needed files to a directory that is writable so they can build the executable files.
   b. Use the **makertc** script to create a copy of the **rtc.o** object file, necessary for building the **bw** executable.
   c. Type **make** to invoke the **makefile**, which compiles **bw.f**, includes **rtc.o**, to create the **bw** executable.
2. Create a file named **host.list** with two entries, one per line, listing the two nodes on which the executable is to be run.
3. Type **./bw.run** [*css_library*] [*SP_name*]

   where the optional parameters are:

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

SP_name is the name of the control workstation for the SP system on which the test is to be run, required if css_library is **us**. If *SP_name* is not supplied, the environment variable **SP_NAME** is used. When running on an SP and submitting from a non-SP node, the **ssp.clients** fileset must be installed on the submitting node. Refer “When to Install ssp.clients (SP Resource Manager)” on page 19 for further information.

Note: The **bw.run** script runs the **bw** executable from the current working directory.

4. Check output on the following screen. The output should finish in about one minute, on the SP system using the User_Space message passing library. The execution time for IP will be 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
```

For the IBM RS/6000 SP with the SP Switch , the bandwidth should be around 32 MB/Sec.

For IP over Ethernet, the bandwidth should be around 1 MB/Sec.

Broadcast Test Sample

The purpose of this sample is to perform a broadcast from task 0 to all of the rest of the nodes in the partition. It is located 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 applications, which is the same information presented here.

bcast.f Sample application FORTRAN source.

makefile makefile for compiling application.

bcast.run. Script for running application.

Note: The FORTRAN compiler must be available.

Verification Steps

Follow these steps to verify your system:

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

   a. Type **cd /usr/lpp/ppe.poe/samples/poetest.cast** to switch to the appropriate directory. If this directory is not writable to the user, the user must copy the needed files to a directory that is writable so they can build the executable files.

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

2. Create a file of execution nodes, named **host.list** with one entry per line.
3. Make sure that the nodes specified in the **host.list** file are initialized for the requested type of message passing.
4. Type **/bcast.run** *ntasks* [*css_library*] [*SP_name*]

   where the required parameter is:

   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 parameters are:

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

   SP_name is the name of the control workstation for the SP system on which the test is to be run, required if *css_library* is **us**. If *SP_name* is not supplied, the environment variable **SP_NAME** is used. When running on an SP and submitting from a non-SP node, the **ssp.clients** fileset must be installed on the submitting node. Refer "When to Install ssp.clients (SP Resource Manager)" on page 19 for further information.

   Note: The **bcast.run** script runs the **bcast** executable from the current working directory.

5. Check the output on the following screen. The output should finish in about one minute if your system does not have more than 64 SP nodes. The actual response time depends on your LAN traffic.

```
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
```

   Note:

   The order of these lines is unpredictable.

   If the test failed, 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 will appear.

MPI Threads Sample Program

Two sample programs have been provided to illustrate the use of the MPI message passing library with user-created threads. They are located in the **/usr/lpp/ppe.poe/samples/threads** directory.

The following files are located in this directory:

README.threads README file containing instructions on running the sample programs.

threaded_ring.c Source for sample program for testing threaded MPI library with user threads.

interrupt_ring.c Sample program for testing the threaded MPI library with a user signal handler.

makefile makefile for compiling the threaded sample programs.

threads.run script for running user threads sample program, **threaded_ring**

interrupt.run script for running threads sample program with a signal handler, **interrupt_ring**

Note: These sample programs require the C compiler to be available.

Verification Steps

Follow these steps to run the threaded sample applications on your system:

1. Create the executables. To do so, log in as a non-root user, and do the following:

   a. Type **cd /usr/lpp/ppe.poe/samples/threads** to switch to the appropriate directory. If you do not have write access to this directory, you will need to copy the needed files to a directory that is writable so that you can build the executable files.

   b. Type **make** to invoke the makefile, which compiles both source programs to create two executables, **threaded_ring** and **interrupt_ring**.

2. Create a file of execution nodes, named **host.list**, with one entry per line.

   Note: This file must contain a minimum of two nodes.

