



PowerVM Lx86 for x86 Linux Applications Administration Guide



PowerVM Lx86 for x86 Linux Applications Administration Guide

Note

Before using this information and the product it supports, read the information in “Notices” on page 101.

Second Edition (April 2008)

This edition applies to the version 1.2.0.0 of IBM PowerVM Lx86 for x86 Linux applications and to all subsequent releases and modifications until otherwise indicated in new editions.

Copyright (c) International Business Machines Corporation, 2007, 2008. All rights reserved.

Portions of this document are based on information supplied under license by Transitive Limited, which is copyright (c) Transitive Limited, 2003-2008.

IBM PowerVM Lx86 applications uses, under license from Transitive Limited, one or more patents cited at <http://www.transitive.com/patents>

© Copyright International Business Machines Corporation 2008. All rights reserved.

US Government Users Restricted Rights – Use, duplication or disclosure restricted by GSA ADP Schedule Contract with IBM Corp.

Contents

About this publication	v
How to send your comments	v

Typographical conventions	vii
--	------------

Part 1. Introduction to PowerVM Lx86	1
---	----------

Chapter 1. PowerVM Lx86 overview	3
---	----------

Chapter 2. PowerVM Lx86 concepts	5
VxE for PowerVM Lx86	5
x86World for PowerVM Lx86	6
Jailing and escapes for PowerVM Lx86	7

Part 2. Installation of PowerVM Lx86	9
---	----------

Chapter 3. System and installation requirements for PowerVM Lx86	11
---	-----------

Chapter 4. Downloading ISOs for PowerVM Lx86	15
Red Hat procedure for downloading PowerVM Lx86 ISO images	15
SUSE procedure for downloading PowerVM Lx86 ISO images	16

Chapter 5. Running the PowerVM Lx86 installation script	19
Installing PowerVM Lx86 and x86World	19
PowerVM Lx86 configuration settings	24
x86World user configuration for PowerVM Lx86	27
Starting the PowerVM Lx86 daemon and x86 daemons	28
Uninstalling PowerVM Lx86	28
Upgrading PowerVM Lx86	30
Upgrading previous versions of PowerVM Lx86	30
Uninstalling previous versions of PowerVM Lx86	31
PowerVM Lx86 installation reference	31

Chapter 6. Installing and migrating x86 applications onto a PowerVM Lx86 system	35
Migrating the x86 system configuration	35
Installing x86 applications	36
Running x86 applications	39
Running x86 applications reference	40
Installing and updating packages in x86 World	41
PowerVM Lx86 log files	46
x86 /etc/init.d support scripts	47
Monitoring x86 applications	48

Maintaining x86 applications	49
--	----

Part 3. Managing users, groups, and passwords with PowerVM Lx86	51
--	-----------

Chapter 7. The solution: A unified view	53
--	-----------

Chapter 8. Periodic checks of the subject x86 World	55
--	-----------

Chapter 9. Known issues with the unified view approach	57
---	-----------

Chapter 10. WORLD_CHECK_OR_SYNC options	59
--	-----------

Chapter 11. PowerVM Lx86 translation process	61
---	-----------

Chapter 12. Default PowerVM Lx86 escapes and virtual files in the x86 World	63
Virtual files and directories	63
Default escaped directories, files, and sockets	66
Syslog support	66

Part 4. PowerVM Lx86 error messages and resolutions	69
--	-----------

Chapter 13. Alerts and errors when managing users, groups, and passwords with PowerVM Lx86	71
---	-----------

Chapter 14. Errors from the x86 /etc/init.d support scripts	73
--	-----------

Chapter 15. Errors from the linkx86 script	77
---	-----------

Chapter 16. Errors from the execve router	79
--	-----------

Chapter 17. Syslog messages	81
--	-----------

Chapter 18. PowerVM Lx86 (p-ave) errors	83
--	-----------

Chapter 19. PowerVM Lx86 log file errors 87

Chapter 20. PowerVM Lx86 daemon (p-ave-daemon) errors 89

Chapter 21. E-mail messages for managing users, groups, and passwords in PowerVM Lx86 91

Part 5. Glossary 95

Part 6. Appendixes. 97

Appendix. Accessibility features. 99

Notices 101

Trademarks 103

Terms and conditions. 104

About this publication

This guide provides information about the use of IBM PowerVM Lx86.

For information about the accessibility features of this product, see “Accessibility features,” on page 99.

How to send your comments

Your feedback is important in helping to provide the most accurate and highest quality information. If you have any comments about this publication, send your comments using Resource Link™ at <http://www.ibm.com/servers/resourcelink>. Click **Feedback** on the navigation pane. Be sure to include the name of the book, the form number of the book, and the specific location of the text you are commenting on (for example, a page number or table number).

Typographical conventions

The following typographical highlighting conventions are used in this book:

Font	Usage
Bold	Identifies commands and graphical objects such as buttons, labels, and icons that you select.
<i>Italics</i>	Identifies parameters whose actual names or values you are to supply.
Monospace	Identifies examples of text similar to what you might see displayed, examples of portions of program code similar to what you might enter, messages from the system, or information you should literally type.

Part 1. Introduction to PowerVM Lx86

This guide explains how to install and configure the PowerVM Lx86 for x86 Linux[®] Applications (Lx86) product on a Linux on POWER[™] system.

The guide has the following parts:

- PowerVM Lx86 overview and concepts gives an overview of the system, including explanations of some of the terms and concepts unique to PowerVM Lx86.
- Installation of PowerVM Lx86 and workload migration explains how to install PowerVM Lx86. Overviews and examples show you how to use PowerVM Lx86 to run your Linux on x86 applications on Linux on POWER.
- Managing users, groups, and passwords with PowerVM Lx86 explains how the system deals with two definitions of users, groups, and passwords on the system because the x86 World and the POWER system both have their own set of own set of passwords, groups, and shadow files in the locations /etc/passwd, /etc/group and /etc/shadow.
- Error messages and resolutions shows errors that can be reported on the terminal by the components of PowerVM Lx86 and details of how to resolve each problem.

At the end of the guide there are also appendixes and a glossary. This document should be used in conjunction with the PowerVM Lx86 release notes, which provide additional information specific to a particular release. This includes known limitations with Linux on x86 tools and any additional installation instructions.

Chapter 1. PowerVM Lx86 overview

PowerVM Lx86 enables POWER machines to run Linux x86 applications alongside native Linux on POWER applications.

The Linux x86 applications run on SUSE or Red Hat Linux x86 distributions within a Virtual x86 Environment (VxE). No modifications or recompilations of the x86 Linux applications are needed. The x86 operating system and applications only need to be installed on or copied to a Linux on POWER machine that has PowerVM Lx86 installed on it.

Installing PowerVM Lx86E on a POWER system makes the system compatible with x86 applications. This extends the application support for Linux on POWER, allowing applications that are available on x86 but not on POWER to be run on the system.

How it works

PowerVM Lx86 creates a VxE within which x86 applications can run. The VxE is created solely within user-space; no modifications to the POWER kernel are required. PowerVM Lx86 does not run the x86 kernel on the POWER machine. Instead, it dynamically translates and maps all requests made from within the VxE to the underlying Linux operating system and POWER processor. The VxE is not a virtual machine; rather, x86 applications are encapsulated so the operating environment appears to be Linux on x86, even though the underlying system is Linux on POWER.

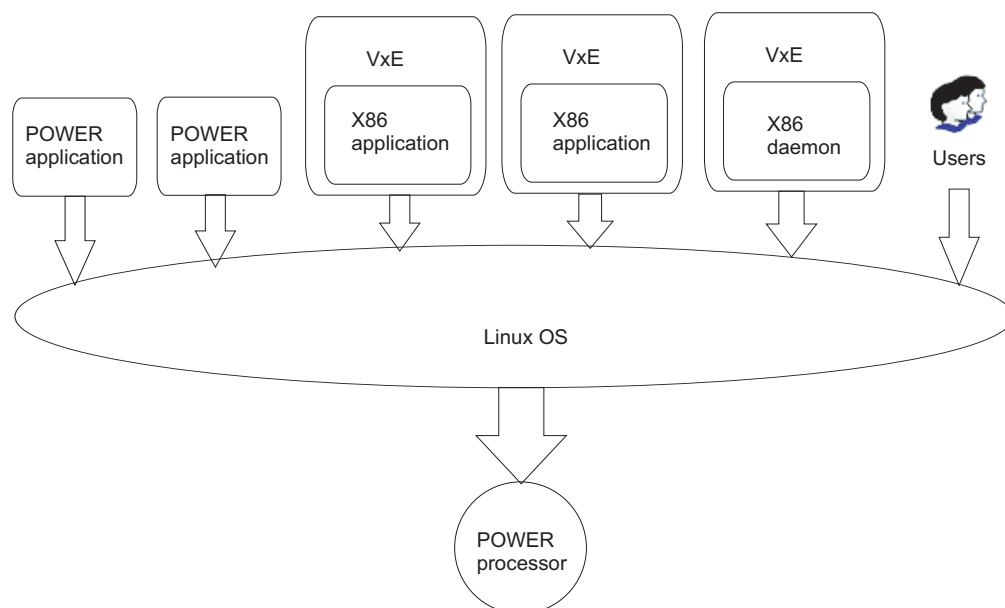


Figure 1. How it works

See Chapter 11, “PowerVM Lx86 translation process,” on page 61 in this guide for a detailed description about how requests made from within a VxE are dynamically translated and mapped to the underlying Linux operating system and POWER processor.

PowerVM Lx86 does not affect Linux on POWER applications

An x86 application running within a VxE appears to be just another user-space POWER process. It has no direct effect on any POWER applications that run natively on the host system.

As an end user

If you want to run x86 applications, you can log on to the POWER system and run the x86 applications within a VxE using PowerVM Lx86. See the “Running x86 applications” on page 39 section in this guide.

PowerVM Lx86 installation

IBM® supplies a semi-automated installation script and RPM packages. See Part 2, “Installation of PowerVM Lx86,” on page 9 in this guide for full installation instructions.

Installation of x86 applications

You can install x86 applications using their original installation scripts from within a VxE. Alternatively, you can copy x86 applications onto the POWER machine. See the “Installing x86 applications” on page 36 section in this guide for details.

Chapter 2. PowerVM Lx86 concepts

PowerVM Lx86 concepts include the VxE, the x86 World, and jailing and escapes.

VxE for PowerVM Lx86

PowerVM Lx86 creates a VxE within which x86 applications can run.

The PowerVM Lx86 software creates a VxE using the following:

- A set of x86 Linux libraries, commands, applications, and other system files that are installed in a directory, known as x86World, on the POWER system.
- A program, known as the translator, that handles the mapping of instructions and requests from the VxE onto the underlying POWER system.
- Selective integration between the VxE and the POWER environment. For example, a subset of the Linux on POWER file system is visible to applications from within the VxE.

Use the **runx86** command to run x86 binaries. For example, to run the x86 binary `myx86Binary` within a VxE, you enter the following command from a native POWER shell on the host system:

```
runx86 ./myx86Binary
```

In this example, when `myx86Binary` finishes executing, the **runx86** command also exits, and the VxE no longer exists.

If you enter the **runx86** command without providing an argument, then **runx86** starts an x86 Bash shell within a new VxE. You then interact with the x86 shell as though you are on an x86 machine. Any commands entered from the x86 shell are then automatically run within a VxE.

Following is an example using the **runx86** command:

```
% uname -srmpi
Linux 2.6.9 ppc64 ppc64 ppc64
% runx86
$ uname -srmpi
Linux 2.6.9 i686 i686 i386
$ exit
%
```

After you finish working with the x86 shell, use the **exit** command as if you were using a native x86 shell on an x86 machine. After you exit, the VxE that ran the x86 shell no longer exists, and you return to the POWER shell.

Notes:

- It is not possible to run an x86 binary directly from a native POWER shell. To ensure that the x86 binary is run within a VxE, the **runx86** command must always be used, either with the x86 binary as an argument, or without any arguments to start an x86 shell. From the shell you can then directly run the x86 binary.

- Not all commands can be run in a VxE. For example, you should do system administration work from a native POWER shell. However, software management should be done in the VxE.

For more information about using runx86, see “Running x86 applications” on page 39.

x86World for PowerVM Lx86

The x86World directory contains the x86 binaries, common x86 libraries, and infrastructure files. X86 applications should also be installed in this directory. This directory is configurable and is specified during the installation process. The default location for x86World is the /i386 directory. The convention *X86WORLD_ROOT* is used in this document to represent the directory where the x86 files are installed.

An x86World can be created using x86 libraries and binaries obtained from an existing Linux distribution for x86.

As described previously, you run x86 binaries within a VxE with the **runx86** command. For an x86 application or shell running within a VxE, the accessible part of the file system is restricted to x86World. This is shown in the following example. The area ringed with a dashed line is x86World.

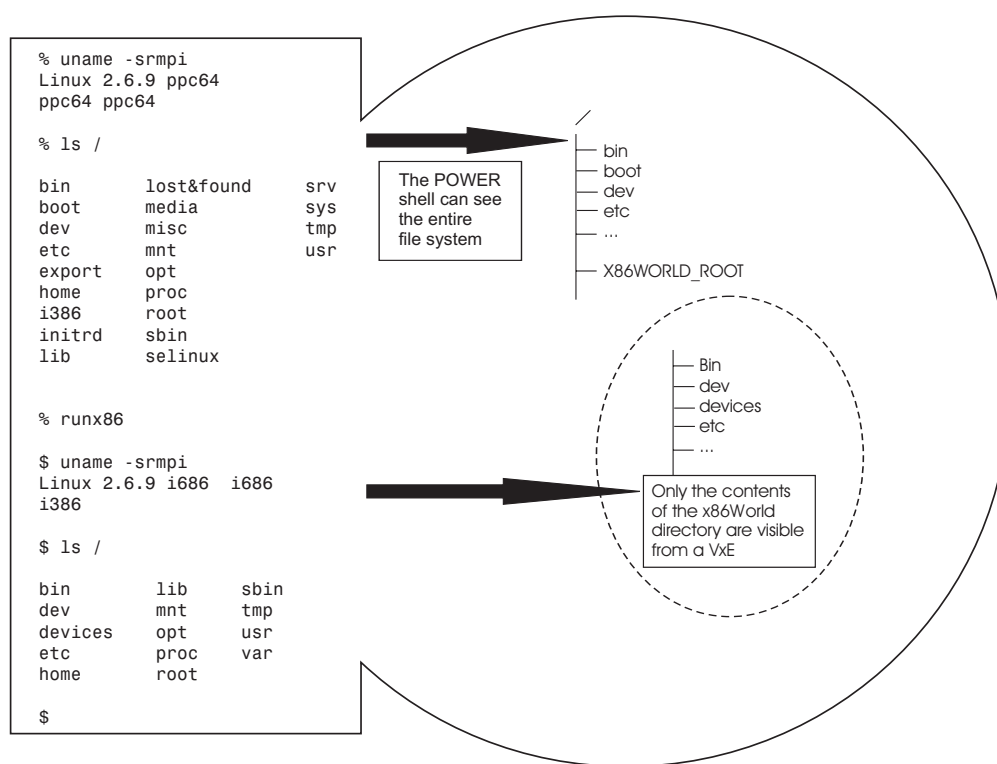


Figure 2. x86World

In this example, when the `ls` command is issued from within the x86 shell, the x86 `/bin/ls` binary is run within a VxE, and it displays the contents of the root of the x86World filesystem. That is `/i386` by default.

This restriction of the view of the Linux file system is called *jailing*, and is similar in concept to UNIX[®] `chroot`.

Jailing and escapes for PowerVM Lx86

Jailing an application ensures that it accesses x86 libraries and infrastructure files, just as if the application is running natively on an x86 machine.

Any application running within a VxE sees the directory structure under `X86WORLD_ROOT` as its root file system. The VxE replaces references to `/` at the start of a file path with `X86WORLD_ROOT`.

Following is an example of a jail:

Assume the x86 application tries to open the `/lib/libc.so.6` library.

If this path is left unmodified and passed through to the POWER kernel, then the application incorrectly opens the POWER version of the C runtime library, which is incompatible with x86.

So, the path is jailed, and the jailed path is passed through to the POWER kernel as `X86WORLD_ROOT/lib/libc.so.6`.

This causes the x86 C runtime library to be opened, and subsequently loaded, allowing the x86 application to run.

Escapes (access to files outside of x86World)

Sometimes it is necessary for an x86 application to access files that are located outside of x86World. For example, the application might need to access data stored on a remote shared filesystem, or it might need to access data that is stored locally but not within x86World. To allow access to files stored locally that are external to x86World, a mechanism known as an *escape* is used.

An escape links a path inside x86World with a path outside x86World. You create escapes with the **linkx86** command supplied with PowerVM Lx86. The **linkx86** command enables PowerVM Lx86 to recognise escapes. The escape is then transparent to an x86 application running within a VxE, similarly to the way a mounted file or directory in Linux is transparent to a Linux application. Escapes persist across reboots of the host POWER system, and they are visible to all x86 applications.

Note: An escape is implemented as a symbolic link that follows a specific naming convention to ensure that access from the VxE functions correctly. It is not recommended to use the **ln** command directly to create escapes. This can cause x86 applications running within a VxE to fail. Due to the nature of its implementation, an escape can be safely removed from the system by deleting the link on a non translated (POWER) shell.

Example Escape

An x86 application needs access to files in /var/accounts

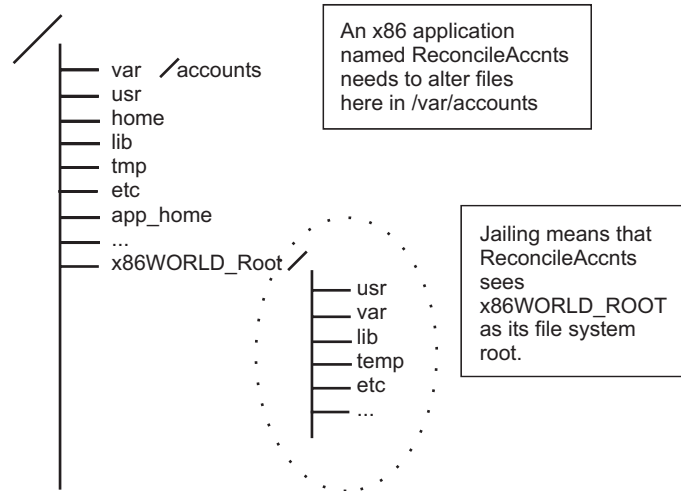


Figure 3. Example escape

You can use the linkx86 command to create an escape to the /var/accounts directory, as the following example demonstrates, starting in a POWER shell:

```
% runx86

$ ls var
cache lib log
$ exit

% linkx86 /var/accounts
% runx86

$ ls /var
accounts cache lib log
$
```

The x86 application now sees the /var/accounts directory.

```
/
var/accounts
usr
x86WORLD_Root
home
lib
tmp
etc
apphome
...
```

This is the file system as seen from the VxE.

