Installation Guide
Version 3.0
DRAFT
Third Edition (September 2007)

This edition applies to the Early Release version 3.0 (program number 5724-S84) of the Software Development Kit for Multicore Acceleration and to all subsequent releases and modifications until otherwise indicated in new editions.

This edition replaces SC33-8323-02.

© Copyright International Business Machines Corporation 2006, 2007 (DRAFT). All rights reserved.
US Government Users Restricted Rights – Use, duplication or disclosure restricted by GSA ADP Schedule Contract with IBM Corp.
# Contents

About this publication .......................... v

Chapter 1. Introduction .......................... 1
New in this release .............................. 1
Related products .................................. 1
Supported platforms .............................. 2
Licenses .......................................... 2

Chapter 2. SDK prerequisites ...................... 3
Hardware prerequisites ............................ 3
Software prerequisites ............................ 3

Chapter 3. Quick start guide ....................... 5

Chapter 4. Operating system installation .......... 7
Installing Fedora 7 Linux ........................ 7
Installing Fedora 7 on an X86, X86_64, or PPC64
machine ......................................... 7
BladeCenter QS20 specifics ....................... 7
Managing a BladeCenter QS20 .................... 7
Installing Fedora 7 Linux on a BladeCenter QS20 9
BladeCenter QS21 Specifics ...................... 14
Creating a Linux network installation for
BladeCenter QS21 .................................. 15
Finishing the Linux installation .................. 19

Chapter 5. SDK components ....................... 21
SDK target platforms ............................ 21
SDK directories .................................. 21
RPMs ............................................. 22
SDK component descriptions ..................... 23
YUM groups ...................................... 26
RPMs by component ................................ 28

Chapter 6. Installing and uninstalling the SDK .... 33
Default SDK installation ......................... 33
Choose a product set ............................. 33
Download the SDK files .......................... 35
Prepare for installation ........................ 35
Install the SDK Installer ......................... 36
Start the SDK installation ....................... 36
Post-install configuration ....................... 37
Installing the SDK using the Pirut GUI .......... 39

The cellsdk script ................................ 45
cellsdk script options .......................... 45
cellsdk script verify .............................. 46
Uninstalling the SDK ............................ 46
Uninstalling SDK version 2.1 ..................... 46
Uninstalling SDK version 3.0 ..................... 46
Building the SPU-Isolation component ........... 47
Configuring the Eclipse IDE ..................... 47
Setting up a YUM server for the SDK .......... 48
Troubleshooting the SDK installation .......... 49

Chapter 7. DaCS for Hybrid-x86 configuration .......... 51
Daemon configuration ............................ 51
Topology configuration ........................... 51
Affinity requirements ............................ 53

Chapter 8. Getting support ......................... 55

Appendix A. Accessibility features ................ 57

Appendix B. cellsdk script SDK installation example .......... 59

Appendix C. cellsdk script SDK verify example .......... 67

Appendix D. cellsdk script SDK uninstallation example .......... 71

Appendix E. Known limitations ..................... 77

Notices .......................................... 79
Edition notices ................................... 81
Trademarks ...................................... 81
Terms and conditions ............................ 82

Related documentation .......................... 83

Glossary .......................................... 85

Index ............................................ 89
About this publication

This is an introduction to the Cell Broadband Engine (Cell BE) Software Development Kit for Multicore Acceleration (SDK) version 3.0.

Note: This early release SDK will be refreshed on October 19th with a final 3.0 release that includes an option for IBM® warranted support. This early release version requires Fedora 7. The final release will add support for RHEL 5.1 and any references to RHEL 5.1 in this book can be ignored for now.