3. Make sure that the nodes specified in the **host.list** file are initialized for the requested type of message passing.

4. To test the threaded MPI library with user threads, type:

   threads.run [**{ip,us}** [*SP_partition*]]

   where the optional parameters are as follows:

   {ip,us} specifies the library to be used. 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.

   SP_partition is the name of an SP partition if running User Space and not logged onto a node in the partition.

   Note: These instructions assume that the current directory is mounted (or automounted) on all execution nodes.

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

5. To test the threaded MPI library with user signal handler type:

   interrupt.run [**{ip,us}** [*SP_partition*]]

   where the optional parameters are as follows:

   {ip,us} specifies the library to be used. 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.

   SP_partition is the name of an SP partition if running User Space and not logged onto a node in the partition.

   Note: These instructions assume that the current directory is mounted (or automounted) on all execution nodes.

   The program issues the following output.

   Note: The order of the outputs from task 0 and task 1 may be different from what appears here; you may receive fewer than the number of "Receiver got..." messages than shown here.

```
0:MPI Initialization Complete
 0:Receiver got thread condition signal
 1:Receiver got thread condition signal
 1:Receiver got thread condition signal
 0:Receiver got thread condition signal
 1:Receiver got thread condition signal
 0:Receiver got thread condition signal
 1:Receiver got thread condition signal
 1:Receiver got thread condition signal
 1:Receiver got thread condition signal
 1:Receiver got thread condition signal
 1:Receiver got thread condition signal
 1:Receiver got thread condition signal
 1:Receiver got thread condition signal
 0:TEST COMPLETE
```

LAPI Sample Programs

There are a number of sample programs provided to illustrate the use of the Low-level Applications Programming Interface (LAPI). Since LAPI runs only with the User Space communications subsystem, it is installed and runs as part of the Parallel System Support Programs (PSSP).

The LAPI sample programs are provided in the **/usr/lpp/ssp/css/samples/lapi** directory. In that directory, refer to the file README.LAPI, which contains instructions on how to compile and execute the sample programs.

Note: These sample programs require the C and Fortran compilers.

Glossary of Terms and Abbreviations

This glossary includes terms and definitions from:

- The *Dictionary of Computing*, New York: McGraw-Hill, 1994.
- The *American National Standard Dictionary for Information Systems*, ANSI X3.172-1990, copyright 1990 by the American National Standards Institute (ANSI). Copies can be purchased from the American National Standards Institute, 1430 Broadway, New York, New York 10018. Definitions are identified by the symbol (A) after the definition.
- The *ANSI/EIA Standard - 440A: Fiber Optic Terminology*, copyright 1989 by the Electronics Industries Association (EIA). Copies can be purchased from the Electronic Industries Association, 2001 Pennsylvania Avenue N.W., Washington, D.C. 20006. Definitions are identified by the symbol (E) after the definition.
- The *Information Technology Vocabulary* developed by Subcommittee 1, Joint Technical Committee 1, of the International Organization for Standardization and the International Electrotechnical Commission (ISO/IEC JTC1/SC1). Definitions of published parts of this vocabulary are identified by the symbol (I) after the definition; definitions taken from draft international standards, committee drafts, and working papers being developed by ISO/IEC JTC1/SC1 are identified by the symbol (T) after the definition, indicating that final agreement has not yet been reached among the participating National Bodies of SC1.

This section contains some of the terms that are commonly used in the Parallel Environment books and in this book in particular.

IBM is grateful to the American National Standards Institute (ANSI) for permission to reprint its definitions from the American National Standard *Vocabulary for Information Processing* (Copyright 1970 by American National Standards Institute, Incorporated), which was prepared by Subcommittee X3K5 on Terminology and Glossary of the American National Standards Committee X3. ANSI definitions are preceded by an asterisk (*).

Other definitions in this glossary are taken from *IBM Vocabulary for Data Processing, Telecommunications, and Office Systems* (GC20-1699).

A

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.