If access is needed to a remote filesystem from within a VxE, then the remote filesystem can be mounted to a mount point under the x86World root or mounted in a native POWER shell with an escape created from the x86World. The mount command and the linkx86 command should both be issued from a native POWER shell, and not from an x86 shell.

Part 2. Installation of PowerVM Lx86

The PowerVM Lx86 installation package consists of RPM packages and scripts.

The scripts are for installing and uninstalling PowerVM Lx86, and can optionally install or uninstall x86World. The `installer.pl` script installs the required PowerVM Lx86 RPM package. With the `installer.pl` script you can install an x86World using an x86 distribution from CD or ISO images. The PowerVM Lx86 translator RPM package and binaries are named `p-ave`.

Release contents

A PowerVM Lx86 release can be provided on a CD or downloaded from IBM's Web site. A typical release contains the following files:

- `installer.pl`: a semi-automated script that installs PowerVM Lx86 and installs and sets up x86World for use by PowerVM Lx86
- PowerVM Lx86 for x86 Linux Applications Administration Guide (*powervm-lx86-admin-guide-1.2.pdf*): this guide in PDF format
- PowerVM Lx86: RPMs and internal installation scripts (*powervm-lx86-installer-1.2.0.0.-1*)
- PowerVM Lx86 Release Notes (*powervm-lx86-release-notes-1.2..txt* and *powervm-lx86-release-notes-1.2..pdf*): amendments to installation instructions and other information specific to the release version. (*release* stands for the version number of the release and file version. The PowerVM Lx86 release notes specify the exact version numbers of a release.)

Chapter 3. System and installation requirements for PowerVM Lx86

Requirements include the necessary access levels, and hardware and software requirements, for the installation.

Prerequisites

Root access when installing the PowerVM Lx86 RPM package and x86World is necessary.

Supported POWER operating systems

- Novell SUSE Linux Enterprise Server 10 (SLES 10)
- Novell SUSE Linux Enterprise Server 10 (SLES 10) Service Pack 1
- Novell SUSE Linux Enterprise Server 9 (SLES 9) Service Pack 3
- Novell SUSE Linux Enterprise Server (SLES9) Service Pack 4
- Red Hat 4 AS (RHEL 4 AS) Linux Update 4
- Red Hat 4 AS (RHEL 4 AS) Linux Update 5
- Red Hat 4 AS (RHEL 4 AS) Linux Update 6

Minimum hardware requirements

PowerVM Lx86 runs on the following hardware platforms:

- System p™ server with POWER5™, POWER6™ or Power PC-compatible processor (for example, OpenPower™, JS20, JS21, p505, p510, p520, p550, or p570)

Note: Only System p servers are supported. System i™ servers are not supported.

Disk space for SLES9 Service Pack 3

- 3 GB disk space for software for a full installation; 600 MB disk space for software for a minimal installation
- 1 GB disk space for the /tmp directory for a full installation; 300MB disk space for the /tmp directory for a minimal installation

Disk space for SLES9 Service Pack 4

- 3 GB disk space for software for a full installation; 600 MB disk space for software for a minimal installation
- 1 GB disk space for the /tmp directory for a full installation; 150 MB disk space for the /tmp directory for a minimal installation

Disk space for SLES10 and SLES 10.1

- 3.2 GB disk space for software for a full installation; 900 MB disk space for software for a minimal installation
- 1 GB disk space for the /tmp directory for a full installation; 300 MB disk space for the /tmp directory for a minimal installation

Disk space for RHEL 4.4, 4.5, 4.6

- 10 GB disk space for software for a full installation; 1.2 GB disk space for software for a minimal installation
- 2.4 GB disk space in the /tmp directory for a full installation; 400 MB for a minimal installation

Other required POWER software

PowerVM Lx86 requires that the following software components be installed. The installation script checks for these components, and notifies you if they are not present.

- Perl 5.8 must be installed on the POWER system for the PowerVM Lx86 installer to function. You can check the Perl version with the **perl --version** command. If you do a full installation of the particular distribution, it will be installed by default. If Perl is not installed on your system, install the version of Perl that is provided with your distribution.

Table 1. Perl requirements for PowerVM Lx86 installer

Power OS	Perl version required
SLES 10.1	perl-5.8.8-14.2
SLES 10	perl-5.8.8-14.2
SLES 9 SP3	perl-5.8.3-32.4
RHEL 4.6	perl-5.8.5-36.RHEL4
RHEL 4.5	perl-5.8.5-36.RHEL4
RHEL 4.4	perl-5.8.5-36.RHEL4

- The 64-bit version of libstdc++.so.6 must be installed on the POWER system for PowerVM Lx86 to function. Each of the supported distributions has its own version of the library, which must be installed. You can check that the library is installed on your system, by checking that **/usr/lib64/libstdc++.so.6** exists.

Table 2. libstdc++.so.6 version required for PowerVM Lx86

Power OS libstdc	package required
SLES 10.1	libstdc++-64bit
SLES 10	libstdc++-64bit
SLES 9 SP4	compat-libstdc++-lsb-64bit
SLES 9 SP3	compat-libstdc++-lsb-64bit
RHEL 4.6	libstdc++-3.4.6-9
RHEL 4.5	libstdc++-3.4.6-8
RHEL4.4	libstdc++-3.4.6-3

You can check the RPM version of your libstdc++.so.6 packages installed with the following command:

```
rpm -qa | grep libstdc
```

Installation media requirements

- For SUSE Linux Enterprise Server (SLES) 9 SP3, the installation requires the following CDs or ISO images:
 - SUSE SLES Version 9 CD (Base CD1); an example is SLES-9-i386-RC5-CD1.iso

- SUSE CORE Version 9 CDs 1 - 4; an example is SLES-9-i386-RC5-CD{2-5}.iso
- SUSE SLES 9 Service-Pack Version CDs 1 - 3; an example is SLES-9-SP-3-i386-GM-CD{1-3}.iso
- For SUSE Linux Enterprise Server (SLES) 9 SP4, the installation requires the following CDs or ISO images:
 - SLES Version 9 for x86 CDs 1 - 4; an example is SLES-9-i386-RC5-CD{1-4}.iso
 - and*
 - SLES Version 9 SP4 for x86 CDs 1 - 2; an example is SLES-9-SP4-CD-i386-GM-CD{ 1-2}.iso
- For SUSE Linux Enterprise Server (SLES) 10 SP1, the installation requires the following CDs, DVD or ISO images:
 - SLES Version 10 SP1 for x86 CDs 1 - 4; an example is SLES-10-SP1-CD-i386-GM-CD{ 1-4}.iso

Or

 - SLES Version 10 SP1 for x86 DVDs 1 - 2; an example is SLES-10-SP1-DVD-i386-GM-DVD{ 1-2}.iso
- For Red Hat Enterprise Linux (RHEL) 4.4 AS, the installation requires the following CDs or ISO images:
 - RHEL 4.4 AS CDs 1 - 4; an example is RHEL4-U4-i386-ES-disc{1-4}.iso
- For Red Hat Enterprise Linux (RHEL) 4.5 AS, the installation requires the following CDs or ISO images:
 - RHEL 4.5 AS CDs 1 - 4; an example is RHEL4-U5-i386-ES-disc{1-4}.iso
- For Red Hat Enterprise Linux (RHEL) 4.6, the installation requires the following DVD or ISO images:
 - RHEL version 4.6 for x86 CDs 1 - 5; an example is RHEL4.6-i386-AS-disc{1-5}.iso

Or

 - RHEL Version 4.6 for x86 DVD; an example is RHEL4.6-i386-AS-DVD.iso

Chapter 4. Downloading ISOs for PowerVM Lx86

Before you proceed with the installation, you must make sure that the x86 CDs or ISO images required by PowerVM Lx86 are available to the installer script.

The Virtual x86 Environment uses a set of x86 libraries, commands, applications and other system files. See Chapter 3, “System and installation requirements for PowerVM Lx86,” on page 11 for the specific CDs or ISOs you need. If you do not already have the CDs or ISOs, you can download the ISO images from the Red Hat or Novell Web sites.

The best method for installation is to download all ISOs to one directory and provide the name of that directory to the installer.

Red Hat procedure for downloading PowerVM Lx86 ISO images

Find the requirements and procedures for downloading Red Hat ISO images from the Red Hat Network (RHN) Web site. Note that these procedures have been verified, but may have changed slightly since the writing of this manual.

Prerequisites

To download ISO images from the RHN Web site, you must have the following items. You can skip this if you already have the ISOs.

- A valid login account for the RHN Web site. If you do not have an RHN account, you can create one on the <https://www.redhat.com/rhn> Web site.
- A system that is registered under your login account. If you need to register a system, follow these steps:
 1. Log in to your system as root, and type the following at the command prompt: `up2date --register`. A window then opens, and you are led through the registration process.
 2. Read the Red Hat privacy statement, and provide your RHN login ID and password. If you do not have an RHN ID, you can create one using this window.
 3. When asked, provide the information you want to have accessible on your system profile.

Once completed, your chosen system profile information is sent to RHN, and you will be able to view this information on the RHN Web site under your account.

- A system with the proper Red Hat entitlement assigned to it.

If you do not have an entitlement, the first entitlement is free for your first registered system.

Thereafter you must purchase additional entitlements to be used for any additional systems.

You can use the same entitlement for different systems with different architectures, so long as only one system is assigned to that entitlement at any given time.

- A system with the proper Red Hat subscription assigned to it.

You must purchase a subscription for a particular system to obtain ISOs for it. Red Hat allows you to have a 30 day free trial period, during which you can

download ISO images. After this period ends, you still have access to the ISO images, but you are no longer able to download updates or to receive support. The software subscription is system and architecture specific. Regular customers need to purchase one subscription to cover ISO images for an x86 architecture and a second subscription to cover the ISO images for a POWER architecture. However, you are not asked to pay for two subscriptions with PowerVM Lx86. The PowerVM Lx86 installation collects the data needed for Red Hat to upgrade the Linux POWER entitlement to include an entitlement for Linux on x86 as well. This additional x86 entitlement is only to be used on a POWER system.

Downloading ISO images from the Red Hat Network

It is simpler to download the RHEL ISO images onto a RHEL system than onto a non-RHEL system. The procedure to download onto a non-RHEL system follows this one, if you need to do so for some reason.

To download ISO images from the RHN onto a RHEL 4 system, complete the following steps:

1. Log in to the Red Hat Network site. As stated previously, if you do not have a login, you can create one at <https://www.redhat.com/rhn>.
2. Click **Downloads** on the top menu.
3. Click **Download your software here** in the Red Hat Enterprise Linux section.
4. Choose the system you want to download the ISO for. The window that opens shows the various ISO installation and source disk images, with their corresponding MD5 checksum values.

To download ISO images from the RHN on to a non-Red Hat system, complete the following steps.

1. Log in to the Red Hat Network site. As stated previously, if you don't have a login, you can create one at <https://www.redhat.com/rhn>.
2. Click **Systems** on the top right corner.
3. Choose the system you want to download the ISO for.
4. Click the **Software** tab located below the system name at the top.
5. Click the **Software Channels** tab. This shows the base channel that your system is subscribed to, along with any other additional channels beneath it.
6. Click the desired channel subscription that you want the ISO image from. The next window that opens shows further details of that subscription.
7. Click the **Downloads** tab. The window that opens shows the various ISO installation and source disk images, with their corresponding MD5 checksum values.

SUSE procedure for downloading PowerVM Lx86 ISO images

Find the requirements and procedures for downloading SUSE ISO images from the Novell Customer Center Account Web site. Note that these procedures have been verified, but may have changed slightly since the writing of this manual.

Prerequisites

To download images from the Novell Web site, you must have the following items:

- A valid login account for the Novell Customer Center Account Web site.

If you do not have an existing Novell Customer Center account, you can create one at <http://www.novell.com>. Click **Customer Center** under **Support**.

- The proper subscription for the SLES distribution version you want to download. To download the distribution version ISOs, you must have purchased the subscription for that particular version. You can obtain an evaluation version that gives you access to the distribution ISO images for a 30-day free trial period on SLES 9 distributions, and 60-day free trial period on SLES 10 distributions. You can use one subscription on varying system architectures, so you can use the same subscription for both x86 and POWER systems.

To get the subscription, follow these steps:

1. Go to <http://www.novell.com/products/server/>.
2. Click **Download Eval**.
3. Click the download that is appropriate for your platform. You get an activation code and a link to download the ISOs.

Downloading ISO images from the Novell Customer Center

To download ISO images from the Novell Customer Center, complete the following steps.

1. Log in to your account on the Novell Customer Center Web site located at <http://www.novell.com>.
2. Click **Customer Center** under **Support**.
3. Click **List of Products and Subscriptions** under **Products and Subscriptions** on the left panel. A window opens with a table of the software subscriptions you currently own.
4. Click **Activate Product/Subscription** under **Products and Subscriptions** on the left panel.
5. Enter the activation code, which is on the registration card you receive when you purchase the subscription. You also receive an activation code if you request an evaluation copy.
6. Double-click the distribution version you want to download.
7. Under **Overview**, scroll down to the **Downloads** section, and click **Media**. There is a list of ISO images that are available for the selected distribution version.
8. Click **proceed to download** on the upper right corner.
9. A page opens with the distribution images, which you can download.

Chapter 5. Running the PowerVM Lx86 installation script

The PowerVM Lx86 installation script installs the software in the standard configuration.

Installing PowerVM Lx86 and x86World

Use this procedure to install PowerVM Lx86 and x86World.

The following procedure assumes that you are installing onto a system for the first time, or that any previous installations of PowerVM Lx86 have been uninstalled. If the default locations are accepted, then the PowerVM Lx86 translation software is installed in the /opt/p-ave directory. You can install it in an alternative location if necessary.

The default location for the x86World libraries and binaries is the /i386 directory, which is frequently referred to as *X86WORLD_ROOT* in this document.

Also note that this procedure uses RHEL as an example, but the SLES installation is very similar.

1. Become root on the POWER system.
2. Download and untar the tar file.
3. Run the PowerVM Lx86 installation script:

```
./installer.pl
```

You are asked to read through and agree to the license agreement before continuing with the installation. The following prompt is displayed:

```
Host OS detected as Red Hat Enterprise Linux release 4, Update 4
```

```
Welcome to the PowerVM Lx86 (System p AVE) Installer
```

```
To run Linux/x86 applications the installation will require Linux/x86
application binaries, libraries and infrastructure files as well as the
PowerVM Lx86 (System p AVE) software.
```

```
Please refer to the PowerVM Lx86 (System p AVE) Administration Guide and
Release Notes for further details of the installation requirements.
```

```
You must agree to the following license to install System p AVE.
If you do not accept the license terms, the installation will abort.
```

```
Page through the license with SPACE.
```

```
Press enter to view the license.
```

4. Press Enter to view the license, and press Enter again to scroll to the next page. The following prompt is displayed:

```
Do you accept the license terms? [y/N]
```

5. Enter y to agree to the terms and to continue the installation. The license text is saved to disk. You can print this file for your records. You will be asked for information that will be sent by e-mail to activate the product. The following prompt is displayed:

```
Registration with IBM
```

```
You must provide some information to IBM before starting the installation
process. This Activation information will be sent to IBM as an email.
```

```
Note: You must provide your company name for the installation to proceed.
```

All other details are optional. The Activation email will also include some details about this system. Once you have entered this information it will be presented to you to review. You will have the option to send the information to IBM via email or abort the installation. Please enter the name of your company:

6. You must enter the name of your company. You are also prompted for your name, address, phone number, and e-mail address; these fields are optional.
7. Enter the requested information, or, to skip any of the questions, press Enter at the prompt. The system model number, serial number, and current time and date information are collected automatically from your system. All the information collected is displayed on the screen, and is similar to the following:

```
The following is about to be sent to IBM:
Date:      Wed Mar 28 15:03:33 GMT 2007
System:    IBM,nnnn-nnn
Serial:    IBM,nnnnnnnnn
Company:   Your Company
Name:      Your name
Address:   Your address
Phone Number: 555-5555
Email:     yourname@example.com
Is the above information correct? [y/N]
```

8. If this information is correct, enter y. The following prompt is displayed:
Send the above information to IBM (pave@us.ibm.com)?
(If you answer 'no' here, the installation will be aborted.) [Y/n]
9. If you agree with sending the information to IBM, press Enter, and you then see this output:

Email sent successfully.

10. (RHEL 4 only) The following prompt is displayed:

```
Registration with Red Hat Inc.
Would you like to Register? [y/N]
```

If you would like to register with Red Hat to upgrade your system to support both Linux on x86 and Linux on POWER, then enter y and go to the next step. If you do not wish to register now, you can register after installation.

11. (RHEL 4 only) The following prompt is displayed:

```
The Linux/x86 binaries and libraries in the x86 World can be kept
up to date using the Red Hat Network. To receive this support,
you must upgrade your current Red Hat subscription to support both
Linux/x86 and Linux/POWER. This upgrade is at no additional charge.
```

To upgrade the subscription on this system, you need to provide some information that will be emailed to Red Hat Inc. Once this has been sent, you will receive an acknowledgement email from Red Hat within approximately the next 24 hours (1 business day). You will then receive another email confirming that the subscription has been upgraded within approximately the next 48 hours (2 business days).

Please enter the name of your company [Your Company]:

To receive an upgraded Red Hat Network Subscription, you must provide your name, your company name, and an e-mail address, and, optionally, your Red Hat login name and your Red Hat account number. Your name, your company name, and an e-mail address are provided as default from the IBM Registration process you entered above.

