IBM Software Development Kit for Multicore Acceleration
Version 3.0

Installation Guide

SC33-8323-03

IBM Software Development Kit for Multicore Acceleration
Version 3.0

Installation Guide

SC33-8323-03

Note

Before using this information and the product it supports, read the information in "Notices" on page 89.

Edition notice

This edition applies to version 3, release 0, modification 0 of the IBM Software Development Kit for Multicore Acceleration (product number 5724-S84) and to all subsequent releases and modifications until otherwise indicated in new editions.

This edition replaces SC33-8323-02.

Contents

About this publication

This book is the Installation Guide for the IBM® Software Development Kit (SDK) for Multicore Acceleration version 3.0.

The SDK is a complete package of tools that help you create applications for hardware platforms built on Cell Broadband Engine Architecture such as the IBM BladeCenter QS21. The SDK is composed of runtime tools such as the Linux® kernel, development tools, software libraries and frameworks, performance tools, a Full-System Simulator, and example source files, all of which fully support the capabilities of the Cell Broadband Engine Architecture.

Packages containing code derived from GPL or LGPL open source software such as GCC are located on the BSC Web site: http://www.bsc.es/projects/deepcomputing/linuxoncell/.

A single integrated installation based on the open source tool *YUM* installs both IBM and BSC open source components. The SDK is available for download from IBM Passport Advantage® and the IBM developerWorks® Cell Broadband Engine Architecture resource center at http://www-128.ibm.com/developerworks/power/cell/. This book describes the details of installing both a prerequisite Linux operating system and the SDK for supported platforms.

Chapter 1. Introduction

New in this release

This release of the SDK contains a number of significant enhancements over previous versions of the SDK and completely replaces those SDK versions.

These enhancements include:

- New installation process based on YUM
- Addition of PPU and SPU Fortran compiler
- Addition of PPU-only GNU Ada compiler
- Addition of single-source XL C/C++ compiler
- All compilers now generate code that is compliant with the SPE Stall App. Note. See http://www.ibm.com/chips/techlib/techlib.nsf/techdocs/C5996EDB722D3A478725728E0074B465.
- GCC toolchain enhancements
- Better integration of Eclipse IDE with compilers, GDB debugger, and Full-System Simulator
- Restructuring of examples and demonstration source code; addition of more examples
- Addition of DaCS and DaCS for Hybrid-x86 programming model
- Major enhancements to ALF framework and addition of ALF for Hybrid-x86
- Complete implementation of SIMDMath library
- Addition of BLAS linear algebra library
- Addition of FFT Library
- Addition of Monte Carlo random number generator library
- Addition of SPU virtual clock and timer services
- Addition of Performance and Debug Tracing tool (PDT and PDTR)
- Updates to Cell Performance Counter and FDPR-Pro performance tools
- Addition of Hybrid performance tooling
- Performance enhancements to the Full-System Simulator
- Updated Full-System Simulator sysroot to Fedora 7

Related products

You can use these related products together with the SDK components to provide additional capability.

Here is a list of related products and where to get them:

- XL C/C++ for Multicore Acceleration for Linux on System p™ (dual-source compiler) and XL C/C++ for Multicore Acceleration for Linux on X86 Systems (dual-source compiler) available from IBM at http://www-306.ibm.com/software/awdtools/ccompilers/
- XL Fortran for Multicore Acceleration for Linux on System p (dual-source compiler) available from IBM at http://www.ibm.com/software/awdtools/fortran/

- Visual Performance Analyzer (VPA) available from alphaWorks® at http://www.alphaworks.ibm.com/tech/vpa/

Supported platforms

Cell Broadband Engine Architecture applications can be developed on these Fedora 7 and RHEL 5.1 platforms:

- X86
- X86_64
- 64-bit PowerPC® (PPC64)
- IBM BladeCenter QS20
- IBM BladeCenter QS21

For specific requirements, see "Hardware prerequisites" on page 3.

Licenses

The source code and binaries that are part of the total SDK package are distributed with different licenses.

The packages on the BSC Web site are generally open source and use either:

- The General Public license (GPL)

  `http://www.gnu.org/copyleft/gpl.html`
- Lesser General Public license (LGPL)

  `http://www.gnu.org/licenses/licenses.html#LGPL`

If you are not familiar with these licenses, visit the Free Software Foundation (FSF) for more information.

The ISO images or physical media available from IBM have several licenses depending upon the package:

- The *RHEL5-Product* ISO or physical media has an International Program License Agreement (IPLA) number **L-SGAN-74USJD**.
- The *RHEL5-Devel* ISO and *Fedora-Devel* ISO have an International License Agreement for Non-Warranted Programs (ILAN) number **L-SGAN 73RPBP**.
- The *RHEL5-Extra* ISO has an International License Agreement for Early Release of Programs (ILAER) number **L-SGAN-76Q4EY**.
- The *RHEL5-Extra* ISO has an International License Agreement for Early Release of Programs (ILAER) number **L-SGAN-76X5BX**.

See http://www.ibm.com/software/sla/sladb.nsf for more information about IBM licenses.

Chapter 2. SDK prerequisites

Hardware prerequisites

The SDK has specific hardware requirements. The following table shows the recommended minimum configuration for each hardware platform.

Table 1. Hardware prerequisites

System	Recommended minimum configuration
X86 or X86_64 architecture	2 GHz Intel® Pentium® 4 processor, or AMD Opteron "F" processor that supports the `RDTSCP` instruction.
PowerPC	1.42 GHz 64-bit PPC. **32-bit PPC platforms are not supported.**
BladeCenter QS20	Revision 31, hardware firmware level QA-06.14.0-0F (7.21). See "Checking the firmware version" on page 5
BladeCenter QS21	Hardware firmware level QB-01.08.0-00

All systems must have:

- Hard disk space: 5 GB (minimum) to install the source package and the accompanying development tools
- 1 GB RAM (minimum) on the host system

Note: If you use the Full System Simulator, the minimum amount of RAM installed must be twice the amount of simulated memory. For example, to simulate a system with 512 MB of RAM, the host system must have at least 1 GB of RAM installed.

Software prerequisites

The SDK requires Fedora 7 or Red Hat Enterprise Linux (RHEL) 5.1, which must be installed before you install the SDK. See Chapter 3, "Operating system installation," on page 5 for information about how to install Linux.

To install the SDK, see Chapter 5, "Installing, uninstalling, and updating the SDK," on page 31.

SELinux

The SELinux policy files that are included in the Fedora 7 base distribution prevent *spufs* from loading correctly on boot. To install the SDK, you must either turn off SELinux or update the `selinux-policy` and `selinux-policy-targeted` RPMs to the latest version. The preferred method is to update the RPMs. To update, type the following commands as root:

```
# yum update selinux-policy selinux-policy-targeted
```

SDK utility software dependencies

The SDK requires the packages *rsync*, *sed*, *TCL*, and *wget*.

To install these dependencies, type the following command as root:

```
# yum install rsync sed tcl wget
```

expat for Hybrid-x86

The DaCS for Hybrid-x86 daemon for the X86_64 platform requires the *expat* XML parsing library. Install expat by typing the following command as root:

```
# yum install expat
```

Chapter 3. Operating system installation

This topic provides detailed information about how to install an operating system supported by the SDK on IBM BladeCenter hardware.

See the instructions provided by the operating system (Fedora 7 or Red Hat Enterprise Linux (RHEL) 5.1) for more information. The SDK is tested with the released versions of the operating system, but it might not be tested with all combinations of the latest software updates.

See "BladeCenter QS20 specifics" for details about installing Fedora 7 on a BladeCenter QS20. See "BladeCenter QS21 Specifics" on page 11 for details about installing either Fedora 7 or RHEL 5.1 on a BladeCenter QS21.

BladeCenter QS20 specifics

This topic describes how to install Fedora 7 on the BladeCenter QS20.

The following topics give specific details:
- "Managing a BladeCenter QS20"
- "Installing Fedora 7 Linux on a BladeCenter QS20" on page 7

Managing a BladeCenter QS20

This topic describes how to manage a BladeCenter QS20.

The following chapters describe how to check the firmware version, boot, shut down, and restart a BladeCenter QS20:
- "Checking the firmware version"
- "Checking which firmware bank was booted" on page 6
- "Booting a BladeCenter QS20" on page 6
- "Recovering from a bad firmware boot" on page 6
- "Shutting down and restarting the BladeCenter QS20" on page 7

Checking the firmware version

This topic describes how to check the firmware version.

To check the firmware version, do one of the following:
1. Access the BladeCenter® Management Module.
2. Click **Monitors → Firmware VPD**. The **Blade Server Firmware VPD** window contains the build identifier, release, and revision.

or:

From the Linux command line, run the following commands:

```
# for file in `ls /proc/device-tree/openprom/*bank*`; do echo $file; \
  cat $file; echo; echo; done
```

The following sample output shows that the blade has been booted from the temporary firmware bank, which contains version 6.14.E of the firmware.


```
/proc/device-tree/openprom/ibm,fw-bank
T
/proc/device-tree/openprom/ibm,fw-perm-bank
CB1FW614E       , 06-26-2006 22:52
/proc/device-tree/openprom/ibm,fw-temp-bank
CB1FW614E       , 06-26-2006 22:52
```

If you want to download newer firmware for the BladeCenter QS20, access the following Web site: http://www-304.ibm.com/jct01004c/systems/support/supportsite.wss/docdisplay?lndocid=MIGR-66645&brandind=5000020

Checking which firmware bank was booted

This topic describes how to check which firmware bank was booted.

To check if the TEMP or PERM firmware was booted, type the following command:

```
# cat /proc/device-tree/openprom/ibm,fw-bank
```

A **T** indicates a boot from the temporary bank and a **P** from the permanent bank.

Booting a BladeCenter QS20

This topic describes how to boot a BladeCenter QS20.

To boot a BladeCenter QS20, do the following:

1. Open the BladeCenter Management Module.
2. Set the appropriate boot device (network, hard disk) for the BladeCenter QS20 by selecting **Blade Tasks** → **Configuration** → **Boot Sequence**.
3. Power on the BladeCenter QS20 by selecting **Blade Tasks** → **Power/Restart** → checkmark the blade → **Power On Blade**.

Note:

- The boot process of the BladeCenter QS20 can only be monitored with a serial console (115200,N,1,8, no handshake) connected to the serial port on the front bezel of the BladeCenter QS20.
- Use a null-modem cable to connect to the serial port.
- To force the BladeCenter QS20 into the OpenFirmware prompt, press **s** on the serial console during the early stages of the boot process.

Recovering from a bad firmware boot

If Linux does not boot with the temporary firmware level, connect a console to the BladeCenter QS20 serial port and reboot using the Management Module.

At the console, stop the firmware boot (press **s** on the console) and type on the console the following command to set the firmware to the permanent side:

```
# 0 set-flashside
```

Next, reject the temporary firmware (which copies the permanent firmware to the temporary location) by typing:

```
# update_flash -r
```

Then type the following command to set the firmware back to the temporary side:

```
# 1 set-flashside
```

and reboot the BladeCenter.

Shutting down and restarting the BladeCenter QS20

This topic describes how to shut down and restart a BladeCenter QS20.

Always shutdown and restart a BladeCenter QS20 that has been booted to the Linux prompt with one of the following commands from a Linux shell on the BladeCenter QS20:

```
# shutdown -g0 -i0 -y
# halt
# reboot
# shutdown -r now
```

Do not use the Blade Center Management Module to power down or restart the Blade (using **Blade Tasks** → **Power/Restart** → checkmark the blade → **Power Off Blade / Restart Blade**) as this can result in a damaged file system: the Blade Center will power off the BladeCenter QS20 without first notifying the operating system.

Installing Fedora 7 Linux on a BladeCenter QS20

This section describes how to install Fedora 7 for PPC64 on a BladeCenter QS20.

Preparing your BladeCenter QS20

If your BladeCenter QS20 comes with InfiniBand option(s) already installed, unplug the PCI-Express cable(s) on the board side or uninstall the InfiniBand option(s), then install the OS and the patched kernel, then refit the PCI-Express cable(s) or reinstall the InfiniBand option(s).

Fedora 7 installation overview

The installation consists of the following steps:

1. Set up a netboot environment.
2. Set up a net installation environment.
3. Perform a manual installation.

The Fedora 7 installation process starts by booting a kernel with the install `initrd` from the network device (this is the only supported installation method on a BladeCenter QS20.) The init process `/sbin/init` (which starts `/sbin/loader`) prompts you for the installation language and installation method. For a network installation, the loader also configures the network and queries the parameters for the install server, before it downloads the secondary stage image `Fedora/base/stage2.img` from the installation server. After mounting the disk image, the loader passes control to the Python script named `anaconda`.

The Anaconda installer is the main installation program for Fedora 7. It performs the remaining steps of the installation, either manually through configuration screens or automatically using the `kickstart` configuration file. This process includes downloading all RPMs, which are selected for installation from the installation server.

The network installation environment

Note:

1. The IP addresses used in the examples below are for illustrative purposes only. Use IP addresses allocated to your network.

2. During installation, you are prompted for the directory containing the Fedora 7 installation files. The illustrations below show `/fedoratree` as the source containing the installation files. Change this to the path containing your installation files.

For the remainder of this document, it is assumed that you have the following environment:

- A BladeCenter QS20 (10.32.5.11). This is the installation target.
- A DHCP/BOOTP server (10.32.0.1).
- An install server (10.32.0.1) running a TFTP server, with the installation source. This server must also be able to run Fedora 7 if the installation material requires modification.
- An NFS, HTTP or FTP server (10.64.0.31) with the installation source.

The NFS/HTTP/FTP server can reside on the same server as the DHCP/BOOTP/TFTP server.

Figure 1 shows a typical network installation environment.

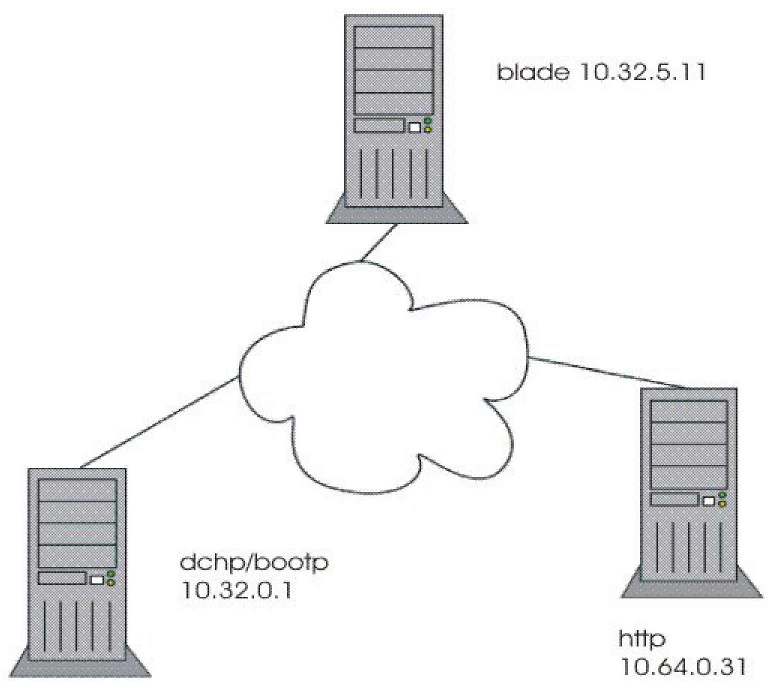


Figure 1. The network installation environment

Setting up a netboot environment

From within the ISO or image server root path, copy the netboot image (images/netboot/ppc64.img) to the TFTP server root directory (for example/tftpboot, see /etc/xinet.d/tftp) and make sure that it matches the respective entry in /etc/dhcpd.conf. For example:

```
host somehostname {
   allow bootp;
   allow booting;
   option host-name     "somehost.localdomain.com";
   hardware ethernet    00:20:9f:12:0f:19;
```

```
   fixed-address       10.3.5.11;
   filename            "ppc64.img";
   next-server         10.32.0.1;
}
```

Setting up a network installation environment

Fedora 7 is installed over the network using TFTP and NFS, or FTP.

Installing Fedora 7

To install Fedora 7, do the following:

1. Either insert a new hard disk into the BladeCenter QS20, or overwrite an existing hard disk with an existing Linux.
2. Connect the BladeCenter QS20 to a serial console (115200,N,1,8, no handshake) and boot it to the firmware prompt.
3. To start the installation, enter the following:

   ```
   > netboot vnc console=hvc0
   ```

4. Select the language you would like to use for the installation.

   ```
   +----------+ Choose a Language +----------+
   |                                          |
   | What language would you like to use      |
   | during the installation process?         |
   |                                          |
   |       Catalan                     ^      |
   |       Chinese(Simplified)         :      |
   |       Chinese(Traditional)        :      |
   |       Croatian                    :      |
   |       Czech                       :      |
   |       Danish                      :      |
   |       Dutch                       :      |
   |    >  English                     :      |
   |                                   v      |
   |                  +----+                  |
   |                  | OK |                  |
   |                  +----+                  |
   |                                          |
   |                                          |
   +------------------------------------------+
   ```

5. Select the media type that contains the installation packages.

   ```
   +------+ Installation Method +------+
   |                                    |
   | What type of media contains the    |
   | packages to be installed?          |
   |                                    |
   |           Local CDROM              |
   |           Hard drive               |
   |           NFS directory            |
   |         > FTP                      |
   |           HTTP                     |
   |                                    |
   |     +----+          +------+       |
   |     | OK |          | Back |       |
   |     +----+          +------+       |
   |                                    |
   |                                    |
   +------------------------------------+
   ```

6. Select the network device. Unless you have a second switch installed in your BladeCenter QS20 chassis, you select `eth0`.

```
+---------------- Networking Device -----------------+
|                                                     |
| You have multiple network devices on this system.   |
| Which would you like to install through?            |
|                                                     |
|  > eth0 - Unknown device 01b3                       |
|    eth1 - Unknown device 01b3                       |
|                                                     |
|         +----+              +------+               |
|         | OK |              | Back |               |
|         +----+              +------+               |
|                                                     |
|                                                     |
+-----------------------------------------------------+
```

7. Select how you wish to configure the network device. Because you have booted from DHCP, it is easiest to leave it set to DHCP. To do this, make sure that **Dynamic IP configuration (DHCP)** is selected. Fedora 7 determines the host name and domain from the dhcp/bootp server.

```
+----------------+ Configure TCP/IP +-----------------+
|                                                      |
| [*] Enable IPv4 support                              |
|        (*) Dynamic IP configuration (DHCP)           |
|        ( ) Manual configuration                      |
| [ ] Enable IPv6 support                              |
|        (*) Automatic neighbor discovery              |
|        ( ) Dynamic IP configuration (DHCPv6)         |
|        ( ) Manual configuration                      |
|                                                      |
|        +----+              +------+                  |
|        | OK |              | Back |                  |
|        +----+              +------+                  |
|                                                      |
|                                                      |
+------------------------------------------------------+
```

8. Enter the network site name and the path where you installed your media during the server setup:

```
+----------------------+ FTP Setup +------------------------+
|                                                            |
| Please enter the following information:                    |
|                                                            |
|     o the name or IP number of your FTP server             |
|     o the directory on that server containing              |
|       Fedora Core for your architecture                    |
|                                                            |
|   FTP site name:    10.32.0.1                              |
|                                                            |
|   Fedora core directory: /install/fedora/core7/ppc         |
|                                                            |
|  [ ] Use non-anonymous ftp                                 |
|                                                            |
|         +----+                     +------+                |
|         | OK |                     | Back |                |
|         +----+                     +------+                |
|                                                            |
|                                                            |
+------------------------------------------------------------+
```

9. The following welcome screen is displayed. Press the OK button to continue with the installation.

```
+------------------+ Fedora +-------------------+
|                                               |
| Welcome to Fedora!                            |
|                                               |
|                                               |
|                   +----+                      |
|                   | OK |                      |
|                   +----+                      |
|                                               |
|                                               |
+-----------------------------------------------+
```

10. The installer displays a status screen similar to the following:

```
+-------------------------------------------------------------------+
| Running anaconda, the Fedora Core system installer - please wait... |
| Framebuffer ioctl failed. Exiting.                                |
| Probing for video card:   Unable to probe                         |
| Probing for monitor type:   Unknown monitor                       |
| Probing for mouse type:   No — mouse                              |
| No video hardware found, assuming headless                        |
| Starting VNC...                                                   |
|                                                                   |
| WARNING!!! VNC server running with NO PASSWORD!                   |
| You can use the vncpassword=<password> boot option                |
| if you would like to secure the server.                           |
|                                                                   |
| The VNC server is now running.                                    |
| Please connect to 10.32.5.11:1 to begin the install...            |
|                                                                   |
| Press <enter> for a shell                                         |
| Starting graphical installation...                                |
+-------------------------------------------------------------------+
```

11. Start a VNC session on another computer in the network. At the command prompt of that computer enter `vncviewer <target IP>:<vnc session id>`, where `<vnc session id>` is the address of the BladeCenter QS20 being installed, for example, 10.32.5.11:1. Continue the installation process from the computer running the vncviewer session, not the BladeCenter QS20 where the installation process is actually taking place.

Rebooting the BladeCenter QS20:

When the installation is complete, reboot the system from the installation screen. The Fedora 7 kernel does not support all of the features of SDK 3.0 and should be replaced with the kernel with SDK 3.0 (see "Software prerequisites" on page 3). A final reboot after the install ensures that the BladeCenter QS20 is using the SDK kernel.

BladeCenter QS21 Specifics

These are specifics about installing Fedora 7 or Red Hat Enterprise Linux (RHEL) 5.1 on the BladeCenter QS21.

Creating a Linux network installation for BladeCenter QS21

Red Hat Enterprise Linux (RHEL) 5.1 or Fedora 7 does not directly support an installation to an NFS-mounted disk. You must create an initial installation on disk. From this initial installation on disk you can create a network installation that can be used by the BladeCenter QS21.

The BladeCenter QS21 does not provide on-board disk space. If SAS-attached storage is available to your BladeCenter QS21, you can install on a SAS disk. Alternatively, you can use any a 64-bit POWER-based system that is supported by RHEL 5.1 or Fedora 7 and has enough disk space for the installation.

Figure 2 illustrates the main steps for creating a network installation for BladeCenter QS21.

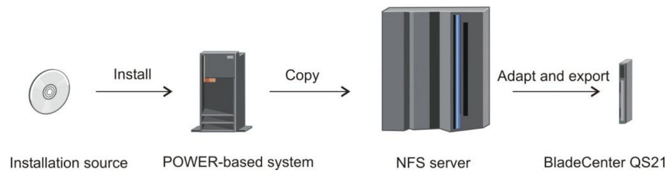


Figure 2. Overview of the network installation procedure

First you install Linux on the local disk space of a 64-bit POWER-based system. You then copy the resulting root file system to a Network File System (NFS) server where you make it network-bootable and adapt it to the specifics of an individual blade server.

See http://fedoraproject.org/wiki/StatelessLinuxCreateClientImage for general information about installing Linux on diskless systems.

This section describes how you can create a network installation for a single blade server. You can create multiple copies of this first installation and adapt each copy for use by a different blade server. See http://www.ibm.com/alphaworks/tech/dim/ for a description of a tooled approach of managing root file systems and boot kernels for numerous blade servers.

Requirements

This topic describes the resources you require for setting up a net-boot environment for BladeCenter QS21.

All of the following resources must be configured and connected through an Ethernet network.

BladeCenter QS21
: You need one or more BladeCenter QS21 blade servers installed in a BladeCenter H unit.

POWER-based installation system
: You need a 64-bit POWER-based system that is supported by Red Hat Enterprise Linux (RHEL) 5.1 or Fedora 7 and has enough disk space for the installation.

NFS server
: You need an NFS exported directory for each blade server that you want to boot from the network. This directory is to hold the root file system for the blade server. The NFS server can but need not run on the installation system.

TFTP server
: You need a Trivial File Transfer Protocol (TFTP) exported directory that is to hold the zImage for booting the blade server. A zImage contains a boot kernel and a suitable initial RAM disk. You need different zImages for

different kernels but blade servers that run the same kernel and same NFS root can all boot using the same zImage. The TFTP server can but need not run on the installation system.

DHCP server
You need a Dynamic Host Configuration Protocol (DHCP) server that supports the Bootstrap Protocol (BOOTP) and has a DHCP configuration for each blade server. BOOTP maps the blade servers to the zImage on the TFTP server. The DHCP server can but need not run on the installation system.

Figure 3 shows a sample setup where the initial Linux installation is performed on one system, the NFS server and TFTP server run on a second system, and the DHCP server runs on a third system.

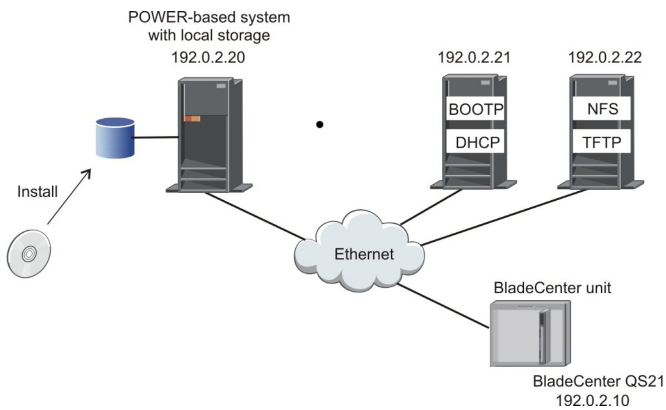


Figure 3. Sample setup with three systems

The examples in the procedures that follow are based on the sample setup of Figure 3.

How the network boot process works

The network boot process works as follows:

1. The BladeCenter server boots from its assigned boot sequence. Ensure that *network* is set as the first boot device in the Management Module software.
2. The BladeCenter server broadcasts its MAC address.
3. BOOTP on the DHCP server uses the MAC address to locate the DHCP configuration for the BladeCenter server. The DHCP server returns the IP address of the BladeCenter server and the name of the zImage to be used for booting the BladeCenter server.
4. The BladeCenter server uses the `server-name` attribute in the DHCP configuration to find the TFTP server and loads the zImage.
5. The BladeCenter server uses the root-path option received during a driven DHCP request to locate the root file system on the NFS server.

Creating a network installation for the BladeCenter QS21

Perform these main steps to set up your network installation.

1. “Install Linux on a POWER-based system” on page 14
2. “Create or download a zImage with NFS support” on page 14
3. “Install the zImage” on page 15
4. “Set up the root file system on the network” on page 15
5. “Boot from the network” on page 16

Install Linux on a POWER-based system:

Install Red Hat Enterprise Linux (RHEL) 5.1 or Fedora 7 as usual.

Create or download a zImage with NFS support:

You must create or download a zImage with an initial RAM disk (initrd) that supports booting from NFS and install it.

For Fedora 7, you can download the zImage from the BSC Web site. The full URL is: http://www.bsc.es/projects/deepcomputing/linuxoncell/cellsimulator/sdk3.0/zImage.initrd-2.6.22-5.20070920bsc.