AIXwindows Environment/6000. A graphical user interface (GUI) for the IBM RS/6000. It has the following components:

- A graphical user interface and toolkit based on OSF/Motif
- Enhanced X-Windows, an enhanced version of the MIT X Window System
- Graphics Library (GL), a graphical interface library for the applications programmer which is compatible with Silicon Graphics' GL interface.

API. Application Programming Interface.

application. The use to which a data processing system is put; for example, topayroll 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.

B

bandwidth. The total available bit rate of a digital channel.

blocking operation. An operation which 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 in which 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 ANSI standards committee for the C language in 1984 and by Uniforum in 1983.

C++. A general purpose programming language, based on C, which 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.

call arc. The representation of a call between two functions within the Xprofiler function call tree. It appears as a solid line between the two functions. The arrowhead indicates the direction of the call; the function it points to is the one that receives the call. The function making the call is known as the *caller*, while the function receiving the call is known as the *callee*.

chaotic relaxation. An iterative relaxation method which uses a combination of the Gauss-Seidel and Jacobi-Seidel methods. The array of discrete values is divided into sub-regions which can be operated on in parallel. The sub-region boundaries are calculated using Jacobi-Seidel, whereas the sub-region interiors are calculated using Gauss-Seidel. 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. It typically provides excellent price/performance.

collective communication. A communication operation which 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 know as *command aliases*.

| **Communication Subsystem (CSS)**. A component of the Parallel System Support Programs that provides software support for the SP Switch. CSS provides two protocols: IP (Internet Protocol) for LAN-based communication and US (user space) as a message passing interface that is optimized for performance over the switch. See also *Internet Protocol* and *User Space.*

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.

control workstation. A workstation attached to the IBM RS/6000 SP SP that serves as a single point of control allowing the administrator or operator to monitor and manage the system using Parallel System Support Programs.

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

core file. A file which 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 either of the PE parallel debuggers, control of the parallel program and the display of its data can be limited to a subset of the tasks that belong 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 decimals.

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.

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

Ethernet. Ethernet is the standard hardware for TCP/IP LANs in the UNIX marketplace. It is a 10 megabit per second baseband type network that uses the contention based CSMA/CD (collision detect) media access method.

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, then no set of circumstances can cause any thread to wait indefinitely for access to the lock.

FDDI. Fiber distributed data interface (100 Mbit/s fiber optic LAN).

file system. In the AIX operating 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 function while updates correct an error in, or enhance, a previously installed product. 2) One or more separately installable, logically grouped units in an installation package. See also *Licensed Program Product* 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. It's 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 call tree. A graphical representation of all the functions and calls within an application, which appears in the Xprofiler main window. The functions are represented by green, solid-filled rectangles called function boxes. The size and shape of each function box indicates its CPU usage. Calls between functions are represented by blue arrows, called call arcs, drawn between the function boxes. See also *call arcs*.

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. One divides the program into independent pieces of functionality which are distributed to independent processors. This 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+1st iteration Jacobi-Seidel uses only values calculated in the ith iteration. Gauss-Seidel uses a mixture of values calculated in the ith and i+1st 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, Pascal, Fortran, or COBOL 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**.

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

H

SP Switch. The high-performance message passing network, of the IBM RS/6000 SP(SP) machine, that connects all processor nodes.

HIPPI. High performance parallel interface.

hook. **hook** is a **pdbx** command that allows you to re-establish control over all task(s) 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, and providing 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 which 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 Parallel Environment for AIX. A program product that provides an execution and development environment for parallel FORTRAN, C, or C++ 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 a IBM RS/6000 workstation or on SP 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 Product*, and *package*.

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

Internet Protocol (IP). 1) The TCP/IP protocol that provides packet delivery between the hardware and user processes. 2) The SP Switch library, provided with the Parallel System Support Programs, that follows the IP protocol of TCP/IP.

IP. See *Internet Protocol*.

J

Jacobi-Seidel. See *Gauss-Seidel*.

job management system.

The software you use to manage the jobs across your system, based on the availability and state of system resources.

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 which 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.