12. (RHEL 4 only) If you are happy with the defaults, press Enter for each prompt, and then, optionally, answer the login and account number prompts. To skip these last two questions, press Enter at the prompt. The system model number, serial number, and current time and date information are now

collected automatically from your system. All the information collected is displayed on the screen. The following prompt is displayed:

```
Date:          Wed Feb 28 15:03:33 GMT 2007
Company:       Your Company
RHN Login:
Account No:
Name:         You name
Email:        yourname@example.com
```

Is the above information correct? [y/N]

13. (RHEL 4 only) If the information is correct, enter y. The following prompt is displayed:

```
Send the above information to Red Hat Inc. (pave@redhat.com)?
(If you answer 'no' here, the installation will be aborted.) [Y/n]
```

If you agree with sending the information to Red Hat to upgrade your subscription, press Enter, and you see this output:

```
Email sent successfully.
```

14. The following menu is displayed:

```
1. Install Software.
2. Upgrade Software.
3. Uninstall Software.
4. Show Installed Products.
5. Configure Software.
6. Quit.
```

Please select the type of operation you would like to perform:

15. Enter 1 to begin the installation. The following menu is displayed:

```
1. Install p-ave + x86 World.
2. Install p-ave.
3. Install x86 World.
4. Go back to the Main Menu.
5. Quit.
```

Please select the type of operation you would like to perform:

16. Enter 1 to begin the installation of the PowerVM Lx86 package and x86 World. The following prompt is displayed:

```
Install p-ave to: [/opt/p-ave]
```

17. Press Enter to accept the default installation location, or enter an alternative location. The following prompt is displayed:

```
Log files created by p-ave will be stored in [/var/opt/p-ave/log]:
```

18. Press Enter to accept the default log file location, or enter an alternative location. The following prompt is displayed:

```
Install x86 World to: [/i386]
```

19. Press Enter to install x86World in the recommended default location, or enter an alternative location. The following prompt is displayed:

```
1. p-ave-<version>-<OS>
2. Go back to the Main Menu.
```

Please select the version of p-ave you would like to install:

20. Select the version of PowerVM Lx86 to install by entering 1. The following prompt is displayed:

```
Which distribution will you be installing for your x86 World?
```

```
1. Red Hat Enterprise Linux AS release 4, Update 4 (full)
2. Red Hat Enterprise Linux AS release 4, Update 4 (minimal)
3. Red Hat...
```

Please select a distribution from the list above:

21. Enter the number of the appropriate distribution. The following prompt is displayed:

Where are the home directories located on this system (type 'skip' if you don't want the x86 World to share your home directories)? [/home]

22. This step escapes the x86World home directory. Press Enter to accept the default home directory location, or enter an alternative location. If you do not want to escape the home directories, enter skip. If your environment uses multiple locations for home directories, such as the /home1 directory and the /home2 directory, then you need to use the linkx86 script to configure additional escapes after the installation script has finished. The following prompt is displayed:

To create the x86 World the p-ave installer will need the appropriate set of x86 RPMs. This set of RPMs is generated from the Red Hat Enterprise Linux AS 4 Update 4 FULL (<install type>) x86 install media (ISO or CD/DVD-ROM).

Please enter the directory or files containing Red Hat Enterprise Linux AS release 4, Update 5 (<install type=">").

- ISO9660 images are accepted (if they end with .iso)
- RPM files are accepted (if they end with .rpm)
- If you have a CD/DVD-ROM, please mount it and enter the mount path

You can enter each file or mount path individually or provide a pattern to select multiple files at once, e.g.
/files/rhel/RHEL4-U5-i386-AS-disc*.iso.

To view a list of missing RPM files, type 'list'
To return to the main menu, type 'skip'

Enter directory or file:

23. To view the list of outstanding RPMs, enter list. You can source the RPMs from CD, DVD, ISO images, or a directory of RPMs. You can add the sources individually, and the required RPMs will be copied from the media. For CD and DVD installation, you can mount the disks individually.
24. (If mounting CDs or DVDs) Mount the first disk, and then enter the mount location. After the RPMs are copied from that disk, unmount the disk and mount the next disk. Type in the new mount location, and continue until all of the RPMs are copied across.

If you are installing from ISO images, you can use a pattern match if the ISOs have similar names. For example, you can add RHEL4-U4-i386-AS-disc1.iso and RHEL4-U4-i386-AS-disc2.iso by typing in the pattern
RHEL4-U4-i386-AS-disc*.iso.

25. (If mounting CDs or DVDs) Enter the location of the media. For each medium, the installer prints the following message:

nnn required RPMs found. Copying...

After all of the required RPMs are copied onto the local system, the RPMs are installed into x86World, and the following output is displayed:

All RPMs found successfully.

About to proceed with the following:

- Install p-ave to /opt/p-ave
- Log files will be stored in /var/opt/p-ave/log
- Install x86 World to /i386
 - Selected distribution is Red Hat Enterprise Linux AS release 4, Update 5 (minimal)
 - Home directories are in /home

1. Continue.
2. Change settings.

3. Go back to the Main Menu.

Please select the type of operation you would like to perform:

26. If the settings are correct, Enter 1 to continue.

p-ave installed successfully.

Installing x86 World...

Step 1/2: Installing base RPMs...

```
100 [=====>]
      [ OK ]
```

Step 2/2: Installing x86 world...

```
100% [=====>]
```

27. Enter 6 to quit the installation. The installation exits, with the following summary of the installation process and details of where the installation log and activation e-mails are stored:

The Activation email data is stored on the system in
/etc/opt/p-ave/activation-detail.

If this system is not set up to to send email, please email this
data file to IBM at <pave@us.ibm.com>.

The Red Hat Subscription email data is stored on the system in
/etc/opt/p-ave/redhat-activation-detail.

If this system is not set up to to send email or you do not get
an acknowledgement email from Red Hat within the next 24 hours,
please email this data file to Red Hat at <pave@redhat.com>. This will
ensure that your Red Hat subscription is upgraded.

License text saved in
/etc/opt/p-ave/redhat-activation-detail. You may print
this file for your records.

Thank you for using the System p AVE Installer

Today you performed the following transactions:

- Accepted license agreement
- Registered with IBM
- Registered with Red Hat
- Installed System p AVE
- Installed installer into /opt/p-ave/installer
- Installed x86 World (<OS><install type>)

End of Transaction log.

To start a translated x86 World shell, run /usr/local/bin/runx86

The full log of your session is in /tmp/p-ave_install_XXXXXX.log

Installation of PowerVM Lx86 and x86World is now complete. For further details about the installation, see “Starting the PowerVM Lx86 daemon and x86 daemons” on page 28 and “Installing x86 applications” on page 36.

To start running x86 applications, see “Running x86 applications” on page 39.

PowerVM Lx86 configuration settings

PowerVM Lx86 can be configured using various switches. The configuration switches supply parameters to the translator and change aspects of the translator's run time behavior. The configuration switches can be set using a configuration file or by setting environment variables.

PowerVM Lx86 configuration file

The configuration file is located on the POWER system at: `/etc/opt/p-ave/config`. Configuration switches can be added to the configuration file. The configuration switches are checked whenever a new process or application is started from an x86 shell. Changes made to the configuration switches will not effect processes that are already running.

The configuration switches take the following form:

`<CONFIGURATION_SWITCH>=<VALUE>`. Each configuration switch must be entered on a separate line in the configuration file.

By default, no configuration file is present when you install PowerVM Lx86. A configuration file will only be created by the PowerVM Lx86 installation process if non-default options are chosen for any one of the following installation options:

- System p AVE directory
- System p AVE log file directory
- x86 World directory

Creating a configuration file

If a configuration file is not present on the system, one can be created using a standard text editor. Save the file as: `/etc/opt/p-ave/config`.

Example configuration file

This is an example of the contents of a configuration file for a PowerVM Lx86 installation with the non-default PowerVM Lx86 directory: `/myp-ave/install-location` and the non-default log file location: `/var/myp-avelogs/log`.

```
FU_OPT_P_AVE=/myp-ave/install-location
LOGFILE_PATH=/var/myp-avelogs/log
LOCALISATION_FILES_DIR=/myp-ave/install-location/locale
```

Note: The `LOCALISATION_FILES_DIR` configuration switch is set when a non-default PowerVM Lx86 directory is chosen.

Configuration file switches

Switches effecting installation directories and log files:

Configuration switch name	Parameters and use
FU_OPT_P_AVE	<p>Type String</p> <p>Parameters An absolute path</p> <p>Use This switch specifies the PowerVM Lx86 installation directory. This should only be updated by the PowerVM Lx86 installer.pl script.</p>

Configuration switch name	Parameters and use
LOGFILE_PATH	<p>Type String</p> <p>Parameters An absolute path</p> <p>Use This switch specifies the PowerVM Lx86 log file directory. If this is changed manually, then a log file directory must exist with the correct permissions (owner and group 'root' and set to 01777) before PowerVM Lx86 will run correctly. This defaults to /var/opt/p-ave/log when no configuration switch is set.</p>
FU_OPT_SUBJECT_WORLD_ROOT	<p>Type String</p> <p>Parameters An absolute path</p> <p>Use This switch specifies the x86 World directory. This should only be updated by the PowerVM Lx86 installer.pl script. The defaults to /i386 when no configuration switch is set.</p>
LOCALISATION_FILES_DIR	<p>Type String</p> <p>Parameters An absolute path</p> <p>Use This switch specifies the PowerVM Lx86 localization files directory. This should only be updated by the PowerVM Lx86 installer.pl script and will be a subdirectory of the directory set by FU_OPT_P_AVE called locale.</p>

Switches effecting networking support:

Note: See Chapter 12, "Default PowerVM Lx86 escapes and virtual files in the x86 World," on page 63 for more details about the use of these configuration switches.

Configuration switch name	Parameters and use
FU_HAVE_SEPARATE_RESOLV_CONF_FILES	<p>Type Boolean</p> <p>Parameters y or n</p> <p>Use The default is n. Setting this to y will cause PowerVM Lx86 to manage the x86 World resolv.conf file separately from the POWER system version of the file. This is an option for advanced users only.</p>

Switches effecting the user id support:

Note: See Part 3, "Managing users, groups, and passwords with PowerVM Lx86," on page 51 for more details about using these configuration switches, in particular the parameters for WORLD_CHECK_OR_SYNC.

Configuration switch name	Parameters and use
FU_HAVE_SEPARATE_PASSWORDS	<p>Type Boolean</p> <p>Parameters y or n</p> <p>Use The default is n. Setting this to y will cause PowerVM Lx86 to favor the POWER system entries for passwords, rather than the entries in the x86 World files.</p>
FU_MERGE_PASSWD_FILES	<p>Type Boolean</p> <p>Parameters y or n</p> <p>Use The default is y. Setting this to n will cause PowerVM Lx86 to manage the /etc/passwd, /etc/group and /etc/shadow (and on Red Hat systems there is also /etc/gshadow) files completely separately by PowerVM Lx86 and the native POWER system. This is not recommended unless it is explicitly known that there are no security risks involved in doing so.</p>
WORLD_CHECK_OR_SYNC	<p>Type String</p> <p>Parameters sync_all, check_all, check_passwd, check_group, force_sync_mtab, none</p> <p>Use The default is check_all. This switch defines whether the user id cron job, which checks for differences between the user files in the x86 World and the POWER system, runs on the system. Setting this to none will disable the cron job and the system administrator will not be notified of any conflicts between the user files in each environment.</p>

Environment variable switches

In addition to setting configuration switches in the configuration file, configuration switches can also be set using environment variables. The configuration switches must be set in a POWER shell before invoking PowerVM Lx86 with the runx86 command. To set a configuration switch, the name of the switch must be preceded with P_AVE_CFG_ to ensure that PowerVM Lx86 detects that the configuration switch has been set in the environment. Set the configuration switch in the environment of a POWER shell and invoke runx86 with the following commands in the POWER shell:

```
% export P_AVE_CFG_<CONFIGURATION_SWITCH>=VALUE
% runx86
```

Any applications that are run from this shell will have this configuration switch set.

To reset the switch to its default value, close the running applications, return to the POWER shell where the environment variable was originally set, unset the value and then reinvoke runx86 with the following command in the POWER shell:

```
% unset P_AVE_CFG_<CONFIGURATION_SWITCH>
% runx86
```

You can also start a new POWER shell and invoke a new instance of PowerVM Lx86 with the runx86 command.

Different x86 applications can be run concurrently with different configuration switches set, by invoking them from different POWER shells with the environment variables set to the appropriate values. Some configuration switches effect the global characteristics of PowerVM Lx86 and cannot be set for each shell using environment variables. The configuration switches that can be set using environment variables will be listed in the release notes for each release.

x86World user configuration for PowerVM Lx86

A menu option is provided in the installer to manage and configure users in x86World.

The installation script provides a menu option to manage and configure users in x86World. Support is provided to configure a NIS name space registry. Only local file (the default) and NIS name space registries are supported with this release. All others, including LDAP, are not supported.

1. Become root on the POWER system before starting the installation process.
2. Run the PowerVM Lx86 installation script with the following command in the POWER shell:

```
./installer.pl
```

The following prompt is displayed:

1. Install Software.
2. Upgrade Software.
3. Uninstall Software.
4. Show Installed Products.
5. Configure Software.
6. Quit.

Please select the type of operation you would like to perform:

3. Enter 5. The following prompt is displayed:

```
=====
Please select the type of operation you would like to perform: 5
```

1. Bind NIS for x86 World.
2. Go back to the Main Menu.

Please select the type of operation you would like to perform:

```
=====
```

4. Enter 1. The following prompt is displayed:

Would you like to bind NIS for x86 World? [y/N]

5. To set NIS user support, enter y. This configures NIS support within x86World. The following output is displayed, and then the main installation menu is displayed again:

Binding NIS...

Done.

1. Install Software.
2. Upgrade Software.

3. Uninstall Software.
4. Show Installed Products.
5. Configure Software.
6. Quit.

Please select the type of operation you would like to perform:

NIS has now been configured.

6. Quit the installer by entering 6.

Starting the PowerVM Lx86 daemon and x86 daemons

You can run x86 daemons in the VxE.

The PowerVM Lx86 translation daemon must be running in order to run x86 applications on the POWER machine.

The PowerVM Lx86 installer starts the PowerVM Lx86 daemon on the system after the installation is completed. PowerVM Lx86 installs the `/etc/init.d/p-ave` script to start the PowerVM Lx86 daemon and any x86 daemons.

This script runs whenever the POWER system is booted or the runlevel changes to 2, 3, or 5. The script triggers the `rc` script for the x86 distribution (`rc` is responsible for starting and stopping services on runlevel changes) and starts the appropriate x86 services according to the current runlevel of the POWER distribution. For example, if the current runlevel on the POWER distro is 3, the x86 distro will start the x86 services configured for runlevel 3.

To start the PowerVM Lx86 daemon manually, run the `/etc/init.d/p-ave` script as root. The output is as follows:

```
% /etc/init.d/p-ave start
Starting p-ave-daemon...ok
Started p-ave-daemon [ OK ]
```

You can now start a simple x86 binary using the **runx86** command. For example, the `ls / x86` command should give output similar to that shown in the following example, run in a POWER shell:

```
% runx86 /bin/ls /
bin dev home lib mnt proc sbin srv tmp var
boot etc initrd media opt root selinux sys usr
```

Uninstalling PowerVM Lx86

The semi-automated `installer.pl` script is used to uninstall PowerVM Lx86 and x86World from your system.

The script can remove the following:

- PowerVM Lx86 software and configuration (optional)
- x86World software and configuration (optional)
- Files in x86World that have been modified or added following the original installation (optional)

You can choose to remove only the PowerVM Lx86 software and configuration without affecting x86World, or also to remove applications installed within x86World. This is useful if you are replacing the PowerVM Lx86 translation software with a newer version.

Notes:

- The **rpm -e** command cannot remove PowerVM Lx86. RPM cannot remove any files added to the installation directories, including any applications that have been installed within x86World. The `installer.pl` script can, if you so choose, completely remove the installation directories and contents.
- You must run the `installer.pl` script from a POWER shell, and not an x86 shell. This is because the `installer.pl` script uses the POWER RPM system.

To uninstall PowerVM Lx86, follow these steps:

1. Run the PowerVM Lx86 uninstall script with the following command:

```
# ./installer.pl
```

The script shows a message similar to the following:

Welcome to the PowerVM Lx86 (System p AVE) Installer

To run Linux/x86 applications the installation will require Linux/x86 application binaries, libraries and infrastructure files as well as the PowerVM Lx86 (System p AVE) software.

Please refer to the PowerVM Lx86 (System p AVE) Administration Guide and Release Notes for further details of the installation requirements.

License file found

Registration with IBM

Existing registration with IBM has been found in
/etc/opt/p-ave/activation-detail.
Would you like to re-register? [y/N]

2. If you want to register again, enter `y`. If not, enter `n` or press the Enter key to accept the default `n`. If you choose to skip reregistration, the system displays the following installation menu:

Registration with Red Hat Inc.
Would you like to register? [y/N]

3. If you want to register again with Red Hat, enter `y`. If not, enter `n` or press the Enter key to accept the default `n`. If you choose to skip the Red Hat reregistration, the system displays the following installation menu:

1. Install Software.
2. Upgrade Software.
3. Uninstall Software.
4. Show Installed Products.
5. Configure Software.
6. Quit.

Please select the type of operation you would like to perform:

4. Select option **3. Uninstall Software** from the menu.

The following menu is displayed:

1. Uninstall p-ave + x86 World.
2. Uninstall p-ave.
3. Uninstall x86 World.
4. Go back to the Main Menu.
5. Quit.

Please select the type of operation you would like to perform.

5. Select option **1. Uninstall p-ave + x86 World** from the menu.
The system prompts you with the following message:
Are you sure you want to uninstall p-ave? [Y/n]
6. Press Enter to accept the default yes, or enter y.
The system prompts you to remove the log file directory:
Remove log file directory (/var/opt/p-ave/log)? [y/N]
7. Enter y and followed by the Enter key. Note that by default, the log file directory is not deleted so that any log files contained in the directory are saved for future reference.
The system prompts you to remove the x86 World:
Delete x86 World (installed at /i386) from the filesystem?
You will lose all custom config and installed applications. [y/N]
8. Enter y to remove the x86 World. All the files stored in the X86WORLD_ROOT are deleted. Back up any critical data before carrying out this step.
The system prompts you to remove the configuration files for PowerVM Lx86:
Remove the installation config directory (/etc/opt/p-ave)? [y/N]
9. Enter y to remove the configuration directory and its contents.
The system shows the following:
Uninstalling p-ave. Please wait...
p-ave installation directory (/opt/p-ave) deleted.
Log file directory (/var/opt/p-ave/log) deleted.
Uninstalling x86 World. Please wait...
Config directory (/etc/opt/p-ave) deleted.

Thank you for using the PowerVM Lx86 (System p AVE) Installer

Today you performed the following transactions:
Uninstalled p-ave
Uninstalled x86 World

End of Transaction log.

The full log of your session is in /tmp/p-ave_install_XXXXXX.log
The uninstall of PowerVM Lx86 and the x86 World are now complete.

Upgrading PowerVM Lx86

You can upgrade earlier versions of PowerVM Lx86.

Upgrading previous versions of PowerVM Lx86

For systems with a previous installation of PowerVM Lx86 either upgrade to the new version of PowerVM Lx86 or uninstall PowerVM Lx86 and the x86 World and install the new version of the software.

Check the PowerVM Lx86 release notes to see if upgrading or reinstallation is recommended for your version of PowerVM Lx86.