For Red Hat Enterprise Linux (RHEL) 5.1, you need to create a zImage using the following procedure. This procedure can also be used for Fedora 7 but is not necessary.

1. Boot the newly-installed Linux image on your POWER-based system.
2. If you do not know it, find the kernel version. Type:
```
# uname -r
```
3. Make sure that `BOOTPROTO=dhcp` is defined in /etc/sysconfig/network-scripts/ifcfg-eth0 so that the correct boot configuration is stored in the zImage.
4. Create an `initrd` in your home directory by typing the following command:
```
# mkinitrd --with=tg3 --rootfs=nfs --net-dev=eth0 \
  --rootdev=<nfs server>:/<path to nfsroot> ~/initrd-<kernel-version>.img \
  <kernel-version>
```
   This `initrd` must be created using the Red Hat Enterprise Linux (RHEL) 5.1 kernel.
5. Create a `zImage` that includes the new initial RAM disk (initrd) by issuing a command of this form:
```
# mkzimage /boot/vmlinuz-<kernel-version> /boot/config-<kernel-version> \
  /boot/System.map-<kernel-version> <initrd> \
  /usr/share/ppc64-utils/zImage.stub <zImage>
```
   In the command:

   `<kernel-version>`
   is the version of your Linux kernel.

   `<initrd>`
   is the `initrd` you created in the previous step.

   `<zImage>`
   is the path and name of the `zImage` to be created.

   For example, to use an initial RAM disk `/boot/initrd-2.6.18-28.el5.img` to create a zImage `/boot/zImage.initrd-2.6.18-28.el5`, type:
```
# mkzimage /boot/vmlinuz-2.6.18-28.el5 /boot/config-2.6.18-28.el5 \
  /boot/System.map-2.6.18-28.el5 /boot/initrd-2.6.18-28.el5.img \
  /usr/share/ppc64-utils/zImage.stub /boot/zImage.initrd-2.6.18-28.el5
```

Install the zImage:

After you have created or downloaded a zImage, you should install it.

Copy the `zImage` to the exported TFTP directory on the TFTP server. Assuming that the exported TFTP directory is named `tftp_zImages` and mounted as `/mnt/tftp_zImages` type for example the following command for RHEL 5.1:

```
# cp /boot/zImage.initrd-2.6.18-28.el5 /mnt/tftp_zImages/
```

If you downloaded the `zImage` for Fedora 7, type for example:

```
# cp /boot/zImage.initrd-2.6.22-5.20070920bsc /mnt/tftp_zImages/
```

Ensure that the `zImage` file name matches the one configured in DHCP.

Set up the root file system on the network:

Copy the root file system to the NFS server and make some changes to adapt it to an individual BladeCenter server.

Perform the following steps to deploy your root file system on the NFS server:

1. On the POWER-based installation system, enable SSH logins by changing the entry `SELINUX='enforcing'` in `/etc/selinux/config` to `SELINUX='permissive'`.
2. Copy the root file system to the NFS directory by typing the following command on the NFS server:

   ```
   # rsync -avp -e ssh -x root@<POWER-server>:/ /<NFS-dir>
   ```

   where *<NFS-dir>* is the directory to which you want to copy the root file system. For example, enter:

   ```
   # rsync -avp -e ssh -x root@192.0.2.20:/ /nfsroot/root_for_192.0.2.10
   ```
3. Make the following changes to the copy of the root file system on the NFS server.
   - To make this image bootable, edit the /etc/fstab file and comment out the entries for "`/`", "`/boot`" and "`swap`". Failure to comment these out will cause errors on startup.
   - Change the first line in `/etc/fstab` so that it specifies your NFS directory as the location of the root file system. For example, change line

     ```
     /dev/VolGroup00/LogVol00 /                       ext3    defaults       1 1
     ```

     to read

     ```
     192.0.2.22:/nfsroot/root_for_192.0.2.10 /        nfs     tcp,nolock     1 1
     ```
   - Change the host name in `/etc/hosts` and `/etc/sysconfig/network` to the host name of your BladeCenter QS21.
   - Remove the `/dev/VolGroup00/LogVol01 swap` line from `/etc/fstab`.

You now have a root file system in place for your BladeCenter QS21.

Tip: By copying this root file system to another directory on your NFS server you can easily create the root file system for further BladeCenter servers. After creating a copy you need to change the host name to that of the respective BladeCenter server.

Boot from the network:

You are now ready to boot your BladeCenter QS21 from the network.

1. Ensure that the DHCP server has a DHCP configuration entry for your BladeCenter QS21. The entry must assign an IP address to the BladeCenter server and include the filename keyword to specify the name of the zImage to be used for booting. Here is an example of the /etc/dhcpd.conf file:

   ```
   # QS21-4614
   host QS21-4614 {
      hardware ethernet 00:1A:64:0E:01:01;
      next-server 192.168.70.100;
      fixed-address 192.168.70.101;
      filename "zImage.initrd-2.6.22-5.20070920bsc";
      option root-path \
         "192.168.70.100:/srv/netboot/QS21/F7sdk30/boot/192.168.70.101/";
   }
   ```

2. After adding the BladeCenter QS21 system(s) to the /etc/dhcpd.conf file, restart DHCP by typing the following the command:

   ```
   # service dhcpd restart
   ```

3. Ensure that the NFS server has an NFS export configuration entry for the BladeCenter QS21. The following is an example /etc/exports file:

   ```
   /nfsroot/root_for_192.0.2.10 192.0.2.10(rw,no_root_squash)
   ```

4. If you modified the /etc/exports file, restart the NFS server by typing the following command:

   ```
   # service nfs restart
   ```

5. Select **Network** as the first device of the boot sequence for the BladeCenter QS21. Make this change in the Management Module. Finally, boot the system.

Finishing the Linux installation

At this point you have Fedora 7 or RHEL 5.1 installed, rebooted, and running on your system.

A default Linux system might not have all of the packages required to install the SDK. To install these packages, follow the instructions in “Software prerequisites” on page 3.

If you want to understand the components that make up the SDK, and optional components you can install, see Chapter 4, “SDK components,” on page 17.

Now proceed to Chapter 5, “Installing, uninstalling, and updating the SDK,” on page 31.

Chapter 4. SDK components

The topic describes the components of the SDK and how they are packaged. Use this information to understand what gets installed and how to configure the installation for your own specific purposes.

SDK target platforms

The SDK can be installed on different target platforms. The development (build) platforms for cross compilation of Cell Broadband Engine Architecture (CBEA) code are X86 and X86_64 machines. The native development platforms are 64-bit PowerPC and CBEA-compliant machines. There are essentially only two execution platforms, either CBEA hardware or the Full-System Simulator when available. Note that you can also run the Full-System Simulator on CBEA hardware to assist with debugging. Executables built on any development platform should run on any execution platform using the same operating system.

The following table summarizes the development and execution platforms available for Red Hat Enterprise Linux (RHEL) 5.1:

Table 2. RHEL 5.1 platforms

Development plaform	CBEA execution platform (BladeCenter QS21)
X86	
X86_64	
PPC64	
CBEA	✓

The following table summarizes the development and execution platforms available for Fedora 7:

Table 3. Fedora 7 platforms

Development plaform	CBEA execution platform (BladeCenter QS20 or BladeCenter QS21)	Full System Simulator execution platform
X86		✓
X86_64		✓
PPC64		✓
CBEA	✓	✓

SDK directories

The SDK installs files into a number of different directories depending on the host platform and file type. This section describes the SDK standards for directories to help you understand where to find the parts of the SDK and how to best use the SDK development environment.

The root directory for the SDK is **/opt/cell**. Most of the SDK files are in this directory. There are three exceptions:

- **/usr** follows the Linux filesystem hierarchy standard (FHS), except for prototype-level code which is placed in the **/opt/cell/sdk/prototype/usr** directory. In some cases, subdirectories are used to store individual components under **include**, **lib** or **lib64**.
- **/opt/ibm/systemsim-cell** contains the IBM Full System Simulator for Cell Broadband Engine Architecture (Fedora 7 only).
- **/opt/ibmcmp** contains the IBM XL C/C++ Alpha Edition for Multicore Acceleration compiler.

There are three main directories under **/opt/cell**:

- **sdk** contains the SDK files.
- **sysroot** contains a *fakeroot* used for cross compilation on X86 and X86_64 architecture systems. There are directories under the **/opt/cell/sysroot** directory that mirror either a native host system (such as **/usr**) or mirror the SDK **/opt/cell/sdk** directory.
- **toolchain** contains the GCC toolchain.

There are various subdirectories for parts of the SDK under the **/opt/cell/sdk** directory:

- **docs** contains the SDK documentation.
- **prototype** contains level 1 components. This is a separate directory to clearly distinguish those parts of the SDK that might change in a future release. Subdirectories of **/opt/cell/sdk/prototype** are similar to peer directories, for example there are **doc**, **src**, and **usr** directories below this directory.
- **src** contains source code such as examples.
- **usr** contains host-based tools.

RPMs

The SDK is distributed as a set of Red Hat Package Manager (RPM) files that can be installed on the target platform. The list of available RPMs that can be installed depends on the host Linux operating system, the target hardware platform, and the options chosen by the user when installing the SDK. The SDK also depends on a number of RPMs provided by the base Linux operating system.

The SDK RPMs follow typical RPM naming conventions including version and revision, and standard name suffixes such as *devel* for development code and *debuginfo* for GDB debugging data. The SDK includes additional conventions that make is easier to identify what the RPM is used for. The following table details these conventions:

Table 4. RPM naming conventions

Convention	Explanation
RPM version number	For IBM-owned code the version number is always 3.0 or 3.0.0 to reflect this version of the SDK.
-source suffix	The RPM contains source code, typically used for examples built using the SDK rather than rpmbuild which uses a SRPM (src.rpm).
-cross-devel suffix	The RPM contains development code for a cross-build environment (X86 or X86_64) rather than a native one.

Table 4. RPM naming conventions (continued)

Convention	Explanation
-trace suffix	The RPM contains libraries that have been enabled for the IBM Performance and Debugging Tool (PDT).
-hybrid suffix	The RPM contains libraries that are used in a hybrid runtime environment where the host is an X86_64 platform and the accelerator is a CBEA platform.
cell- prefix	The RPM is oriented for CBEA platforms and can be used to differentiate the RPM from a standard implementation.
ppu- prefix	The RPM contains a PPU-only library.
spu- preifx	The RPM contains a SPU-only library.

The SDK RPMs also use a number of different RPM targets. They are listed in the following table:

Table 5. SDK target platforms

Architecture/Platform	Explanation
PPC, PPC64	A CBEA application can be either 32-bit or 64-bit. Regular PowerPC platforms are treated as native for CBEA code only for development. Execution of this code still needs either a CBEA -compliant hardware platform or the Full-System Simulator.
i386, i686 (X86)	This is native code that executes on a 32-bit X86 platform.
X86_64	This native code only executes on a 64-bit X86 platform and is used for the hybrid programming model.
noarch	*noarch* is generally used to indicate an architecture-neutral RPM. For the SDK, noarch has the additional meaning that the RPM contains PPC or PPC64 target code that is to be installed on an X86 or X86_64 system for cross compilation. The noarch target is used so that the file will install without complaints from RPM or YUM.
src	The source code for some SDK components are available as SRPMs (src.rpm).

SDK component descriptions

The SDK is divided into components, each of which is at a particular level of development. Some components are prototype code and others have been fully tested and are warranted by IBM with the appropriate purchased license.

The following table details the component development levels:

Table 6. Component development levels

Development level	Description
1	Prototype-level code. There is no guarantee that the features and API will not change in a future release. IBM is particularly interested in customer feedback about this component.
2	Beta-level code that is stable.
3	Product-level code that is stable. However the function, which is typically example code, is provided on an *as-is* basis and might not be maintained or upgraded by IBM.

Table 6. Component development levels (continued)

Development level	Description
4	Product-level code that is stable and has been fully tested. This code is warranted on certain platforms and is fully supported by IBM through standard support channels.
5	GPL and LGPL open source code that is not directly supported by IBM but has been tested with the SDK.

The following table provides the list of SDK components with license, development level, and functional descriptions:

Table 7. SDK component list

Component	Level	License	Description
ALF for Cell BE	4	IBM	Accelerator Library and Framework (ALF) provides for ease of use in multi-core computing by simplifying the data distribution and work queue management for multiple tasks. The host is the PPU and the SPUs are the accelerators. The source and examples are available under a BSD license.
ALF for Hybrid-x86	1	IBM	This version of ALF is directed toward a hybrid computing environment with an X86_64 host and CBEA hardware accelerators.
BLAS	4	IBM	BLAS library for single and double precision linear algebra functions. The examples are available under a BSD license.
Cell Performance Counter	2	IBM	The cell-perf-counter (cpc) tool is used for setting up and using the hardware performance counters in the Cell BE processor. These counters allow you to see how many times certain hardware events occur, which is useful if you are analyzing the performance of software running on a Cell BE system. This component is only available for the Fedora 7 platform.
Crash SPU Commands	5	GPL	Crash extension with specific commands for analyzing Cell Broadband Engine Architecture SPU run control state. This component is only available for the Fedora 7 platform.
DaCS for Cell BE	4	IBM	The Data Communication and Synchronization (DaCS) library contains functions for process management, data movement, data and process synchronization, topology features (such as the group concept), and error handling. DaCS is used only on CBEA hardware. The source and examples are available under a BSD license.
DaCS for Hybrid-x86	1	IBM	The Data Communication and Synchronization (DaCS) library contains functions for process management, data movement, data and process synchronization, topology features (such as the group concept), and error handling. DaCS for Hybrid-x86 is used between an X86_64 host and CBEA hardware.

Table 7. SDK component list (continued)

Component	Level	License	Description
Documentation	4	IBM	Documentation consists of man pages, PDFs, and README files in individual directories. The PDFs for the SDK are installed into directories under the /opt/cell/sdk/docs directory.
Examples	3	IBM	This component contains example code including example libraries, demos, and a tutorial. The source code is available under a BSD license.
FDPR-Pro	4	IBM	The Feedback-directed post-link program optimization tool allows you to instrument a program, run the instrumented version to collect its profile, and create a semantically-equivalent optimized version using that profile.
GCC Toolchain	4	GPL	The GNU Toolchain packages provide a full development tool chain (GCC compiler, assembler, linker, debugger, binary utilities, and runtime library) to generate and debug code for the Cell BE PPE and SPE processor cores. The toolchain is provided both as native version running on Cell Broadband Engine Architecture and other PowerPC Linux systems, and as a cross toolchain hosted on X86 or X86_64 Linux systems. See the *SDK 3.0 Programmer's Guide* for more information on how to use the GNU Toolchain.
Hybrid Performance Tools	1	IBM	These tools are designed to help you use a number of the performance tools in a hybrid system with more than one processor architecture. In particular, the Cell Broadband Engine is used as an accelerator for a host system with a different architecture.
IDE	4	IBM	Eclipse-based integrated development environment for the SDK.
Kernel	5	GPL	The operating system kernel with Cell Broadband Engine Architecture support. The kernel is included in RHEL 5.1. For Fedora 7, the kernel is part of the SDK.
LIBFFT	1	IBM	This library provides a wide range of 1D and 2D Fast Fourier Transforms.
LIBSPE1/LIBSPE2	5	LGPL	A low level library that defines the user space API to program for Cell Broadband Engine Architecture applications. LIBSPE2 is supplied with RHEL 5.1. For Fedora 7, LIBSPE2 and backwards compatibility support of LIBSPE1 is provided.
MASS Library	4	IBM	The Mathematical Acceleration Subsystem (MASS) consists of libraries of mathematical intrinsic functions, which are tuned specifically for optimum performance on the Cell BE processor. Currently 32-bit, 64-bit PPU, and SPU libraries are supported.
netpbm	5	GPL	This graphics bitmap library is used by the Julia example. A cross development version is provided in the SDK for use on X86 and X86_64 platforms.
numactl	5	LGPL	A library for tuning Non-Uniform Memory Access (NUMA) machines.

Table 7. SDK component list (continued)

Component	Level	License	Description
OProfile	5	GPL	OProfile is a tool for profiling user and kernel level code. It uses the hardware performance counters to sample the program counter every *N* events. This component is only available for the Fedora 7 platform.
PDT	4	IBM	The Performance Debugging Tool (PDT) provides the ability to trace events of interest during application execution, and record data related to these events from the SPEs, the PPE, and the AMD Opteron processor.
PDTR	4	IBM	`pdtr` is a command line tool that reads and post-processes PDT traces.
Random Number Library	1	IBM	A random number generator library suitable for simulation.
SIMDMath	4	IBM	A math library that takes advantage of the Single Instruction, Multiple Data (SIMD) instructions in CBEA-compliant hardware.
Simulator	2	IBM	A full system simulation infrastructure and tools for the Cell Broadband Engine™ processor. This component is only available for the Fedora 7 platform.
SPU-Isolation	1	IBM	SPU-Isolation provides a build and runtime environment for signing and encrypting SPE applications. This component is only available for the Fedora 7 platform.
SPU-Timer	1	IBM	The SPU timer library provides virtual clock and timer services for SPU applications.
SPU-Timing Tool	2	IBM	The SPU static timing tool *spu_timing* annotates an SPU assembly file with scheduling, timing, and instruction issue estimates assuming a straight, linear execution of the program.
Sysroot Image	5	GPL/ LGPL	The system root image for the Full System Simulator is a file that contains a disk image of Fedora 7 files, libraries, and binaries that can be used within the simulator. This component is only available for the Fedora 7 platform.
XL C/C++	2	IBM	The IBM XL C/C++ Alpha Edition for Multicore Acceleration for Linux is an advanced, high-performance cross compiler that is tuned for the Cell Broadband Engine Architecture. It allows users to compile and link PPU and SPU code segments with a single compiler invocation.

Notes about the table:

1. For RHEL 5.1, the Kernel and LIBSPE components are supplied with the operating system and not the SDK.
2. The following components are not available for RHEL 5.1 because the necessary prerequisites are not available:
   - Crash SPU commands
   - Cell Performance Counter

- OProfile
- SPU-Isolation
- Full-System Simulator and Sysroot Image
3. The following components are provided for Fedora 7 as a convenience or to ensure correct functionality of a dependent component:
   - netbpm cross development package
4. For the SDK, components that are at level 3 or 4 are installed by default except for the IDE, Crash SPU commands, PDT, and PDTR which are optional. Components that are at level 1 or 2 are optional. RPMs that have the *-trace* or *-debuginfo* suffixes are also optional and not installed by default.

YUM groups

YUM provides the ability to group RPMs together to facilitate installing a number of RPMs simultaneously and for categorization in the Pirut GUI.

The following groups are defined in the YUM metadata files located in /opt/cell/yum-repos:

- Cell Runtime Environment
- Cell Development Libraries
- Cell Development Tools
- Cell Performance Tools
- Cell Programming Examples
- Cell Simulator

The *Cell Runtime Environment* group contains the RPMs that are only needed for runtime execution of Cell Broadband Engine Architecture applications. It does not contain any development libraries, tools or example code. When you use the `--runtime option` of the cellsdk script, only the *Cell Runtime Environment* group is included in the install, update or uninstall.

You can use the following YUM group commands to find out which RPMs are in a group and which groups are already installed:

- `groupinstall group1 [group2] [...]`
- `groupupdate group1 [group2] [...]`
- `grouplist [hidden]`
- `groupremove group1 [group2] [...]`
- `groupinfo group1 [...]`

You can display the contents of these groups using Pirut, or by running the `cellsdk` script with the --gui option.

The following table lists each component and the YUM group that contains its RPMs. In general, components are typically defined in only one group. One exception is that if a *Cell Development Library* has a runtime RPM, then that RPM is in the *Cell Runtime Environment* group. Also if a *Cell Development Library* includes example code then that example code RPM is in the *Cell Programming Examples* group.

Table 8. YUM groups for each SDK component

Component	YUM group
ALF for Cell BE	Cell Development Libraries
ALF for Hybrid-x86	Cell Development Libraries
BLAS	Cell Development Libraries
Cell Performance Counter	Cell Performance Tools
Crash SPU Commands	Cell Runtime Environment
DaCS for Cell BE	Cell Development Libraries
DaCS for Hybrid-x86	Cell Development Libraries
Documentation	Cell Development Libraries
Examples	Cell Programming Examples
FDPR-Pro	Cell Performance Tools
GCC Toolchain	Cell Development Tools
Hybrid Performance Tools	Cell Performance Tools
IDE	Cell Development Tools
Kernel	Cell Runtime Environment
LIBFFT	Cell Development Libraries
LIBSPE/LIBSPE2	Cell Development Libraries
MASS Library	Cell Development Libraries
netpbm	Cell Development Libraries
numactl	Cell Development Libraries
OProfile	Cell Performance Tools
PDT	Cell Performance Tools
PDTR	Cell Performance Tools
Random Number Library	Cell Development Libraries
SIMDMath	Cell Development Libraries
Simulator	Cell Simulator
SPU-Isolation	Cell Development Libraries
SPU-Timer	Cell Development Libraries
SPU-Timing Tool	Cell Performance Tools
Sysroot Image	Cell Simulator
XL C/C++	Cell Development Tools

After installing the SDK you might want to install, update or uninstall components or individual RPMs in the SDK. You can install an RPM such as *alf-hybrid-devel* by typing the following command:

```
# yum install alf-hybrid-devel
```

YUM uses its repository information to ensure that you can only install the correct RPM on each platform. Some RPMs are only available with a target platform of X86_64 because they are needed for building X86_64 code using a host-based compiler such as GCC. The SDK contains several hybrid programming model libraries and performance tools.

RPMs by component

This topic provides information about the list of RPMs for each component of the SDK.

YUM defines the install type of an RPM as follows:

Table 9. YUM installation choices

Option	Result
m (mandatory)	The group will not install if any mandatory RPMs are missing. Mandatory RPMs are not displayed in Pirut.
d (default)	Installed by default. Automatically selected in Pirut.
o (optional)	Does not install automatically. Must be selected in Pirut or individually installed from the command line.

The following are examples of how to select and install the RPMs for an optional component:

- If you are interested in working with Monte Carlo simulations, install the Random Number Generator library. This component is optional and has development packages for both native and cross compilation. If you are developing code on an X86 machine, type the following command to install the Random Number Generator library:

  ```
  # yum install libmc-rand-cross-devel
  ```

  Because this component is listed as level 1 (prototype code), the development headers and libraries are installed into the /opt/cell/sdk/prototype/src/usr directory. For cross compilation, this directory is prefixed by the cross directory /opt/cell/sysroot. The example code is placed in the /opt/cell/sdk/prototype/src/examples directory. See "SDK directories" on page 17 for more information.
- If you are interested in developing applications using the ALF programming model but in a hybrid host-accelerator environment, install the optional ALF for Hybrid-x86 component. This component has both runtime and development RPMs. The runtime RPMs are needed on an X86_64 machine for the host and a BladeCenter QS20 or BladeCenter QS21 for the accelerator.

  To develop applications on an X86_64 system, the development RPMs are needed. Because ALF for Hybrid-x86 depends on ALF for Cell BE, YUM will install these dependencies if they are not already installed.

  Therefore, on the X86_64 system type the following command:

  ```
  # yum install alf-hybrid alf-hybrid-devel alf-hybrid-cross-devel \
    alf-hybrid-examples-source
  ```

  On a BladeCenter QS20 or BladeCenter QS21, type the following command:

  ```
  # yum install alf-hybrid
  ```

  You might also want to install the ALF man pages that are provided in the *alfman* RPM.

  Note: ALF for Hybrid-x86 depends on DaCS for Hybrid-x86 which should be configured after installation. See Chapter 6, "DaCS for Hybrid-x86 configuration," on page 59.

The following table lists every RPM name by component and install type. Use this information to select additional RPMs that you want to install or uninstall.