The dimension-free version of Laplace's equation is:

$$\nabla^2 u = 0$$

The two-dimensional version of Laplace's equation may be written as:

$$\frac{\partial^2 u}{\partial x^2} + \frac{\partial^2 u}{\partial y^2} = 0$$

latency. The time interval between the instant at which an instruction control unit initiates a call for data transmission, and the instant at which the actual transfer of data (or receipt of data at the remote end) begins. Latency is related to the hardware characteristics of the system and to the different layers of software that are involved in initiating the task of packing and transmitting the data.

Licensed Program Product (LPP). A collection of software packages, sold as a product, that customers pay for to license. It can consist of packages and filesets a customer would install. These packages and filesets bear a copyright and are offered under the terms and conditions of a licensing agreement. See also *fileset* and *package*.

| **LoadLeveler**. A job management system that works
| with POE to allow users to run jobs and match
| processing needs with system resources, in order to
| better utilize 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 which makes multiple copies of the body of a loop, placing the copies also 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

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 using the AIX Message Facility 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.

MIMD (Multiple Instruction Multiple Data). A parallel programming model in which different processors perform different instructions on different sets of data.

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

MPI. Message Passing Interface; a standardized API for implementing the message passing model.

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.

node. (1) In a network, the point where one or more functional units interconnect transmission lines. A computer location defined in a network. (2) In terms of the IBM RS/6000 SP, a single location or workstation in a network. An SP node is a physical entity (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, which returns immediately whether or not the operation was completed. For example, a nonblocking receive will not wait until a message is sent, but a blocking receive will wait. A nonblocking receive will return a status value that indicates whether or not a message was received.

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.

optimization. A not strictly accurate but widely used 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 filesets that have been collected into a single installable image of program products, or LPPs. Multiple filesets can be bundled together for installing groups of software together. See also *fileset* and *Licensed Program Product*.

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) In terms of the IBM RS/6000 SP, a logical definition of nodes to be viewed as one system or domain. System partitioning is a method of organizing the SP 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. **pdbx** is the parallel, symbolic command line debugging facility of PE. **pdbx** is based on the **dbx** debugger and has a similar interface.

PE. The IBM Parallel Environment for AIX program product.

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

POE. See Parallel Operating Environment.

pool. Groups of nodes on an SP that are known to the Resource Manager, and are identified by a number.

point-to-point communication. A communication operation which 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 via 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 which produces an execution profile of an application or program. It is useful to identifying 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.

Program Marker Array. An X-Windows run time monitor tool provided with Parallel Operating Environment, used to provide immediate visual feedback on a program's execution.

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

R

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

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

remote shell (rsh). A command supplied with both AIX and the Parallel System Support Programs that lets you issue commands on a remote host.

Report. In Xprofiler, a tabular listing of performance data that is derived from the gmon.out files of an application. There are five types of reports that are generated by Xprofiler, and each one presents different statistical information for an application.

Resource Manager. A server that runs on one of the nodes of a IBM RS/6000 SP (SP) machine. It prevents parallel jobs from interfering with each other, and reports job-related node information.

RISC. Reduced Instruction Set Computing (RISC), the technology for today's high-performance personal computers and workstations, was invented in 1975.

S

shell script. A sequence of commands that are to be executed by a shell interpreter such as C shell, korn shell, or Bourne shell. Script commands are stored in a file in the same form as if they were typed at a terminal.

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

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

signal handling. A type of communication that is used by message passing libraries. Signal handling involves using AIX signals as an asynchronous way to move data in and out of message buffers.

source line. A line of source code.

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

SP. IBM RS/6000 SP; a scalable system from two to 128 processor nodes, arranged in various physical configurations, that provides a high-powered computing environment.

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

standard input (STDIN). In the AIX operating system, 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). In the AIX operating system, 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.

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 may 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.

System Data Repository. A component of the Parallel System Support Programs software that provides configuration management for the SP system. It manages the storage and retrieval of system data across the control workstation, file servers, and nodes.

System Status Array. An X-Windows run time monitor tool, provided with the Parallel Operating Environment, that lets you quickly survey the utilization of processor nodes.

T

task. A unit of computation analogous to an AIX process.