To upgrade PowerVM Lx86 to the new version, use the following instructions:

1. Run the installer.pl script.
2. Select option **2. Upgrade Software** from the menu.
3. Select option **1. Upgrade a p-ave installation**.
4. Select option **1. version name** to upgrade from the current version of the software.

5. Select option **6. Quit** from the menu.

Uninstalling previous versions of PowerVM Lx86

To uninstall the previous version of PowerVM Lx86 and the x86 World before installing PowerVM Lx86 release 1.2.0.0, use either the installer provided with PowerVM Lx86 release 1.2.0.0 or the one provided with the previous release.

Uninstalling the x86 World will delete all the files within the x86 World. Linked-in files such as the /home directory will not be deleted. You must back up any critical files before uninstalling. You must answer yes when prompted to remove the various directories from the previous installation.

To uninstall a previous version, use the following instructions:

1. Check that all running x86 processes have terminated, then run the installer.pl script.
2. To uninstall the x86 World and PowerVM Lx86, select option **1. Uninstall Software**, and then select option **1. Uninstall p-ave + x86 World**.
3. Answer y to uninstall p-ave.

The following messages appear:

```
Uninstalling p-ave. Please wait...  
p-ave installation directory (/opt/p-ave) deleted.  
Log file directory (/var/opt/p-ave/log) deleted.
```

4. Answer y to remove the log file directory. This is optional. Only remove the directory if you do not require any PowerVM Lx86 log files contained in the directory.
5. Answer y to delete the x86 World.
6. Answer y to remove the installation configuration directory.

The installer script will check if any x86 processes are still running and then shutdown the PowerVM Lx86 daemon (p-ave-daemon) before uninstalling the software.

Now install PowerVM Lx86 release 1.2.0.0 and the x86 World using the installation script provided with the release by running the installer.pl script. See Installing PowerVM Lx86 and x86World.

PowerVM Lx86 installation reference

The details of a PowerVM Lx86 installation include PowerVM Lx86 directories and default configurations.

PowerVM Lx86 directory structure

The RPM files will create and install the software to a pre-set directory structure, as shown in this chart:

Directory	File or sub-directory	Description
/opt/p-ave/bin/	p-ave p-ave-daemon i386-router runx86 p-ave-world-sync	Core translator program PowerVM Lx86 translator daemon Router to launch x86 applications Script to invoke PowerVM Lx86 Script to manager local user files
/opt/p-ave/installer	installer.pl Maiken.pm Maiken/ resources/	Local copy of PowerVM Lx86 installer Auxiliary script for installer Auxiliary installer globalization text and files Installer RPMs and license files
/opt/p-ave/lib/perl5	pAVEscripts	PowerVM Lx86 globalization support files
/opt/p-ave/locale	<locale_directories>	PowerVM Lx86 globalization files
/etc/init.d	p-ave p-ave-rc2 p-ave-rc3 p-ave-rc5 p-ave-rccommon	Start-up script to start PowerVM Lx86 daemon x86 init.d runlevel script x86 init.d runlevel script x86 init.d runlevel script x86 init.d runlevel script
/etc/opt/p-ave	config	Configuration file, created if PowerVM Lx86 or x86 World are installed into non-default locations
/etc/opt/p-ave	activation-detail	Contents of IBM activation e-mail created during installation.
/etc/opt/p-ave/license	license	Copy of license agreed to during installation
/var/opt/p-ave	log/	Contains PowerVM Lx86 debug log files
/var/opt/p-ave/daemon	p-ave-daemon.log	Contains event logs for the PowerVM Lx86 daemon
/var/opt/p-ave/daemon	p-ave-daemon.lock	The lock file exists while the daemon is running.
/var/opt/p-ave/daemon	cache	PowerVM Lx86 daemon cache

Directory	File or sub-directory	Description
/i386	<x86World files>	The default directory for the x86World files. When x86 applications are run, they see this directory as root (/)
/usr/bin	linkx86 runx86	Script to creates escapes from the x86World Script to invoke PowerVM Lx86
<i>PowerVM Lx86 install location/installer</i>	installer.pl lib/ resources/	Local copy of PowerVM Lx86 installer Auxiliary installer files and installer globalization text Installer RPMs and, license files, and documentation
<i>PowerVM Lx86 install location/doc</i>	<i>Release notes</i> <i>Administration Guide</i>	Symlink to the Release Notes and the PowerVM Lx86 Administration Guide (all locales)

Default PowerVM Lx86 links and escapes

The following escapes and links out of x86World are created when PowerVM Lx86 is installed. These are required for PowerVM Lx86 to function.

86World path	Escapes to
/dev	/dev
/home	/home (The escape to /home means that home directories are shared by x86 and POWER applications.)
/media	/media
/mnt	/mnt
/selinux	/selinux
/sys	/sys
/etc/group	/etc/group
/etc/hosts	/etc/hosts
/etc/resolv.conf	/etc/resolv.conf
/tmp	/tmp
/var/yp/binding	/var/yp/binding

Chapter 6. Installing and migrating x86 applications onto a PowerVM Lx86 system

This section explains how to install x86 applications onto a POWER platform, and how to migrate existing applications from an x86 platform to a POWER platform.

The general approach to migration is to make the x86 application and data accessible from the POWER machine. This typically means installing the application and copying or mounting the application files. No alteration of application binaries is necessary, and no conversion of the data is necessary.

The combination of an application and the data that it needs is defined here as a *workload*.

The migration has two parts:

1. Migrating the x86 system configuration
2. Installing x86 applications

Migrating the x86 system configuration

User authentication, remote file systems, x86 configuration and daemons, and environment variables are parts of the x86 system that might need migrating.

The installation script installs x86 libraries, commands, utilities, and infrastructure files in x86World. For more information, see Chapter 5, “Running the PowerVM Lx86 installation script,” on page 19. You can add additional packages later to x86World (see Updating the packages and files in x86World in “Running x86 applications reference” on page 40). This section summarizes the following areas of the x86 system that might need migrating:

- User authentication
- Remote file systems
- x86 configuration and daemons
- Environment variables

User authentication

The users within x86World are separate from the users on the POWER system.

The root password for x86World can be the same as the root password for the POWER system, but the root password for x86World is stored and maintained in the password files in x86World.

If you choose to add NIS support to x86World during installation, then all NIS user accounts are visible in x86World and on the POWER system.

If an x86 application creates a new user (for example, during installation), then this user is created in x86World and is not available on the POWER system.

By default, home directories on the POWER system are escaped from x86World during the installation process. All user home directories are then visible to both

x86 and POWER applications. You can optionally set the home directory to not escape if you prefer.

Remote file systems

Non-local file systems can be made accessible from the VxE. This is done by mounting the file system on the POWER side. You then mount the remote file system directly to a mountpoint within x86World, or to a mountpoint external to x86World, and then you use the **linkx86** command from the POWER side to create an escape to the mountpoint. Note that you can only use the **linkx86** command to create an escape on the POWER side, and not the x86 side.

x86 configuration and daemons

It might be necessary to migrate x86 daemons onto the POWER system. They can be automatically started in a VxE on the host system by inserting a suitable script in the *X86WORLD_ROOT/etc/init.d* directory and setting up symbolic links to it in *X86WORLD_ROOT/etc/rc{2-5}.d*. You can use the tools provided by the supported distribution to perform this configuration (chkconfig or YAST). The PowerVM Lx86 startup script runs scripts in this directory when the system boots according to the current runlevel of the POWER distribution. For example, if the current runlevel on the POWER distribution is 3, the x86 distribution will start the x86 services configured for runlevel 3. By default, on runlevels 2, 3, 4, and 5 the standard installation only starts syslogd. For more information about the PowerVM Lx86 startup script, see Chapter 5, “Running the PowerVM Lx86 installation script,” on page 19.

Environment variables

Environment variables set from within an x86 shell apply only to that shell. If more than one x86 shell is open on a system, then the environment variables are completely independent of each other.

Installing x86 applications

This procedure describes how to install x86 applications onto a POWER system.

Installing an x86 application onto a POWER system is the next step after you install PowerVM Lx86 and configure the system. Installing x86 applications consists of the following tasks:

- Install, copy, or set access to the x86 application binaries.
- Transfer data, or set access to application data.
- Create access to required devices, such as remote file systems.

Install, copy, or set access to the x86 application binaries

The x86 application binaries must be either installed in x86World, or made accessible from x86World.

To install binaries into x86World, you can copy them directly into x86World (or to a location accessible from x86World through an escape or a mount point). If x86 application installation scripts or packages exist, then you can copy them into x86World, or an accessible location, and run them within a VxE using the **runx86** command.

You can make application binaries that are already installed on an x86 machine in the network accessible from x86World either by creating an escape from x86World, or by mounting a remote filesystem to a mount point within x86World.

Java™ installation scripts will need the Java runtime libraries installed first (see Installing Java applications in this guide), although many ISV applications provide a Java runtime library as part of the installation.

Transfer data, or set access to application data

If an x86 application requires access to specific data, the data must be made available. You can do this by copying the data to a location within x86World, or making it available through a mount or an escape.

Data stored in files on disk can be transferred between x86 machines and POWER machines without the need for any conversion. Provided the necessary mounts or escapes are created, an x86 application running with a VxE is able to access data stored in files that reside on both x86 and POWER file systems.

Create access to required devices

Some applications might require access to specific devices, such as tape drives. It is recommended that access to these devices are set in the Linux on POWER operating system.

Most applications will not require access to specific devices to be configured. Many common devices such as file storage and network interfaces appear to applications as files or directories in the VxE filesystem.

Standard devices that are not accessible by default, such as CD devices, should be mounted from the POWER system directly to a mountpoint within x86World. Alternatively, you can mount to a mount point external to x86World and then use the **linkx86** command to create an escape to the mountpoint. It is possible to mount the device in the VxE, but this is not the best method.

Following are examples of creating access to required devices.

Example: Creating access to a CD drive

In Linux, CD drives appear as devices in the `/dev` directory, such as the `/dev/cdrom` drive. The CD drives are accessed natively by mounting to a directory in the file system. To make a CD drive accessible from the VxE, you can mount it at a mountpoint accessible from x86World using the POWER **mount** command, or using the x86 **mount** command from within a VxE.

Mounting from within a POWER shell:

The following is an example of mounting a directory for x86World from a POWER shell, and then listing the mounted directory in the translated x86 shell:

```
% mkdir X86WORLDROOT/cdrom
% mount /dev/cdrom X86WORLDROOT/cdrom
mount: block device /dev/cdrom is write-protected, mounting read-only
% runx86
$ ls /cdrom
Copyright README installer ...
```

Mounting from within an x86 shell:

The following is an example of starting an x86 shell from a POWER shell, and then mounting a directory in the translated x86 shell:

```
% runx86
$ mkdir /cdrom
$ mount /dev/cdrom /cdrom
mount: block device /dev/cdrom is write-protected, mounting read-only
$ ls /cdrom
Copyright README installer ...
```

Installing Java applications in x86World

PowerVM Lx86 can run Java applications on a POWER system. Java applications are run using an x86 Java Virtual Machine (JVM) that is running in the VxE.

Note: The correct x86 Java Runtime Environment (JRE) must be installed in x86World before Java applications can be run.

It is possible to have an entirely different POWER JRE installed on the host system that is running PowerVM Lx86. These libraries do not affect the x86 Java applications.

Supported Java versions

PowerVM Lx86 supports the x86 versions of Java 2 Platform Standard Edition (J2SE) version 1.4 and higher from both Sun and IBM.

To run Java applications, you must install either of the following libraries: J2SE Runtime Environment (JRE) 1.4 or higher, or J2SE Development Kit (JDK) 1.4 or higher.

Configuring x86World for Java

The only configuration required is for the JRE to be installed in x86World, just as on a native x86 machine. Typically this means that it is installed in the `X86WORLD_ROOT/usr/bin/` directory. Note that Java must be installed from an x86 shell.

Example: installing IBM's J2SE 1.4 runtime binaries:

1. Download the `IBMJava2-142-ia32-JRE-1.4.2-8.0.i386.rpm` file from <http://www.ibm.com/developerworks/java/jdk/linux/download.html> to `X86WORLDROOT`.
2. Start an x86 shell by entering the following command in a POWER shell:
`runx86`
3. Install the RPM by running the following command in the translated x86 shell:
`rpm -ivh IBMJava2-142-ia32-JRE-1.4.2-8.0.i386.rpm`
4. Follow the installation prompts. Remember that the x86 shell is jailed, so the `X86WORLDROOT/usr/bin/` directory appears as `/usr/bin/`.

Running Java applications

You run Java applications using the **runx86** command, just like any other x86 applications. No special switches or system daemons are required.

Running x86 applications

You can use the **runx86** command from a native POWER shell.

All x86 applications must be run within a VxE. The **runx86** command must always be used to ensure that an application, command, or utility is run within a VxE.

Applications can be run in a VxE in either of the following ways:

- Use the **runx86** command to start an x86 shell. From within the x86 shell, run the x86 application as you normally would on an x86 system.
- Run x86 applications from a native POWER shell by using the **runx86** command with the x86 application as an argument to the command.

Notes:

- When invoking an x86 application, the current working directory must be accessible from x86World. Either the application must be installed within x86World or be visible from x86World. You can do this by either mounting the application directly to a mount point within x86World, or to a mount point external to x86World. Then, use the **linkx86** command to create an escape to the mount point.
- The **runx86** command does not automatically start x86 system daemons. If your application requires a specific daemon that is not already available and running on the POWER host system, you might need to install and run the x86 daemon before the application will run. Where possible, it is best that the POWER version of a daemon is run. It is possible to configure your system so that x86 daemons are started when Linux starts.

Running x86 applications from an x86 shell

Using an x86 shell is the most flexible way of running x86 applications, but it has disadvantages in that the starting process is manual, and in that it uses a command-line interface. For these reasons, running applications from an x86 shell is generally only suited to applications that are run by experienced users or system administrators. An example of this is a middleware application that is run on an application server.

Example: Running applications from an x86 shell

This example shows how to run an application named TradeOffice. It normally runs on a Linux on x86 machine on a network. TradeOffice monitors a designated remote file system, processes files from the file system, and sends files to another remote file system.

In a POWER shell, enter the following command:

```
runx86
```

The **runx86** command creates a VxE and starts an x86 shell from the native shell.

In the translated x86 shell, enter the following command:

```
TradeOffice
```

This starts the TradeOffice application from the x86 shell.

For applications that must be run “on demand,” it is best to run x86 applications from a native POWER shell.

Running x86 applications from a native POWER shell

Applications can be started directly from a native POWER shell by passing them to the **runx86** command as a parameter. The path to the executable must be a subpath relative to the x86World root (for example, /bin/ls). Arguments are passed directly to the x86 application, so any paths that are passed as arguments should be subpaths below the x86World root (for example, /tmp rather than *X86WORLD_ROOT*/tmp).

This method has the advantage that it can be set up as a script that an end user runs. The end users do not need to know that they are running the application in a VxE on a POWER machine.

Running applications directly from a native POWER shell

This example translates the x86World /bin/ls binary, and lists the contents of the directory.

You must enter the command from a directory that is visible from x86World. See “PowerVM Lx86 installation reference” on page 31 in this guide for a list of directories that are automatically visible from x86World. For example, you can run the following command from a POWER shell:

```
runx86 /bin/ls /tmp
```

This command creates a VxE, translates the **ls** command, shows the results of the **ls** command, and then closes the VxE.

Note: x86 applications cannot be run directly from a POWER shell without invoking the **runx86** command.

Running x86 applications reference

Running x86 applications reference describes the commands that are involved in getting x86 applications operating within a VxE.

There are two items discussed here:

- The PowerVM Lx86 start-up script that starts the PowerVM Lx86 daemon and then starts any optional x86 daemons running within a VxE.
- The **runx86** command, which starts an x86 application running within a VxE.

The PowerVM Lx86 start-up script

The PowerVM Lx86 installation package installs a startup script for PowerVM Lx86 named /etc/init.d/p-ave. This startup script runs automatically when Linux boots, but you can also run it manually (see the example later in this section).

The PowerVM Lx86 startup script starts the PowerVM Lx86 daemon. The PowerVM Lx86 daemon must be active for PowerVM Lx86 to run x86 applications. It enables x86 processes running within VxEs to communicate with each other.

When the PowerVM Lx86 daemon starts, it creates a /var/opt/p-ave directory. This directory must exist, and have full read and write permissions for all users, for PowerVM Lx86 to work.

PowerVM Lx86 start-up script parameters

The PowerVM Lx86 script takes the following arguments:

- **start:** Checks if the PowerVM Lx86 daemon is started already, if not, then starts the daemon.
- **stop:** Halts the PowerVM Lx86 daemon.
- **restart:** Halts the PowerVM Lx86 daemon, then starts the PowerVM Lx86E daemon again.
- **status:** Reports the current status of the PowerVM Lx86 daemon.

Example of using the PowerVM Lx86 startup script

To stop the PowerVM Lx86 daemon, run the following command in a POWER shell:

```
/etc/init.d/p-ave stop
```

To restart the PowerVM Lx86 daemon, run the following command in a POWER shell:

```
/etc/init.d/p-ave restart
```

Note: You must have root access rights to run the PowerVM Lx86E script.

Stopping the PowerVM Lx86 daemon while x86 applications are running will cause them to shut down.

Installing and updating packages in x86 World

Find information on how to manage the software packages within the x86 World. As with any normal system, refer to your system administrator for advice and best practices before adding and upgrading software in the x86 World.

The x86 World using PowerVM Lx86 on a POWER system should be managed just as if it were an independent x86 system. The x86 World contains a set of x86 libraries, command line tools, applications and other system files, just like a native Linux on x86 file system. You can install new packages and update existing packages using the standard x86 package management tools, such as RPM. More advanced package management tools, such as system-config-packages (RHEL), up2date (RHEL), and YaST2 (SLES) are also supported.

Note: When updating packages in the x86 World, ensure that you do not update them to a newer version of the Linux distribution than the underlying POWER system. For example, if you are running Red Hat Enterprise Linux 4.4 on the POWER system, ensure that you upgrade the POWER system to Red Hat 4.5 before upgrading the x86 World to that version. See Chapter 3, “System and installation requirements for PowerVM Lx86,” on page 11 for more details on which x86 World Linux distribution versions are supported under PowerVM Lx86 with which POWER operating system versions.

Installing and updating packages for Red Hat (RHEL 4) in x86 World

For Red Hat, the up2date tool is recommended for managing packages. You can use up2date to install new packages and download updates from the internet via the Red Hat Network.