Table 10. RPMs by component

Component	RPM Name	Install Type
ALF for Cell BE	alf	Default
ALF for Cell BE	alf-cross-devel	Default
ALF for Cell BE	alf-debuginfo	Optional
ALF for Cell BE	alf-devel	Default
ALF for Cell BE	alf-examples-source	Default
ALF for Cell BE	alf-trace	Optional
ALF for Cell BE	alf-trace-devel	Optional
ALF for Hybrid-x86	alf-hybrid	Optional
ALF for Hybrid-x86	alf-hybrid-cross-devel	Optional
ALF for Hybrid-x86	alf-hybrid-devel	Optional
ALF for Hybrid-x86	alf-hybrid-examples-source	Optional
ALF for Hybrid-x86	alf-hybrid-trace	Optional
ALF for Hybrid-x86	alf-hybrid-trace-devel	Optional
BLAS	blas	Default
BLAS	blas-cross-devel	Default
BLAS	blas-devel	Default
Cell Performance Counter	cellperfctr-tools	Optional (See Note 2)
Crash SPU Commands	crash-spu-commands	Optional (See Note 2)
Crash SPU Commands	crash-spu-commands-debuginfo	Optional (See Note 2)
DaCS for Cell BE	dacs	Default
DaCS for Cell BE	dacs-cross-devel	Default
DaCS for Cell BE	dacs-debuginfo	Optional
DaCS for Cell BE	dacs-devel	Default
DaCS for Cell BE	dacs-trace	Optional
DaCS for Cell BE	dacs-trace-devel	Optional
DaCS for Hybrid-x86	dacs-hybrid	Optional
DaCS for Hybrid-x86	dacs-hybrid-cross-devel	Optional
DaCS for Hybrid-x86	dacs-hybrid-devel	Optional
DaCS for Hybrid-x86	dacs-hybrid-trace	Optional
DaCS for Hybrid-x86	dacs-hybrid-trace-devel	Optional
Documentation	alfman	Default
Documentation	cell-documentation	Default
Documentation	dacsman	Default
Documentation	libspe2man	Default
Documentation	simdman	Default
Examples	cell-buildutils	Default
Examples	cell-compliance-tests	Optional
Examples	cell-compliance-tests-cross	Optional

Table 10. RPMs by component (continued)

Component	RPM Name	Install Type
Examples	cell-compliance-tests-source	Optional
Examples	cell-demos	Default
Examples	cell-demos-cross	Default
Examples	cell-demos-source	Default
Examples	cell-examples	Default
Examples	cell-examples-cross	Default
Examples	cell-examples-source	Default
Examples	cell-libs	Default
Examples	cell-libs-cross	Default
Examples	cell-libs-cross-devel	Default
Examples	cell-libs-devel	Default
Examples	cell-libs-source	Default
Examples	cell-tutorial	Default
Examples	cell-tutorial-cross	Default
Examples	cell-tutorial-source	Default
FDPR-Pro	fdprpro	Optional
GCC Toolchain	ppu-binutils	Default
GCC Toolchain	ppu-binutils-debuginfo	Optional
GCC Toolchain	ppu-gcc	Default
GCC Toolchain	ppu-gcc-c++	Default
GCC Toolchain	ppu-gcc-debuginfo	Optional
GCC Toolchain	ppu-gcc-fortran	Optional
GCC Toolchain	ppu-gcc-gnat	Optional
GCC Toolchain	ppu-gdb	Default
GCC Toolchain	ppu-gdb-debuginfo	Optional
GCC Toolchain	ppu-sysroot	Default
GCC Toolchain	ppu-sysroot64	Default
GCC Toolchain	spu-binutils	Default
GCC Toolchain	spu-binutils-debuginfo	Optional
GCC Toolchain	spu-gcc	Default
GCC Toolchain	spu-gcc-c++	Default
GCC Toolchain	spu-gcc-debuginfo	Optional
GCC Toolchain	spu-gcc-fortran	Optional
GCC Toolchain	spu-gdb	Default
GCC Toolchain	spu-gdb-debuginfo	Optional
GCC Toolchain	spu-newlib	Default
GCC Toolchain	spu-newlib-debuginfo	Optional
GCC Toolchain	spu-tools	Optional
GCC Toolchain	spu-tools-debuginfo	Optional

Table 10. RPMs by component (continued)

Component	RPM Name	Install Type
Hybrid Performance Tools	cell-perf-hybrid-tools	Optional
IDE	alf-ide-template	Optional
IDE	cellide	Optional
Kernel	kernel	Mandatory (See Note 1)
Kernel	kernel-debuginfo	Optional (See Note 1)
LIBFFT	libfft	Optional
LIBFFT	libfft-cross-devel	Optional
LIBFFT	libfft-devel	Optional
LIBFFT	libfft-examples-source	Optional
LIBSPE/LIBSPE2	elfspe2	Mandatory (See Note 1)
LIBSPE/LIBSPE2	libspe	Mandatory (See Note 2)
LIBSPE/LIBSPE2	libspe2	Mandatory (See Note 1)
LIBSPE/LIBSPE2	libspe2-adabinding-devel	Optional (See Note 1)
LIBSPE/LIBSPE2	libspe2-cross-devel	Default (See Note 1)
LIBSPE/LIBSPE2	libspe2-debuginfo	Optional (See Note 1)
LIBSPE/LIBSPE2	libspe2-devel	Default (See Note 1)
MASS Library	mass-cross-devel	Default
MASS Library	ppu-mass-devel	Default
MASS Library	spu-mass-devel	Default
netpbm	netpbm-cross-devel	Default (See Note 1)
numactl	numactl	Default
numactl	numactl-cross-devel	Default
numactl	numactl-devel	Default
OProfile	oprofile	Default (See Note 2)
PDT	pdt	Optional
PDT	pdt-cross-devel	Optional
PDT	pdt-devel	Optional
PDT	pdt-module	Optional
PDT	trace-cross-devel	Optional
PDT	trace-devel	Optional
PDTR	pdtr	Optional
Random Number Library	libmc-rand-cross-devel	Optional
Random Number Library	libmc-rand-devel	Optional
SIMDMath	ppu-simdmath	Default
SIMDMath	ppu-simdmath-devel	Default
SIMDMath	simdmath-cross-devel	Default
SIMDMath	simdmath-debuginfo	Optional
SIMDMath	spu-simdmath-devel	Default

Table 10. RPMs by component (continued)

Component	RPM Name	Install Type
Simulator	systemsim-cell	Default (See Note 2)
SPU-Isolation	cell-spu-isolation-cross-devel	Optional (See Note 2)
SPU-Isolation	cell-spu-isolation-devel	Optional (See Note 2)
SPU-Isolation	cell-spu-isolation-emulated-samples	Optional (See Note 2)
SPU-Isolation	cell-spu-isolation-loader	Optional (See Note 2)
SPU-Isolation	cell-spu-isolation-loader-cross	Optional (See Note 2)
SPU-Isolation	cell-spu-isolation-tool	Optional (See Note 2)
SPU-Isolation	cell-spu-isolation-tool-source	Optional (See Note 2)
SPU-Timer	spu-timer-cross-devel	Optional
SPU-Timer	spu-timer-devel	Optional
SPU-Timing Tool	cell-spu-timing	Optional
Sysroot Image	sysroot_image	Default (See Note 2)
XL C/C++	cell-xlc-ssc-cmp	Optional
XL C/C++	cell-xlc-ssc-lib	Optional
XL C/C++	cell-xlc-ssc-omp	Optional
XL C/C++	cell-xlc-ssc-rte	Optional
XL C/C++	cell-xlc-ssc-rte-lnk	Optional

Notes:

1. This RPM is provided by Red Hat Enterprise Linux (RHEL) 5.1 and is not included in the SDK for the RHEL 5.1 platform. This RPM is included in the SDK for the Fedora 7 platform.
2. This RPM is only available for the Fedora 7 version of the SDK.

Chapter 5. Installing, uninstalling, and updating the SDK

This topic describes how to install, uninstall, and update the SDK.

Upgrading to SDK 3.0 from a previous version

Upgrading from a previous version to version 3.0 is not supported. If you already have SDK 2.1 or the SDK 3.0 early release installed, you must first uninstall it. See "Uninstalling the SDK" on page 50..

Default SDK installation

This topic describes the steps to perform a default installation of the SDK.

Follow these steps to install the SDK:
1. "Choose a product set"
2. "Download the SDK files" on page 34
3. "Prepare for installation" on page 35
4. "Install the SDK Installer" on page 37
5. "Start the SDK installation" on page 37
6. Do "Post-install configuration" on page 38

These steps assume you have already installed a supported Linux operating system and have satisfied the prerequisites listed in Chapter 2, "SDK prerequisites," on page 3. There are extra steps specific to RHEL 5.1.

Choose a product set

A product set is a formal grouping of RPMs that compose the SDK. It is further defined as a YUM repository for a specific environment and operating system. Some product sets are packaged as ISO images to distribute the SDK. The YUM repository for each product set is installed and then enabled or disabled as part of installing the *cell-install* RPM.

Product sets are categorized as follows:

Table 11. Product set group descriptors

Descriptor	Options	Rationale
Distributor	IBM or BSC	All GPL or LGPL code is distributed by the Barcelona Supercomputing Center (BSC) and is in separate product sets from the IBM-owned code that is distributed using ISO images from either developerWorks or Passport Advantage.
Operating system	RHEL 5.1 or Fedora 7	As noted in Chapter 4, "SDK components," on page 17, not all of the components are distributed for RHEL 5.1. The SDK requires different product sets for each supported operating system.
License	IPLA, ILAN or ILAER.	Different licences apply to different components.

Product sets with *Open* in the name are not downloadable as an ISO image but are accessed directly by YUM from a directory on the BSC Web site. The product sets without *Open* in the name are distributed as ISO images that you can download from the developerWorks or Passport Advantage Web sites. For example, the ISO for the Devel-Fedora product set is named *CellSDK-Devel-Fedora.iso*.

There are subdirectories on an ISO image for each target platform that contains files linked back to a common RPMs subdirectory on the ISO image.

SRPMs (source RPMs) are also available either on the BSC Web site or on the ISO images. These SRPMs must be installed manually using the rpm command and are not installed by YUM.

The following table lists the components in each Fedora 7 product set:

Table 12. Fedora 7 product set component details

Component	Devel-Fedora	Extras-Fedora	Open-Fedora
ALF for Cell BE	✓		
ALF for Hybrid-x86		✓	
Basic Linear Algebra Subprograms (BLAS)	✓		
Cell Performance Counter		✓	
Crash SPU Commands			✓
DaCS for Cell BE	✓		
DaCS for Hybrid-x86		✓	
Documentation	✓		
Examples	✓		
FDPR-Pro	✓		
GCC Toolchain			✓
Hybrid Performance Tools		✓	
IDE	✓		
Kernel			✓
LIBFFT		✓	
LIBSPE/LIBSPE2			✓
MASS Library	✓		
netpbm			✓
numactl			✓
OProfile			✓
PDT	✓		
PDTR	✓		
Random Number Library		✓	
SIMDMath	✓		
Simulator		✓	
SPU-Isolation		✓	
SPU-Timer		✓	
SPU-Timing Tool		✓	

Table 12. Fedora 7 product set component details (continued)

Component	Devel-Fedora	Extras-Fedora	Open-Fedora
Sysroot Image			✓
XL C/C++		✓	

The following table lists the components in each RHEL 5.1 product set:

Table 13. RHEL 5.1 product set component details

Component	Product-RHEL, Devel-RHEL	Extras-RHEL	Open-RHEL
ALF for Cell BE	✓		
ALF for Hybrid-x86		✓	
Basic Linear Algebra Subprograms (BLAS)	✓		
Cell Performance Counter		N/A	
Crash SPU Commands			N/A
DaCS for Cell BE	✓		
DaCS for Hybrid-x86		✓	
Documentation	✓		
Examples	✓		
FDPR-Pro	✓		
GCC Toolchain			✓
Hybrid Performance Tools		✓	
IDE	✓		
Kernel			Included in RHEL 5.1
LIBFFT		✓	
LIBSPE/LIBSPE2			Included in RHEL 5.1
MASS Library	✓		
netpbm			Included in RHEL 5.1
numactl			✓
OProfile			N/A
PDT	✓		
PDTR	✓		
Random Number Library		✓	
SIMDMath	✓		
Simulator		N/A	
SPU-Isolation		N/A	
SPU-Timer		✓	
SPU-Timing Tool		✓	
Sysroot Image			N/A
XL C/C++		✓	

Download the SDK files

This topic describes how to download the SDK files needed for installation. You can skip this step if you have physical media for the SDK such as a CD.

YUM automatically downloads most of these RPMs so it is not necessary for you to manually download them.

The developerWorks Web site and the Passport Advantage Web site provide the IBM-licensed code and its documentation as ISO images. Passport Advantage is an IBM Web site that gives you information about software maintenance, product upgrades and technical support under a single, common set of agreements, processes and tools.

To download the SDK perform the following steps:

1. Create a temporary directory for the images and the cell-install RPM by typing the following commands:
```
# mkdir -p /tmp/cellsdkiso
# cd /tmp/cellsdkiso
```
2. Download the cell-install RPM from developerWorks or Passport Advantage Web site and place it into the /tmp/cellsdkiso directory that you created in the previous step.
3. Download the ISO images into the same directory.

Here are the choices for ISO images for each supported Linux distribution:

Table 14. ISO images for Red Hat Enterprise Linux (RHEL) 5.1

The *Product* package contains all the mature technologies in SDK 3.0 plus access to IBM Support and is intended for production purposes.	*CellSDK-Product-RHEL_3.0.0.1.0.iso*	http://www.ibm.com/software/howtobuy/passportadvantage/
The *Developer* package is intended for evaluation of the SDK in a non-production environment and contains all the mature technologies in SDK 3.0	*CellSDK-Devel-RHEL_3.0.0.1.0.iso*	http://www.ibm.com/developerworks/power/cell/downloads.html
The *Extras* package contains the "latest and greatest" technologies in the SDK These packages tend to be less mature or are technology preview code that may or may not become part of the generally available product in the future.	*CellSDK-Extra-RHEL_3.0.0.1.0.iso*	http://www.ibm.com/developerworks/power/cell/downloads.html

Table 15. ISO images for Fedora 7

Product set	ISO name	Location
The *Developer* package is intended for evaluation of the SDK in a non-production environment and contains all the mature technologies in SDK 3.0	*CellSDK-Devel-Fedora_3.0.0.1.0.iso*	http://www.ibm.com/developerworks/power/cell/downloads.html
The *Extras* package contains the "latest and greatest" technologies in the SDK These packages tend to be less mature or are technology preview code that may or may not become part of the generally available product in the future.	*CellSDK-Extra-Fedora_3.0.0.1.0.iso*	http://www.ibm.com/developerworks/power/cell/downloads.html

You can verify the integrity of the files using the md5sum command. Checksums are provided on the download Web page.

Prepare for installation

This topic will help you prepare your system for installation of the SDK.

Prepare your system by following these steps:

1. If you have installed an older version of the SDK, you must remove it before you can install this version. See "Uninstalling the SDK" on page 50.
2. If necessary, install or upgrade your operating system. See Chapter 3, "Operating system installation," on page 5.
3. Verify that your BladeCenter QS20 or BladeCenter QS21 has the right firmware level. See "Hardware prerequisites" on page 3.
4. The YUM updater daemon must not be running when installing the SDK. To see if it is running, type the following command:

   ```
   # /etc/init.d/yum-updatesd status
   ```

   If the command returns a result similar to:

   ```
   # /etc/init.d/yum-updatesd status
   yum-updatesd (pid 12260) is running...
   ```

   then type the command:

   ```
   # /etc/init.d/yum-updatesd stop
   ```

   You will see a result similar to:

   ```
   # /etc/init.d/yum-updatesd stop
   Stopping yum-updatesd:                                      [  OK  ]
   ```

   Later in the installation process you will restart the daemon.
5. If you previously added an exclude clause in the /etc/yum.conf file that includes the numactl, numactl-devel, blas, blas-debuginfo, blas-devel, oprofile

or oprofile-debuginfo packages, temporarily remove the clauses to ensure that these packages are installed for the SDK.

6. If you plan to install the FDPR-Pro component, it requires the compat-libstdc++ RPM. For RHEL 5.1 only, this RPM should be installed first, otherwise the install of the FDPR-Pro RPM will fail.
7. The LIBSPE2 libraries are required to execute Cell BE applications. For RHEL 5.1 Power architecture and CBEA-compliant systems, the LIBSPE2 libraries from the RHEL 5.1 supplementary CD are prerequisites for installing some of the SDK RPMs. Type the following commands to install these RPMs:

   ```
   # rpm -ivh libspe2-2.2.0.85-1.el5.ppc.rpm
   # rpm -ivh libspe2-2.2.0.85-1.el5.ppc64.rpm
   ```

8. For RHEL 5.1 cross compilation only (X86 and X86_64), you must build and install the compiler *sysroot* RPMs as described below. That is because these RPMs cannot be provided by SDK and can only be built if you have a RHEL 5.1 license. For Fedora 7, these RPMs are supplied by the SDK and YUM installs them automatically. Follow these steps to build and install the sysroot on an X86 or X86_64 Red Hat Enterprise Linux (RHEL) 5.1 system:
   a. Download the *ppu-sysroot.spec* file from the Barcelona Supercomputing Center Web site: http://www.bsc.es/projects/deepcomputing/linuxoncell/cellsimulator/sdk3.0/sources/toolchain/rhel5-ppu-sysroot/ppu-sysroot.spec. Place this file in the /usr/src/redhat/SPECS directory.
   b. Copy the following RHEL 5.1 PowerPC binary packages (from your PPC architecture RHEL 5.1 distribution medium, such as a CD) into the /usr/src/redhat/SOURCES directory:

      ```
      glibc-2.5-18.ppc.rpm
      glibc-2.5-18.ppc64.rpm
      glibc-devel-2.5-18.ppc.rpm
      glibc-devel-2.5-18.ppc64.rpm
      glibc-headers-2.5-18.ppc.rpm
      kernel-headers-2.6.18-52.el5.ppc.rpm
      gmp-4.1.4-10.el5.ppc.rpm
      gmp-4.1.4-10.el5.ppc64.rpm
      gmp-devel-4.1.4-10.el5.ppc.rpm
      gmp-devel-4.1.4-10.el5.ppc64.rpm
      ```

      Note: These RPM version numbers are correct at the time of writing. They might change with the final release of RHEL 5.1.
   c. You might have to edit the ppu-sysroot.spec file to match the version numbers of the final RHEL 5.1 release RPMs. Verify or edit the version numbers in the following 3 lines:

      ```
      %define glibc_version 2.5-18
      %define kernheaders_version 2.6.18-52.el5
      %define gmp_version 4.1.4-10.el5
      ```

   d. Type the following command to build the RHEL 5.1 ppu-sysroot RPMs:

      ```
      # rpmbuild -ba --target=noarch /usr/src/redhat/SPECS/ppu-sysroot.spec
      ```

      This will create the following sysroot RPMs:

      ```
      /usr/src/redhat/RPMS/noarch/ppu-sysroot-rhel5-2.noarch.rpm
      /usr/src/redhat/RPMS/noarch/ppu-sysroot64-rhel5-2.noarch.rpm
      ```

   e. Type the following commands to install them:

      ```
      # cd /usr/src/redhat/RPMS/noarch/
      # rpm -ivh ppu-sysroot-rhel5-2.noarch.rpm ppu-sysroot64-rhel5-2.noarch.rpm
      ```

      The installation places the target library files into the /opt/cell/sysroot directory.

Note: If you want to install the sysroot on multiple RHEL 5.1 host systems, you do not have to build the RPMs on each system. Just copy the compiler sysroot RPMs to each system and install them.

Install the SDK Installer

This topic shows you how to install the SDK Installer.

The SDK Installer requires the *rsync*, *sed*, *tcl*, and *wget* packages. If they are not installed on your system, type the following command:

```
# yum install rsync sed tcl wget
```

To install the SDK, first install the SDK Installer which is provided by the cell-install RPM package. The naming convention for this file is `cell-install-<rel>-<ver>.noarch.rpm`, where `<rel>` represents the release, and `<ver>` represents the version. If you have physical media such as a CD, you can find this RPM in the root directory.

To install this RPM, type the following command:

```
# rpm -ivh cell-install-3.0.0.1.0.noarch.rpm
```

Note: You cannot use YUM to install this RPM because it is not part of any YUM repository.

Note: If a message appears similar to:

```
warning: cell-install: Header V3 DSA signature: NOKEY, key ID 9ac02885
```

it is because the cellsdk RPM GPG key is not yet installed. The cellsdk script will install it automatically the first time it is run.

Start the SDK installation

This topic describes how to install the SDK. The cellsdk script is a *wrapper* around YUM. Install the SDK by following these steps:

1. Use the cellsdk script to install the SDK.
   - If you installing from an ISO image, type:
     ```
     # cd /opt/cell
     # ./cellsdk --iso /tmp/cellsdkiso install
     ```
   - If you are installing from a local server (see "Setting up a YUM server for the SDK" on page 56) type:
     ```
     # cd /opt/cell
     # ./cellsdk install
     ```
   - If you are installing from a physical CD, load the CD and mount it on the `/media` directory. Now type:
     ```
     # cd  /opt/cell
     # ./cellsdk install
     ```

   Note: You can pass the `--gui` flag to cellsdk to install the SDK using a GUI. See "Installing the SDK using the Pirut GUI" on page 41.
2. Read the SDK licenses.

   There are several licenses that you must agree to. First are the GPL and LGPL licenses. Answer '**yes**' to the license question if you agree to the license terms. The second is either the International Program License Agreement (IPLA) or International License Agreement for Non-Warranted Programs (ILAN). Follow

the on-screen menu to agree to the license. This IBM license is installed into the /opt/cell/license file for later reference. If you downloaded the *Extras* ISO into the ISO directory, then you will also be presented with the International License Agreement for Early Release of Programs (ILAER). Again, follow the on-screen menu to agree to the license.
3. After you agree to both licenses, YUM will install the RPM files. Answer '**y**' to the package install question from YUM.

See Appendix B, "cellsdk script SDK installation example," on page 67 for an example of installing the SDK.

Post-install configuration

After the SDK is installed, finish the installation and configure your system to use the SDK. If you have installed the SDK on RHEL 5.1, you must make specific changes.

Preventing automatic updates from overwriting SDK components

If you are installing on an IBM BladeCenter QS20 or BladeCenter QS21, add the following clause to the /etc/yum.conf file in the [Main] section to prevent a YUM update from overwriting the SDK versions of these runtime RPMs:

```
exclude=blas kernel numactl oprofile
```

For PowerPC platforms, add the following clause to the /etc/yum.conf file:

```
exclude=blas
```

In the future, the YUM update daemon might attempt to update SDK packages with a version not enhanced for the SDK. The exclude line will prevent this from occurring.

Note: If you exclude a package from regular updates, YUM will not automatically update it when a newer version becomes available. If a new version containing security updates or bug fixes is released, you must manually update the RPM.

Installing the Linux Kernel

If you are installing Fedora 7 on BladeCenter hardware, the kernel must be manually installed. First, download the kernel from the Barcelona Supercomputing Center Web site. The kernel RPM URL is http://www.bsc.es/projects/deepcomputing/linuxoncell/cellsimulator/sdk3.0/kernel-2.6.22-5.20070920bsc.ppc64.rpm

Next, install the kernel by typing the following command:

```
# rpm -ivh --force kernel-2.6.22-5.20070920bsc.ppc64.rpm
```

You must reboot to activate this kernel.

Installing the elfspe utility (RHEL 5.1 only)

For CBEA-compliant hardware such as the IBM BladeCenter QS20 and BladeCenter QS21, install the elfspe2 RPM from the RHEL 5.1 supplementary CD. For example, type the following command:

```
# rpm -ivh elfspe2-2.2.0.85-1.el5.rpm
```

After you install the elfspe2 RPM on an IBM BladeCenter QS20 or BladeCenter QS21, ensure that spufs (the SPU File System) is loaded correctly:

- Create the /spu directory by typing the following command:
  ```
  # mkdir -p /spu
  ```
- Add the following line to /etc/fstab if it does not already exist:
  ```
  spufs  /spu  spufs defaults  0  0
  ```

  spufs will now mount automatically at boot. To mount spufs immediately, type the following command:
  ```
  # mount /spu
  ```

Note: The version of SELinux included with RHEL 5.1 might prevent spufs from mounting at boot. If spufs did not mount at boot, type the following command:

```
# mount /spu
```

Disable SELinux if you want spufs automatically mounted at boot.

Installing required development libraries (RHEL 5.1 only)

For RHEL 5.1, you should install additional libraries. This step is not necessary for Fedora 7 because it is done automatically by the installation program.

The versions of the libspe2-devel packages for PPC and PPC64 are needed for application development. These RPMs can be found on the RHEL 5.1 supplementary CD.

To install on X86 or X86_64 architecture systems, type the following commands:

```
# rpm -ivh --force --nodeps --noscripts --ignorearch --root=/opt/cell/sysroot \
   libspe2-2.2.0.85-1.el5.ppc.rpm
# rpm -ivh --force --nodeps --noscripts --ignorearch --root=/opt/cell/sysroot \
   libspe2-2.2.0.85-1.el5.ppc64.rpm
# rpm -ivh --force --nodeps --noscripts --ignorearch --root=/opt/cell/sysroot \
   libspe2-devel-2.2.0.85-1.el5.ppc.rpm
# rpm -ivh --force --nodeps --noscripts --ignorearch --root=/opt/cell/sysroot \
   libspe2-devel-2.2.0.85-1.el5.ppc64.rpm
```

To install on PowerPC architecture or CBEA-compliant systems, type the following commands:

```
# rpm -ivh libspe2-devel-2.2.0.85-1.el5.ppc.rpm
# rpm -ivh libspe2-devel-2.2.0.85-1.el5.ppc64.rpm
```

The PPC version of the netbpm-devel package is needed to compile the Julia demo example code (see /opt/cell/sdk/src/demos/julia_set). To install on X86 or X86_64 architecture systems, type the following command:

```
# rpm -ivh --force --nodeps --noscripts --ignorearch --root=/opt/cell/sysroot \
   netpbm-devel-10.35-6.fc6.ppc.rpm
```

To install on PowerPC architecture or CBEA-compliant systems, type the following command:

```
# rpm -ivh netpbm-devel-10.35-6.fc6.ppc.rpm
```

Installing runtime RPMs on CBEA-compliant systems

If you installed the SDK on a CBEA-complaint system using the --runtime option, the blas and numactl RPMs must be replaced by the SDK versions by following this procedure:

1. Type the following commands to uninstall the operating system blas and numactl packages:

```
# rpm -e --nodeps --allmatches blas
# rpm -e --nodeps --allmatches numactl
```

2. Install the SDK versions of the RPMs. The cellsdk script attempts to put copies of the SDK versions of the RPMs in the /tmp/cellsdk/openSrc directory. The blas RPMs are also in the rpms/cbea subdirectory on the ISO image. The numactl RPMs are available from the BSC Web site at http://www.bsc.es/projects/deepcomputing/linuxoncell/cellsimulator/sdk3.0/CellSDK-Open-RHEL/cbea/. Install these RPMs by typing the following commands:

```
# rpm -ivh blas-3.0-6.ppc.rpm
# rpm -ivh blas-3.0-6.ppc64.rpm
# rpm -ivh numactl-0.9.10-1.ppc.rpm
# rpm -ivh numactl-0.9.10-1.ppc64.rpm
```

Restarting automatic updates

Start the YUM updates daemon by typing the following command as root:

```
# /etc/init.d/yum-updatesd start
```

Adding SDK components

After the SDK is installed, you can install optional packages. See "Installing additional SDK components" on page 50.

The Eclipse IDE

If you have installed the Eclipse IDE RPM, see "Configuring the Eclipse IDE" on page 55 for instructions about how to finish the installation for the IDE.

Hybrid-x86 daemon configuration

If you installed the DaCS for Hybrid-x86 or the ALF for Hybrid-x86 component, see Chapter 6, "DaCS for Hybrid-x86 configuration," on page 59.

Simulator Sysroot

If you have installed the Full-System Simulator and the sysroot image on a Fedora 7 system, ensure that the sysroot image is updated by typing the following command:

```
# /opt/cell/cellsdk_sync_sdk_simulator install
```

See the *Software Development Kit for Multicore Acceleration Version 3.0 Programmer's Guide* for more information.

The SPU-Isolation RPMs

If you have installed the SPU-Isolation RPMs on Fedora 7, you should build the isolation tool. See "Building the SPU-Isolation component" on page 55 for more information about the cellsdk_sync_simulator script.

Installing the SDK using the Pirut GUI

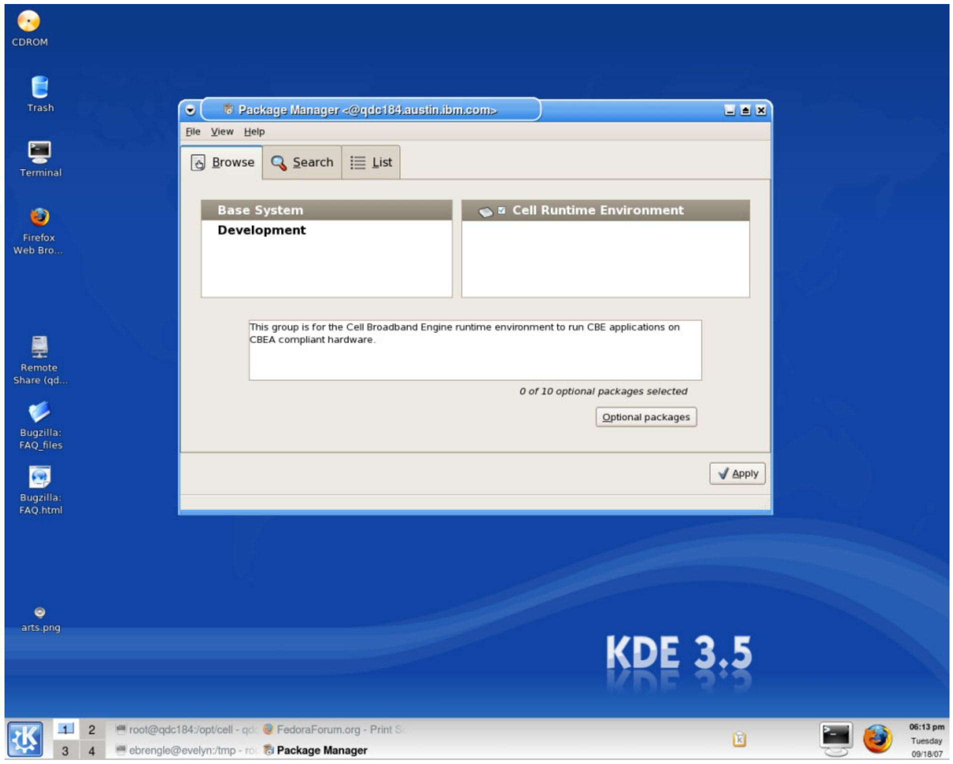
You can install the SDK using the Pirut graphical package manager.