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

tracing. In PE, the collection of data for the Visualization Tool (VT). The program is *traced* by collecting information about the execution of the program in trace records. These records are then accumulated into a trace file which a user visualizes with VT.

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 may not be appropriate). These records are then accumulated into a trace file which allows the Visualization Tool to visually display the communications patterns from the program.

U

unrolling loops. See *loop unrolling*.

US. See *user space*.

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

user space (US). A version of the message passing library that is optimized for direct access to the SP Switch , that maximizes the performance capabilities of the SP hardware.

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.

view. (1) In an information resource directory, the combination of a variation name and revision number that is used as a component of an access name or of a descriptive name.

Visualization Tool. The PE Visualization Tool. This tool uses information that is captured as your parallel program executes, and presents a graphical display of the program execution. For more information, see *IBM Parallel Environment for AIX: Operation and Use, Volume 2*.

VT. See *Visualization Tool*.

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.

xpdbx. This is the former name of the PE graphical interface debugging facility, which is now called **pedb**.

Xprofiler. An AIX tool that is used to analyze the performance of both serial and parallel applications, via a graphical user interface. Xprofiler provides quick access to the profiled data, so that the functions that are the most CPU-intensive can be easily identified.

Index

Communicating Your Comments to IBM

IBM Parallel Environment for AIX
Installation
Version 2 Release 4

Publication No. GC28-1981-02

If you especially like or dislike anything about this book, please use one of the methods listed below to send your comments to IBM. Whichever method you choose, make sure you send your name, address, and telephone number if you would like a reply.

Feel free to comment on specific errors or omissions, accuracy, organization, subject matter, or completeness of this book. However, the comments you send should pertain to only the information in this manual and the way in which the information is presented. To request additional publications, or to ask questions or make comments about the functions of IBM products or systems, you should talk to your IBM representative or to your IBM 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.

If you are mailing a reader's comment form (RCF) from a country other than the United States, you can give the RCF to the local IBM branch office or IBM representative for postage-paid mailing.

- If you prefer to send comments by mail, use the RCF at the back of this book.
- If you prefer to send comments by FAX, use this number:
  - FAX: (International Access Code)+1+914+432-9405
- If you prefer to send comments electronically, use this network ID:
  - IBM Mail Exchange: USIB6TC9 at IBMMAIL
  - Internet e-mail: mhvrcfs@us.ibm.com
  - World Wide Web: http://www.s390.ibm.com/os390

Make sure to include the following in your note:

- Title and publication number of this book
- Page number or topic to which your comment applies

Optionally, if you include your telephone number, we will be able to respond to your comments by phone.

Reader's Comments — We'd Like to Hear from You

IBM Parallel Environment for AIX
Installation
Version 2 Release 4

Publication No. GC28-1981-02

You may use this form to communicate your comments about this publication, its organization, or subject matter, with the understanding that IBM may use or distribute whatever information you supply in any way it believes appropriate without incurring any obligation to you. Your comments will be sent to the author's department for whatever review and action, if any, are deemed appropriate.

Note: Copies of IBM publications are not stocked at the location to which this form is addressed. Please direct any requests for copies of publications, or for assistance in using your IBM system, to your IBM representative or to the IBM branch office serving your locality.

Today's date: ____________________________________

What is your occupation?

Newsletter number of latest Technical Newsletter (if any) concerning this publication:

How did you use this publication?

[ ] As an introduction
[ ] As a reference manual
[ ] For another purpose (explain)

[ ] As a text (student)
[ ] As a text (instructor)

____

____

Is there anything you especially like or dislike about the organization, presentation, or writing in this manual? Helpful comments include general usefulness of the book; possible additions, deletions, and clarifications; specific errors and omissions.

Page Number: Comment:

____
Name

____
Company or Organization

____
Phone No.

____
Address

____

____

Reader's Comments — We'd Like to Hear from You
GC28-1981-02

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
522 South Road
Poughkeepsie NY 12601-5400

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

Cut or Fold
Along Line

Program Number: 5765-543

Printed in the United States of America
on recycled paper containing 10%
recovered post-consumer fiber.

GC28-1981-02