In addition to `up2date`, the `rpm` and `system-config-packages` tools can also be used to add packages to the x86 World. The `system-config-packages` tool is not installed by default in a minimum x86 World install.

Configuring up2date for the first time

1. Enter the `X86WORLDROOT`, for example by entering the following command in a POWER shell:
`cd /i386`
2. Run PowerVM Lx86 by entering the following command in a POWER shell:
`runx86`
3. Become root in a translated x86 shell with the command:
`su`
4. Run the `up2date` tool in the translated x86 shell with the command:
`up2date --config`

If you need access to the internet via a proxy, enter it under `httpProxy` (option 11) and then enable the proxy (option 3). Save the settings by pressing Enter.

5. If you see a prompt to install a GPG key, do that by entering the following command in the translated x86 shell:
`rpm --import /usr/share/rhn/RPN-GPG-KEY`

Registering the system with the Red Hat Network (RHN)

This process only needs to be done once per PowerVM Lx86 installation.

1. Become root in a translated x86 shell by typing:
`su`
2. Run the `up2date` in the translated x86 shell tool by typing:
`up2date`

Follow the prompts on the screen. Enter your Red Hat Network registration details. Once complete, you will see the message: "You have successfully registered this System Profile on Red Hat Network."

Adding packages to the x86 World

The `up2date` tool is used to add packages and their dependencies into the x86 World.

1. Become root in a translated x86 shell by typing:
`su`
2. To install a package, use the `-i` command line option to `up2date`. For example, to install `gcc` (and its dependencies) by entering the following command in the translated x86 shell:
`up2date -i gcc`

Updating packages within x86 World

The `up2date` tool can also update packages within the x86 World.

1. Become root in a translated x86 shell by typing:
`su`
2. Perform an online update of the x86 World by entering the following command in the translated x86 shell:

```
up2date --update
```

Installing and updating packages for Novell SLES 10 in x86 World

For Novell SLES 10, the recommended method for managing (adding and updating) packages is to use the YaST tool.

A media source must be set up to allow YaST to manage packages in the x86 World. The media source contains the SLES 10 Linux distribution ISO images from which YaST can access all the x86 packages. The media source must be either on the local file system or on a shared server.

Accessing the SLES 10 media source

The instructions assume that the media source has been created on a shared server (fileserver) which can be accessed by each system that needs additional packages installing.

Copy the SLES 10 Linux distribution ISO images into a suitable directory on the shared server. The instructions assume the ISO are located in `/fileserver/isos/sles10x86`.

The shared server is assumed to already have been mounted on the POWER system to the `/fileserver` directory.

First, ensure the `/fileserver` directory is accessible from within the x86 World:

1. Become root in a POWER shell by typing:
`su`
2. Run the `linkx86` command in the POWER shell to create the escape to `/fileserver`:
`/usr/local/bin/linkx86 /fileserver`

Next, verify that the shared server is accessible from with the x86 World:

1. Enter the `X86WORLDROOT`, by entering the following command in a POWER shell:
`cd /i386`
2. Run PowerVM Lx86 by entering the following command in a POWER shell:
`runx86`
3. List the contents of the media source by typing entering the following command in the translated x86 shell:
`ls /fileserver/isos/sles10x86`

The output from this command should list the ISO images on the shared server. If not, check that the shared server is accessible from a POWER shell and check the steps above.

Configuring YaST to access the media source

Now that the media source is accessible in the x86 World, the next step is to configure YaST to access the media source so that it can find the x86 packages.

1. Become root in a translated x86 shell by typing:
`su`
2. Run YaST by entering the following command in a POWER shell:

yast

The YaST Control Centre will start and you will see a graphical text screen.

3. Select **Software** from the main menu on the left, and press Enter to confirm.
4. Select **Change Source of Installation** from the Software list on the right, and press Enter to confirm.
5. Press Tab to select the **Add** menu and press Enter to confirm.
6. Press Tab to select **Local Directory** from the list, and press Enter to confirm.
7. Press Tab to select **ISO image**, and press Enter to confirm.
8. Press Tab to select **Browse**, and press Enter to confirm.
9. Navigate to the SUSE SLES10 ISO image (/fileservers/isos/sles10x86) from the browse list using the Tab and Arrow keys and use the Enter key to highlight your selection.
10. Press Tab to select **Ok**, and press Enter to confirm. 12.
11. Press Tab to select **Next**, and press Enter to confirm.
12. A License agreement will display. Press Tab to select **Yes** if you agree to the license, and press Enter to confirm.
13. Press Tab to select **Next**, and press Enter to confirm.
14. Add additional media sources if required by repeating steps 5-13 or press Tab to select **Finish**, and press Enter to confirm.

Managing packages

Now that YaST is aware of the media sources, it is possible to add or update packages in the x86 World.

1. From the YaST Software menu, select **Software Management**, and press Enter to confirm.
2. Press Tab to select **Filter**, then **Search**.
3. In the **Search Phrase** field, enter the name of the package you want to install, for example, gcc.
4. In the list of available packages, use the arrow keys to navigate and use the enter key to select packages you want to install. Package dependencies will be automatically resolved.
5. Repeat the previous two steps for all packages you want to install.
6. After you have selected all the packages you want to install, use the Tab key to select the **Accept** button, and press Enter to confirm. A prompt to display the resolved dependencies may appear; press the enter key to confirm. YaST will now install the selected packages.
7. At the **Install or remove more packages** prompt, press Tab to select **No**, and press Enter to confirm.
8. After the installation is complete, you can exit YaST by using Tab to select **Quit** and pressing Enter to confirm.

Installing and updating packages for Novell SLES 9 SP3 in x86 World

For Novell SLES 9 SP3, the recommended method for managing (adding and updating) packages is to use the YaST tool.

A media source must be set up to allow YaST to manage packages in the x86 World. The ISO images for the distribution must be mounted and set up in a particular structure before running YaST.

Package management for SLES 9 using YaST requires the following media:

- “SUSE SLES Version 9” CD (Base CD1)
For example: SLES-9-i386-RC5-CD1.iso
- “SUSE CORE Version 9” CD1 - 4
For example: SLES-9-i386-RC5-CD{2-5}.iso
- “SUSE SLES 9 Service-Pack Version 3” CD1 - 3
For example: SLES-9-SP-3-i386-RC4-CD{1-3}.iso

Creating a media source

All of the ISO images must be mounted in a location that is accessible by PowerVM Lx86 before proceeding with the package management in YaST. In order to do this, each of the ISO images will be mounted on the POWER system to a directory visible from the x86 World.

1. For each of the ISO images above, create a directory in /mnt and mount the ISO image from the POWER machine as shown in this example. Enter the following commands in a POWER shell:
2. After mounting the ISO images into separate directories, another single directory must be created to store the contents of the SUSE CORE files which exist on core discs 2-5 and must be copied from the mounted ISOs, as shown in this example. Enter the following command in a POWER shell:

```
cd /mnt
mkdir SLES-9-i386-RC5-CD1
mount -t iso9660 SLES-9-i386-RC5-CD1.iso SLES-9-i386-RC5-CD1 -o loop
```

3. Copy the files from each of the SUSE CORE ISO images into this directory, as in the following example. Enter the following command in a POWER shell:

```
mkdir /mnt/SUSE-CORE
```

```
cp -r /mnt/SLES-9-i386-RC5-CD2/* /mnt/SUSE-CORE
```

Perform this step for each of the SUSE CORE ISO images, CD2, CD3 and CD4 (ISO image numbers 2 – 5).

Configuring YaST to access the media source

Now that the media source is accessible in the x86 World, the next step is to configure YaST to access the media source so it can find the x86 packages.

1. Become root in a translated x86 shell by typing:
su
2. Run YaST by entering the following command in the translated x86 shell:
yast

The YaST Control Center will start and you will see a graphical text screen.

3. Select **Software** in the main menu on the left, and press Enter to confirm.
4. Select **Change Source of Installation** from the Software list on the right side of the screen, and press Enter to confirm.
5. Press Tab to select the **Add** menu, and press Enter to confirm.
6. Select **Local Directory...** from the list using the Arrow keys, and press Enter to confirm.
7. Press Tab to select **Browse**, and press Enter to confirm.
8. Press Tab and the Arrow keys to find the SUSE SLES 9 Service-Pack Version 3 directory (for example, /mnt/SLES-9-SP-3-i386-RC4-CD1) from the browse list and press Enter to highlight your selection.

9. Confirm your selection by pressing Tab to select **OK**, and press Enter to confirm. Then, confirm your directory path by pressing Tab to select **OK**, and press Enter to confirm.
10. The following line will display on the YaST summary screen:
SUSE SLES 9 Service-Pack Version 3
11. Select the **Add** menu, and press Enter to confirm.
12. Select **Local Directory** from the list, and press Enter to confirm.
13. Select the SUSE SLES Version 9 directory (for example, /mnt/SLES-9-i386-RC5-CD1) from the list, and press Enter. The following message will appear on the screen:
SUSE SLES Version 9
14. Press Tab to select the **Add** menu, and press Enter to confirm.
15. Use the Arrow keys to select **Local Directory...** from the list, and press Enter to confirm.
16. Press Tab to select **Browse**, and press Enter to confirm.
17. Using Tab and the Arrow keys, select the SUSE CORE directory created earlier in the installation process (for example, /mnt/SUSE-CORE) from the browse list, and press Enter to highlight your selection.
18. Confirm your selection by pressing Tab to select **OK**, and press Enter to confirm.
19. The following message will be displayed:
SUSE CORE Version 9
20. Press Tab to select **Finish**, and press Enter to confirm. The Finish button will flash while YaST processes the media sources. Then, the menu will return to the YaST Control Center.

Managing packages

Now that the media source is accessible in the x86 World, the next step is to configure YaST to access the media source so it can find the x86 packages.

1. From the YaST main menu, select **Install and Remove Software**, and press Enter to confirm.
2. Press Tab to select **Filter**, then select **Search**.
3. In the **Search Phrase** field, enter the name of the package you want to install, for example, gcc.
4. In the list of available packages, use the Arrow keys to navigate and press Enter to select the packages you want to install. Package dependencies will be automatically resolved.
5. After you have selected all the packages you want to install, use the Tab key to move to the **Accept** button, and press Enter to confirm. A prompt to display the resolved dependencies may appear, confirm using the enter key. YaST will now install the selected packages.
6. After the installation is complete, you can exit YaST by pressing Tab to select **Quit**, and pressing Enter to confirm.

PowerVM Lx86 log files

PowerVM Lx86 creates log files for the PowerVM Lx86 daemon and any x86 processes that have errors. These log files are not automatically deleted, so you might periodically need to clean them out.

The PowerVM Lx86 daemon log file is created in the `/var/opt/p-ave/daemon` directory, and is always given the name `p-ave-daemon.log`. A new log file is started for the PowerVM Lx86 daemon every time it starts. The `p-ave-daemon.log` file lists communication failures between the translated x86 applications and the PowerVM Lx86 daemon, and internal errors, such as being out of memory.

x86 process log files are created for x86 processes if they produce an error message, warning message, or fail as they are running within a VxE. The log files are created in the `/var/opt/p-ave/log` directory. The filename takes the form `p-ave.log.<process_name>.<process_id>.<unique_id>`.

x86 /etc/init.d support scripts

On a Linux system, the `/etc/init.d` directory contains initialization and termination scripts for setting up subsystems or start-stop services.

Introduction

Every kernel runlevel has a corresponding directory in `/etc/rc{0-6}.d` (for example, `rc0.d`, `rc1.d` etc.) where symbolic links are created to the scripts located under `/etc/init.d/`. When the system boots, reboots, or there is any other runlevel change, the symbolic links are called to start and stop services.

The scripts are called by the `rc` script according to a priority number and script name. In a system with PowerVM Lx86 installed, there are two sets of `init.d` scripts. One set corresponds to the host POWER system, and the other set corresponds to the x86 World.

PowerVM Lx86 contains a number of utilities that enable the execution of `init.d` scripts in the x86 World with every runlevel change on the host system. This infrastructure allows PowerVM Lx86 to start x86 services with `init.d` scripts in the same way as if they were running on the original x86 platform in a completely transparent way for the system administrator.

If a newly installed x86 application adds entries to the x86 World `init.d` scripts, it is handled correctly, with the new services being started or stopped when the system boots or reboots, or when the runlevel changes.

Implementation

A new installation of PowerVM Lx86, and corresponding x86 World, only have a limited number of services enabled. These are only `dbus` and `syslog`. During installation of the x86 World, once all the x86 RPMs are installed, the installer clears the `/etc/rc{0-6}.d` directories, leaving only the necessary services for initial operation of PowerVM Lx86.

After installation, the administrator has the opportunity to start any installed service using the normal utilities provided with the distribution. These typically include `chkconfig` and other distribution specific tools, such as SUSE's `YaST2`. During normal operation, every time the POWER system boots, reboots, or changes runlevel, a PowerVM Lx86 script is run to trigger the execution of the corresponding `init.d` scripts in the x86 World. This PowerVM Lx86 script acts as a wrapper for the `rc` script in the x86 World. These PowerVM Lx86 scripts are named `p-ave-rc{2-5}`, and are installed in the `/etc/init.d` directory in the host system.

Some of the services that normally run on an x86 environment are not necessary in a PowerVM Lx86 x86 World. This implementation checks that these unnecessary services, or any services that conflict with already running POWER services, are disabled. In addition, SUSE allows dependencies to be defined between the scripts. Therefore, as part of the installation process and later as part of the maintenance tasks, a dependency checker will be run to edit the `init.d` scripts and delete any known unnecessary dependencies (for example, `boot.*`, `acpid`, `haldemon`, and so on).

The perl script that edits the `init.d` scripts is named `dependency_checker.pl`, and the installer calls it as the final step in the installation process, and during normal operation when a change in the x86 World `/etc/init.d` directory is detected. Notification of changes in the x86 World `/etc/init.d` directory are sent to a directory monitor called `rc_monitor`. This program receives events when new files are added or when the permissions are changed on existing files. These events trigger the execution of the `dependency_checker.pl` script.

The directory monitor, `rc_monitor`, is installed in `/etc/init.d/p-ave-rcmonitor`. Although this is invoked automatically by the `/etc/init.d/p-ave` script (which also invokes the `p-ave-daemon`), the `rc_monitor` can be controlled independently of the `/etc/init.d/p-ave` script by running `/etc/init.d/p-ave-rcmonitor` manually.

The `p-ave-rcmonitor` has the following usage options: `/etc/init.d/p-ave-rcmonitor [start|stop|force-reload|restart|status]`

Monitoring x86 applications

You can monitor x86 applications using x86 commands.

x86 commands must be run from a VxE or x86 shell. These x86 commands display information about any processes running in an x86 shell or a VxE. POWER processes are not shown.

Processes running within a VxE or x86 shell can also be seen from the POWER host using commands such as the `ps` command and the `top` command. The output is more verbose and shows the translator processes running the x86 applications. These details might not be required if you are only checking which x86 processes are running. However, you might prefer to use a POWER tool that you know is running natively, and use a script to filter out unwanted verbosity.

Following is an example that shows the use of commands to monitor x86 applications. (On this system, the only x86 processes running are `bash` and `ps`.)

From a translated x86 shell, enter the following:

```
ps -A
```

The output is similar to the following:

```
PID  TTY    TIME  CMD
16180 pts/13 00:00:00 bash
16176 pts/11 00:00:00 ps
```

From a POWER shell, enter the following:

```
ps w w ax
```

The output is similar to the following:

```
16097 pts/13  Ss    0:00 -bash
16179 pts/13  S      0:00 /bin/bash /usr/bin/runx86
16180 pts/13  S1     0:01 /opt/p-ave/bin/p-ave /i386/bin/bash
16230 pts/13  S1+    0:00 /opt/p-ave/bin/p-ave -f3ff -argv0 top /i386/usr/bin/top
16252 pts/11  R+     0:00 ps w w ax
```

Maintaining x86 applications

Maintaining x86 applications includes debugging and x86 core dump files.

Debugging

When an in-house x86 application is migrated to POWER, developers might need to build or support the application on the POWER system. Developers cannot use native Linux on POWER debugging tools when the application is running in a VxE, because results in the translator program itself being debugged. Instead, developers should use x86 debugging tools running within a VxE on the POWER machine.

The x86 command line debugging tools `strace` and `ltrace` are supported within a VxE.

Note: Debugging of x86 applications, using a debugger such as `gdb`, is not supported in this version of PowerVM Lx86.

x86 core dump files

Core dump files are supported for x86 processes running in a VxE. If an x86 process crashes unexpectedly while running within a VxE, it can produce a core dump file. If the crash was caused by a problem with the translator, then an error log will also be generated. The translator can also produce a core dump.

Part 3. Managing users, groups, and passwords with PowerVM Lx86

With the creation of the x86 World there are now two definitions of users, groups, and passwords on the system.

Conflicting definitions of users, groups, and passwords

The x86 World that is installed with PowerVM Lx86 comes with its own set of passwords, groups, and shadow files that are normally found on a native POWER system in the locations `/etc/passwd`, `/etc/group` and `/etc/shadow`, respectively. Therefore, with the introduction of the x86 World there are now two definitions of users, groups and passwords on the system.

This can be confusing to the system administrator and to end users, and there is also a potential security risk. Supposing the x86 World is installed in the location `/i386`, consider the two possible scenarios outlined below:

Scenario 1: Consider two users named fred and bob, both of whom share the same user ID. User fred exists in the native POWER `/etc/passwd` file, and user bob exists in the x86 World `/i386/etc/passwd` file. Suppose you change to user fred in a native POWER shell and then run PowerVM Lx86. In the x86 World you are now user bob, since both fred and bob share the same user ID. Not only is this confusing (try running the `id` command, and you can see that your username has changed from fred to bob), but it can pose security issues, because users fred and bob might have different primary groups.

Scenario 2: Now consider that a user named fred exists in both `/etc/passwd` and `/i386/etc/passwd`, but they have different user IDs. In a native POWER shell you log in as user fred and create a file in `/home/fred` that is only readable by fred. Now suppose you run PowerVM Lx86 as a normal user, and then change to user fred and try to read that file. You won't be able to read it because you have different user IDs.

These two scenarios also apply in a similar way to groups. In its default installation, PowerVM Lx86 attempts to manage these problems transparently by presenting the user with a consistent *one system*, or unified, view of users and groups.

Chapter 7. The solution: A unified view

PowerVM Lx86 attempts to unify the definitions of users, groups, and passwords by collecting information from both the native POWER system and the x86 World and producing a merged view of the `/etc/passwd`, `/etc/group` and `/etc/shadow` files (and on Red Hat there is also `/etc/gshadow`).