Following the procedure in “Start the SDK installation” on page 37, pass the -g or –gui option to the cellsdk script to launch Pirut. For example, to install from an ISO image using Pirut, type the following commands as root:

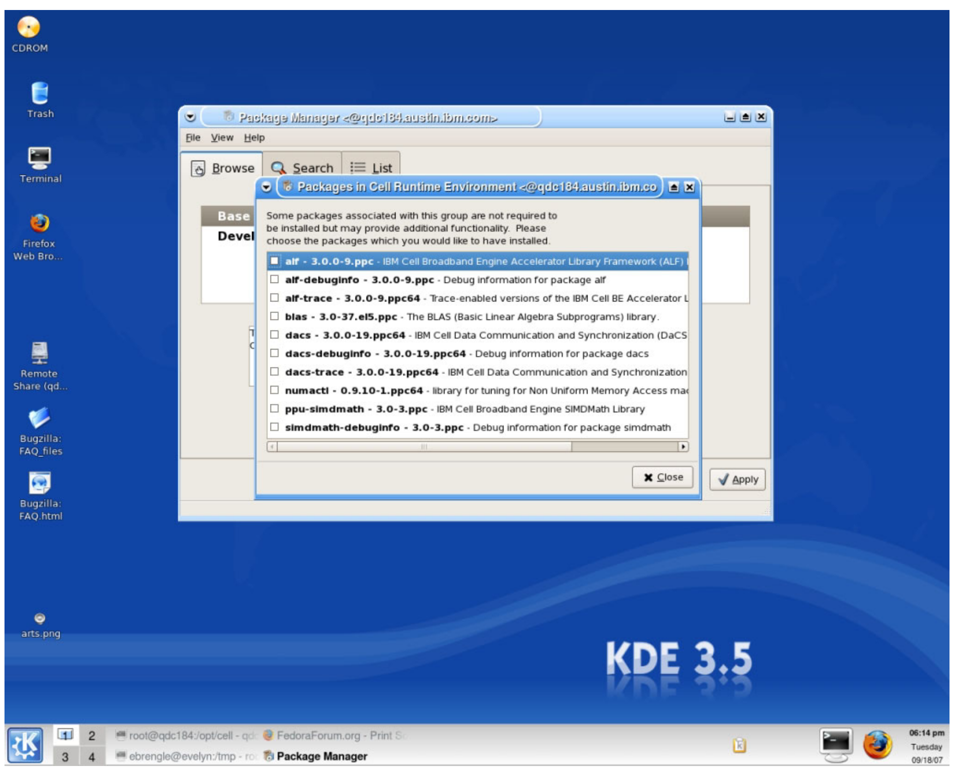
```
# cd /opt/cell
# cellsdk --gui --iso /tmp/cellsdkiso install
```

The following screens are demonstrated on a Red Hat Enterprise Linux (RHEL) 5.1 system. The installation process is similar on Fedora 7.

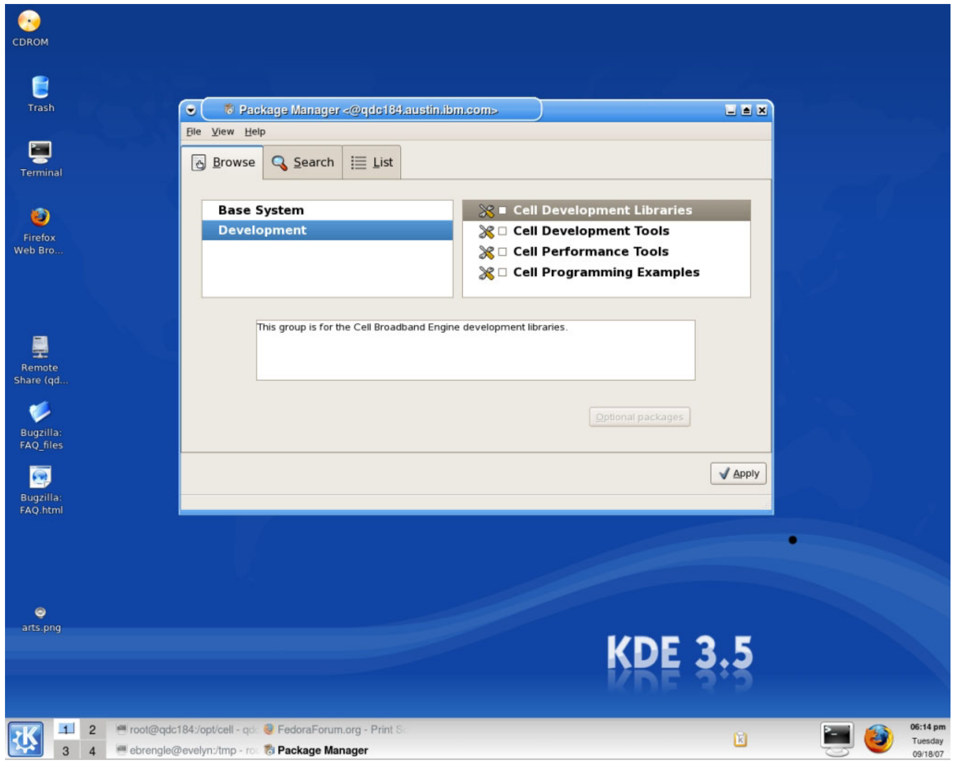
1. This is the first screen you see after starting Pirut. It shows that the *Cell Runtime Environment* is selected to be installed.



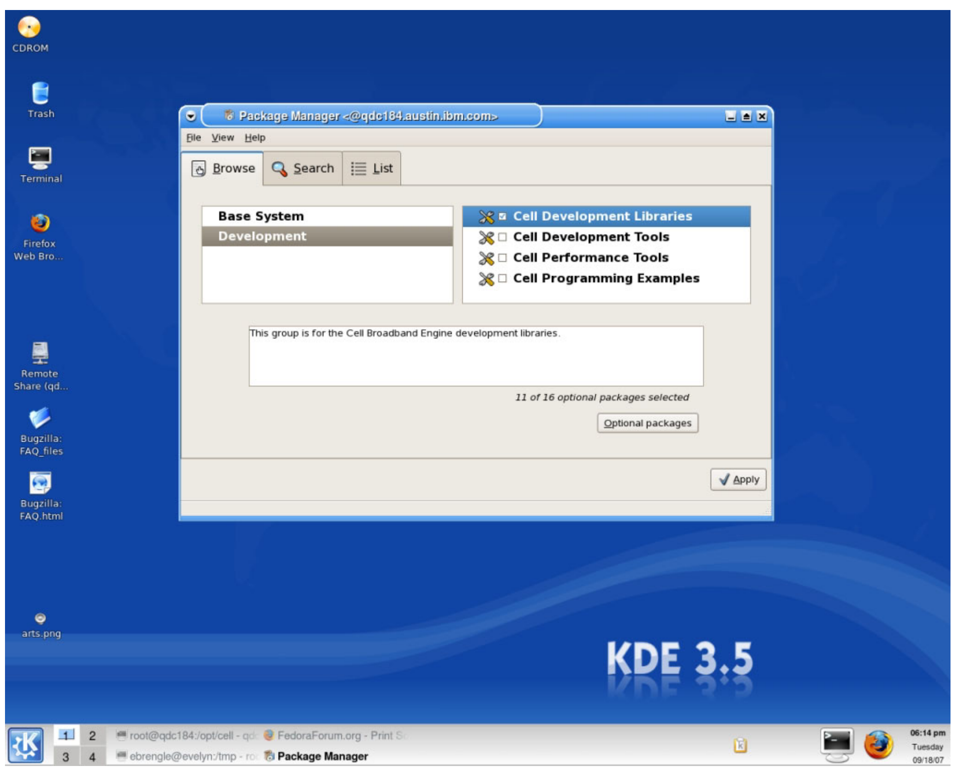
2. After you click **Apply**, Pirut displays the package choices for *Cell Runtime Environment*. Tick the packages you want to install, and click **Close**.



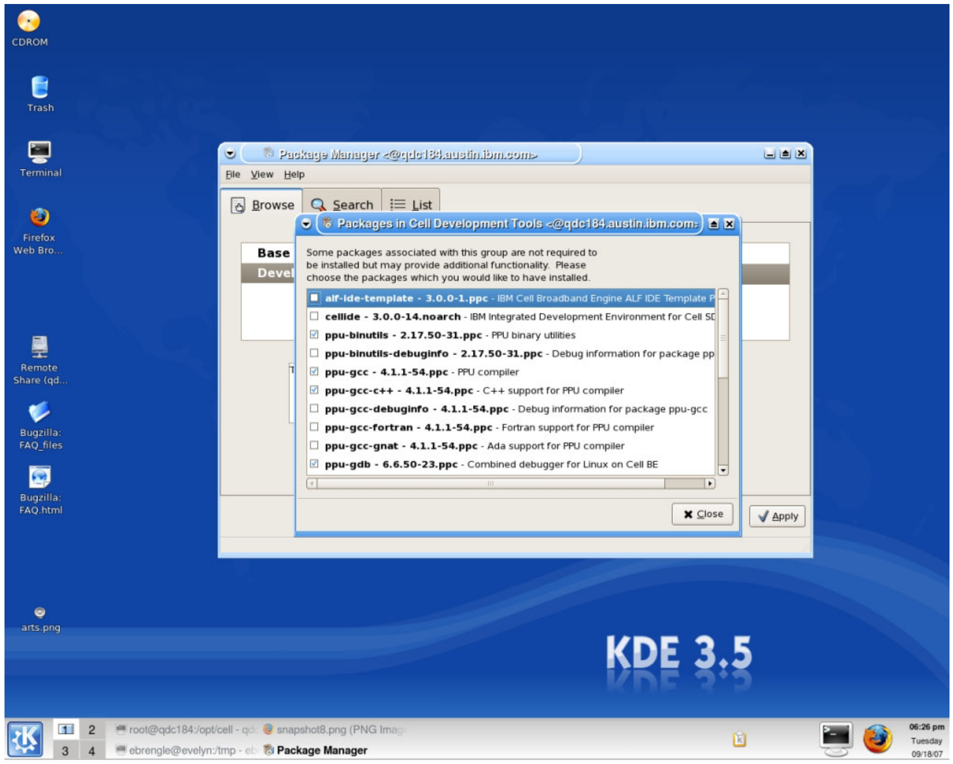
3. Click on *Development* to select development packages:



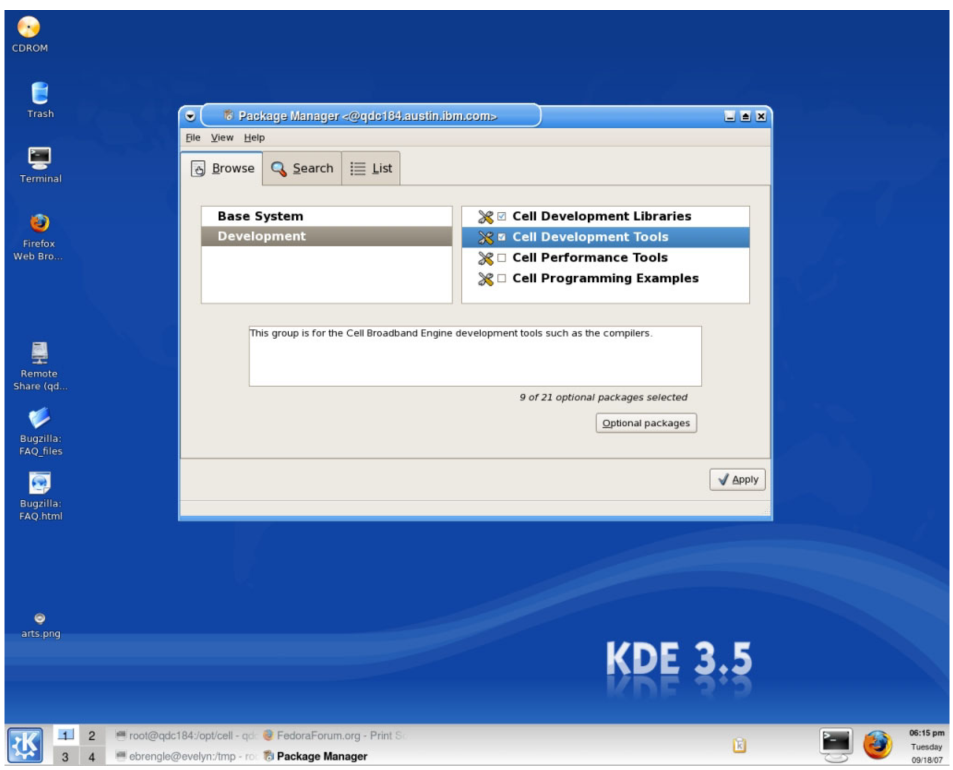
4. Tick the checkbox for *Cell Development Libraries* to have the default Cell BE development libraries installed:



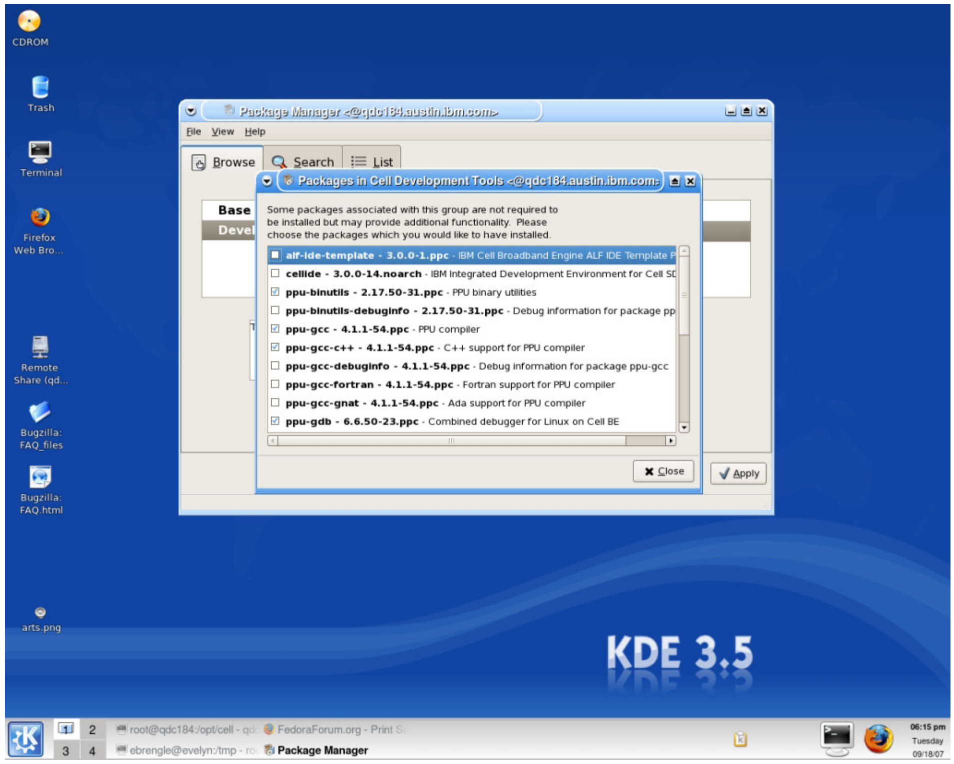
5. Click on *Optional Packages* and make your selection to add or remove individual RPMs:



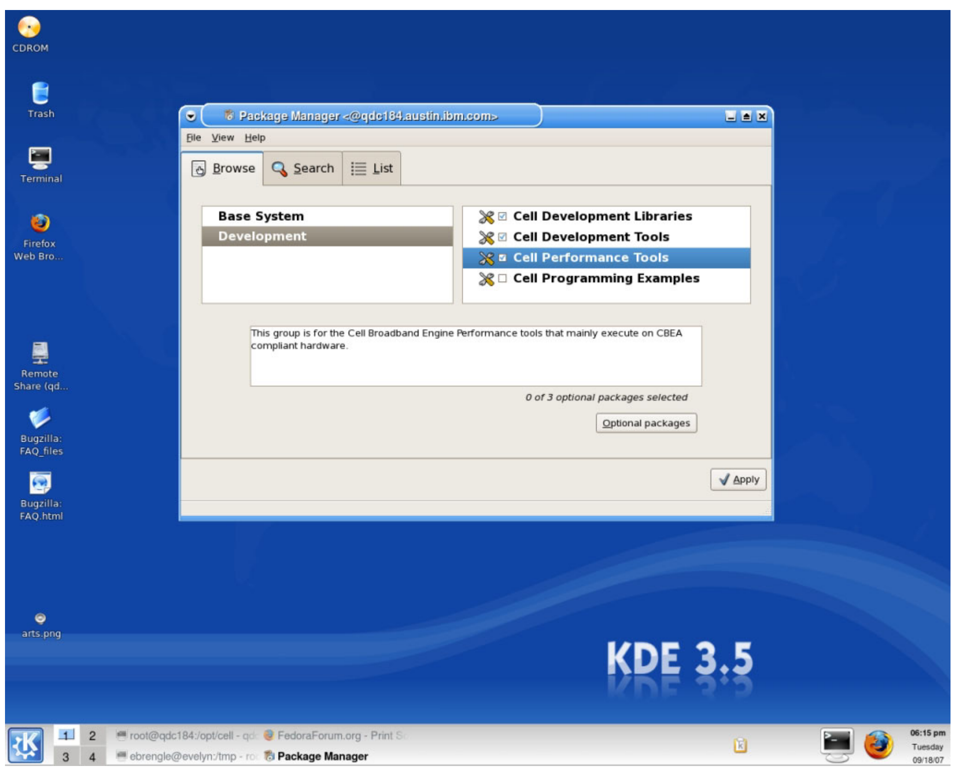
6. Tick the checkbox for *Cell Development Tools* to have the default Cell BE development tools installed:



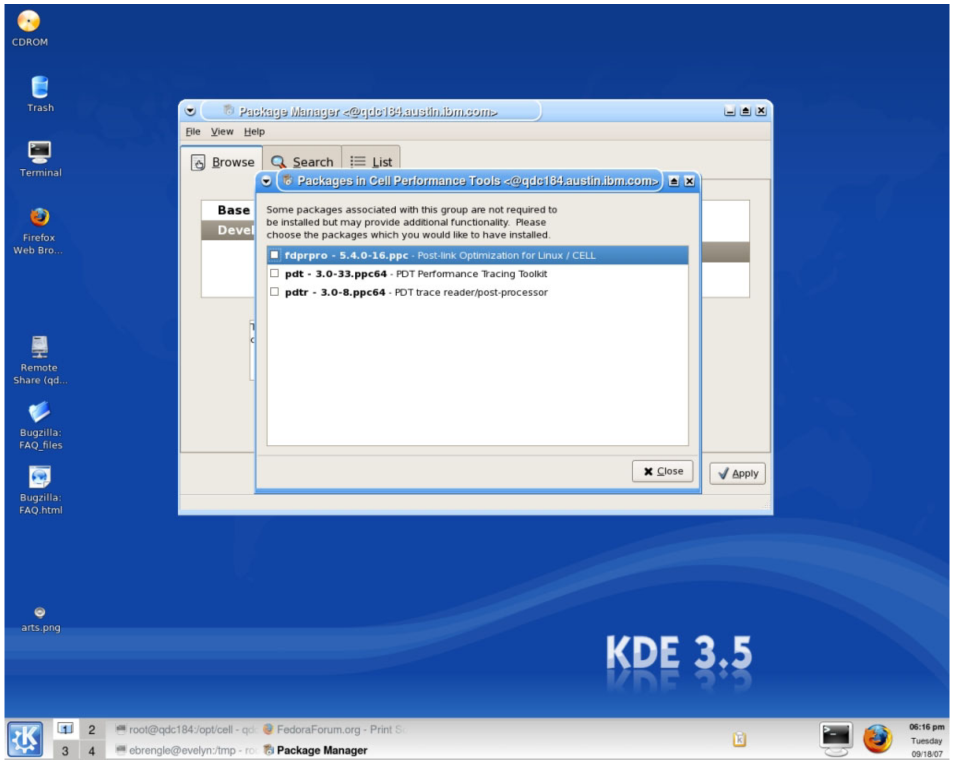
7. Click *Optional Packages* and make your selection:



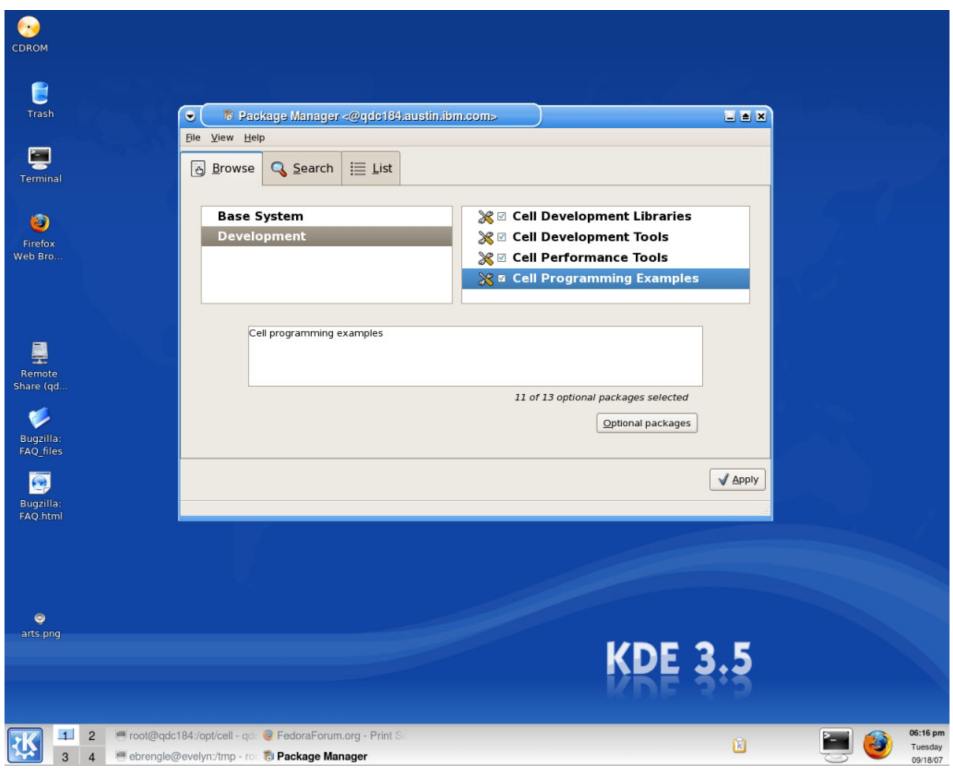
8. Tick the checkbox for *Cell Performance Tools* to have the default Cell BE performance tools installed:



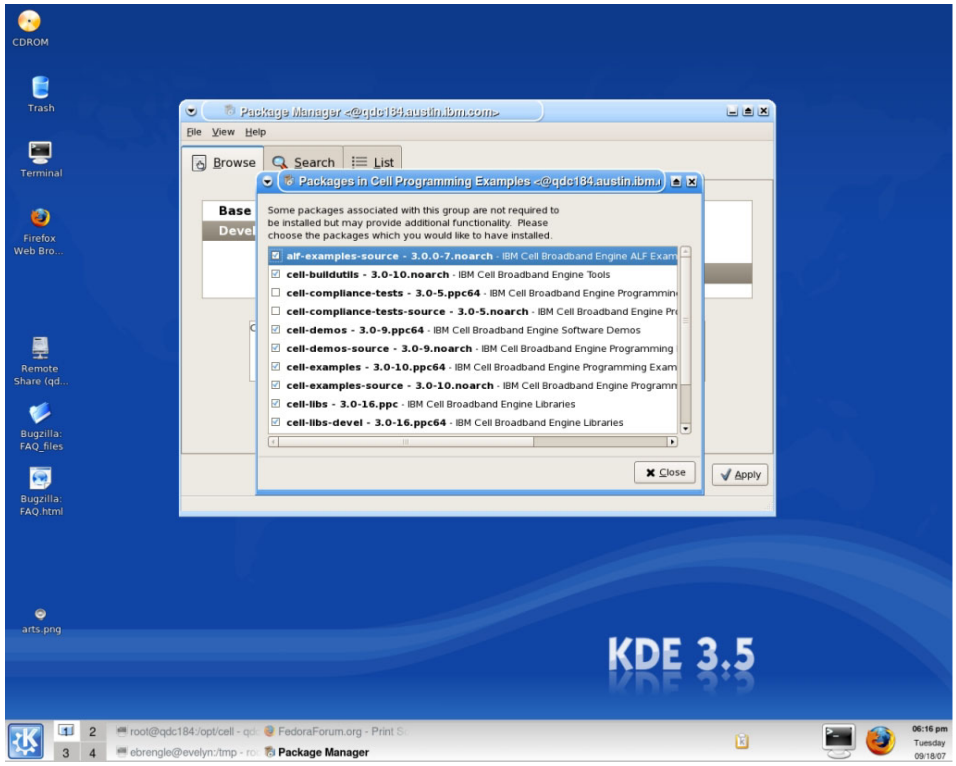
9. Click **Optional packages** and make your selection:



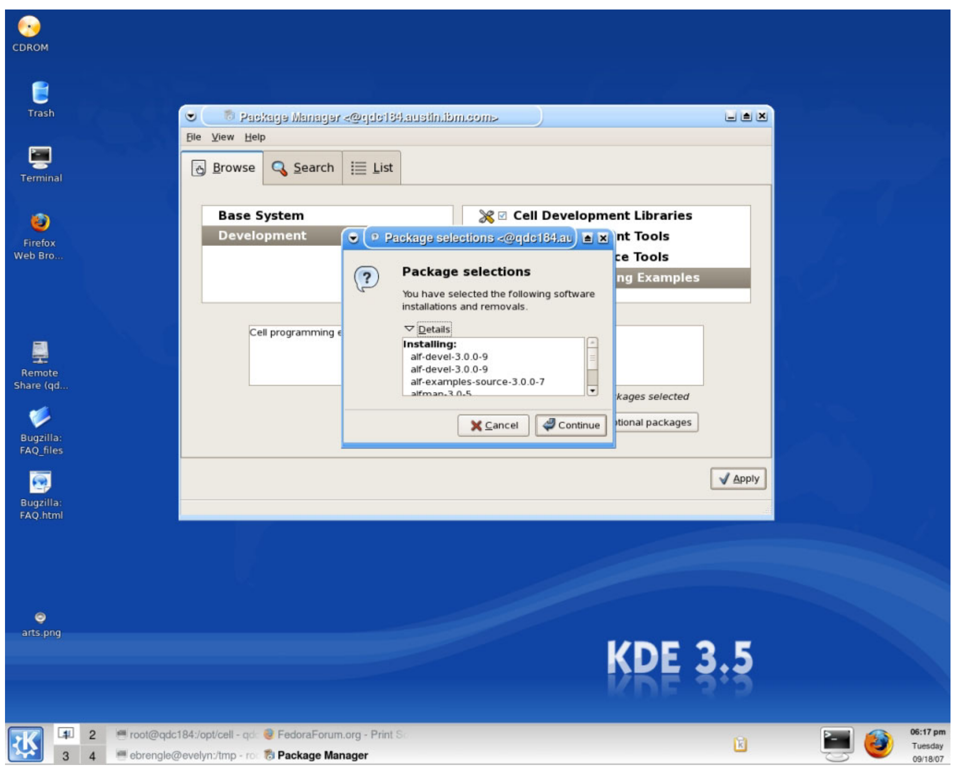
10. Tick the checkbox for *Cell Programming Examples* to have the default Cell BE programming examples installed:



11. Click *Optional packages* and make your selection:

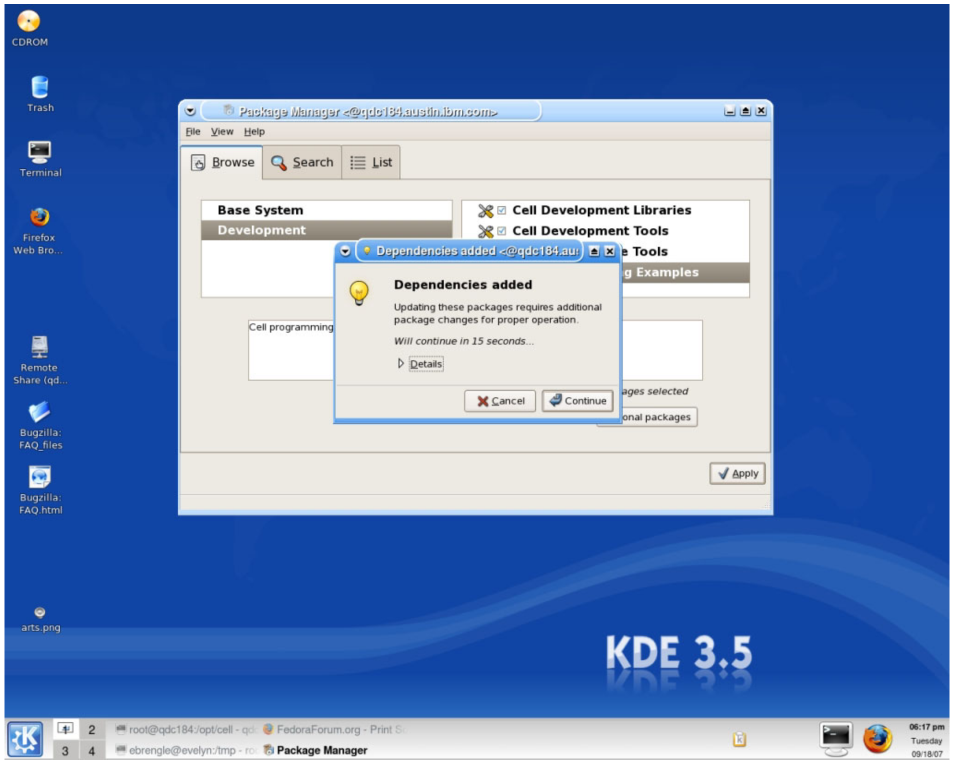


12. Click the **Apply** button. The *Package Selections* window appears:



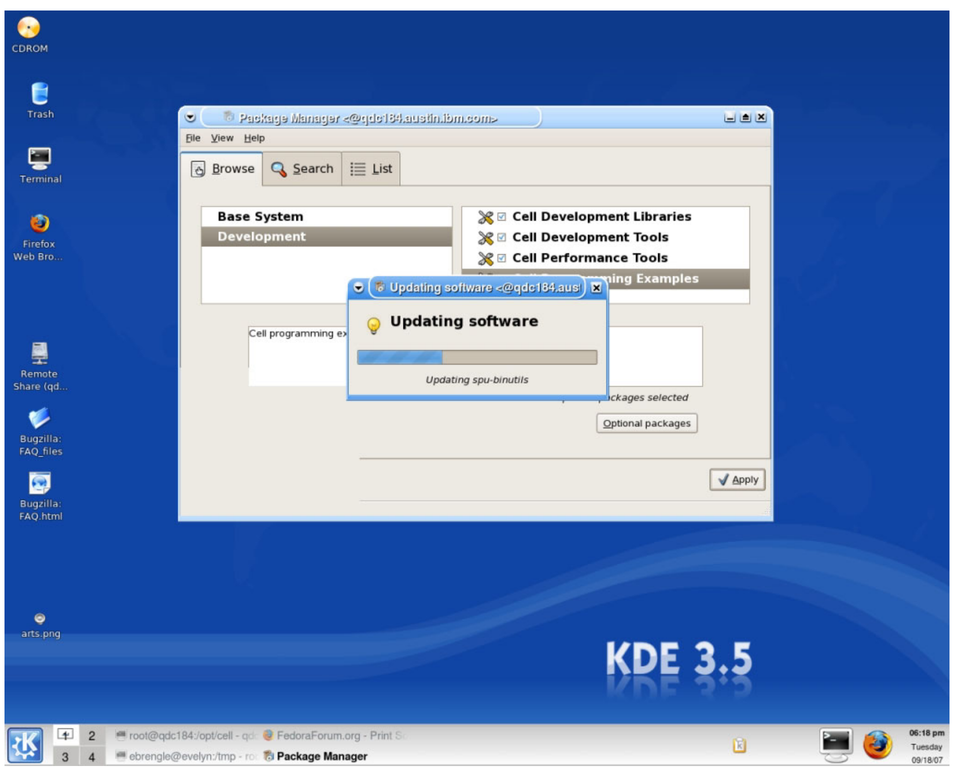
Click the **Continue** button to install the selected SDK packages.

13. Pirut will automatically calculate any required dependencies for the SDK. If dependencies are found, Pirut will display a window similar to:

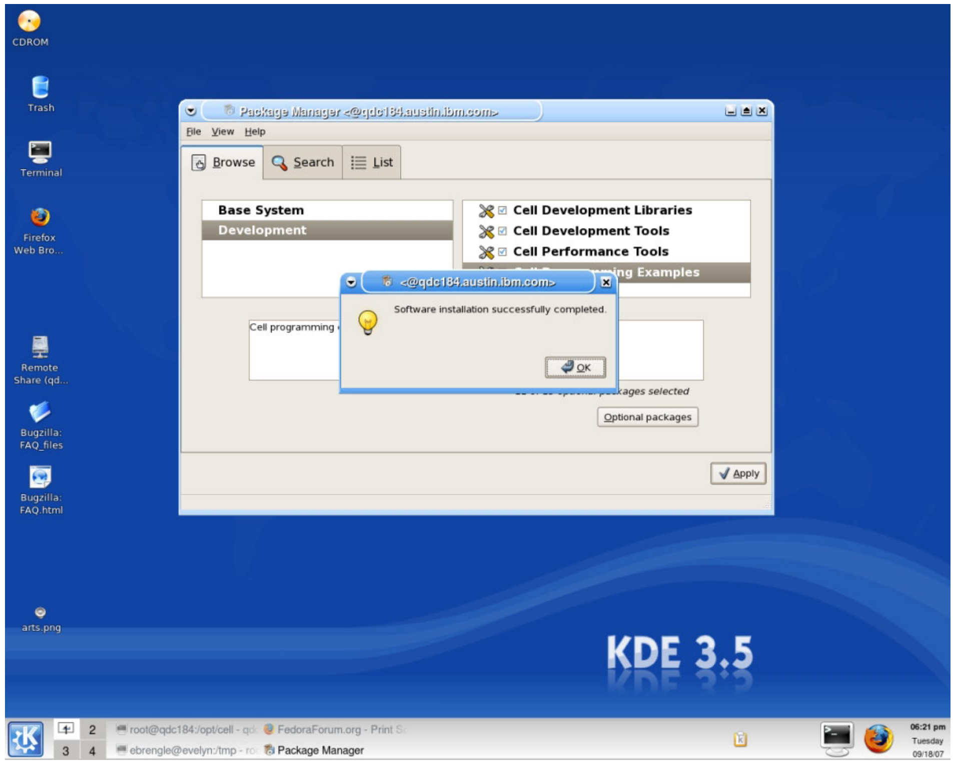


Press the **Details** selector if you want to see individual dependencies. Next, press **Continue**.

14. Pirut displays the *Updating software* window:



15. When the installation finishes, Pirut displays the *Installation success* window:



Click **OK**.

After you have installed the SDK, close Pirut.

The cellsdk script

The cellsdk script is used to install, update or uninstall the SDK. This script is a wrapper with most of the work done by YUM-based tools.

cellsdk script options

When called without options or parameters, the cellsdk script displays an option list. The following is an example:

```
Usage: cellsdk [OPTIONS] [--iso ISO_DIR] TASK
   ISO_DIR is the directory where cellsdk iso's have been downloaded.
      If not specified, network or cdrom install is assumed.
   TASK is one of install, update, uninstall, verify, mount, unmount, \
      removeUpdate

The main tasks are:
   install:      ./cellsdk [--gui] install     (starts pirut or yum)
   update:       ./cellsdk [--gui] update      (starts pup or yum)
   uninstall:    ./cellsdk [--gui] uninstall   (starts pirut or yum)
   verify:       ./cellsdk verify              (lists RPMs installed)
   mount:        ./cellsdk -iso ISO_DIR mount  (mounts cellsdk iso images)
   unmount:      ./cellsdk unmount             (unmounts cellsdk iso images)
   removeUpdate: ./cellsdk removeUpdate        (uninstalls a cellsdk update)

The cellsdk script uses the YUM-based tools according to the following options:
   no flags             starts yum using groupinstall
   -g, --gui            start up pirut (install, uninstall) or pup (update).
                          The gui tools take no arguments, so --gui prohibits
                          using --runtime or --auto.
   -r, --runtime        only uses the YUM group Cell Runtime Environment
```

```
-a, --auto          starts yum using the -y (yes to everything) flag
-V, --version       display version of the SDK
-q, --quiet         no messages
-v, --verbose       verbose message
-vv, --very-verbose  very verbose messages
```

cellsdk script verify

The verify option lists the SDK RPMs installed or available to be installed on your system. Pass the verify option to the cellsdk script to list the SDK RPMs installed on your system. For an example of the output produced by verify, see Appendix C, "cellsdk script SDK verify example," on page 75.

Upgrading from RHEL5 Developer to RHEL5 Product

It is possible to "upgrade" the SDK from the *RHEL5 Developer* to the *RHEL5 Product* version simply by downloading the ISO image for *RHEL5 Product* from Passport Advantage and accepting the license.

To upgrade, type the following command:

```
# ./cellsdk --iso /tmp/cellsdkiso install
```

and accept the IPLA license when it is displayed. The rest of the installation process is similar to before. Because the content of the two ISOs is exactly the same, this second install will not install any additional RPMs unless explicitly selected in the Pirut GUI.

Installing additional SDK components

You can install additional components by either running the cellsdk script again with the `install` task or by using the `yum install` command. ISO images must be mounted (for example after a system reboot) for YUM to find the RPMs. To mount an image, type the following command:

```
# ./cellsdk --iso /tmp/cellsdkiso mount
```

See "RPMs by component" on page 25 for a complete list of SDK RPMs that you can install using YUM.

Uninstalling the SDK

The following topics describe how to uninstall the SDK.

Uninstalling SDK version 2.1

If you previously installed version 2.1 of the SDK from the IBM alphaWorks Web site, save any files you need from the /opt/ibm/cell-sdk directory.

Uninstall the SDK by typing the following commands as the user root:

1. `# /opt/ibm/cell-sdk/prototype/cellsdk uninstall`
2. `# rm -rf /opt/ibm/systemsim-cell`
3. `# rm -rf /opt/ibm/cell-sdk`
4. `# rm -rf /opt/cell`
5. `# rm -rf /opt/ibmcmp`
6. `# umount /mnt/cellsdk`
7. `# rmdir /mnt/cellsdk`

Uninstalling SDK version 3.0

If you installed version 3.0 of the SDK using the instructions in this book, first save any files you need from the /opt/cell directory and the /opt/ibm/systemsim directory.

Uninstall the SDK by following these steps:

1. Uninstall the SDK RPMs using YUM. Use the uninstall option of the cellsdk script. For example type `# ./cellsdk uninstall`.

   For an example, see Appendix E, "cellsdk script SDK uninstallation example," on page 81. Answer **'y'** when asked by YUM to uninstall the packages.

   After YUM has uninstalled all of the SDK RPMs, there are a series of questions about how much cleanup you want to do for other directories used by the SDK. To perform a full uninstall, answer **'y'** to all questions.
2. Clean up the YUM cache.
   a. Type the following commands:
   ```
   # yum clean all
   # rm -rf /var/cache/yum/CellSDK*
   ```
3. Clean up the YUM configuration.
   a. Remove the SDK exclude clause added to the /etc/yum.conf file. See "Preventing automatic updates from overwriting SDK components" on page 38.
4. Uninstall the Eclipse IDE.
   a. Start Eclipse
   b. Click **Help -> Software Updates -> Manage Configuration**
   c. Click **Cell IDE feature**
   d. Click the right mouse button
   e. From the popup menu, click **disable**
   f. Click **uninstall**

Updating the SDK

The SDK can be updated with a new version using the cellsdk script `update` option.

The most likely reason to update the SDK is to apply an IBM Fix Pack or interim fix to the SDK. All fixes are cumulative. Fixes to the SDK, if available, are only for RHEL 5.1 product installations and are supplied on an ISO image with the name *CellSDK-Updates-RHEL*. Note that the version numbering for both the ISO image and the cell-install RPM uses the following 5 digit numbering scheme:

`version.release.product.fixpack.interimfix`

For example `3.0.0.2.0` is fix pack #2 for SDK version 3.0.0.

Download the new cell-install RPM and the CellSDK-Updates-RHEL ISO image from Passport Advantage and save them to a directory on your machine. You can either manually install the new cell-install rpm, or it will be done for you as part of the update process.

Update the SDK by typing the following command as root:

```
# /opt/cell/cellsdk --iso /tmp/cellsdkiso update
```

In this example, the /tmp/cellsdkiso directory contains the downloaded update ISO image and the new cell-install RPM.

You must accept the SDK licenses each time you apply an update.

Only installed RPMs are updated during the update process. For example, if there is an update for the cellide RPM but you do not have cellide installed, the update process will skip cellide and not install it. If you want to install it later, mount the update ISO by typing the following command:

```
# cellsdk --iso /tmp/cellsdkiso mount
```

`/tmp/cellsdkiso` is the directory where the update ISO image is mounted.

At the end of the update process, the cellsdk script will prompt you to save the initial RPMs in case you later want to remove the updates and restore the original versions. If you answer yes, the cellsdk script will attempt to find the RPMs in the original install on an ISO image or through the network, and store them in the /opt/cell/updates directory. If you say no, and later you decide to remove the update, you must mount the original install ISO so that the cellsdk script can locate the replacement RPMs.

After applying a Fix Pack or interim fix, it can be backed out. See "The SDK backout procedure" on page 55.

cellsdk script update

Use the update option to update the SDK.

For example, to update the SDK using an ISO image, type the following command:

```
# /opt/cell/cellsdk --iso /tmp/cellsdkiso update
```

See Appendix D, "cellsdk script SDK update example," on page 77 for a complete example.

Updating the SDK with pup

You can update the SDK using the *pup* package manager. pup is provided by the Pirut RPM.

1. Launch pup by typing the following command as root:

   ```
   # /cellsdk --gui --iso /tmp/cellsdkiso update
   ```

2. pup displays a window that shows available updates:

If you want more detail about the updates, click the *Update Details* selector. Otherwise, press the **Apply updates** button to begin the update.

3. The *Downloading packages* window appears:

4. After all packages have downloaded, pup begins updating software:

5. When the update finishes, pup displays the success window:

After you have updated the SDK, close pup.

The SDK backout procedure

This topic describes how to back out a Fix Pack or interim fix that was applied using the cellsdk update option.

For more information on updating the SDK, see "Updating the SDK" on page 51.

In order to remove an update, the cellsdk script must be able to locate the earlier version of each RPM being removed, including the cell-install RPM itself. If you did not save a copy of the RPMs when you applied the update, use the `--iso` flag to tell cellsdk where the original ISO image is:

```
# cellsdk --iso /tmp/cellsdkiso removeUpdate
```

where `/tmp/cellsdkiso` is the directory where the original ISO image is stored.

If you did save a copy of the original RPMs, you do not need the `--iso` flag:

```
# cellsdk removeUpdate
```

Building the SPU-Isolation component

This topic describes the procedures to perform after installing the SPU Isolation component.

After you have installed the optional SPU-Isolation component, you can finish the installation by building the spu-isolated-app tool and example code using these steps:

1. Run the make command to build and install the *spu-isolated-app* tool. Type the following commands as the user root:

   ```
   # cd /opt/cell/sdk/prototype/usr/src/spu-isolated-app/
   # make
   ```

2. Build the samples by typing the following commands:

   ```
   # cd /opt/cell/sdk/prototype/src/examples/isolation/
   # make
   ```

Configuring the Eclipse IDE

This topic describes how to finish the installation of the Eclipse IDE.

If you have installed the optional Eclipse IDE component, you should finish the installation by following these steps:

1. Install Java™ *1.4* from IBM (http://www.ibm.com/developerworks/java/jdk/linux/download.html) or Sun (http://java.sun.com/javase/downloads/index.jsp).
   a. The default Java VM (GCJ) that comes installed with Fedora 7 is not sufficient.
   b. If you have a PPC64 system, you must install and use a 32bit Java runtime environment, because Eclipse for PPC is compiled for the 32-bit architecture.
   c. Update JAVA_HOME and PATH. For example:

      ```
      JAVA_HOME=/usr/java/j2sdk1.4.2_13/jre
      PATH=$PATH:$JAVA_HOME/bin
      ```

   d. To change your PATH variable, edit the file `.bash_profile` located in your home directory, for example: `/home/user/.bash_profile`.

2. Install Eclipse version 3.2.x. Eclipse can be downloaded from http://www.eclipse.org/. Since version 3.2 is not the latest version, download this version by following the menu prompts: **DOWNLOADS (on top) -> By Project (left) -> Eclipse Platform (center)** .
3. Install the CDT version 3.1 plug-ins. CDT is the Eclipse C/C++ Development Tools project. It is a C/C++ IDE that also serves as a platform for others to provide tools for C/C++ developers. You can download CDT from http://www.eclipse.org/cdt/downloads.php.
4. Install the SDK Eclipse IDE for plug-ins using the update manager:
   a. In Eclipse, click **Help** → **Software Updates** → **Find and Install...**
   b. Click **Search for new features to install**, click **Next**.
   c. Click **New Local Site...**
   d. Go to `/opt/cell/ide`.
   e. Select **com.ibm.celldt.update**, and click **OK**.
   f. Click **Finish** and follow the on-screen instructions.
5. Install the ALF for Cell BE IDE template package by typing the following command:
   ```
   # yum install alf-ide-template
   ```
6. Restart your system to make sure all settings take effect.

For more information about the IDE, see the Eclipse IDE help topic. To access the IDE help, in Eclipse click: **Help** → **Help Contents** → **IDE for Cell Broadband Engine SDK**.

Setting up a YUM server for the SDK

This topic is for advanced users who want to set up a local YUM server. A YUM server allows multiple users to access the SDK files without having to download them from the Barcelona Supercomputing Center Web site or use the ISO images. A YUM server is useful if your company has a firewall that prevents direct access to the Internet.

Follow these steps to set up a local YUM server.
1. Install an HTTP server and preferably enable FTP access to a directory for downloading the RPMs.
2. Create a directory for the SDK files on the server. For example,
   ```
   [root@myserver]# mkdir /var/www/sdk30
   [root@myserver]# cd /var/www/sdk30
   ```
   Create the sdk30 directory below the directory (in this example `/var/www/`) that your web server uses to serve files. In the following instructions, it is assumed that the directory created by the previous step is sdk30. Substitute the actual directory name created by the preceding command in subsequent examples.
3. Copy all the files from the source material, for example the ISO images and the BSC Web site, to the sdk30 directory.
4. Create updated SDK YUM *repo* files which you have edited to point to the internal server by setting the `baseurl` paths. For example, the /etc/yum.repos.d/cellsdk-Fedora.repo file might contain the following:
   ```
   [CellSDK-Devel-Fedora-x86]
   baseurl=ftp://myserver.com/sdk30 \
      file:///opt/cell/yum-repos/CellSDK-Devel-Fedora/x86
   ```

```
[CellSDK-Open-Fedora-x86]
baseurl=ftp://myserver.com/sdk30 \
   file:///opt/cell/yum-repos/CellSDK-Open-Fedora/x86
```

Note: Different protocols can be used to retrieve the files from the server including FTP, HTTP or a local file directory on your own system.

5. Decide how to distribute these new repo files to your users. A simple option is to instruct them to install the cell-install RPM and then overwrite the repo files in the /etc/yum.repos.d directory with the new versions.

You can keep a local copy of the RPMs on your system and use the localinstall or localupdate YUM options. The advantage of this approach is that YUM manages the dependencies and uses the configured repositories to resolve dependencies. The following is an example using the localinstall option:

```
# yum localinstall /tmp/sdk30/spu-gcc-fortran-4.1.1-*.i686.rpm
```

Troubleshooting the SDK installation

This topic describes what to do if things go wrong when using the cellsdk script.

YUM continues to install the SDK packages even if some of the RPMs were not completely downloaded from the BSC Web site. The failure messages from YUM do not clearly state this failure. To verify if all files were installed correctly, type the command:

```
# ./cellsdk verify
```

In the list output by this command, verify that all default RPMs were installed. If they were not, retype the `./cellsdk install` command. YUM will attempt to download any required RPMs that were not downloaded during a past attempt to install the SDK. YUM resumes the download process from the previous failure point.

Sometimes YUM operates incorrectly. It writes files in /var/cache/yum, and sometimes these no longer reflect the correct state of the command. If this happens, type the command:

```
# yum clean metadata
```

This will remove the incorrect status files.

There are other options to YUM that are useful to use if things go wrong. If the previous command did not restore correct operation of YUM, try typing the command:

```
# yum clean all
```

This will remove additional state files, and might cause the YUM installation process to succeed on the next invocation.

If the preceding commands do not restore correct operation of the YUM installation process, manually remove the cached state files. To do this, type the following command:

```
# rm -rf /var/cache/yum/CellSDK
```

To see more information about YUM, set the options:

```
debuglevel=10
errorlevel=10
```

in /etc/yum.conf. You an get an equivalent result by adding the string **-d 10 -e 10** to any YUM command.

You can specify the flags **-v** for verbose output, or **-vv** for very verbose output when typing the cellsdk script. YUM writes a log to /var/log/yum.log. The cellsdk install script writes a log to /var/log/cellsdk.log. Looking at these files might provide helpful information about what went wrong.

As the last step of the install, the cellsdk script locates the SDK version of some open source RPMs. The RPMs may include some, none or all of the *blas*, *blas-debuginfo*, *blas-devel*, *numactl*, *numactl-devel*, *oprofile*, and *oprofile-debuginfo* packages. It places the RPMs it finds in the /tmp/cellsdk/openSrc directory. Then cellsdk removes the installed version of each RPM, and installs the SDK version of the RPM in its place. Finally, it creates a *done* file in the /tmp/cellsdk/openSrc directory that prevents this install step from repeating if cellsdk is run again. If you want to reinstall the SDK version of the RPMs at a later time, you can remove the done file and run the `cellsdk install` command again. See "Preventing automatic updates from overwriting SDK components" on page 38 for more information.

A potential problem is that the Fedora 7 version of *blas* is installed instead of the SDK version because YUM sees that it has a higher version number. To fix this problem, delete the /tmp/cellsdk/openSrc/done file and reinstall *blas* as described above.

Chapter 6. DaCS for Hybrid-x86 configuration

The following topics describe the configuration information you need to use DaCS for Hybrid-x86.

- "Daemon configuration"
- "Topology configuration"
- "Affinity requirements" on page 61

Note: ALF for Hybrid-x86 depends on DaCS for Hybrid-x86, therefore this configuration information is also needed for Hybrid ALF.

Daemon configuration

This topic describes DaCS daemon configuration.

The host daemon service is named *hdacsd* and the accelerator daemon service is named *adacsd*. Both daemons are configured by editing the /etc/dacsd.conf file on the respective system.

Default versions of these files are installed by the daemon RPMs. These default files contain comments about the supported parameters and values. Back up your configuration files before making changes.

Changes will not take effect until the daemon is restarted. Start and stop the daemon using the service command in the/sbin directory. To stop the host daemon, type the following command as root:

```
# /sbin/service hdacsd stop
```

To start the host daemon, type:

```
# /sbin/service hdacsd start
```

To stop the accelerator daemon, type:

```
# /sbin/service adacsd stop
```

To start the accelerator daemon, type:

```
# /sbin/service adacsd start
```

See the service man page for more details about controlling daemons.

Topology configuration

This topic describes DaCS topology configuration.

The topology configuration file /etc/dacs_topology.config is only used by the host daemon service. Back up this file before changing it. Changes will not take effect until the daemon is restarted.

The host DaCS daemon might stop if there is a configuration error in the dacs_topology.config file. Check the log file specified by the dacsd.conf file (default is /var/log/hdacsd.log) for configuration errors.

The topology configuration file identifies the hosts and accelerators and their relationship to one another. The host can contain more than one CPU core, for example a Ridgeback contains four cores. The host can be attached to one or more accelerators, for example to a Cell BE BladeCenter. The topology configuration file allows you to specify a number of configurations for this hardware. For example, it can be configured such that each core is assigned one Cell Broadband Engine or it might be configured so that each core can reserve any (or all) of the Cell Broadband Engines.