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/](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 the IBM developerWorks® Cell BE resource center at [http://www-128.ibm.com/developerworks/power/cell/](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. If you are an experienced user, jump to Chapter 3, “Quick start guide,” on page 5.
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
- Minor enhancements to XL C/C++ compiler
- Addition of single-source XL C/C++ compiler
- GCC toolchain enhancements:
  - GCC C/C++ compilers support infix operations on vector data types
  - GCC support of additional PPU VMX intrinsics
  - GCC performance enhancements
  - Link time estimation of SPU stack consumption
  - Transparent SPE embedding
  - SPE function descriptor support for embedded executables
  - Additional POSIX API support in the SPE runtime library
  - Addition of SPE direct access of PPE address space using _ea qualified data types. This feature is supported by the GCC C++ compiler only.
  - Combined debugger enhancements
- 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 SPU virtual clock and timer services
- Addition of Performance and Debug Tracing tool (PDT and PDTR)
- Updates to Cell Performance Counter, OProfile 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 products together with the SDK components. They provide you with additional capability.

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

**Supported platforms**

Cell Broadband Engine Architecture applications can be developed on these Fedora 7 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](http://www.gnu.org/copyleft/gpl.html)
- Lesser General Public license (LGPL) [http://www.gnu.org/licenses/licenses.html#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.

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

<table>
<thead>
<tr>
<th>System</th>
<th>Recommended minimum configuration</th>
</tr>
</thead>
<tbody>
<tr>
<td>x86 or x86-64</td>
<td>2GHz Pentium® 4 processor</td>
</tr>
<tr>
<td>PowerPC</td>
<td>64-bit PPC with a clock speed of 1.42 GHz. 32-bit PPC platforms are not supported.</td>
</tr>
<tr>
<td>BladeCenter QS20</td>
<td>Revision 31 or greater and minimum firmware level of QA-06.14.0-0F (7.21). See “Checking the firmware version” on page 7</td>
</tr>
<tr>
<td>BladeCenter QS21</td>
<td>Minimum hardware firmware level of QB-01.08.0-00</td>
</tr>
</tbody>
</table>

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, which must be installed before you install the SDK. See Chapter 4, “Operating system installation,” on page 7 for information about how to install Linux.

To install the SDK, see Chapter 3, “Quick start guide,” on page 5, or Chapter 6, “Installing and uninstalling the SDK,” on page 33 for full details.

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

expat

The DaCS for Hybrid-x86 daemon for both X86_64 and the BladeCenter QS20 and BladeCenter QS21 platforms requires the expat XML parsing library. Install expat by typing the following command as root:
yum install expat