Again, assuming the x86 World is installed in the `/i386` directory, the solution for `/etc/passwd` follows:

- Whenever a translated x86 program tries to open `/i386/etc/passwd` (the `passwd` file in the x86 World), PowerVM Lx86 instead opens both the native POWER file `/etc/passwd` and the x86 World file `/i386/etc/passwd` at the same time and attempts to merge all the entries together to provide a single view in such a way that all conflicts and inconsistencies are resolved.

For example, if PowerVM Lx86 finds that user bob exists in both `/i386/etc/passwd` and `/etc/passwd` files, then Sytem p AVE favors the bob entry in `/i386/etc/passwd`, and ignores the bob entry in `/etc/passwd`. This solves the problem described in Scenario 2 above.

- Or if PowerVM Lx86 finds users in both `/i386/etc/passwd` and `/etc/passwd` sharing the same user ID PowerVM Lx86 will always favor the native POWER entry and ignore the x86 version. This solves the problem described in scenario 1 above.

Operations on `/i386/etc/group` are treated in a similar manner, which solves both of the issues mentioned previously. However, when conflicting users are found in the shadow files on both the native POWER system and the x86 World, PowerVM Lx86 favors the x86 World entry by default. This is because it is quite reasonable for the root user to require different passwords in both worlds. It is possible to configure PowerVM Lx86 to always favor the POWER shadow entries by setting the `FU_HAVE_SEPARATE_PASSWORDS=y` configuration variable in the standard p AVE configuration file.

By default, PowerVM Lx86 operates in this one system mode; however, it is possible to revert PowerVM Lx86 back to a two-system mode by setting the `FU_MERGE_PASSWD_FILES=n` configuration variable in the standard PowerVM Lx86 configuration file. This means that the `/etc/passwd`, `/etc/group`, and `/etc/shadow` (and on Red Hat there is also `/etc/gshadow`) files are handled completely separately by PowerVM Lx86 and the native POWER system. This is not recommended unless it is explicitly known that there are no security risks involved in doing so.

Chapter 8. Periodic checks of the subject x86 World

As part of the PowerVM Lx86 installation, a cron job is installed in `/etc/cron.d/p-ave` that invokes the `p-ave-world-sync` script (by default located in `/usr/sbin`). This periodically checks the x86 World to see if the password, group, or shadow files have become inconsistent (that is, there is a difference between the corresponding x86 World and native POWER file).

From a security perspective, PowerVM Lx86 checks for aliased user IDs (different user names with the same user ID) and new users present in the x86 World but not in the native POWER system. It is the responsibility of the system administrator to adjust the periodicity and timing of the cron job.

By default, the cron job is installed to check the environments every twenty minutes. If a problem is found, then a message is logged to `/var/log/messages`, and an e-mail is sent to the root user. This e-mail contains clear guidance about how best to resolve the inconsistency by using standard Linux utilities available on the native POWER system.

If the system administrator does not care that certain users or groups are present in the x86 World but not in the native POWER system, then it is possible to configure the cron job to not report about such users or groups. This can be achieved by creating a *white list* of users and groups in the `/etc/opt/p-ave/user_ignore` and `/etc/opt/p-ave/group_ignore` files, respectively. For example, if the system administrator knew about users fred, jane, and bob, but did not want to be warned about them, the administrator could generate the `/etc/opt/p-ave/user_ignore` file with the following contents:

```
fred
bob
jane
```

Each user needs to be on a new line. The same applies for groups.

It is possible to disable this cron job by editing the PowerVM Lx86 configuration file to set the `WORLD_CHECK_OR_SYNC=none` variable. However, this causes the system administrator to have no visibility of issues as they arise. For further details of the options for the `WORLD_CHECK_OR_SYNC` configuration switch, see Chapter 10, “`WORLD_CHECK_OR_SYNC` options,” on page 59.

Although the cron job does not modify any of the native POWER or x86 World password, group, or shadow files, there are cases when PowerVM Lx86 will update the actual x86 World files on disk. In these circumstances, the x86 World files will be synchronized physically on disk by PowerVM Lx86 with the equivalent merged view. Three cases where this could occur are:

- A user manually adds, deletes or modifies a user or group.
- A user manually changes a user or group password.
- Users or groups are added automatically as part of an application installation such as WebSphere® or DB2®.

PowerVM Lx86 will always present a consistent merged view of the password, shadow, or group files even though this virtual file is not synchronized physically

on disk all the time. Any updates to the physical files will be reflected in the merged view. In any of these cases, the cron job will detect the presence of new users or groups and warn the system administrator accordingly.

Chapter 9. Known issues with the unified view approach

Find some known issues with this unified view approach that the system administrator should understand.

- It is possible to enable NIS for the x86 World. If NIS is enabled in the x86 World, NIS users will be visible when running PowerVM Lx86. If NIS support is not enabled in the x86 World, PowerVM Lx86 will ignore any NIS entries from the native POWER system.
- PowerVM Lx86 always favors the native password entries. In the case of a user ID clash there is a danger that the user's home directory will not be visible within the subject world. That is, consider these entries: /etc/passwd: 'fred:x:30003:12113::/fred:/bin/bash' /i386/etc/passwd: 'bob:x:30003:12113::/bob:/bin/bash' Since PowerVM Lx86 favors user fred from the native POWER system, it is possible that the /i386/fred directory does not actually exist within the x86 World. However, the cron job should detect any such issues, and provide the system administrator with a means to resolve them.
- It is possible for entries to appear and disappear in a translated x86 shell. For example, consider the following sequence of events:
 1. A system administrator adds user fred in the x86 World and then logs in as user fred. The administrator then runs the id utility, which shows output such as: 'uid=30001(fred) gid=500(some company) groups=17(audio),500(some company)'
 2. The administrator then adds user bob in a native POWER shell, which happens to be allocated the same user ID (30001). The system administrator then runs a translated x86 shell and again runs the id utility, now producing the following output: 'uid=30001(bob) gid=500(some company) groups=18(uucp),500(some company)'
 3. If the system administrator now returns to a native POWER shell, deletes user bob, then runs the 'id' utility again in a translated x86 shell, the output will be as before: 'uid=30001(fred) gid=500(some company) groups=17(audio),500(some company)'. As above, the cron job will periodically check the x86 World for any such inconsistencies and warn the system administrator

Chapter 10. WORLD_CHECK_OR_SYNC options

The WORLD_CHECK_OR_SYNC configuration switch has multiple options which effects which of the system files are checked by PowerVM Lx86.

The options are shown in the following chart:

Configuration switch value	Effect
check_all	The default value. The cron will check the passwd and group files for any issues.
check_passwd	The cron will only check the passwd files for any issues. The group files will be not be checked for issues with group ids.
none	The cron will only check the group files for any issues. The passwd files will not be checked for issues with user ids.
sync_all	This will disable the cron job without removing the cron files. No checking of the passwd or group files or updates to the /etc/mtab files will occur.
force_sync_mtab	Performs the same checks are the check_all option, but in addition the cron will keep the /etc/mtab file in the x86 World up to date with entries found in POWER /proc/mounts whenever the cron runs.
	The cron will only keep the /etc/mtab file in the x86 World up to date with entries found in POWER /proc/mounts whenever the cron runs. No checking of the passwd or group files will occur.

It is possible to invoke the p-ave-world-sync script directly with the force_sync_mtab option. This forces the x86 World mtab to be updated in sync with the POWER version of the file. This is only expected to be used in a situation where the x86 World mtab file has become corrupt or out of date.

Invoke the script as root with the command: /usr/sbin/p-ave-world-sync force_sync_mtab

Note: The PowerVM Lx86 daemon must be running for this operation to succeed.

Chapter 11. PowerVM Lx86 translation process

The PowerVM Lx86 translation process is multi-stage and iterative.

Once an x86 application is loaded into memory, it undergoes a continuous process of translation and optimization. This is shown in the following diagram.

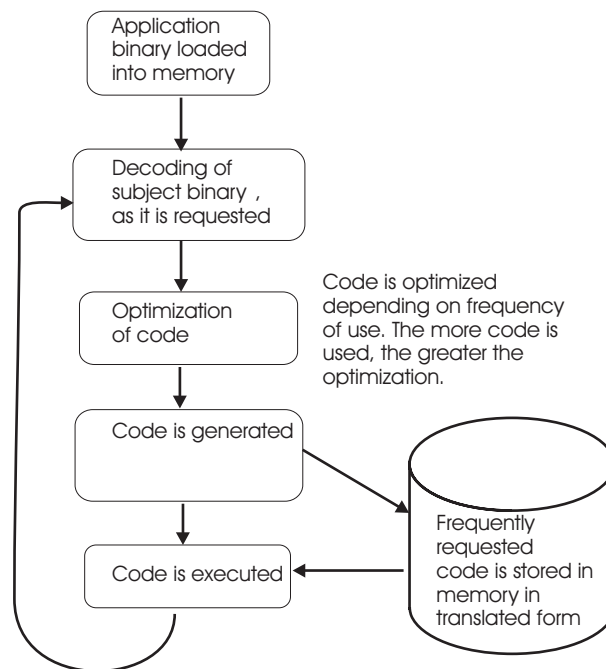


Figure 4. PowerVM Lx86 translation process

As the x86 application runs, PowerVM Lx86 dynamically translates the x86 code into POWER code. The translation is a three-stage process:

1. Decoding of subject binary: x86 binary instructions are decoded as the application requests them.
2. Optimization: The optimization is iterative, so more optimization is done on frequently-used code.
3. Generation of POWER code: Frequently-used code is stored in memory, so it does not need retranslating the next time it runs.

Linux on x86 system calls

x86 applications use system call instructions to request services from the x86 kernel.

The translator maps x86 system call instructions to their POWER equivalents.

System resources, binaries, and files

x86 applications need access to system resources, and to be able to address their own data and files as if they reside on an x86 system. They also need to be able to access files in the Linux on POWER system.

System resources such as x86 applications are translated as they are run. This means that the x86 applications can interact with POWER system resources as if they were a native POWER application. System resources include graphics, disk access, users, and network devices.

x86 binaries and libraries are all installed in one location on the same machine as the translator. The translator ensures that x86 applications can access the binaries and libraries that they require. See Chapter 2, “PowerVM Lx86 concepts,” on page 5 in this guide for more information about x86 World, jailing, and escapes.

Linux on POWER files and resources can be accessed by x86 applications. This may require some configuration.

Chapter 12. Default PowerVM Lx86 escapes and virtual files in the x86 World

This section describes how virtual files are handled and how they are useful for system administrators managing a system with PowerVM Lx86 installed. In addition, the PowerVM Lx86 installer sets up some escapes by default that allow PowerVM Lx86 access to certain directories, files, and sockets on the POWER system.

Virtual files and directories

PowerVM Lx86 maintains some system files as virtual files.

Passwd, group, and shadow files

See Part 3, “Managing users, groups, and passwords with PowerVM Lx86,” on page 51 for information on how PowerVM Lx86 manages user administration files within the x86 World. PowerVM Lx86 manages the files in the following table:

Name	Type of virtual file	Directory (D), File (F), or Socket (S)
/etc/passwd	Merged	F
/etc/group (optional during installation)	Merged	F
/etc/gshadow (RHEL only)	Merged	F
/etc/shadow	Merged	F

These files are virtual, and the contents are managed by PowerVM Lx86. In addition, physical files exist for these in the x86 World. The virtual file is a merged file, meaning that the contents are generated by merging the contents of the x86 World physical file and the POWER system version of the file.

If the files are written by a translated x86 application or shell, the underlying physical file in the x86 World will be updated. The change will be visible in the combined virtual view generated by PowerVM Lx86. In addition, any changes in this file will be detected by the user ID cron job, and the system administrator may be notified of any inconsistencies between the x86 World file and the POWER system file. See Part 3, “Managing users, groups, and passwords with PowerVM Lx86,” on page 51 for more information.

utmp and wtmp

Name	Type of virtual file	Directory (D), File (F), or Socket (S)
/var/run/utmp	Contents as POWER system	F
/var/log/wtmp	Contents as POWER system	F

Both of these files are created during installation of the x86 World. In addition, physical files exist for these in the x86 World. In this case, the physical files are just

stubs and will not be updated by any translated x86 application or shell. The virtual files have exactly the same contents as the equivalent files on the POWER system.

Writing to these from a translated x86 application, or while in a translated x86 shell causes an update to the file on the POWER system. There is effectively only one version of the file on the entire system – the one on the POWER system.

Updates to the POWER system version of the file are reflected in the virtual view of the file from within the x86 World.

If the physical x86 World files are opened from a non-translated shell (for example, by running `vi /i386/var/run/utmp`), then any changes are written back to the physical stub, but PowerVM Lx86 ignores the changes, and they are not visible to translated applications. They will only see the contents of the virtual file.

Do not delete the stub files. If you delete the `utmp` or `wtmp` x86 stub files this will normally have no effect, but it could cause a failure in applications which attempt to find the files, since the files will not appear in a directory listing.

Note:

- It is only possible to delete either of the stub files using a native POWER shell (for example, `rm /i386/var/run/utmp`). If you attempt to delete the files from a translated x86 shell, it will fail.
- Editing the stub files from an x86 shell is not recommended. They are binary files and contain specific data structures. Editing the files manually may corrupt the data structures and the underlying POWER versions of the files.

`/etc/resolv.conf`

Name	Type of virtual file	Directory (D), File (F), or Socket (S)
<code>/etc/resolv.conf</code>	Contents as POWER system	F

This file is virtual, PowerVM Lx86 manages the contents, and it is the same as the POWER system version of the file. In addition, a physical file exists in the x86 World. The contents of the physical file are an exact copy of the POWER system version of the file at the point when PowerVM Lx86 is installed.

The virtual file can be read from a translated x86 application, or while in a translated x86 shell, but it cannot be written to. If you need to update the contents of the file, you must update the POWER system version of the file. Once the file has been updated, the change is seen by applications in the x86 World.

It is possible for PowerVM Lx86 to manage the x86 World `/etc/resolv.conf` file independently of the POWER system version in advanced mode. In this case, the physical file is visible to translated x86 applications and from a translated shell.

`FU_HAVE_SEPARATE_RESOLV_CONF_FILES=y`

The configuration switch to enable advanced mode. This causes the contents of the x86 file to be used. Any changes to the x86 file will be seen by translated x86 applications. In this mode, if you change the POWER file, this will not be seen by translated applications.

`FU_HAVE_SEPARATE_RESOLV_CONF_FILES=n`

To disable advanced mode, either set the switch to this or remove the line

from the configuration file. The contents of the POWER file will be visible to translated x86 applications. The x86 file will not be affected.

If you try to delete the /etc/resolv.conf file from an x86 shell you will get an error. The x86 version of the file must exist so that it can be opened, but its contents are virtual and appear the same as the POWER file. The x86 version of the file can be deleted from a POWER shell (for example, `. rm /i386/etc/resolv.conf`). Once deleted, the file will not be visible to x86 applications. After deleting the x86 file, the user can recreate the file from a POWER shell. The contents of the x86 file are ignored and will always have the contents of the POWER file.

/proc

Name	Type of virtual file	Directory (D), File (F), or Socket (S)
/proc	Entirely virtual	D

The /proc directory is created by the PowerVM Lx86 installer. There are no physical files in the /proc directory. If inspected from a non-translated shell, the directory appears to be empty (for example, `ls /i386/proc`). The entire contents of the /proc directory are virtual and managed by PowerVM Lx86 to represent the detailed processor and system information of an x86 platform. The specific contents of the /proc tree vary depending on the specific OS distribution that is installed for the x86 World and on the underlying POWER system.

Some of the directories and files in /proc are very specific to the x86 hardware and are not supported by PowerVM Lx86. PowerVM Lx86 will deny access if any attempts made to access these directories and files.

The following /proc directories and files are not supported by PowerVM Lx86:

Unsupported /proc file	Directory (D), File (F), or Socket (S)
/proc/acpi	D
/proc/asound	D
/proc/bus	D
/proc/dri	D
/proc/driver	D
/proc/irq	D
/proc/apm	F
/proc/config.gz	F
/proc/cpufreq	F
/proc/iomem	F
/proc/ioports	F
/proc/mm	F
/proc/mtrr	F
/proc/pci	F
/proc/sysrq-trigger	F

Default escaped directories, files, and sockets

Directories and files are created as escapes when PowerVM Lx86 is installed. These are required for PowerVM Lx86 to function.

x86 World path or filename	Directory (D), File (F), or Socket (S)
/dev	D
/home (optional during installation)	D
/media	D
/mnt	D
/selinux (RHEL only)	D

As an example, /home in the x86 World (the directory itself is visible as /i386/home from a POWER shell) is escaped to /home on the POWER system. This means that x86 and POWER applications share home directories on the system. See “Jailing and escapes for PowerVM Lx86” on page 7 for more details.

Syslog support

Support for logging system messages is handled in a special manner for PowerVM Lx86 so that system messages in the x86 World and the POWER system are kept separate.

Kernel messages are only logged by the POWER system and are found in /var/log/messages by default. System messages generated by POWER applications are also logged there by default. System messages generated by translated x86 applications are logged in /var/log/messages in the x86 World (for example, /i386/var/log/messages from a POWER shell).

System logging background

It is possible to run the system logging daemons within x86 World, but the sockets and files they use are handled specially by PowerVM Lx86. The /etc/init.d/syslog script starts two daemons: klogd and syslogd. The klogd daemon is responsible for collecting any messages that have come directly from the kernel. It can do this by either reading /proc/kmsg (default) or by making the syslog system call. If there is no data to read in /proc/kmsg klogd blocks and waits for data to appear. When klogd gets messages from the kernel, it passes them to the syslogd daemon via the /dev/log socket. The syslogd daemon waits on the /dev/log socket for data which can come from either klogd or directly from a user program such as initlog or logger. The messages are then written to the /var/log/messages file.

System logging with PowerVM Lx86

In the x86 World, the syslog system call, /proc/kmsg file, and /dev/log files are handled in a special manner. If a translated x86 klogd process tries to read data from /proc/kmsg, PowerVM Lx86 will never read /proc/kmsg, but instead blocks the process by never returning any data. Therefore, kernel messages are not logged into the x86 World /var/log/messages file.

Kernel messages are only logged by the POWER system and will be found in /var/log/messages by default.

Any translated x86 processes such as syslogd that perform operations on the /dev/log socket will work as normal, however, PowerVM Lx86 will not open /dev/log, but instead will open the file /var/opt/p-ave/devLog instead. All operations on /dev/log will map directly to the socket file /var/opt/p-ave/devLog. If an x86 process tries to delete /dev/log it will actually correspond to deleting /var/opt/p-ave/devLog. The system logs will be written to /var/log/messages in the x86 World (for example, /i386/var/log/messages from a POWER shell). Only messages from translated x86 applications are logged in the /var/log/messages file in the x86 World. All other messages are logged in /var/log/messages on the POWER system. This ensures that translated x86 processes cannot collect messages either from the kernel or other POWER processes.