The default topology configuration file is for a host that has four cores and is attached to a single Cell BE BladeCenter:

```
<DaCS_Topology version="1.0">
  <hardware>
    <de tag="OB1" type="DACS_DE_SYSTEMX" ip="192.168.1.100">
      <de tag="OC1" type="DACS_DE_SYSTEMX_CORE"></de>
      <de tag="OC2" type="DACS_DE_SYSTEMX_CORE"></de>
      <de tag="OC3" type="DACS_DE_SYSTEMX_CORE"></de>
      <de tag="OC4" type="DACS_DE_SYSTEMX_CORE"></de>
    </de>
    <de tag="CB1" type="DACS_DE_CELLBLADE" ip="192.168.1.101">
      <de tag="CBE11" type="DACS_DE_CBE"></de>
      <de tag="CBE12" type="DACS_DE_CBE"></de>
    </de>
  </hardware>
  <topology>
    <canreserve he="OC1" ae="CB1"/>
    <canreserve he="OC2" ae="CB1"/>
    <canreserve he="OC3" ae="CB1"/>
    <canreserve he="OC4" ae="CB1"/>
  </topology>
</DaCS_Topology>
```

The `<hardware>` section identifies the host system with its four cores (OC1-OC4) and the Cell BE BladeCenter (CB1) with its two Cell Broadband Engines (CBE11 and CBE12).

The `<topology>` section identifies what each core (host) can use as an accelerator. In this example, each core can reserve and use either the entire Cell BE BladeCenter (CB1) or one or more of the Cell Broadband Engines on the BladeCenter. The ability to use the Cell BE is implicit in the `<canreserve>` element. This element has an attribute **only** which defaults to false. When it is set to true, only the Cell BE BladeCenter can be reserved. If the fourth `<canreserve>` element was changed to `<canreserve he="OC4" ae="CB1" only="TRUE"></canreserve>`, then OC4 can only reserve the Cell BE BladeCenter. The usage can be made more restrictive by being more specific in the `<canreserve>` element. If the fourth `<canreserve>` element is changed to `<canreserve he="OC4" ae="CBE12"></canreserve>`, then OC4 can only reserve CBE12 and can not reserve the Cell BE BladeCenter.

Modify the topology configuration file to match your hardware configuration. Make a copy of the configuration file before changing it. At a minimum, you must update the IP addresses of the **ip** attributes to match the interfaces between the host and the accelerator. You might need to add additional entries if you have a second BladeCenter. The following is an example of the topology configuration file changed to have valid IP addresses and to add a second BladeCenter:

```
<DaCS_Topology version="1.0">
  <hardware>
    <de tag="OB1" type="DACS_DE_SYSTEMX" ip="192.168.1.100">
      <de tag="OC1" type="DACS_DE_SYSTEMX_CORE"></de>
      <de tag="OC2" type="DACS_DE_SYSTEMX_CORE"></de>
      <de tag="OC3" type="DACS_DE_SYSTEMX_CORE"></de>
```

```
      <de tag="OC4" type="DACS_DE_SYSTEMX_CORE"></de>
    </de>
    <de tag="CB1" type="DACS_DE_CELLBLADE" ip="192.168.1.101">
      <de tag="CBE11" type="DACS_DE_CBE"></de>
      <de tag="CBE12" type="DACS_DE_CBE"></de>
    </de>
    <de tag="CB2" type="DACS_DE_CELLBLADE" ip="192.168.1.102">
      <de tag="CBE21" type="DACS_DE_CBE"></de>
      <de tag="CBE22" type="DACS_DE_CBE"></de>
    </de>
  </hardware>
  <topology>
    <canreserve he="OC1" ae="CB1"></canreserve>
    <canreserve he="OC1" ae="CB2"></canreserve>
    <canreserve he="OC2" ae="CB1"></canreserve>
    <canreserve he="OC2" ae="CB2"></canreserve>
    <canreserve he="OC3" ae="CB1"></canreserve>
    <canreserve he="OC3" ae="CB2"></canreserve>
    <canreserve he="OC4" ae="CB1"></canreserve>
    <canreserve he="OC4" ae="CB2"></canreserve>
  </topology>
</DaCS_Topology>
```

Affinity requirements

This topic describes the affinity requirements for DaCS.

A DaCS for Hybrid-x86 application on the host (X86_64) must have processor affinity to start. This can be done

- on the command line.
- in mpirun.
- through the sched_setaffinity function.

The following is a command line example to set affinity of the shell to the first processor (numbering starts at 0):

```
# taskset -pc 0 &&
```

The following example launches a DaCS application using `taskset`:

```
# taskset -c 0 HelloDaCSApp Mike
```

The application program name is HelloDaCSApp which is passed an argument of Mike.

The man page for `taskset` states that a user must have *CAP_SYS_NICE* permission to change CPU affinity.

Chapter 7. Getting support

The SDK is supported through the CBEA architecture forum on the developerWorks Web site at http://www.ibm.com/developerworks/power/cell/.

Commercial support from IBM is available if you purchased the SDK from Passport Advantage.

The XL C/C++ compilers are supported through the XL compiler Web site. See http://www.ibm.com/software/awdtools/xlcpp/support/.

The XL Fortran compiler is supported through the XL compiler Web site. See http://www.ibm.com/software/awdtools/fortran/support/.

This version of the SDK supersedes all versions of the SDK that were available from alphaWorks.

If you have a problem on the IBM BladeCenter QS21 that you think is caused by running the Barcelona Supercomputing Center kernel on Fedora 7, report a bug to the public `cbe-oss-dev@ozlabs.org` mailing list. Archives and subscription information for this list are available from https://ozlabs.org/mailman/listinfo/cbe-oss-dev/. Since Fedora 7 is not a supported IBM product, IBM provides no guaranteed reply or target dates for fixes for this configuration. Commercial support is available for Red Hat Enterprise Linux (RHEL) 5.1.

Appendix A. 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.

Appendix B. cellsdk script SDK installation example

This is an example of using the cellsdk script to install the SDK on a BladeCenter QS21 running Red Hat Enterprise Linux (RHEL) 5.1.

The following output is the result of typing the commands:

```
cd /opt/cell
./cellsdk --iso /tmp/cellsdkiso install
```

Here is the output:

```
cellsdk logs to /var/log/cellsdk.log

The IBM SDK 3.0 for Multicore Acceleration contains components governed by
the following Open Source licenses:

GNU Public License (GPL) - see http://www.gnu.org/licenses/gpl.html.
  GPL licensed components include GCC Toolchain and numactl.

Installation and use of this software requires you to certify you have read the
license above, and accept its terms.  To accept these terms, type 'yes' at
the prompt below.  If you do not wish to accept, type 'no' or press 'ctrl-C' to
abort this program without installing.

After typing 'yes', you will be presented with licenses for IBM owned code,
Follow the instructions on the screen to accept the IBM Licenses and proceed with
installation of the SDK.

I have read and accept the licenses above [no/yes]: cellsdk INFO-2028: license
 accepted
cellsdk INFO-2023: Trying to install/update cell-product-license
cellsdk INFO-2023: Trying to install/update cell-devel-license
cellsdk INFO-2024: cell-devel-license is installed
International License Agreement for Non-Warranted Programs

Part 1 - General Terms

BY DOWNLOADING, INSTALLING, COPYING, ACCESSING, OR USING THE PROGRAM YOU AGREE
TO THE TERMS OF THIS AGREEMENT. IF YOU ARE ACCEPTING THESE TERMS ON BEHALF OF
ANOTHER PERSON OR A COMPANY OR OTHER LEGAL ENTITY, YOU REPRESENT AND WARRANT
THAT YOU HAVE FULL AUTHORITY TO BIND THAT PERSON, COMPANY, OR LEGAL ENTITY TO
THESE TERMS. IF YOU DO NOT AGREE TO THESE TERMS,

- DO NOT DOWNLOAD, INSTALL, COPY, ACCESS, OR USE THE PROGRAM; AND

- PROMPTLY RETURN THE PROGRAM AND PROOF OF ENTITLEMENT TO THE PARTY FROM WHOM
YOU ACQUIRED IT TO OBTAIN A REFUND OF THE AMOUNT YOU PAID. IF YOU DOWNLOADED
THE PROGRAM, CONTACT THE PARTY FROM WHOM YOU ACQUIRED IT.

"IBM" is International Business Machines Corporation or one of its


Press Enter to continue viewing the license agreement, or, Enter "1" to accept
the agreement, "2" to decline it or "99" to go back to the previous screen,
"3" Print, "4" Read non-IBM terms.
cellsdk INFO-2014: License accepted.
cellsdk INFO-2023: Trying to install/update cell-extras-RHEL-license
cellsdk INFO-2019: yum groupinstall  'Cell Runtime Environment'
 'Cell Development Tools' 'Cell Development Libraries' 'Cell Programming Examples'
 'Cell Simulator' 'Cell Performance Tools'
Loading "installonlyn" plugin
Loading "security" plugin
```

```
Loading "rhnplugin" plugin
Setting up Group Process
Setting up repositories
Setting up repositories
Reading repository metadata in from local files
Package cell-devel-license - 3.0.0-1.0.noarch already installed and latest version
Warning: Group Cell Simulator does not exist.
Resolving Dependencies
--> Populating transaction set with selected packages. Please wait.
---> Downloading header for libspe2man to pack into transaction set.
---> Package libspe2man.noarch 0:2.2.0-5 set to be updated
---> Downloading header for cell-tutorial-source to pack into transaction set.
---> Package cell-tutorial-source.noarch 0:3.0-6 set to be updated
---> Downloading header for ppu-mass-devel to pack into transaction set.
---> Package ppu-mass-devel.ppc 0:4.5.0-10 set to be updated
---> Downloading header for spu-gcc-c++ to pack into transaction set.
---> Package spu-gcc-c++.ppc 0:4.1.1-107 set to be updated
---> Downloading header for dacsman to pack into transaction set.
---> Package dacsman.noarch 0:3.0-6 set to be updated
---> Downloading header for ppu-binutils to pack into transaction set.
---> Package ppu-binutils.ppc 0:2.17.50-32 set to be updated
---> Downloading header for blas-devel to pack into transaction set.
---> Package blas-devel.ppc64 0:3.0-37.el5 set to be updated
---> Downloading header for cell-libs to pack into transaction set.
---> Package cell-libs.ppc64 0:3.0-16 set to be updated
---> Downloading header for blas-devel to pack into transaction set.
---> Package blas-devel.ppc 0:3.0-37.el5 set to be updated
---> Downloading header for cell-demos to pack into transaction set.
---> Package cell-demos.ppc64 0:3.0-10 set to be updated
---> Downloading header for ppu-gcc to pack into transaction set.
---> Package ppu-gcc.ppc 0:4.1.1-57 set to be updated
---> Downloading header for dacs to pack into transaction set.
---> Package dacs.ppc64 0:3.0.0-19 set to be updated
---> Downloading header for spu-mass-devel to pack into transaction set.
---> Package spu-mass-devel.ppc 0:4.5.0-10 set to be updated
---> Downloading header for cell-examples-source to pack into transaction set.
---> Package cell-examples-source.noarch 0:3.0-11 set to be updated
---> Downloading header for alfman to pack into transaction set.
---> Package alfman.noarch 0:3.0-8 set to be updated
---> Downloading header for ppu-simdmath to pack into transaction set.
---> Package ppu-simdmath.ppc 0:3.0-5 set to be updated
---> Downloading header for alf-examples-source to pack into transaction set.
---> Package alf-examples-source.noarch 0:3.0.0-7 set to be updated
---> Downloading header for alf to pack into transaction set.
---> Package alf.ppc 0:3.0.0-9 set to be updated
---> Downloading header for cell-libs-devel to pack into transaction set.
---> Package cell-libs-devel.ppc 0:3.0-16 set to be updated
---> Downloading header for alf to pack into transaction set.
---> Package alf.ppc64 0:3.0.0-9 set to be updated
---> Downloading header for alf-devel to pack into transaction set.
---> Package alf-devel.ppc64 0:3.0.0-9 set to be updated
---> Downloading header for alf-devel to pack into transaction set.
---> Package alf-devel.ppc 0:3.0.0-9 set to be updated
---> Downloading header for simdman to pack into transaction set.
---> Package simdman.noarch 0:3.0-6 set to be updated
---> Downloading header for ppu-simdmath-devel to pack into transaction set.
---> Package ppu-simdmath-devel.ppc64 0:3.0-5 set to be updated
---> Downloading header for ppu-gdb to pack into transaction set.
---> Package ppu-gdb.ppc 0:6.6.50-28 set to be updated
---> Downloading header for cell-examples to pack into transaction set.
---> Package cell-examples.ppc64 0:3.0-11 set to be updated
---> Downloading header for cell-libs-source to pack into transaction set.
---> Package cell-libs-source.noarch 0:3.0-16 set to be updated
---> Downloading header for dacs-devel to pack into transaction set.
---> Package dacs-devel.ppc64 0:3.0.0-19 set to be updated
---> Downloading header for spu-gdb to pack into transaction set.
---> Package spu-gdb.ppc 0:6.6.50-12 set to be updated
```

```
---> Downloading header for spu-newlib to pack into transaction set.
---> Package spu-newlib.ppc 0:1.15.0-81 set to be updated
---> Downloading header for ppu-gcc-c++ to pack into transaction set.
---> Package ppu-gcc-c++.ppc 0:4.1.1-57 set to be updated
---> Downloading header for cell-libs-devel to pack into transaction set.
---> Package cell-libs-devel.ppc64 0:3.0-16 set to be updated
---> Downloading header for spu-binutils to pack into transaction set.
---> Package spu-binutils.ppc 0:2.17.50-33 set to be updated
---> Downloading header for blas to pack into transaction set.
---> Package blas.ppc 0:3.0-37.el5 set to be updated
---> Downloading header for spu-simdmath-devel to pack into transaction set.
---> Package spu-simdmath-devel.ppc 0:3.0-5 set to be updated
---> Downloading header for cell-documentation to pack into transaction set.
---> Package cell-documentation.noarch 0:3.0-5 set to be updated
---> Downloading header for ppu-simdmath-devel to pack into transaction set.
---> Package ppu-simdmath-devel.ppc 0:3.0-5 set to be updated
---> Downloading header for cell-tutorial to pack into transaction set.
---> Package cell-tutorial.ppc 0:3.0-6 set to be updated
---> Downloading header for cell-demos-source to pack into transaction set.
---> Package cell-demos-source.noarch 0:3.0-10 set to be updated
---> Downloading header for cell-buildutils to pack into transaction set.
---> Package cell-buildutils.noarch 0:3.0-11 set to be updated
---> Downloading header for cell-libs to pack into transaction set.
---> Package cell-libs.ppc 0:3.0-16 set to be updated
---> Downloading header for blas to pack into transaction set.
---> Package blas.ppc64 0:3.0-37.el5 set to be updated
---> Downloading header for ppu-simdmath to pack into transaction set.
---> Package ppu-simdmath.ppc64 0:3.0-5 set to be updated
---> Downloading header for ppu-mass-devel to pack into transaction set.
---> Package ppu-mass-devel.ppc64 0:4.5.0-10 set to be updated
---> Downloading header for spu-gcc to pack into transaction set.
---> Package spu-gcc.ppc 0:4.1.1-107 set to be updated
--> Running transaction check
--> Processing Dependency: numactl for package: cell-demos
--> Processing Dependency: libnuma.so.1 for package: cell-demos
--> Restarting Dependency Resolution with new changes.
--> Populating transaction set with selected packages. Please wait.
---> Downloading header for numactl to pack into transaction set.
---> Package numactl.ppc 0:0.9.10-1 set to be updated
---> Downloading header for numactl to pack into transaction set.
---> Package numactl.ppc64 0:0.9.10-1 set to be updated
--> Running transaction check

Dependencies Resolved

=============================================================================
 Package                 Arch       Version          Repository        Size
=============================================================================
Installing:
 alf                     ppc        3.0.0-9          CellSDK-Devel-RHEL-cbea   33 k
 alf                     ppc64      3.0.0-9          CellSDK-Devel-RHEL-cbea   38 k
 alf-devel               ppc64      3.0.0-9          CellSDK-Devel-RHEL-cbea   39 k
 alf-devel               ppc        3.0.0-9          CellSDK-Devel-RHEL-cbea   93 k
 alf-examples-source     noarch     3.0.0-7          CellSDK-Devel-RHEL-cbea   91 k
 alfman                  noarch     3.0-8            CellSDK-Devel-RHEL-cbea   49 k
 blas                    ppc        3.0-37.el5       rhel-ppc-server-5  307 k
 blas                    ppc64      3.0-37.el5       rhel-ppc-server-5  318 k
 blas-devel              ppc64      3.0-37.el5       rhel-ppc-server-5  173 k
 blas-devel              ppc        3.0-37.el5       rhel-ppc-server-5  164 k
 cell-buildutils         noarch     3.0-11           CellSDK-Devel-RHEL-cbea   19 k
 cell-demos              ppc64      3.0-10           CellSDK-Devel-RHEL-cbea   12 M
 cell-demos-source       noarch     3.0-10           CellSDK-Devel-RHEL-cbea   12 M
 cell-documentation      noarch     3.0-5            CellSDK-Devel-RHEL-cbea   40 M
 cell-examples           ppc64      3.0-11           CellSDK-Devel-RHEL-cbea  354 k
 cell-examples-source    noarch     3.0-11           CellSDK-Devel-RHEL-cbea  339 k
 cell-libs               ppc64      3.0-16           CellSDK-Devel-RHEL-cbea  8.3 k
 cell-libs               ppc        3.0-16           CellSDK-Devel-RHEL-cbea   83 k
```

```
 cell-libs-devel        ppc       3.0-16          CellSDK-Devel-RHEL-cbea  691 k
 cell-libs-devel        ppc64     3.0-16          CellSDK-Devel-RHEL-cbea   97 k
 cell-libs-source       noarch    3.0-16          CellSDK-Devel-RHEL-cbea  234 k
 cell-tutorial          ppc       3.0-6           CellSDK-Devel-RHEL-cbea   47 k
 cell-tutorial-source   noarch    3.0-6           CellSDK-Devel-RHEL-cbea   16 k
 dacs                   ppc64     3.0.0-19        CellSDK-Devel-RHEL-cbea   24 k
 dacs-devel             ppc64     3.0.0-19        CellSDK-Devel-RHEL-cbea  131 k
 dacsman                noarch    3.0-6           CellSDK-Devel-RHEL-cbea   62 k
 libspe2man             noarch    2.2.0-5         CellSDK-Devel-RHEL-cbea   52 k
 ppu-binutils           ppc       2.17.50-32      CellSDK-Open-RHEL-cbea  8.2 M
 ppu-gcc                ppc       4.1.1-57        CellSDK-Open-RHEL-cbea  4.3 M
 ppu-gcc-c++            ppc       4.1.1-57        CellSDK-Open-RHEL-cbea   18 M
 ppu-gdb                ppc       6.6.50-28       CellSDK-Open-RHEL-cbea  3.1 M
 ppu-mass-devel         ppc       4.5.0-10        CellSDK-Devel-RHEL-cbea  250 k
 ppu-mass-devel         ppc64     4.5.0-10        CellSDK-Devel-RHEL-cbea  258 k
 ppu-simdmath           ppc       3.0-5           CellSDK-Devel-RHEL-cbea   26 k
 ppu-simdmath           ppc64     3.0-5           CellSDK-Devel-RHEL-cbea   31 k
 ppu-simdmath-devel     ppc64     3.0-5           CellSDK-Devel-RHEL-cbea   99 k
 ppu-simdmath-devel     ppc       3.0-5           CellSDK-Devel-RHEL-cbea   94 k
 simdman                noarch    3.0-6           CellSDK-Devel-RHEL-cbea  158 k
 spu-binutils           ppc       2.17.50-33      CellSDK-Open-RHEL-cbea  3.6 M
 spu-gcc                ppc       4.1.1-107       CellSDK-Open-RHEL-cbea  2.7 M
 spu-gcc-c++            ppc       4.1.1-107       CellSDK-Open-RHEL-cbea   13 M
 spu-gdb                ppc       6.6.50-12       CellSDK-Open-RHEL-cbea  2.3 M
 spu-mass-devel         ppc       4.5.0-10        CellSDK-Devel-RHEL-cbea   98 k
 spu-newlib             ppc       1.15.0-81       CellSDK-Open-RHEL-cbea  1.3 M
 spu-simdmath-devel     ppc       3.0-5           CellSDK-Devel-RHEL-cbea  178 k
Installing for dependencies:
 numactl                ppc       0.9.10-1        CellSDK-Open-RHEL-cbea   82 k
 numactl                ppc64     0.9.10-1        CellSDK-Open-RHEL-cbea   70 k

Transaction Summary
=============================================================================
Install     47 Package(s)
Update       0 Package(s)
Remove       0 Package(s)

Total download size: 126 M
Is this ok [y/N]: y
Downloading Packages:
Running Transaction Test
Finished Transaction Test
Transaction Test Succeeded
Running Transaction

  Installing: blas                         ####################### [ 1/47]
  Installing: numactl                      ####################### [ 2/47]
  Installing: blas                         ####################### [ 3/47]
  Installing: spu-binutils                 ####################### [ 4/47]
  Installing: numactl                      ####################### [ 5/47]
  Installing: ppu-gdb                      ####################### [ 6/47]
  Installing: cell-tutorial                ####################### [ 7/47]
  Installing: spu-gdb                      ####################### [ 8/47]
  Installing: ppu-binutils                 ####################### [ 9/47]
  Installing: ppu-simdmath                 ####################### [10/47]
  Installing: cell-buildutils              ####################### [11/47]
  Installing: cell-libs                    ####################### [12/47]
  Installing: alf                          ####################### [13/47]
  Installing: cell-libs                    ####################### [14/47]
  Installing: spu-gcc                      ####################### [15/47]
  Installing: dacs                         ####################### [16/47]
  Installing: ppu-gcc                      ####################### [17/47]
  Installing: libspe2man                   ####################### [18/47]
  Installing: cell-tutorial-source         ####################### [19/47]
  Installing: ppu-mass-devel               ####################### [20/47]
  Installing: spu-gcc-c++                  ####################### [21/47]
  Installing: dacsman                      ####################### [22/47]
```

```
  Installing: blas-devel                   ####################### [23/47]
  Installing: blas-devel                   ####################### [24/47]
  Installing: cell-demos                   ####################### [25/47]
  Installing: spu-mass-devel               ####################### [26/47]
  Installing: cell-examples-source         ####################### [27/47]
  Installing: alfman                       ####################### [28/47]
  Installing: ppu-simdmath                 ####################### [29/47]
  Installing: alf-examples-source          ####################### [30/47]
  Installing: alf                          ####################### [31/47]
  Installing: cell-libs-devel              ####################### [32/47]
  Installing: alf-devel                    ####################### [33/47]
  Installing: alf-devel                    ####################### [34/47]
  Installing: simdman                      ####################### [35/47]
  Installing: ppu-simdmath-devel           ####################### [36/47]
  Installing: cell-examples                ####################### [37/47]
  Installing: cell-libs-source             ####################### [38/47]
  Installing: dacs-devel                   ####################### [39/47]
  Installing: spu-newlib                   ####################### [40/47]
  Installing: ppu-gcc-c++                  ####################### [41/47]
  Installing: cell-libs-devel              ####################### [42/47]
  Installing: spu-simdmath-devel           ####################### [43/47]
  Installing: cell-documentation           ####################### [44/47]
  Installing: ppu-simdmath-devel           ####################### [45/47]
  Installing: cell-demos-source            ####################### [46/47]
  Installing: ppu-mass-devel               ####################### [47/47]

Installed: alf.ppc 0:3.0.0-9 alf.ppc64 0:3.0.0-9 alf-devel.ppc64 0:3.0.0-9
 alf-devel.ppc 0:3.0.0-9 alf-examples-source.noarch 0:3.0.0-7
 alfman.noarch 0:3.0-8 blas.ppc 0:3.0-37.el5 blas.ppc64 0:3.0-37.el5
 blas-devel.ppc64 0:3.0-37.el5 blas-devel.ppc 0:3.0-37.el5
 cell-buildutils.noarch 0:3.0-11 cell-demos.ppc64 0:3.0-10
 cell-demos-source.noarch 0:3.0-10 cell-documentation.noarch 0:3.0-5
 cell-examples.ppc64 0:3.0-11 cell-examples-source.noarch 0:3.0-11
 cell-libs.ppc64 0:3.0-16 cell-libs.ppc 0:3.0-16 cell-libs-devel.ppc 0:3.0-16
 cell-libs-devel.ppc64 0:3.0-16 cell-libs-source.noarch 0:3.0-16
 cell-tutorial.ppc 0:3.0-6 cell-tutorial-source.noarch 0:3.0-6
 dacs.ppc64 0:3.0.0-19 dacs-devel.ppc64 0:3.0.0-19 dacsman.noarch 0:3.0-6
 libspe2man.noarch 0:2.2.0-5 ppu-binutils.ppc 0:2.17.50-32
 ppu-gcc.ppc 0:4.1.1-57 ppu-gcc-c++.ppc 0:4.1.1-57 ppu-gdb.ppc 0:6.6.50-28
 ppu-mass-devel.ppc 0:4.5.0-10 ppu-mass-devel.ppc64 0:4.5.0-10
 ppu-simdmath.ppc 0:3.0-5 ppu-simdmath.ppc64 0:3.0-5
 ppu-simdmath-devel.ppc64 0:3.0-5 ppu-simdmath-devel.ppc 0:3.0-5
 simdman.noarch 0:3.0-6 spu-binutils.ppc 0:2.17.50-33 spu-gcc.ppc 0:4.1.1-107
 spu-gcc-c++.ppc 0:4.1.1-107 spu-gdb.ppc 0:6.6.50-12
 spu-mass-devel.ppc 0:4.5.0-10 spu-newlib.ppc 0:1.15.0-81
 spu-simdmath-devel.ppc 0:3.0-5
Dependency Installed: numactl.ppc 0:0.9.10-1 numactl.ppc64 0:0.9.10-1
Complete!
cellsdk INFO-2026:     blas-3.0-6.ppc.rpm
cellsdk INFO-2026:     blas-3.0-6.ppc64.rpm
cellsdk INFO-2026:     blas-devel-3.0-6.ppc.rpm
cellsdk INFO-2026:     blas-devel-3.0-6.ppc64.rpm
cellsdk INFO-2026:     numactl-0.9.10-1.ppc.rpm
cellsdk INFO-2026:     numactl-0.9.10-1.ppc64.rpm
cellsdk INFO-2026:     numactl-devel-0.9.10-1.ppc.rpm
cellsdk INFO-2026:     numactl-devel-0.9.10-1.ppc64.rpm
cellsdk INFO-2027: Installing cellsdk versions of openSource rpms
Loading "installonlyn" plugin
Loading "security" plugin
Loading "rhnplugin" plugin
Setting up Local Package Process
Examining /tmp/cellsdk/openSrc/blas-devel-3.0-6.ppc.rpm: blas-devel - 3.0-6.ppc
Examining /tmp/cellsdk/openSrc/blas-devel-3.0-6.ppc64.rpm: blas-devel - 3.0-6.ppc64
Examining /tmp/cellsdk/openSrc/numactl-devel-0.9.10-1.ppc64.rpm: numactl-devel -
 0.9.10-1.ppc64
Examining /tmp/cellsdk/openSrc/numactl-0.9.10-1.ppc.rpm: numactl - 0.9.10-1.ppc
Examining /tmp/cellsdk/openSrc/numactl-devel-0.9.10-1.ppc.rpm: numactl-devel -
```

```
 0.9.10-1.ppc
Examining /tmp/cellsdk/openSrc/numactl-0.9.10-1.ppc64.rpm: numactl -
 0.9.10-1.ppc64
Examining /tmp/cellsdk/openSrc/blas-3.0-6.ppc64.rpm: blas - 3.0-6.ppc64
Examining /tmp/cellsdk/openSrc/blas-3.0-6.ppc.rpm: blas - 3.0-6.ppc
Marking /tmp/cellsdk/openSrc/blas-devel-3.0-6.ppc.rpm to be installed
Setting up repositories
Reading repository metadata in from local files
Marking /tmp/cellsdk/openSrc/blas-devel-3.0-6.ppc64.rpm to be installed
Marking /tmp/cellsdk/openSrc/numactl-devel-0.9.10-1.ppc64.rpm to be installed
Marking /tmp/cellsdk/openSrc/numactl-0.9.10-1.ppc.rpm to be installed
Marking /tmp/cellsdk/openSrc/numactl-devel-0.9.10-1.ppc.rpm to be installed
Marking /tmp/cellsdk/openSrc/numactl-0.9.10-1.ppc64.rpm to be installed
Marking /tmp/cellsdk/openSrc/blas-3.0-6.ppc64.rpm to be installed
Marking /tmp/cellsdk/openSrc/blas-3.0-6.ppc.rpm to be installed
Resolving Dependencies
--> Populating transaction set with selected packages. Please wait.
---> Package numactl-devel.ppc64 0:0.9.10-1 set to be updated
---> Package blas-devel.ppc64 0:3.0-6 set to be updated
---> Package blas.ppc 0:3.0-6 set to be updated
---> Package blas-devel.ppc 0:3.0-6 set to be updated
---> Package numactl.ppc64 0:0.9.10-1 set to be updated
---> Package numactl-devel.ppc 0:0.9.10-1 set to be updated
---> Package blas.ppc64 0:3.0-6 set to be updated
---> Package numactl.ppc 0:0.9.10-1 set to be updated
--> Running transaction check

Dependencies Resolved

=============================================================================
 Package                Arch       Version          Repository        Size
=============================================================================
Installing:
 blas                   ppc        3.0-6
         /tmp/cellsdk/openSrc/blas-3.0-6.ppc.rpm  1.3 M
 blas                   ppc64      3.0-6
         /tmp/cellsdk/openSrc/blas-3.0-6.ppc64.rpm  1.4 M
 blas-devel             ppc64      3.0-6
         /tmp/cellsdk/openSrc/blas-devel-3.0-6.ppc64.rpm   0.0
 blas-devel             ppc        3.0-6
         /tmp/cellsdk/openSrc/blas-devel-3.0-6.ppc.rpm  741 k
 numactl                ppc64      0.9.10-1
         /tmp/cellsdk/openSrc/numactl-0.9.10-1.ppc64.rpm  180 k
 numactl                ppc        0.9.10-1
         /tmp/cellsdk/openSrc/numactl-0.9.10-1.ppc.rpm  186 k
 numactl-devel          ppc64      0.9.10-1
         /tmp/cellsdk/openSrc/numactl-devel-0.9.10-1.ppc64.rpm   12 k
 numactl-devel          ppc        0.9.10-1
         /tmp/cellsdk/openSrc/numactl-devel-0.9.10-1.ppc.rpm   12 k

Transaction Summary
=============================================================================
Install      8 Package(s)
Update       0 Package(s)
Remove       0 Package(s)

Total download size: 3.8 M
Downloading Packages:
Running Transaction Test
Finished Transaction Test
Transaction Test Succeeded
Running Transaction

  Installing: numactl                      ######################### [1/8]
  Installing: blas                         ######################### [2/8]
  Installing: numactl                      ######################### [3/8]
  Installing: numactl-devel                ######################### [4/8]
```

```
  Installing: blas-devel                  ######################### [5/8]
  Installing: blas                        ######################### [6/8]
  Installing: blas-devel                  ######################### [7/8]
  Installing: numactl-devel               ######################### [8/8]

Installed: blas.ppc 0:3.0-6 blas.ppc64 0:3.0-6 blas-devel.ppc64 0:3.0-6
blas-devel.ppc 0:3.0-6 numactl.ppc64 0:0.9.10-1 numactl.ppc 0:0.9.10-1
numactl-devel.ppc64 0:0.9.10-1 numactl-devel.ppc 0:0.9.10-1
Complete!
```

Appendix C. cellsdk script SDK verify example

This is an example of using the cellsdk script to verify the SDK on a BladeCenter QS21 running Red Hat Enterprise Linux (RHEL) 5.1.

The following output is a result of typing the command:

```
./cellsdk verify
```

Here is the output:

```
repository=CellSDK-Devel-RHEL-cbea
    default   CellDevelopmentLibraries  alf-devel-3.0.0-9.ppc
    default   CellDevelopmentLibraries  alf-devel-3.0.0-9.ppc64
    default   CellDevelopmentLibraries  alfman-3.0-8.noarch
    default   CellDevelopmentLibraries  blas-devel-3.0-6.ppc
    default   CellDevelopmentLibraries  blas-devel-3.0-6.ppc64
    default   CellDevelopmentLibraries  dacs-devel-3.0.0-19.ppc64
    default   CellDevelopmentLibraries  dacsman-3.0-6.noarch
    default   CellDevelopmentLibraries  libspe2man-2.2.0-5.noarch
    default   CellDevelopmentLibraries  ppu-mass-devel-4.5.0-10.ppc
    default   CellDevelopmentLibraries  ppu-mass-devel-4.5.0-10.ppc64
    default   CellDevelopmentLibraries  ppu-simdmath-devel-3.0-5.ppc
    default   CellDevelopmentLibraries  ppu-simdmath-devel-3.0-5.ppc64
    default   CellDevelopmentLibraries  simdman-3.0-6.noarch
    default   CellDevelopmentLibraries  spu-mass-devel-4.5.0-10.ppc
    default   CellDevelopmentLibraries  spu-simdmath-devel-3.0-5.ppc
    default   CellProgrammingExamples   alf-examples-source-3.0.0-7.noarch
    default   CellProgrammingExamples   cell-buildutils-3.0-11.noarch
    default   CellProgrammingExamples   cell-demos-3.0-10.ppc64
    default   CellProgrammingExamples   cell-demos-source-3.0-10.noarch
    default   CellProgrammingExamples   cell-examples-3.0-11.ppc64
    default   CellProgrammingExamples   cell-examples-source-3.0-11.noarch
    default   CellProgrammingExamples   cell-libs-3.0-16.ppc
    default   CellProgrammingExamples   cell-libs-3.0-16.ppc64
    default   CellProgrammingExamples   cell-libs-devel-3.0-16.ppc
    default   CellProgrammingExamples   cell-libs-devel-3.0-16.ppc64
    default   CellProgrammingExamples   cell-libs-source-3.0-16.noarch
    default   CellProgrammingExamples   cell-tutorial-3.0-6.ppc
    default   CellProgrammingExamples   cell-tutorial-source-3.0-6.noarch
    default   CellRuntimeEnvironment    alf-3.0.0-9.ppc
    default   CellRuntimeEnvironment    alf-3.0.0-9.ppc64
    default   CellRuntimeEnvironment    blas-3.0-6.ppc
    default   CellRuntimeEnvironment    blas-3.0-6.ppc64
    default   CellRuntimeEnvironment    dacs-3.0.0-19.ppc64
    default   CellRuntimeEnvironment    ppu-simdmath-3.0-5.ppc
    default   CellRuntimeEnvironment    ppu-simdmath-3.0-5.ppc64
    mandatory CellDevelopmentLibraries  cell-documentation-3.0-5.noarch
    mandatory CellRuntimeEnvironment    cell-devel-license-3.0.0-1.0.noarch
    optional  CellDevelopmentLibraries  alf-trace-devel not installed
    optional  CellDevelopmentLibraries  dacs-trace-devel not installed
    optional  CellDevelopmentLibraries  pdt-devel not installed
    optional  CellDevelopmentLibraries  trace-devel not installed
    optional  CellDevelopmentTools      alf-ide-template not installed
    optional  CellDevelopmentTools      cellide not installed
    optional  CellPerformanceTools      fdprpro not installed
    optional  CellPerformanceTools      pdt not installed
    optional  CellPerformanceTools      pdtr not installed
    optional  CellProgrammingExamples   cell-compliance-tests not installed
    optional  CellProgrammingExamples   cell-compliance-tests-source not installed
    optional  CellRuntimeEnvironment    alf-debuginfo not installed
    optional  CellRuntimeEnvironment    alf-trace not installed
    optional  CellRuntimeEnvironment    dacs-debuginfo not installed
```

```
    optional  CellRuntimeEnvironment    dacs-trace not installed
    optional  CellRuntimeEnvironment    simdmath-debuginfo not installed

repository=CellSDK-Open-RHEL-cbea
    default   CellDevelopmentTools      ppu-binutils-2.17.50-32.ppc
    default   CellDevelopmentTools      ppu-gcc-4.1.1-57.ppc
    default   CellDevelopmentTools      ppu-gcc-c++-4.1.1-57.ppc
    default   CellDevelopmentTools      ppu-gdb-6.6.50-28.ppc
    default   CellDevelopmentTools      spu-binutils-2.17.50-33.ppc
    default   CellDevelopmentTools      spu-gcc-4.1.1-107.ppc
    default   CellDevelopmentTools      spu-gcc-c++-4.1.1-107.ppc
    default   CellDevelopmentTools      spu-gdb-6.6.50-12.ppc
    default   CellDevelopmentTools      spu-newlib-1.15.0-81.ppc
    optional  CellDevelopmentLibraries  numactl-devel-0.9.10-1.ppc
    optional  CellDevelopmentLibraries  numactl-devel-0.9.10-1.ppc64
    optional  CellDevelopmentTools      ppu-binutils-debuginfo not installed
    optional  CellDevelopmentTools      ppu-gcc-debuginfo not installed
    optional  CellDevelopmentTools      ppu-gcc-fortran not installed
    optional  CellDevelopmentTools      ppu-gcc-gnat not installed
    optional  CellDevelopmentTools      ppu-gdb-debuginfo not installed
    optional  CellDevelopmentTools      spu-binutils-debuginfo not installed
    optional  CellDevelopmentTools      spu-gcc-debuginfo not installed
    optional  CellDevelopmentTools      spu-gcc-fortran not installed
    optional  CellDevelopmentTools      spu-gdb-debuginfo not installed
    optional  CellDevelopmentTools      spu-newlib-debuginfo not installed
    optional  CellRuntimeEnvironment    numactl-0.9.10-1.ppc
    optional  CellRuntimeEnvironment    numactl-0.9.10-1.ppc64
```

Appendix D. cellsdk script SDK update example

This is an example of using the cellsdk script to update the SDK on a BladeCenter QS21 running Red Hat Enterprise Linux (RHEL) 5.1.

The following output is a result of typing the command:

```
# /opt/cell/cellsdk --iso /tmp/cellsdkiso update
```

Here is the output:

```
# /opt/cell/cellsdk --iso /tmp/cellsdkiso update

cellsdk INFO-2023: Trying to install or update cell-prod-license
cellsdk INFO-2024: cell-prod-license is installed
cellsdk INFO-2023: Trying to install or update cell-extras-license
cellsdk INFO-2024: cell-extras-license is installed
Preparing...                ##################################################
cell-install                ##################################################
cellsdk INFO-2006: spawning a new cellsdk using the newer cell-install version
cellsdk logs to /var/log/cellsdk.log

The IBM SDK 3.0 for Multicore Acceleration contains components governed by
the following Open Source licenses:

GNU Public License (GPL) - see http://www.gnu.org/licenses/gpl.html.
  GPL licensed components include GCC Toolchain and numactl.

Installation and use of this software requires you to certify you have read the
license above, and accept its terms.  To accept these terms, type 'yes' at
the prompt below.  If you do not wish to accept, type 'no' or press 'ctrl-C' to
abort this program without installing.

After typing 'yes', you will be presented with licenses for IBM owned code,
Follow the instructions on the screen to accept the IBM Licenses and proceed with
installation of the SDK.

I have read and accept the licenses above [no/yes]: cellsdk INFO-2028: GPL/LGPL
 license accepted
cellsdk INFO-2023: Trying to install or update cell-product-license
cellsdk INFO-2024: cell-product-license is installed
International Program License Agreement

Part 1 - General Terms

BY DOWNLOADING, INSTALLING, COPYING, ACCESSING, OR USING THE PROGRAM YOU AGREE
TO THE TERMS OF THIS AGREEMENT. IF YOU ARE ACCEPTING THESE TERMS ON BEHALF OF
ANOTHER PERSON OR A COMPANY OR OTHER LEGAL ENTITY, YOU REPRESENT AND WARRANT
THAT YOU HAVE FULL AUTHORITY TO BIND THAT PERSON, COMPANY, OR LEGAL ENTITY TO
THESE TERMS. IF YOU DO NOT AGREE TO THESE TERMS,

- DO NOT DOWNLOAD, INSTALL, COPY, ACCESS, OR USE THE PROGRAM; AND

- PROMPTLY RETURN THE PROGRAM AND PROOF OF ENTITLEMENT TO THE PARTY FROM WHOM
YOU ACQUIRED IT TO OBTAIN A REFUND OF THE AMOUNT YOU PAID. IF YOU DOWNLOADED
THE PROGRAM, CONTACT THE PARTY FROM WHOM YOU ACQUIRED IT.

"IBM" is International Business Machines Corporation or one of its


Press Enter to continue viewing the license agreement, or, Enter "1" to accept
the agreement, "2" to decline it or "99" to go back to the previous screen,
"3" Print, "4" Read non-IBM terms.
```

```
cellsdk INFO-2014: product license accepted.
cellsdk INFO-2023: Trying to install or update cell-extras-RHEL-license
cellsdk INFO-2019: yum groupupdate  'Cell Runtime Environment'
 'Cell Development Tools' 'Cell Development Libraries' 'Cell Programming Examples'
 'Cell Simulator' 'Cell Performance Tools'
Loading "installonlyn" plugin
Loading "security" plugin
Loading "rhnplugin" plugin
Setting up Group Process
Setting up repositories
Setting up repositories
Reading repository metadata in from local files
Excluding Packages in global exclude list
Finished
Package cell-product-license - 3.0.0-1.0.noarch already installed and latest
 version
Package cell-prod-license - 3.0.0-0.0.noarch already installed and latest version
Package alf - 3.0.0-9.ppc64 already installed and latest version
Package alf - 3.0.0-9.ppc already installed and latest version
Package dacs - 3.0.0-19.ppc64 already installed and latest version
Package ppu-gdb - 6.6.50-23.ppc already installed and latest version
Package ppu-gcc-c++ - 4.1.1-54.ppc already installed and latest version
Package ppu-binutils - 2.17.50-31.ppc already installed and latest version
Package spu-newlib - 1.15.0-76.ppc already installed and latest version
Package spu-gdb - 6.6.50-10.ppc already installed and latest version
Package ppu-gcc - 4.1.1-54.ppc already installed and latest version
Package spu-binutils - 2.17.50-31.ppc already installed and latest version
Package spu-gcc-c++ - 4.1.1-100.ppc already installed and latest version
Package spu-gcc - 4.1.1-100.ppc already installed and latest version
Package spu-mass-devel - 4.5.0-10.ppc already installed and latest version
Package libspe2man - 2.2.0-4.noarch already installed and latest version
Package dacs-devel - 3.0.0-19.ppc64 already installed and latest version
Package alf-devel - 3.0.0-9.ppc64 already installed and latest version
Package alf-devel - 3.0.0-9.ppc already installed and latest version
Package ppu-mass-devel - 4.5.0-10.ppc64 already installed and latest version
Package ppu-mass-devel - 4.5.0-10.ppc already installed and latest version
Package cell-libs - 3.0-16.ppc64 already installed and latest version
Package cell-libs - 3.0-16.ppc already installed and latest version
Package cell-demos - 3.0-10.ppc64 already installed and latest version
Package cell-libs-source - 3.0-16.noarch already installed and latest version
Package cell-tutorial - 3.0-6.ppc already installed and latest version
Package cell-tutorial-source - 3.0-6.noarch already installed and latest version
Package cell-examples - 3.0-11.ppc64 already installed and latest version
Package cell-examples-source - 3.0-11.noarch already installed and latest version
Package alf-examples-source - 3.0.0-7.noarch already installed and latest version
Package cell-libs-devel - 3.0-16.ppc64 already installed and latest version
Package cell-libs-devel - 3.0-16.ppc already installed and latest version
Package cell-demos-source - 3.0-10.noarch already installed and latest version
Warning: Group Cell Simulator does not exist.
Resolving Dependencies
--> Populating transaction set with selected packages. Please wait.
---> Package ppu-simdmath-devel.ppc64 0:3.0-5 set to be updated
---> Package spu-simdmath-devel.ppc 0:3.0-5 set to be updated
---> Package cell-documentation.noarch 0:3.0-5 set to be updated
---> Package ppu-simdmath-devel.ppc 0:3.0-5 set to be updated
---> Package alfman.noarch 0:3.0-8 set to be updated
---> Package dacsman.noarch 0:3.0-6 set to be updated
---> Package cell-buildutils.noarch 0:3.0-11 set to be updated
---> Package ppu-simdmath.ppc 0:3.0-5 set to be updated
---> Package ppu-simdmath.ppc64 0:3.0-5 set to be updated
---> Package simdman.noarch 0:3.0-6 set to be updated
--> Running transaction check

Dependencies Resolved

=============================================================================
 Package                 Arch       Version          Repository        Size
```

```
=============================================================================
Updating:
 alfman                 noarch     3.0-8      CellSDK-Updates-RHEL-cbea   49 k
 cell-buildutils        noarch     3.0-11     CellSDK-Updates-RHEL-cbea   19 k
 cell-documentation     noarch     3.0-5      CellSDK-Updates-RHEL-cbea   40 M
 dacsman                noarch     3.0-6      CellSDK-Updates-RHEL-cbea   62 k
 ppu-simdmath           ppc        3.0-5      CellSDK-Updates-RHEL-cbea   26 k
 ppu-simdmath           ppc64      3.0-5      CellSDK-Updates-RHEL-cbea   31 k
 ppu-simdmath-devel     ppc64      3.0-5      CellSDK-Updates-RHEL-cbea   99 k
 ppu-simdmath-devel     ppc        3.0-5      CellSDK-Updates-RHEL-cbea   94 k
 simdman                noarch     3.0-6      CellSDK-Updates-RHEL-cbea  158 k
 spu-simdmath-devel     ppc        3.0-5      CellSDK-Updates-RHEL-cbea  178 k

Transaction Summary
=============================================================================
Install      0 Package(s)
Update      10 Package(s)
Remove       0 Package(s)

Total download size: 41 M
Is this ok [y/N]: y
Downloading Packages:
Running Transaction Test
Finished Transaction Test
Transaction Test Succeeded
Running Transaction
  Updating  : ppu-simdmath                 ####################### [ 1/20]
  Updating  : ppu-simdmath-devel           ####################### [ 2/20]
  Updating  : spu-simdmath-devel           ####################### [ 3/20]
  Updating  : cell-documentation           ####################### [ 4/20]
  Updating  : ppu-simdmath-devel           ####################### [ 5/20]
  Updating  : alfman                       ####################### [ 6/20]
  Updating  : dacsman                      ####################### [ 7/20]
  Updating  : cell-buildutils              ####################### [ 8/20]
  Updating  : ppu-simdmath                 ####################### [ 9/20]
  Updating  : simdman                      ####################### [10/20]

Updated: alfman.noarch 0:3.0-8 cell-buildutils.noarch 0:3.0-11
 cell-documentation.noarch 0:3.0-5 dacsman.noarch 0:3.0-6 ppu-simdmath.ppc 0:3.0-5
 ppu-simdmath.ppc64 0:3.0-5 ppu-simdmath-devel.ppc64 0:3.0-5
 ppu-simdmath-devel.ppc 0:3.0-5 simdman.noarch 0:3.0-6
 spu-simdmath-devel.ppc 0:3.0-5
Complete!
cellsdk INFO-2025: Copying files to
 /opt/cell/updates/3.0.0-1.1/CellSDK-Updates-RHEL-cbea
cellsdk INFO-2026:     ppu-simdmath-3.0-5.ppc.rpm
cellsdk INFO-2026:     ppu-simdmath-3.0-5.ppc64.rpm
cellsdk INFO-2026:     cell-product-license-3.0.0-1.0.noarch.rpm
cellsdk INFO-2026:     cellide-3.0.0-16.noarch.rpm
cellsdk INFO-2026:     alfman-3.0-8.noarch.rpm
cellsdk INFO-2026:     cell-documentation-3.0-5.noarch.rpm
cellsdk INFO-2026:     dacsman-3.0-6.noarch.rpm
cellsdk INFO-2026:     ppu-simdmath-devel-3.0-5.ppc.rpm
cellsdk INFO-2026:     ppu-simdmath-devel-3.0-5.ppc64.rpm
cellsdk INFO-2026:     spu-simdmath-devel-3.0-5.ppc.rpm
cellsdk INFO-2026:     simdman-3.0-6.noarch.rpm
cellsdk INFO-2026:     pdt-devel-3.0-37.ppc.rpm
cellsdk INFO-2026:     pdt-devel-3.0-37.ppc64.rpm
cellsdk INFO-2026:     trace-devel-3.0-37.ppc.rpm
cellsdk INFO-2026:     cell-buildutils-3.0-11.noarch.rpm
cellsdk INFO-2026:     pdt-3.0-37.ppc.rpm
cellsdk INFO-2026:     pdt-3.0-37.ppc64.rpm

Would you like to save the initial rpms in case
you want to remove this update later? [y/n] y
cellsdk INFO-2025: Copying files to
 /opt/cell/updates/3.0.0-1.0/CellSDK-Product-RHEL-cbea
```

```
cellsdk INFO-2026:     alf-3.0.0-9.ppc.rpm
cellsdk INFO-2026:     alf-3.0.0-9.ppc64.rpm
cellsdk INFO-2026:     alf-debuginfo-3.0.0-9.ppc.rpm
cellsdk INFO-2026:     alf-debuginfo-3.0.0-9.ppc64.rpm
cellsdk INFO-2026:     alf-trace-3.0.0-9.ppc.rpm
cellsdk INFO-2026:     alf-trace-3.0.0-9.ppc64.rpm
cellsdk INFO-2026:     dacs-3.0.0-19.ppc64.rpm
cellsdk INFO-2026:     dacs-debuginfo-3.0.0-19.ppc64.rpm
cellsdk INFO-2026:     dacs-trace-3.0.0-19.ppc64.rpm
cellsdk INFO-2026:     ppu-simdmath-3.0-4.ppc.rpm
cellsdk INFO-2026:     ppu-simdmath-3.0-4.ppc64.rpm
cellsdk INFO-2026:     simdmath-debuginfo-3.0-4.ppc.rpm
cellsdk INFO-2026:     simdmath-debuginfo-3.0-4.ppc64.rpm
cellsdk INFO-2026:     blas-3.0-6.ppc.rpm
cellsdk INFO-2026:     blas-3.0-6.ppc64.rpm
cellsdk INFO-2026:     cell-prod-license-3.0.0-0.0.noarch.rpm
cellsdk INFO-2026:     cellide-3.0.0-15.noarch.rpm
cellsdk INFO-2026:     alf-ide-template-3.0.0-1.ppc.rpm
cellsdk INFO-2026:     alf-devel-3.0.0-9.ppc.rpm
cellsdk INFO-2026:     alf-devel-3.0.0-9.ppc64.rpm
cellsdk INFO-2026:     alf-trace-devel-3.0.0-9.ppc.rpm
cellsdk INFO-2026:     alf-trace-devel-3.0.0-9.ppc64.rpm
cellsdk INFO-2026:     blas-devel-3.0-6.ppc.rpm
cellsdk INFO-2026:     blas-devel-3.0-6.ppc64.rpm
cellsdk INFO-2026:     dacs-trace-devel-3.0.0-19.ppc64.rpm
cellsdk INFO-2026:     alfman-3.0-6.noarch.rpm
cellsdk INFO-2026:     cell-documentation-3.0-4.noarch.rpm
cellsdk INFO-2026:     dacs-devel-3.0.0-19.ppc64.rpm
cellsdk INFO-2026:     dacsman-3.0-5.noarch.rpm
cellsdk INFO-2026:     ppu-mass-devel-4.5.0-10.ppc.rpm
cellsdk INFO-2026:     ppu-mass-devel-4.5.0-10.ppc64.rpm
cellsdk INFO-2026:     spu-mass-devel-4.5.0-10.ppc.rpm
cellsdk INFO-2026:     ppu-simdmath-devel-3.0-4.ppc.rpm
cellsdk INFO-2026:     ppu-simdmath-devel-3.0-4.ppc64.rpm
cellsdk INFO-2026:     spu-simdmath-devel-3.0-4.ppc.rpm
cellsdk INFO-2026:     simdman-3.0-5.noarch.rpm
cellsdk INFO-2026:     libspe2man-2.2.0-4.noarch.rpm
cellsdk INFO-2026:     pdt-devel-3.0-36.ppc.rpm
cellsdk INFO-2026:     pdt-devel-3.0-36.ppc64.rpm
cellsdk INFO-2026:     trace-devel-3.0-36.ppc.rpm
cellsdk INFO-2026:     alf-examples-source-3.0.0-7.noarch.rpm
cellsdk INFO-2026:     cell-buildutils-3.0-10.noarch.rpm
cellsdk INFO-2026:     cell-tutorial-3.0-6.ppc.rpm
cellsdk INFO-2026:     cell-tutorial-source-3.0-6.noarch.rpm
cellsdk INFO-2026:     cell-demos-3.0-10.ppc64.rpm
cellsdk INFO-2026:     cell-demos-source-3.0-10.noarch.rpm
cellsdk INFO-2026:     cell-compliance-tests-3.0-6.ppc64.rpm
cellsdk INFO-2026:     cell-compliance-tests-source-3.0-6.noarch.rpm
cellsdk INFO-2026:     cell-examples-3.0-11.ppc64.rpm
cellsdk INFO-2026:     cell-examples-source-3.0-11.noarch.rpm
cellsdk INFO-2026:     cell-libs-3.0-16.ppc.rpm
cellsdk INFO-2026:     cell-libs-3.0-16.ppc64.rpm
cellsdk INFO-2026:     cell-libs-devel-3.0-16.ppc.rpm
cellsdk INFO-2026:     cell-libs-devel-3.0-16.ppc64.rpm
cellsdk INFO-2026:     cell-libs-source-3.0-16.noarch.rpm
cellsdk INFO-2026:     alf-hybrid-examples-source-3.0.0-7.noarch.rpm
cellsdk INFO-2026:     fdprpro-5.4.0-16.ppc.rpm
cellsdk INFO-2026:     pdt-3.0-36.ppc.rpm
cellsdk INFO-2026:     pdt-3.0-36.ppc64.rpm
cellsdk INFO-2026:     pdtr-3.0-11.ppc64.rpm
```

Appendix E. cellsdk script SDK uninstallation example

This is an example of using the cellsdk script to uninstall the SDK on a BladeCenter QS21 running Red Hat Enterprise Linux (RHEL) 5.1.