Note: The /dev directory is escaped by default in the x86 World to the /dev directory on the POWER system. The /dev/log file is a special case and is not escaped.

x86 World log socket	Maps to	Directory (D), File (F) or Socket (S)
/dev/log	/var/opt/p-ave/devLog	S

Part 4. PowerVM Lx86 error messages and resolutions

See the error messages that can be reported on the terminal by the components of PowerVM Lx86 and details of how to resolve each problem.

If the resolution section for each error does not help you to resolve the problem, report the failure by contacting IBM Support.

There are several components of PowerVM Lx86 that can produce error messages:

- The translator itself (p-ave)
- The translator daemon (p-ave-daemon)
- The x86 /etc/init.d scripts
- The user id support (p-ave-world-sync)
- The PowerVM Lx86 installer

Error message template

The PowerVM Lx86 errors messages use the following template: [Module][Error: xxxx]<error text>

Where [Module] is p-ave, p-ave-daemon, or similar, xxxx in [Error: xxxx] is a unique error number for that module starting from 0001, and <error text> is plain text describing the error and its possible resolution.

Chapter 13. Alerts and errors when managing users, groups, and passwords with PowerVM Lx86

This section describes alert and error messages, including cause and resolution.

E-mail alerts

The following alerts are sent via e-mail by the p-ave-world-sync script to root and logged in /var/log/messages on the POWER system when conflicts are found with the users, groups and passwords with PowerVM Lx86 and the underlying POWER system.

Alert message	New user account (<useraccount>) found in <file>
Cause	A new user account has been added to the file (for example, /etc/passwd) in the x86 World.
Resolution	The root user will be e-mailed specific details about how to resolve the issues with this event. See Chapter 21, "E-mail messages for managing users, groups, and passwords in PowerVM Lx86," on page 91.

Alert message	Aliased user id (<userid>) found in <file>
Cause	A username has been found in the file (for example, /etc/passwd) which has the same user id as an entry in the POWER version of the file.
Resolution	The root user will be e-mailed specific details about how to resolve the issues with this event. See Chapter 21, "E-mail messages for managing users, groups, and passwords in PowerVM Lx86," on page 91.

Alert message	New group (<group>) found in <file>
Cause	A new group has been added to the file (for example, /etc/group) in the x86 World.
Resolution	The root user will be e-mailed specific details about how to resolve the issues with this event. See Chapter 21, "E-mail messages for managing users, groups, and passwords in PowerVM Lx86," on page 91.

Alert message	Aliased group id (<groupid>) found in <file>
Cause	A group name has been found in the file (for example, /etc/group) which has the same group id as an entry in the POWER version of the file.
Resolution	The root user will be e-mailed specific details about how to resolve the issues with this event. See Chapter 21, "E-mail messages for managing users, groups, and passwords in PowerVM Lx86," on page 91.

Errors from the p-ave-world-sync script

Error message	[p-ave-world-sync][Error: 0001] You must be root to run this script
Cause	The p-ave-world-sync script has been invoked by a non-root user.
Resolution	The cron job /etc/cron.d/p-ave invokes p-ave-world-sync as root. If this error message occurs, contact IBM support.

Error message	[p-ave-world-sync][Error: 0002] Failed to get current set of mount entries
Cause	The p-ave-world-sync script has been invoked with the sync_all argument, but was unable to access the POWER system /proc/mounts file.
Resolution	Contact IBM support.

Error message	[p-ave-world-sync][Error: 0003] Cannot open <filename>: <errorcode>
Cause	The p-ave-world-sync script failed to open a user management file, for example /etc/passwd.
Resolution	Check the file exists in the x86 World and check that it is owned by root, group root, and the permissions are set to 644 (RW owner, R group, and R others).

Error message	[p-ave-world-sync][Error: 0004] The p-ave-daemon is not running. Please start the p-ave-daemon.
Cause	The p-ave-daemon is not running and is required for managing users, groups and passwords with PowerVM Lx86
Resolution	Invoke PowerVM Lx86 daemon as root. First become root, then invoke the daemon with the following command: /etc/init.d/p-ave start

Error message	[p-ave-world-sync][Error: 0005] Unrecognised option 'WORLD_CHECK_OR_SYNC=<option>'
Cause	The WORLD_CHECK_OR_SYNC configuration file option has been set to an unrecognized value.
Resolution	The valid options are: sync_all, check_all, check_passwd, check_group, force_sync_mtab, none. check_all is the default. The WORLD_CHECK_OR_SYNC can also be enabled by default to check_all by removing the WORLD_CHECK_OR_SYNC line from the configuration file.

Chapter 14. Errors from the x86 /etc/init.d support scripts

This section describes error messages from the x86 /etc/init.d support scripts, including cause and resolution.

Errors from the /etc/init.d/p-ave script

Error message	[/etc/init.d/p-ave][Error: 0001] You must be root to run this script.
Cause	The p-ave script has been launched by a non-root user.
Resolution	The p-ave script must be run by root. Ensure that you are running as root and run the script again.

Error message	[/etc/init.d/p-ave][Error: 0002] /etc/opt/p-ave/config must be owned by root.
Cause	The p-ave script needs to be owned by root to ensure that it cannot be tampered with by any non-root user. The config file is not currently owned by root.
Resolution	Check that the config file is owned by root. Contact IBM support if you have any further problems.

Error message	[/etc/init.d/p-ave][Error: 0003] /etc/opt/p-ave/config must be writable only by root.
Cause	The p-ave script must be writable only by root to ensure that it cannot be tampered with by any non-root user. The config file is currently writable by non-root users.
Resolution	Check that the config file is writable only by root. Contact IBM support if you have any further problems.

Error message	[/etc/init.d/p-ave][Error: 0004] Kernel doesn't contain binfmt_misc, and loading it as a module failed. Please check your kernel configuration and ensure binfmt_misc is available.
Cause	The p-ave script requires that the binfmt_misc kernel feature is enabled on the POWER system.
Resolution	Contact IBM support for further help.

Error message	[/etc/init.d/p-ave][Error: 0005] Could not mount /proc/sys/fs/binfmt_misc (from binfmt_misc)
Cause	The p-ave script requires that the binfmt_misc kernel feature is enabled on the POWER system. The script failed to mount the binfmt_misc file in /proc.
Resolution	Contact IBM support for further help.

Error message	[/etc/init.d/p-ave][Error: 0006] Failed to register i386 handler with binfmt_misc.
Cause	The p-ave script failed to register the i386 handler with binfmt_misc on the POWER system.
Resolution	Contact IBM support for further help.

Error message	[/etc/init.d/p-ave][Error: 0007] Failed to register i386so handler with binfmt_misc.
Cause	The p-ave script failed to register the i386so handler with binfmt_misc on the POWER system.
Resolution	Contact IBM support for further help.

Error message	[/etc/init.d/p-ave][Error: 0008] Unable to source init-script functions.
Cause	The p-ave script failed to launch.
Resolution	Contact IBM support.

Errors from the p-ave-rc runlevel scripts

The p-ave-rc runlevel scripts (/etc/init.d/p-ave-rc2, /etc/init.d/p-ave-rc3, and /etc/init.d/p-ave-rc5) are invoked when the system runlevel changes.

Error message	[/etc/init.d/p-ave-rc<number>][Error: 0001] You must be root to run this script
Cause	The p-ave-rc script was launched by a non-root user.
Resolution	The p-ave-rc scripts must be run by root. The scripts are not expected to be run manually. Contact IBM support.

Error message	[/etc/init.d/p-ave-rc<number>][Error: 0002] Error changing x86 runlevel
Cause	The p-ave-rc script failed to change the x86 runlevel.
Resolution	Contact IBM support.

Error message	[/etc/init.d/p-ave-rc<number>][Error: 0003] Unable to source init-script functions.
Cause	The p-ave-rc script failed to launch when invoked.
Resolution	Contact IBM support.

Errors from the /etc/init.d/p-ave-rcmonitor script

Error message	[/etc/init.d/p-ave-rcmonitor][Error: 0001] You must be root to run this script
Cause	The p-ave-rcmonitor script has been launched by a non-root user.
Resolution	The p-ave-rcmonitor script must be run by root. Ensure that you are running as root and run the script again.

Error message	<code>/etc/init.d/p-ave-rcmonitor</code> [Error: 0002] Unable to source init-script functions.
Cause	The p-ave-rcmonitor script failed to launch.
Resolution	Contact IBM support.

Errors from the `dependency_checker.pl` script

Error message	<code>[dependency_checker.pl]</code> [Error: 0001] You must be root to run this script
Cause	The <code>dependency_checker.pl</code> script has been invoked by a non-root user.
Resolution	The p-ave-rcmonitor script normally invokes this script. The script should not normally be invoked manually. Contact IBM support for further help.

Error message	<code>[dependency_checker.pl]</code> [Error: 0002] Directory <code><directoryName></code> does not exist.
Cause	The <code>dependency_checker.pl</code> script failed to find the main <code>init.d</code> directory.
Resolution	Check if the directory exists. Contact IBM support for further help.

Error message	<code>[dependency_checker.pl]</code> [Error: 0003] Cannot open <code><file></code> for reading: <code><errornumber></code> .
Cause	The <code>dependency_checker.pl</code> script failed to open a file in the main <code>init.d</code> directory.
Resolution	Check that the file exists and the permissions on the file. Contact IBM support for further help.

Error message	<code>[dependency_checker.pl]</code> [Error: 0004] Cannot open <code><file></code> for writing: <code><errornumber></code> .
Cause	The <code>dependency_checker.pl</code> script has failed to open a file in the main <code>init.d</code> directory.
Resolution	Check that the file exists and the permissions on the file. Contact IBM support for further help.

Chapter 15. Errors from the linkx86 script

The linkx86 script may produce the errors when being used to create an escape from the x86 World to a directory on the POWER system.

Error message	[linkx86][Error: 0001] linkx86 must not be run under translation.
Cause	The linkx86 script has been invoked from within the x86 environment, probably from a translated x86 shell.
Resolution	linkx86 can only be invoked from a POWER shell. Check that you are using a POWER shell and invoke the linkx86 script again.

Error message	[linkx86][Error: 0002] Superuser privileges are required for this script
Cause	The linkx86 script has been invoked by a non-root user.
Resolution	Ensure that you are root and try again.

Error message	[linkx86][Error: 0003] Path must be absolute.
Cause	The path provided as an argument to linkx86 is not an absolute path. It is likely that you provided a relative path as an argument.
Resolution	Invoke linkx86 with an absolute path as an argument.

Error message	[linkx86][Error: 0004] Path must not be the root '/' directory.
Cause	The path provided as an argument to linkx86 was the root ('/') directory.
Resolution	Invoke linkx86 with an absolute path as an argument that is not the root ('/') directory.

Error message	[linkx86][Error: 0005] <path> does not exist.
Cause	The path that you are trying to create an escape directory to does not exist on the POWER system.
Resolution	Check that the path exists on the POWER system. If not, create the directory on the POWER system. Check that you have correctly typed the name of the path and invoke linkx86 again.

Error message	[linkx86][Error: 0006] Unable to create <path/filename> because it already exists.
Cause	The escape link cannot be created because something with that name already exists in the x86 World.
Resolution	Ensure that the link you are trying to create does not already exist. Check that you have correctly typed the name of the link and invoke linkx86 again.

Error message	[linkx86][Error: 0007] Unable to create <path/filename>. Please check you have the necessary permission.
Cause	The escape link cannot be created because you do not have the correct permissions.
Resolution	Check the permissions on the directory where the file is being created and ensure that users have write permissions.

Chapter 16. Errors from the execve router

The execve router is responsible for launching certain processes for PowerVM Lx86. If in the unlikely circumstances that it failed to invoke PowerVM Lx86, one of the following errors may be seen:

Error message	[p-ave exec router][Error 0001] p-ave exec router could not invoke p-ave, (error <errornumber>)
Cause	The p-ave-world-sync script failed to invoke the PowerVM Lx86 binary (p-ave). The p-ave binary may not exist or the configuration file (if one exists) may have FU_OPT_P_AVE set to an incorrect location.
Resolution	Check that the p-ave binary exists in the default installation location or where you chose to install the binary if to a non-default location. If installed to a non-default location, check that FU_OPT_P_AVE configuration switch in /etc/opt/p-ave/config correctly points to the p-ave binary. If you cannot resolve the problem, contact IBM Support.

Error message	[p-ave exec router][Error 0002] Path to p-ave binary is too long (<number>)
Cause	The path to the p-ave binary is too long, for example /opt/<many_character_directory_name>/p-ave.
Resolution	Ensure that the PowerVM Lx86 binary is installed into a directory with a reasonably short length directory path.

Chapter 17. Syslog messages

Support for logging system messages is handled in a special manner for PowerVM Lx86 so that system messages in the x86 World and the POWER system are kept separate.

See Chapter 12, “Default PowerVM Lx86 escapes and virtual files in the x86 World,” on page 63 for detail of how system logs and error messages are handled by PowerVM Lx86.

Chapter 18. PowerVM Lx86 (p-ave) errors

This section lists error messages for PowerVM Lx86 (p-ave), including cause and resolution.

Error message	[p-ave][Error: 0001] Process received signal <signalname> (<signalnumber>).
Cause	One of the running x86 applications received a signal unexpectedly.
Resolution	Normally an x86 application will produce an error, report or log file when this problem occurs. Contact IBM support for further help.

Error message	[p-ave][Error: 0002] Access denied for x86 binary 'binaryname'. Please check the permissions on the file.
Cause	You do not have permissions to access the binary file.
Resolution	Check the permissions on the binary you attempted to execute and try again.

Error message	[p-ave][Error: 0003] File '<filename>' is not a valid x86 binary. The file may be a POWER binary. Please check the type of the file.
Cause	The binary may not be a valid x86 binary.
Resolution	Check that the binary is a valid x86 binary, for example by running the command line tool 'file'. PowerVM Lx86 can only execute Linux/x86 elf binary files.

Error message	[p-ave][Error: 0004] Cannot read x86 binary '<filename>'. Please check the permissions on the file.
Cause	You do not have permissions to read the binary file. This case should be handled correctly when running within a translated x86 shell.
Resolution	Ensure that you are running within a translated x86 shell and attempt to execute the binary again.

Error message	[p-ave][Error: 0005] Current working directory is not visible from the x86 World root. Please "cd "<path>" and try again.
Cause	The current working directory must be one of the following: <ul style="list-style-type: none">• The <X86WORLD_ROOT> directory or any of its subdirectories, for example, /i386 or /i386/etc• An escape directory or any of its subdirectories, for example, /home/mike or /home/mike/myDirectory

Error message	[p-ave][Error: 0005] Current working directory is not visible from the x86 World root. Please "cd "<path>" and try again.
Resolution	<p>Ensure that the current working directory is visible to the Linux/x86 environment, by ensuring it meets the criteria in the Cause section above.</p> <p>Ensure that you are invoking the runx86 script correctly and check the default list of escapes for the x86 World and any escapes you have added to the x86 World since installation.</p>

Error message	[p-ave][Error: 0006] x86 binary '<binaryname>' is not a valid binary. It may be a data file. Please check that the file is an executable binary.
Cause	The binary may not be a valid Linux/x86 elf binary.
Resolution	Check that the binary is a valid Linux/x86 elf binary, for example by running the command line tool 'file'. PowerVM Lx86 can only execute valid Linux/x86 elf binary files.

Error message	[p-ave][Error: 0007] x86 binary '<binaryname>' is not a valid file. It may be a directory. Please check that the file is an executable binary.
Cause	You may have attempted to execute a directory instead of a binary file within that directory, for example, /home/user instead of /home/user/myDirectory/myBinary.
Resolution	Ensure that you have correctly typed the name of the binary file you want to execute.

Error message	[p-ave][Error: 0008] Cannot access x86 binary '<binaryname>'. Please check that the file exists, the path to the file is valid and that the permissions on the path are correct.
Cause	You may not have permissions to access the binary file or the file may not exist or the path to the file may not be valid.
Resolution	Check that the file exists, the path to the file is valid and that the permissions on the path are correct and try again.

Error message	[p-ave][Error: 0009] Too many symbolic link loops encountered for x86 binary '<binaryname>'. Please check for loops in any symbolic links in the path to the binary and try again.
Cause	When attempting to resolve the path to the file, too many (more than 20) symbolic links were encountered. This may be caused a symbolic link loop, for example, a symbolic link pointing back to itself.
Resolution	Ensure that a symbolic link loop has not been created for the file you are trying to access.

Error message	[p-ave][Error: 0010] Unknown problem with file '<filename>'. Please save the log file '<logfile>' and contact IBM support.
Cause	Unknown.

Error message	[p-ave][Error: 0010] Unknown problem with file '<filename>'. Please save the log file '<logfile>' and contact IBM support.
Resolution	Contact IBM support with details of how the error occurred and send any log files that were generated.

Error message	[p-ave][Error: 0011] Could not contact the p-ave-daemon (error ('<errorname>', '<errornumber>')). Please check that the p-ave-daemon is running and try again.
Cause	PowerVM Lx86 has failed to connect to the PowerVM Lx86 daemon (p-ave-daemon).
Resolution	Check that the p-ave-daemon is running by using the command: /etc/init.d/p-ave status If the p-ave-daemon is not running, then start the p-ave-daemon by using the command: /etc/init.d/p-ave start If the p-ave-daemon is running, then try launching the x86 application again. If PowerVM Lx86 still cannot contact the p-ave-daemon, then contact IBM support.

Error message	[p-ave][Error: 0012] p-ave cannot write to the specified log file '<logfile>' (error (<errorname>, <errornumber>))
Cause	An error has occurred and PowerVM Lx86 has attempted but failed to write to the log file.
Resolution	<ul style="list-style-type: none"> • Check that the directory where the log file is being written to exists, if not, create a directory with the same name and try again. • Check that permissions have been granted to allow access to the directory.

Error message	[p-ave][Error: 0013] p-ave has terminated unexpectedly. Please save the log file '<filename>' and contact IBM support.
Cause	An error has occurred in PowerVM Lx86 or the x86 application, causing the translated process to be terminated.
Resolution	Save the log files and contact IBM support.

Error message	[p-ave][Error: 0014] Please invoke System p AVE using the runx86 script.
Cause	The p-ave binary has been invoked directly (for example, /opt/p-ave/bin/p-ave).
Resolution	To invoke PowerVM Lx86 (System p AVE), use the runx86 script, for example, /usr/local/bin/runx86.

Chapter 19. PowerVM Lx86 log file errors

This section lists the log file error messages, including cause and resolution.