The following output is the result of typing the commands:

```
cd /opt/cell
./cellsdk --iso /tmp/cellsdkiso uninstall
```

Here is the output:

```
cellsdk INFO-2019: yum groupremove  'Cell Runtime Environment'
   'Cell Development Tools' 'Cell Development Libraries'
   'Cell Programming Examples' 'Cell Simulator' 'Cell Performance Tools'
Loading "installonlyn" plugin
Loading "security" plugin
Loading "rhnplugin" plugin
Setting up Group Process
Setting up repositories
No package matched to remove
No group named Cell Simulator exists
No package matched to remove
Resolving Dependencies
--> Populating transaction set with selected packages. Please wait.
---> Package numactl-devel.ppc64 0:0.9.10-1 set to be erased
---> Package dacs.ppc64 0:3.0.0-19 set to be erased
---> Package cell-tutorial-source.noarch 0:3.0-6 set to be erased
---> Package ppu-mass-devel.ppc 0:4.5.0-10 set to be erased
---> Package spu-gcc-c++.ppc 0:4.1.1-107 set to be erased
---> Package dacsman.noarch 0:3.0-6 set to be erased
---> Package ppu-binutils.ppc 0:2.17.50-32 set to be erased
---> Package cell-libs.ppc 0:3.0-16 set to be erased
---> Package cell-libs.ppc64 0:3.0-16 set to be erased
---> Package cell-devel-license.noarch 0:3.0.0-1.0 set to be erased
---> Package blas.ppc64 0:3.0-6 set to be erased
---> Package cell-demos.ppc64 0:3.0-10 set to be erased
---> Package ppu-gcc.ppc 0:4.1.1-57 set to be erased
---> Package spu-gdb.ppc 0:6.6.50-12 set to be erased
---> Package alf-examples-source.noarch 0:3.0.0-7 set to be erased
---> Package numactl.ppc 0:0.9.10-1 set to be erased
---> Package cell-examples-source.noarch 0:3.0-11 set to be erased
---> Package alfman.noarch 0:3.0-8 set to be erased
---> Package ppu-simdmath.ppc 0:3.0-5 set to be erased
---> Package blas-devel.ppc 0:3.0-6 set to be erased
---> Package alf.ppc 0:3.0.0-9 set to be erased
---> Package numactl.ppc64 0:0.9.10-1 set to be erased
---> Package alf.ppc64 0:3.0.0-9 set to be erased
---> Package alf-devel.ppc64 0:3.0.0-9 set to be erased
---> Package spu-mass-devel.ppc 0:4.5.0-10 set to be erased
---> Package simdman.noarch 0:3.0-6 set to be erased
---> Package ppu-simdmath-devel.ppc64 0:3.0-5 set to be erased
---> Package cell-examples.ppc64 0:3.0-11 set to be erased
---> Package blas.ppc 0:3.0-6 set to be erased
---> Package cell-libs-source.noarch 0:3.0-16 set to be erased
---> Package dacs-devel.ppc64 0:3.0.0-19 set to be erased
---> Package ppu-simdmath.ppc64 0:3.0-5 set to be erased
---> Package spu-newlib.ppc 0:1.15.0-81 set to be erased
---> Package blas-devel.ppc64 0:3.0-6 set to be erased
---> Package ppu-gcc-c++.ppc 0:4.1.1-57 set to be erased
---> Package cell-libs-devel.ppc64 0:3.0-16 set to be erased
---> Package spu-binutils.ppc 0:2.17.50-33 set to be erased
---> Package cell-tutorial.ppc 0:3.0-6 set to be erased
```



```
---> Package spu-simdmath-devel.ppc 0:3.0-5 set to be erased
---> Package cell-documentation.noarch 0:3.0-5 set to be erased
---> Package alf-devel.ppc 0:3.0.0-9 set to be erased
---> Package ppu-simdmath-devel.ppc 0:3.0-5 set to be erased
---> Package libspe2man.noarch 0:2.2.0-5 set to be erased
---> Package cell-demos-source.noarch 0:3.0-10 set to be erased
---> Package cell-buildutils.noarch 0:3.0-11 set to be erased
---> Package numactl-devel.ppc 0:0.9.10-1 set to be erased
---> Package ppu-gdb.ppc 0:6.6.50-28 set to be erased
---> Package ppu-mass-devel.ppc64 0:4.5.0-10 set to be erased
---> Package cell-libs-devel.ppc 0:3.0-16 set to be erased
---> Package spu-gcc.ppc 0:4.1.1-107 set to be erased
--> Running transaction check

Dependencies Resolved

=============================================================================
 Package                 Arch       Version          Repository        Size
=============================================================================
Removing:
 alf                     ppc        3.0.0-9          installed          71 k
 alf                     ppc64      3.0.0-9          installed          96 k
 alf-devel               ppc64      3.0.0-9          installed         136 k
 alf-devel               ppc        3.0.0-9          installed         304 k
 alf-examples-source     noarch     3.0.0-7          installed         1.0 M
 alfman                  noarch     3.0-8            installed          38 k
 blas                    ppc64      3.0-6            installed         1.4 M
 blas                    ppc        3.0-6            installed         1.3 M
 blas-devel              ppc        3.0-6            installed         741 k
 blas-devel              ppc64      3.0-6            installed          0.0
 cell-buildutils         noarch     3.0-11           installed          76 k
 cell-demos              ppc64      3.0-10           installed          18 M
 cell-demos-source       noarch     3.0-10           installed          19 M
 cell-devel-license      noarch     3.0.0-1.0        installed         936 k
 cell-documentation      noarch     3.0-5            installed          55 M
 cell-examples           ppc64      3.0-11           installed         1.4 M
 cell-examples-source    noarch     3.0-11           installed         1.4 M
 cell-libs               ppc        3.0-16           installed         172 k
 cell-libs               ppc64      3.0-16           installed          20 k
 cell-libs-devel         ppc64      3.0-16           installed         542 k
 cell-libs-devel         ppc        3.0-16           installed         3.3 M
 cell-libs-source        noarch     3.0-16           installed         2.2 M
 cell-tutorial           ppc        3.0-6            installed         145 k
 cell-tutorial-source    noarch     3.0-6            installed         130 k
 dacs                    ppc64      3.0.0-19         installed          67 k
 dacs-devel              ppc64      3.0.0-19         installed         597 k
 dacsman                 noarch     3.0-6            installed          50 k
 libspe2man              noarch     2.2.0-5          installed          43 k
 numactl                 ppc        0.9.10-1         installed         186 k
 numactl                 ppc64      0.9.10-1         installed         180 k
 numactl-devel           ppc64      0.9.10-1         installed          12 k
 numactl-devel           ppc        0.9.10-1         installed          12 k
 ppu-binutils            ppc        2.17.50-32       installed          19 M
 ppu-gcc                 ppc        4.1.1-57         installed          12 M
 ppu-gcc-c++             ppc        4.1.1-57         installed          99 M
 ppu-gdb                 ppc        6.6.50-28        installed         7.9 M
 ppu-mass-devel          ppc        4.5.0-10         installed         559 k
 ppu-mass-devel          ppc64      4.5.0-10         installed         675 k
 ppu-simdmath            ppc        3.0-5            installed         156 k
 ppu-simdmath            ppc64      3.0-5            installed         190 k
 ppu-simdmath-devel      ppc64      3.0-5            installed         707 k
 ppu-simdmath-devel      ppc        3.0-5            installed         622 k
 simdman                 noarch     3.0-6            installed         133 k
 spu-binutils            ppc        2.17.50-33       installed         7.4 M
 spu-gcc                 ppc        4.1.1-107        installed         7.0 M
 spu-gcc-c++             ppc        4.1.1-107        installed          60 M
 spu-gdb                 ppc        6.6.50-12        installed         5.1 M
```

```
 spu-mass-devel         ppc        4.5.0-10         installed        413 k
 spu-newlib             ppc        1.15.0-81        installed        4.7 M
 spu-simdmath-devel     ppc        3.0-5            installed        1.0 M

Transaction Summary
=============================================================================
Install      0 Package(s)
Update       0 Package(s)
Remove      50 Package(s)

Is this ok [y/N]: y
Downloading Packages:
Running Transaction Test
Finished Transaction Test
Transaction Test Succeeded
Running Transaction

Removed: alf.ppc 0:3.0.0-9 alf.ppc64 0:3.0.0-9 alf-devel.ppc64 0:3.0.0-9
 alf-devel.ppc 0:3.0.0-9 alf-examples-source.noarch 0:3.0.0-7
 alfman.noarch 0:3.0-8 blas.ppc64 0:3.0-6 blas.ppc 0:3.0-6
 blas-devel.ppc 0:3.0-6 blas-devel.ppc64 0:3.0-6 cell-buildutils.noarch 0:3.0-11
 cell-demos.ppc64 0:3.0-10 cell-demos-source.noarch 0:3.0-10
 cell-devel-license.noarch 0:3.0.0-1.0 cell-documentation.noarch 0:3.0-5
 cell-examples.ppc64 0:3.0-11 cell-examples-source.noarch 0:3.0-11
 cell-libs.ppc 0:3.0-16 cell-libs.ppc64 0:3.0-16 cell-libs-devel.ppc64 0:3.0-16
 cell-libs-devel.ppc 0:3.0-16 cell-libs-source.noarch 0:3.0-16
 cell-tutorial.ppc 0:3.0-6 cell-tutorial-source.noarch 0:3.0-6
 dacs.ppc64 0:3.0.0-19 dacs-devel.ppc64 0:3.0.0-19 dacsman.noarch 0:3.0-6
 libspe2man.noarch 0:2.2.0-5 numactl.ppc 0:0.9.10-1 numactl.ppc64 0:0.9.10-1
 numactl-devel.ppc64 0:0.9.10-1 numactl-devel.ppc 0:0.9.10-1
 ppu-binutils.ppc 0:2.17.50-32 ppu-gcc.ppc 0:4.1.1-57 ppu-gcc-c++.ppc 0:4.1.1-57
 ppu-gdb.ppc 0:6.6.50-28 ppu-mass-devel.ppc 0:4.5.0-10
 ppu-mass-devel.ppc64 0:4.5.0-10 ppu-simdmath.ppc 0:3.0-5
 ppu-simdmath.ppc64 0:3.0-5 ppu-simdmath-devel.ppc64 0:3.0-5
 ppu-simdmath-devel.ppc 0:3.0-5 simdman.noarch 0:3.0-6
 spu-binutils.ppc 0:2.17.50-33 spu-gcc.ppc 0:4.1.1-107 spu-gcc-c++.ppc 0:4.1.1-107
 spu-gdb.ppc 0:6.6.50-12 spu-mass-devel.ppc 0:4.5.0-10 spu-newlib.ppc 0:1.15.0-81
 spu-simdmath-devel.ppc 0:3.0-5
Complete!
cellsdk INFO-2009: looking for still-installed cellsdk rpms
installedCount=0

Completely remove cellsdk from the system [y/n]? y
/tmp/sdk/CellSDK-Devel-RHEL/x86
/tmp/sdk/CellSDK-Open-RHEL/x86
/tmp/sdk/CellSDK-Devel-RHEL/x86_64
/tmp/sdk/CellSDK-Open-RHEL/x86_64
/tmp/sdk/CellSDK-Devel-RHEL/ppc64
/tmp/sdk/CellSDK-Open-RHEL/ppc64
/tmp/sdk/CellSDK-Devel-RHEL/cbea
/tmp/sdk/CellSDK-Open-RHEL/cbea

If you have files in the /opt/cell you want to keep, answer no.
Completely remove /opt/cell [y/n]? y

If you have files in the /opt/ibm/systemsim you want to keep, answer no.
Completely remove /opt/ibm/systemsim [y/n]? y
If you have files in the /tmp/cellsdk/rpms you want to keep, answer no.
Completely remove /tmp/cellsdk/rpms [y/n]? y
If you have files in the /tmp/cellsdk/openSrc you want to keep, answer no.
Completely remove /tmp/cellsdk/openSrc [y/n]? y
```

Appendix F. Known limitations

These are the known limitations and restrictions in this version of the SDK.

General limitations:

- For applications that link with the SDK example shared libraries (such as the libraries whose source is found in /opt/cell/sdk/src/lib) these libraries are not located in a standard library search path. To locate the libraries, set *LD_LIBRARY_PATH* or specify an *rpath* when linking the application. For example:

  ```
  LDFLAGS += -R/opt/cellsdk/sdk/usr/lib     # for 32-bit apps
  LDFLAGS += -R/opt/cellsdk/sdk/usr/lib64   # for 64-bit apps
  ```

  Or, use the *SDKEXRPATH* variable that make.footer defines:

  ```
  LDFLAGS += -R${SDKEXRPATH}
  ```

- The cellsdk_select_compiler bash script will allow you to select a compiler, even if it is not installed. The most common case is the xlc option which only functions correctly when the separately available *IBM XL C/C++ Alpha Edition for Multicore Acceleration for Linux, V0.9* package is installed.
- When using large pages, there is a risk that your application will terminate when you run it. To remove this problem, ensure that the number of large pages you request does not exceed the number available. To check, type the command `` `cat /proc/meminfo` `` and examine the line that says `HugePages_Free:` near the bottom of the output. Notify your administrator if this number is insufficient.
- When installing the SDK on X86_64 systems with RHEL 5.1, you might get a transaction check error when trying to install numactl because of a file conflict. The workaround is to first uninstall the RHEL 5.1-supplied numactl and then restart the SDK installation. To uninstall numactl, type this command:

  ```
  # rpm -e --nodeps --allmatches numactl
  ```

ALF *Black Scholes* example application:

- The ALF *Black Scholes* example application has the following command line parameters:

  ```
  ./BlackScholes_ALF_PPU <naccel> <double_flag> <ncycles> <size>
  ```

- The input parameters are:

  naccel How many accelerators the program will use, specified as the log2 of the number of accelerators. Valid values are between 0 and 4 inclusive:

  0 1 accelerator

  1 2 accelerators

  2 4 accelerators

  3 8 accelerators

  4 16 accelerators

  double_flag
  Specifies the calculation precision. Valid values are 0 or 1.

  0 Run in single floating point mode

1 Run in double floating point mode

ncyles Specifies how many times the computational kernel should run. This is for the program to return performance results. The valid range for this parameter is 1 to 512.

size Specifies the size of input parameters in kilobytes. The value will be the amount of the samples with the same type. Must be a multiple of 64.

- The following is an example:

```
# ./BlackScholes_ALF_PPU 3 0 128 64
```

DaCS for Cell BE and DaCS for Hybrid-x86 limitations:

- Performing a `dacs_runtime_exit()` call immediately after a DaCS for Hybrid-x86 communication operation without an intervening test for communication completion may cause the application to hang. The hang occurs because the exiting application may complete (and terminate communication) before the communication completes, leaving the other application (the one the exiting function was communicating with) in a hung state awaiting the communication. To avoid this problem ensure that your applications allow for communication to complete by testing for completion prior to calling `dacs_runtime_exit()`.
- When using the `dacs_de_start` API, it returns DACS_ERR_SYSTEM instead of INVALID_DACS_ERROR_CODE if the file doesn't exist on the remote machine. To diagnose the problem, set the environment variable *DACS_HYBRID_DEBUG=Y* and examine the associated log files on the accelerator(s) for the *adacsd*, typically /var/log/adacsd.log for the failing errno.

GCC Toolchain limitations:

- In the SPE, the `%l` qualifier for `printf/scanf` is not supported.
- When debugging a multi-threaded application, the debugger may get confused if, during a single-step operation on one thread, some other thread reports an event that causes debugging to stop. An example is a new thread is created, a shared library or SPE context is loaded, and some breakpoint is hit. Subsequent single-step operations on the original thread may behave incorrectly. When this occurs, you can restore correct operation of the debugger by not single-stepping and instead continuing the program until it hits another breakpoint.
- GDB has trouble correctly tracing an SPE context that is created with events enabled (*SPE_EVENTS_ENABLE*).
- Attempting to build the Linux kernel for Cell BE using the ppu-gcc compiler will fail with an Internal Compiler Error. As a workaround, build the kernel using the system compiler gcc.

Networking limitations:

- The eth1 network port on a BladeCenter QS20 may be slower than eth0 when transmitting UDP streams.

Performance Debugging Tool limitations:

- PDT tracing in a hybrid system is only supported on AMD Opteron chips that support the *RDTSCP* (read time-stamp counter and processor ID instruction) introduced in the NPT Family `0Fh` processors.

Full-System Simulator limitations:

- The libsim library has been removed from the SDK. Sample source code is provided with the simulator that allows users to build their own version of libsim. The source code can be found in /opt/ibm/systemsim-cell/sample/cell/

libsim. The first execution of the /opt/cell/cellsdk_sync_simulator script will build the libsim.a library and export it to the /opt/cell/sysroot/usr/spu/lib directory.

- The sim_printf.h header file has been removed from the SDK. A sample header file containing a prototype for sim_printf is shipped with the simulator and can be found in the /opt/ibm/systemsim-cell/sample/cell/libsim/spu/libsim.h file. The first execution of the /opt/cell/cellsdk_sync_simulator script will export the libsim.h header file to the /opt/cell/sysroot/usr/spu/include directory.
- The level 3 BLAS examples cause a BUS ERROR when they are executed in the Full-System Simulator.

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:

alphaWorks
BladeCenter
developerWorks
IBM
Passport Advantage
POWER™
PowerPC
PowerPC Architecture™
System p

Cell Broadband Engine and Cell BE are trademarks of Sony Computer Entertainment, Inc., in the United States, other countries, or both and is used under license therefrom

Intel, MMX, and Pentium are trademarks of Intel Corporation in the United States, other countries, or both.

Java and all Java-based trademarks and logos are trademarks or registered 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.

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.

Related documentation

This topic helps you find related information.

Document location

Links to documentation for the SDK are provided on the developerWorks Web site located at:

`http://www.ibm.com/developerworks/power/cell/`

Click the **Docs** tab.

The following documents are available, organized by category:

Architecture

- *Cell Broadband Engine Architecture*
- *Cell Broadband Engine Registers*
- *SPU Instruction Set Architecture*

Standards

- *C/C++ Language Extensions for Cell Broadband Engine Architecture*
- *Cell Broadband Engine Linux Reference Implementation Application Binary Interface Specification*
- *SIMD Math Library Specification for Cell Broadband Engine Architecture*
- *SPU Application Binary Interface Specification*
- *SPU Assembly Language Specification*

Programming

- *Cell Broadband Engine Programmer's Guide*
- *Cell Broadband Engine Programming Handbook*
- *Cell Broadband Engine Programming Tutorial*

Library

- *Accelerated Library Framework for Cell Programmer's Guide and API Reference*
- *Accelerated Library Framework for Hybrid-x86 Programmer's Guide and API Reference*
- *Basic Linear Algebra Subprograms Programmer's Guide and API Reference*
- *Cell Broadband Engine Monte Carlo Library API Reference Manual*
- *Data Communication and Synchronization for Cell Programmer's Guide and API Reference*
- *Data Communication and Synchronization for Hybrid-x86 Programmer's Guide and API Reference*
- *Example Library API Reference*
- *Mathematical Acceleration Subsystem (MASS)*
- *SIMD Math Library API Reference*
- *SPE Runtime Management Library*
- *SPE Runtime Management Library Version 1 to Version 2 Migration Guide*
- *SPU Timer Library*

Installation

- *SDK for Multicore Acceleration Version 3.0 Installation Guide*

IBM XL C/C++ and IBM XL Fortran Compilers

Documentation for the compilers is available from the following Web sites:

- http://www.ibm.com/software/awdtools/xlcpp/
- http://www.ibm.com/software/awdtools/fortran/

IBM Full-System Simulator

- *IBM Full-System Simulator User's Guide*
- *Performance Analysis with the IBM Full-System Simulator*

Tools

- *Getting Started - XL C/C++ Advanced Edition for Linux*
- *Compiler Reference - XL C/C++ Advanced Edition for Linux*
- *Language Reference - XL C/C++ Advanced Edition for Linux*
- *Programming Guide - XL C/C++ Advanced Edition for Linux*
- *Installation Guide - XL C/C++ Advanced Edition for Linux*
- *Getting Started - XL Fortran Advanced Edition for Linux*
- *Compiler Reference - XL Fortran Advanced Edition for Linux*
- *Language Reference - XL Fortran Advanced Edition for Linux*
- *Optimization and Programming Guide - XL Fortran Advanced Edition for Linux*
- *Installation Guide - XL Fortran Advanced Edition for Linux*
- *Using the single-source compiler*
- *Performance Analysis with the IBM Full-System Simulator*
- *IBM Full-System Simulator User's Guide*
- *IBM Visual Performance Analyzer User's Guide*

PowerPC Base

- *PowerPC Architecture Book*
  - *Book I: PowerPC User Instruction Set Architecture*
  - *Book II: PowerPC Virtual Environment Architecture*
  - *Book III: PowerPC Operating Environment Architecture*
- *PowerPC Microprocessor Family: Vector/SIMD Multimedia Extension Technology Programming Environments Manual*

Glossary

This glossary provides definitions for terms included in the *SDK Installation Guide*.

ALF

Accelerated Library Framework. This an API that provides a set of services to help programmers solving data parallel problems on a hybrid system. ALF supports the multiple-program-multiple-data (MPMD) programming style where multiple programs can be scheduled to run on multiple accelerator elements at the same time. ALF offers programmers an interface to partition data across a set of parallel processes without requiring architecturally-dependent code.

Barcelona Supercomputing Center

Spanish National Supercomputing Center, supporting Bladecenter and Linux on cell.

BE

Broadband Engine.

BOOTP

Bootstrap Protocol. A UDP network protocol used by a network client to obtain its IP address automatically. Replaced in many networks by DHCP.

Broadband Engine

See *CBEA*.

CBEA

Cell Broadband Engine Architecture. A new architecture that extends the 64-bit PowerPC Architecture. The CBEA and the Cell Broadband Engine are the result of a collaboration between Sony, Toshiba, and IBM, known as STI, formally started in early 2001.

Cell BE processor

The Cell BE processor is a multi-core broadband processor based on IBM's Power Architecture.

Cell Broadband Engine processor

See *Cell BE*.

DaCS

The Data Communication and Synchronization (DaCS) library provides functions that focus on process management, data movement, data synchronization, process synchronization, and error handling for processes within a hybrid system.

DaCS Element

A general or special purpose processing element in a topology. This refers specifically to the physical unit in the topology. A DE can serve as a Host or an Accelerator.

DHCP

Dynamic Host Configuration Protocol. Similar to BOOTP, DHCP is a protocol for assigning IP addresses to client devices on a network.

FDPR-Pro

Feedback Directed Program Restructuring. A feedback-based post-link optimization tool.

Fedora

Fedora is an operating system built from open source and free software. Fedora is free for anyone to use, modify, or distribute. For more information about Fedora and the Fedora Project, see the following Web site: http://fedoraproject.org/.

firmware

A set of instructions contained in ROM usually used to enable peripheral devices at boot.

GNU

GNU is Not Unix. A project to develop free Unix-like operating systems such as Linux.

GPL

GNU General Public License. Guarantees freedom to share, change and distribute free software.

GUI

Graphical User Interface. User interface for interacting with a computer which employs graphical images and widgets in addition to text to represent the information and actions available to the user. Usually the actions are performed through direct manipulation of the graphical elements.

host

A general purpose processing element in a hybrid system. A host can have multiple accelerators attached to it. This is often referred to as the master node in a cluster collective.

HTTP

Hypertext Transfer Protocol. A method used to transfer or convey information on the World Wide Web.

Hybrid

A module comprised of two Cell BE cards connected via an AMD Opteron processor.

hypervisor

A control (or virtualization) layer between hardware and the operating system. It allocates resources, reserves resources, and protects resources among (for example) sets of SPEs that may be running under different operating systems. The Cell Broadband Engine has three operating modes: user, supervisor and hypervisor. The hypervisor performs a meta-supervisor role that allows multiple independent supervisors' software to run on the same hardware platform. For example, the hypervisor allows both a real-time operating system and a traditional operating system to run on a single PPE. The PPE can then operate a subset of the SPEs in the Cell Broadband Engine with the realtime operating system, while the other SPEs run under the traditional operating system.

IDE

Integrated Development Environment. Integrates the Cell/B.E. GNU tool chain, compilers, the Full-System Simulator, and other development components to provide a comprehensive, Eclipse-based development platform that simplifies Cell/B.E. development.

initrd

A command file read at boot

ISO image

Commonly a disk image which can be burnt to CD. Technically it is a disk image of and ISO 9660 file system.

kernel

The core of an operating which provides services for other parts of the operating system and provides multitasking. In Linux or UNIX operating system, the kernel can easily be rebuilt to incorporate enhancements which then become operating-system wide.

LGPL

Lesser General Public License. Similar to the *GPL*, but does less to protect the user's freedom.

Makefile

A descriptive file used by the `make`command in which the user specifies: (a) target program or library, (b) rules about how the target is to be built, (c) dependencies which, if updated, require that the target be rebuilt.

netboot

Command to boot a device from another on the same network. Requires a TFTP server.

NUMA

Non-uniform memory access. In a multiprocessing system such as the Cell/B.E., memory is configured so that it can be shared locally, thus giving performance benefits.

Oprofile

A tool for profiling user and kernel level code. It uses the hardware performance counters to sample the program counter every N events.

PDF

Portable document format.

PPC

See *Power PC*.

PPC-64

64 bit implementation of the *PowerPC Architecture*.

proxy

Allows many network devices to connect to the internet using a single IP address. Usually a single server, often acting as a firewall, connects to the internet behind which other network devices connect using the IP address of that server.

RPM

Originally an acronym for Red Hat Package Manager, and RPM file is a packaging format for one or more files used by many Linux systems when installing software programs.

SDK

Software development toolkit. A complete package of tools for application development. The Cell/B.E. SDK includes sample software for the Cell Broadband Engine.

SIMD

Single Instruction Multiple Data. Processing in which a single instruction operates on multiple data elements that make up a vector data-type. Also known as vector processing. This style of programming implements data-level parallelism.

SMP

Symmetric Multiprocessing. This is a multiprocessor computer architecture where two or more identical processors are connected to a single shared main memory.

Tcl

Tool Command Language. An interpreted script language used to develop GUIs, application prototypes, Common Gateway Interface (CGI) scripts, and other scripts. Used as the command language for the Full System Simulator.

TFTP

Trivial File Transfer Protocol. Similar to, but simpler than the Transfer Protocol (FTP) but less capable. Uses UDP as its transport mechanism.

topology

A topology is a configuration of DaCS elements in a system. The topology specifies how the different processing elements in a system are related to each other. DaCS assumes a tree topology: each DE has at most one parent.

x86

Generic name for Intel-based processors.

yaboot

Linux utility which is a boot loader for PowerPC-based hardware.

yum

Yellow dog Updater, Modified. A package manager for RPM-compatible Linux systems.

Index

Printed in USA

SC33-8323-03