Error message	[p-ave][Error: 0015] Cannot write to specified log directory '<logdirectory>'. Please check the permissions on the directory.
Cause	PowerVM Lx86 is attempting to write to the log file directory but failed.
Resolution	Check that the directory has write permissions.

Error message	[p-ave][Error: 0016] Cannot open logfile. Unexpected error occurred while initializing '<logfile>'. Please contact IBM support.
Cause	Unknown.
Resolution	Contact IBM support.

Error message	[p-ave][Error: 0017] Failed to create unique logfile name.
Cause	<p>PowerVM Lx86 attempts to create a uniquely numbered log file name each time it generates a new log file, p-ave.log.<binaryname>.<processID>.<uniquenumber> (for example, p-ave.log.perl.23724.4)</p> <p>PowerVM Lx86 has been unable to create a file with a new unique number.</p>
Resolution	Check the log directory to see if a particular process has generated an unexpected number of log files. Do not delete the log files unless you are certain they are not required to help resolve the issue further. If the problem persists, contact IBM support.

Error message	[p-ave][Error: 0018] Cannot open logfile - filesystem is full.
Cause	The file system where PowerVM Lx86 stores the log files appears to be full.
Resolution	Check for free space on the file system and make more space available if none is free.

Error message	[p-ave][Error: 0019] Cannot open logfile - filesystem is not writable. Please check the permission on the file.
Cause	The log file is not writable because the whole file system is read-only.
Resolution	Either change the log file directory location in the config file or remount the file system where the log file is located with read-write permissions.

Chapter 20. PowerVM Lx86 daemon (p-ave-daemon) errors

This section describes PowerVM Lx86 daemon (p-ave-daemon) error messages, including cause and resolution.

Error message	[p-ave-daemon] p-ave-daemon not starting due to errors. Please correct the problem and try again.
Cause	An error has occurred while invoking the PowerVM Lx86 daemon.
Resolution	A more detailed error message will be printed with this message. Follow the instructions in that message.

Error message	[p-ave-daemon][Error: 0001] Usage: /etc/init.d/p-ave [start stop force-reload restart status]
Cause	<ul style="list-style-type: none">• The /etc/init.d/p-ave script has been invoked with an invalid argument not listed above.• The p-ave-daemon binary has been invoked directly on the command line with an argument (it can only be invoked directly if no arguments are provided and this will invoke the daemon with the start option)
Resolution	Invoke the p-ave-daemon using the /etc/init.d/p-ave script with one of the arguments listed above. If you want to invoke p-ave-daemon directly, then do not provide any arguments to the binary on the command line.

Error message	[p-ave-daemon][Error: 0002] This machine model is not supported. Please check the system requirements in the System p AVE Administration Guide.
Cause	The PowerVM Lx86 daemon failed to start as it appears that the system is not an IBM System p server.
Resolution	Ensure that you are running PowerVM Lx86 on a supported platform by checking the system requirements in the PowerVM Lx86 (System p AVE) Administration Guide.

Error message	[p-ave-daemon][Error: 0003] Failed to open lock file '<lockfile>' (error ('<errorname>', <errornumber>)). Please check the permissions on the directory and lock file, then try again.
Cause	The PowerVM Lx86 daemon failed to start because it cannot open the lock file.
Resolution	Check the permissions on the directory and lock file. The lock file is located in /var/opt/p-ave/daemon/p-ave-daemon.lock.

Error message	[p-ave-daemon][Error: 0004] p-ave-daemon is already running (lock file detected). You do not need to invoke p-ave-daemon again.
Cause	The PowerVM Lx86 daemon is already running and you have attempted to start another instance.

Error message	[p-ave-daemon][Error: 0004] p-ave-daemon is already running (lock file detected). You do not need to invoke p-ave-daemon again.
Resolution	Check that the p-ave-daemon is running with the following command: <code>/etc/init.d/p-ave status</code> If the p-ave-daemon is running, then continue to launch x86 applications as normal. If not, then start the p-ave-daemon with the following command: <code>/etc/init.d/p-ave start</code>

Error message	[p-ave-daemon][Error: 0005] Failed to open log file '<logfile>' (error ('errorname', <errornumber>)). Please check the permissions on the directory and log file, then try again.
Cause	The PowerVM Lx86 daemon could not open the log file.
Resolution	Check that the directory where the log file is being written to exists, if not, create a directory with that name and try again. Check the permissions on the directory (which should be writable by daemon) and the log file (which should be owned by daemon and readable and writable by daemon) and try again.

Error message	[p-ave-daemon][Error: 0006] Failed to make directory '<directoryname>' (error ('errorname', <errornumber>)). Please check the permissions on the directory and try again.
Cause	The PowerVM Lx86 daemon could not create a directory on the system.
Resolution	Check the permissions on the parent directory (which should be writable daemon) where the directory is being created and try again.

Error message	[p-ave-daemon][Error: 0007] Socket directory '<socketdirectory>' must be owned by user '<user1>' (id <userid1>) but is owned by user '<user2>' (id <userid2>). Please correct the ownership on the directory and try again.
Cause	The ownership of the PowerVM Lx86 daemon socket is not correct.
Resolution	Change the ownership of the socket directory from <user2> to <user1> and try again.

Error message	[p-ave-daemon][Error: 0008] p-ave-daemon cannot change user or group. Please invoke p-ave-daemon as root.
Cause	The PowerVM Lx86 daemon was not invoked by root.
Resolution	Invoke PowerVM Lx86 daemon as root. First become root, then invoke the daemon with the following command: <code>/etc/init.d/p-ave start</code>

Chapter 21. E-mail messages for managing users, groups, and passwords in PowerVM Lx86

These tables show the templates for the e-mails sent to the root user when a user ID alert occurs. The text with square brackets ([_number]) represent specific files and details to the alert.

Alert message	New user account ('<useraccount>') found in <file>.
Example e-mail	<p>A new user account has been found in the x86 World password file ([_1]) that is not present in the POWER password file. The relevant password entry found in [_2] is shown here:</p> <p>[_3]</p> <p>where the fields represent the user account, password, user id, primary group id, comment, home directory and default shell, respectively. Also, the output of the translated command '/usr/bin/id [_4]' is shown below</p> <p>[_5]</p> <p>where the second field shows the primary group and the third field shows the full list of groups to which user [_6] belongs, including the primary and supplementary groups.</p> <p>In order to stop receiving this email in future you have two options. The first option is to add the user [_7] to the user whitelist file found in /etc/opt/p-ave/user_ignore. In this case you don't need to add the user to the POWER side and this user will be ignored in future. The second option is to add the user on the POWER side yourself. One way of doing this (but not necessarily the most complete) is to execute the following command in a native POWER shell:</p> <p>/usr/sbin/useradd -m -d <home directory> [_8]</p> <p>To replicate the x86 World setup completely you will have to ensure that all the groups to which user [_9] is a member exist on the POWER side and then set both the primary and supplementary groups for the user [_10] accordingly.</p> <p>(This mail was generated by the cron script [_1] and can be disabled by setting the configuration variable WORLD_CHECK_OR_SYNC=none in /etc/opt/p-ave/config)</p>

Alert message	Aliased user id ('<userid>') found in <file>.
Example e-mail	<p>A user account has been found in the x86 World password file ([_1]) that shares the same user id ([_2]) with an account found in the POWER password file (/etc/passwd). The relevant password entry found in the x86 World file ([_3]) is shown here:</p> <p>[_4] and the relevant password entry found in the POWER file (/etc/passwd) is shown here:</p> <p>[_5]</p> <p>where the fields represent the user account, password, user id, primary group id, comment, home directory and default shell, respectively.</p> <p>Although highly unlikely, this could represent a serious security risk, since the identity corresponding to user id [_6] is ambiguous.</p> <p>In order to stop receiving this email in future you have two options. If you believe there to be no security risk involved then you can add the user id [_7] to the whitelist file found in /etc/opt/p-ave/uid_ignore. Otherwise, it is recommended that you rectify the problem by changing the user id of [_8] in the x86 World. However, before doing so you MUST first shut down PowerVM Lx86 by running the command:</p> <pre>/etc/init.d/p-ave stop</pre> <p>You must do this as there may currently be Lx86 processes running as user [_9]. Next, you need to choose a new, unique user id for user [_10], ensuring that this user id does not already exist in either the x86 World or POWER password files. The simplest way to change the user id of user [_11] is to manually edit the x86 World password file, updating the user id accordingly. Finally, you must update the ownership of all files in the x86 World that are owned by user id [_12]. There are several ways to update the ownership on files in the x86 World, however here is one simple command you can run:</p> <pre>/bin/chown --from=[_13] -R <new user> [_14]</pre> <p>CAUTION: Any mistakes made at this stage could damage both your x86 World and your POWER system. You should then be able to restart PowerVM Lx86 with the following command:</p> <pre>/etc/init.d/p-ave start</pre> <p>(This mail was generated by the cron script [_1] and can be disabled by setting the configuration variable WORLD_CHECK_OR_SYNC=none in /etc/opt/p-ave/config)</p>

Alert message	New group ('<group>') found in <file>.
Example e-mail	<p>A new group account has been found in the x86 World group file ([_1]) that is not present in the POWER group file. The relevant group entry found in [_2] is shown here:</p> <p>[_3]</p> <p>where the fields represent the group account, password, group id, and users that are members of this group, respectively. In order to stop receiving this email in future you have two options. The first option is to add the group [_4] to the group whitelist file found in /etc/opt/p-ave/group_ignore. In this case you don't need to add the group to the POWER side and this group will be ignored in future. The second option is to add the group on the POWER side yourself. One way of doing this (but not necessarily the most complete) is to execute the following command in a native POWER shell:</p> <pre>/usr/sbin/groupadd [_5]</pre> <p>(This mail was generated by the cron script [_1] and can be disabled by setting the configuration variable WORLD_CHECK_OR_SYNC=none in /etc/opt/p-ave/config)</p>

Alert message	Aliased group id ('<groupid>') found in <file>.
Example e-mail	<p>A group account has been found in the x86 World group file ([_1]) that shares the same group id ([_2]) with an account found in the POWER group file (/etc/group). The relevant group entry found in the x86 World file ([_3]) is shown here:</p> <p>[_4] and the relevant group entry found in the POWER file (/etc/group) is shown here:</p> <p>[_5] where the fields represent the group account, password, group ID, and users that are members of this group, respectively.</p> <p>Although highly unlikely, this could represent a serious security risk, since the identity corresponding to group id [_6] is ambiguous.</p> <p>In order to stop receiving this email in future you have two options. If you believe there to be no security risk involved then you can add the group id [_7] to the whitelist file found in /etc/opt/p-ave/gid_ignore Otherwise, it is recommended that you rectify the problem by changing the group id of [_8] in the x86 World. However, before doing so you MUST first shut down PowerVM Lx86 by running the command:</p> <pre>/etc/init.d/p-ave stop</pre> <p>since there may currently be PowerVM Lx86 processes running as group [_9]. Next, you need to choose a new, unique group id for group [_10], ensuring that this group id does not already exist in either the x86 World or POWER group files. The simplest way to change the group id of group [_11] is to manually edit the x86 World group file, updating the group id accordingly. Finally, you must update the ownership of all files in the x86 World that are owned by group id [_12]. There are several ways to update the ownership on files in the x86 World, however here is one simple command you can run:</p> <pre>/bin/chown --from=[_13] -R :<new group id> [_14]</pre> <p>CAUTION: Any mistakes made at this stage could damage both your x86 world and your POWER system.</p> <p>You should then be able to restart PowerVM Lx86 with the following command:</p> <pre>/etc/init.d/p-ave start</pre> <p>(This mail was generated by the cron script [_1] and can be disabled by setting the configuration variable WORLD_CHECK_OR_SYNC=none in /etc/opt/p-ave/config)</p>

Part 5. Glossary

This is a glossary for the PowerVM Lx86 for x86 Linux Applications Administration Guide.

escape A mechanism that allows access to files on the local Linux on POWER file system, which are external to x86World from the VxE.

host system The POWER system on which PowerVM Lx86 has been installed. It is able to run x86 applications within a VxE.

jailing The restriction of the view of the Linux file system from the VxE. Similar in concept to UNIX chroot.

Linux on POWER system A system with a POWER CPU running the Linux operating system.

Linux on x86 system A system with an x86 CPU running the Linux operating system.

Linux on POWER application A Linux application compiled for a Linux on POWER system.

Linux on x86 application A Linux application compiled for a Linux on x86 system.

native POWER application A Linux on POWER application running natively on a Linux on POWER system.

native x86 application A Linux on x86 application running natively on a Linux on x86 system.

native POWER shell The Linux shell running natively on the Linux on POWER host system.

native x86 shell The Linux shell running natively on a Linux on x86 system.

runx86 The command that runs an x86 binary within a Virtual x86 Environment.

p-ave The program that translates x86 applications so they can run on POWER systems.

p-ave-daemon The daemon program that p-ave uses to communicate between translated x86 processes on the POWER system.

PowerVM Lx86 A product that enables POWER systems to run x86 applications alongside native POWER applications. No modifications, recompiling, or changes are needed to the x86 applications.

translator The p-ave program that handles the mapping of instructions and requests from the VxE onto the underlying Linux on POWER system.

Virtual x86 Environment (VxE) The method that PowerVM Lx86 uses to add Linux on x86 compatibility to Linux on POWER systems. Linux on x86

applications are encapsulated so the operating environment appears to be x86, even though the underlying system is POWER. This is achieved using the files and libraries in x86World, the translator, and selective integration between the VxE and the POWER host system.

x86 application A Linux on x86 application running within a VxE on a Linux on POWER host system.

x86 shell The Linux shell running within a VxE on a Linux on POWER host system. Linux on x86 commands entered from the x86 shell prompt will also be run within a VxE.

x86World A set of Linux on x86 libraries, commands, applications, and other system files that are installed in a directory on the POWER system.

Part 6. Appendixes

Appendix. Accessibility features

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

The following list includes the major accessibility features:

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

IBM and accessibility

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

Notices

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

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

The manufacturer may have patents or pending patent applications covering subject matter described 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 the manufacturer.

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

The following paragraph does not apply to the United Kingdom or any other country where such provisions are inconsistent with local law: THIS INFORMATION IS PROVIDED "AS IS" WITHOUT WARRANTY OF ANY KIND, EITHER EXPRESS OR IMPLIED, INCLUDING, BUT NOT LIMITED TO, THE IMPLIED WARRANTIES OF NON-INFRINGEMENT, MERCHANTABILITY OR FITNESS FOR A PARTICULAR PURPOSE. Some states do not allow disclaimer of express or implied warranties in certain transactions, therefore, this statement may not apply to you.

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

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

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

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

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

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

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

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

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

The manufacturer's prices shown are the manufacturer's suggested retail prices, are current and are subject to change without notice. Dealer prices may vary.

This information is for planning purposes only. The information herein is subject to change before the products described become available.

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

COPYRIGHT LICENSE:

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

CODE LICENSE AND DISCLAIMER INFORMATION:

The manufacturer grants you a nonexclusive copyright license to use all programming code examples from which you can generate similar function tailored to your own specific needs.

SUBJECT TO ANY STATUTORY WARRANTIES WHICH CANNOT BE EXCLUDED, THE MANUFACTURER, ITS PROGRAM DEVELOPERS AND SUPPLIERS, MAKE NO WARRANTIES OR CONDITIONS EITHER EXPRESS OR

IMPLIED, INCLUDING BUT NOT LIMITED TO, THE IMPLIED WARRANTIES OR CONDITIONS OF MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE, AND NON-INFRINGEMENT, REGARDING THE PROGRAM OR TECHNICAL SUPPORT, IF ANY.

UNDER NO CIRCUMSTANCES IS THE MANUFACTURER, ITS PROGRAM DEVELOPERS OR SUPPLIERS LIABLE FOR ANY OF THE FOLLOWING, EVEN IF INFORMED OF THEIR POSSIBILITY:

1. LOSS OF, OR DAMAGE TO, DATA;
2. SPECIAL, INCIDENTAL, OR INDIRECT DAMAGES, OR FOR ANY ECONOMIC CONSEQUENTIAL DAMAGES; OR
3. LOST PROFITS, BUSINESS, REVENUE, GOODWILL, OR ANTICIPATED SAVINGS.

SOME JURISDICTIONS DO NOT ALLOW THE EXCLUSION OR LIMITATION OF DIRECT, INCIDENTAL, OR CONSEQUENTIAL DAMAGES, SO SOME OR ALL OF THE ABOVE LIMITATIONS OR EXCLUSIONS MAY NOT APPLY TO YOU.

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

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

If you are viewing this information in softcopy, the photographs and color illustrations may not appear.

Trademarks

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

DB2
IBM
OpenPower
POWER
POWER5
POWER6
Resource Link
System i
System p
WebSphere

Adobe®, Acrobat®, Portable Document Format (PDF), and PostScript® are either registered trademarks or trademarks of Adobe Systems Incorporated in the United States, other countries, or both.

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

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

Red Hat, the Red Hat "Shadow Man" logo, and all Red Hat-based trademarks and logos are trademarks or registered trademarks of Red Hat, Inc., in the United States and other countries.

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

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

Terms and conditions

Permissions for the use of these publications is granted subject to the following terms and conditions.

Personal Use: You may reproduce these publications for your personal, noncommercial use provided that all proprietary notices are preserved. You may not distribute, display or make derivative works of these publications, or any portion thereof, without the express consent of the manufacturer.

Commercial Use: You may reproduce, distribute and display these publications solely within your enterprise provided that all proprietary notices are preserved. You may not make derivative works of these publications, or reproduce, distribute or display these publications or any portion thereof outside your enterprise, without the express consent of the manufacturer.

Except as expressly granted in this permission, no other permissions, licenses or rights are granted, either express or implied, to the publications or any data, software or other intellectual property contained therein.

The manufacturer reserves the right to withdraw the permissions granted herein whenever, in its discretion, the use of the publications is detrimental to its interest or, as determined by the manufacturer, the above instructions are not being properly followed.

You may not download, export or re-export this information except in full compliance with all applicable laws and regulations, including all United States export laws and regulations.

THE MANUFACTURER MAKES NO GUARANTEE ABOUT THE CONTENT OF THESE PUBLICATIONS. THESE PUBLICATIONS ARE PROVIDED "AS-IS" AND WITHOUT WARRANTY OF ANY KIND, EITHER EXPRESSED OR IMPLIED, INCLUDING BUT NOT LIMITED TO IMPLIED WARRANTIES OF MERCHANTABILITY, NON-INFRINGEMENT, AND FITNESS FOR A PARTICULAR PURPOSE.



Printed in USA

SA38-0650-01