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
```
Chapter 3. Quick start guide

This is a brief overview of installation tasks for experienced readers who are eager to get started. You must have a supported operating system installed and have satisfied the hardware and software prerequisites. Many details are skipped. If an issue arises during the quick start installation process, consult the additional material provided in the relevant chapters of this guide for help.

Note: If you are unfamiliar with the terms or procedures in this section, see Chapter 6, “Installing and uninstalling the SDK,” on page 33.

The installation process consists of these steps:
1. Choose a product set appropriate for your operating system and environment.
2. Download the cell-install rpm and the corresponding product set ISO images.
3. Prepare for installation
4. Install the SDK Installer
5. Start the SDK installation
6. Do post-installation configuration

Choose a product set

A product set is a formal grouping of SDK RPMs for a specific environment and operating system. See “Choose a product set” on page 33 to determine the corresponding ISO images to download.

Download the SDK files

Follow the instructions located in “Download the SDK files” on page 35. An installation requires the cell-install RPM and one or more ISO images that contain the SDK.

Prepare for installation

Prepare your system for installation by following these steps:
1. Uninstall any old versions of the SDK. See “Uninstalling the SDK” on page 46.
2. If necessary, install Fedora 7. See Chapter 4, “Operating system installation,” on page 7.
3. Verify that your BladeCenter QS20 or BladeCenter QS21 is at the right firmware level. See “Hardware prerequisites” on page 3.
4. Stop the YUM updates daemon:
   `/etc/init.d/yum-updatesd stop`
5. Some SDK packages have open source versions which must be removed before installing the SDK. These packages are numactl, numactl-devel, blas, blas-debuginfo, blas-devel, oprofile and oprofile-debuginfo. See “Prepare for installation” on page 35.

Note: If you previously added exclude clauses to the /etc/yum.conf file, temporarily remove the clauses to ensure that these RPMs are installed by YUM.
Install the SDK Installer

Install the downloaded cell-install RPM by typing the following command as root:
```
rpm -ivh cell-install-3.0.0-0.0.noarch.rpm
```

Start the SDK install

Install the SDK by typing the following command as root:
```
/opt/cell/cellsdk --iso /tmp/cellsdkiso install
```

In this example, `/tmp/cellsdkiso` is the directory that contains the downloaded ISO images and the cell-install RPM.

Do post-install configuration

After the SDK is installed, you can install optional SDK components. Type the following command to see a list of RPMs available for your operating system and platform:
```
/opt/cell/cellsdk verify
```

See “SDK component descriptions” on page 23 for more information about the contents of the SDK and “RPMs by component” on page 28 for a complete list of RPMs in the SDK.

If you are installing on an IBM BladeCenter QS20 or BladeCenter QS21, add exclude clauses to the `/etc/yum.conf` file to prevent YUM from upgrading these packages to a later version that is not supported by the SDK. The following is an example:
```
exclude=blas kernel numactl oprofile
```
Chapter 4. Operating system installation

This topic provides detailed information about how to install an operating system on supported hardware.

Installing Fedora 7 Linux

This topic describes how to install Fedora 7 Linux.

Before you install the SDK, install a supported operating system.

The following sections describe how to install Fedora 7 Linux:
- “Installing Fedora 7 on an X86, X86_64, or PPC64 machine”

Installing Fedora 7 on an X86, X86_64, or PPC64 machine

This topic describes how to install Fedora 7 on an X86, X86_64, or PPC64 machine.

If you have a suitable workstation or server (see “Hardware prerequisites” on page 5), you can install Fedora 7 Linux from the installation media or downloaded files. Follow the instructions located at the following Web site: [http://docs.fedoraproject.org/install-guide/f7/en_US/](http://docs.fedoraproject.org/install-guide/f7/en_US/)

After you install Fedora 7, install the required prerequisite packages. See “Software prerequisites” on page 3.

Now proceed to “Default SDK installation” on page 33.

BladeCenter QS20 specifics

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

The following chapters give specific details:
- “Managing a BladeCenter QS20”
- “Installing Fedora 7 Linux on a BladeCenter QS20” on page 9

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 8
- “Booting a BladeCenter QS20” on page 8
- “Recovering from a bad firmware boot” on page 8
- “Shutting down and restarting the BladeCenter QS20” on page 9

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 command

```bash
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
```


**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
```

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.

**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 install 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 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 parameter for the install server, before it downloads the secondary stage image Fedora/base/stage2.img from the install server. After mounting the disk image, loader passes control to the Python script anaconda.
The Anaconda installer is the main installation program for Fedora 7 and it performs the remaining steps of the installation, either manually through configuration screens or automatically using the kickstart configuration file. This includes downloading all RPMs, which are selected for installation from the install 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 on page 11](#) shows a typical network installation environment.
Setting up a netboot environment

Copy the netboot image /srv/repos/p/F7/images/netboot/ppc64.img file to the /tftpboot directory of the TFTP/BOOTP server and make sure that it matches the respective entry in /etc/dhcpd.conf. For example:

```bash
host somehostname {
    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:
   ```bash
   > netboot vnc console=hvc0
   ```
4. Select the language you would like to use for the installation.

Figure 1. The network installation environment
Welcome to Fedora

Choose a Language

What language would you like to use during the installation process?

Catalan
Chinese (Simplified)
Chinese (Traditional)
Croatian
Czech
Danish
Dutch
> English

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 image
> FTP
HTTP

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 102f:01b3
eth1 - Unknown device 102f:01b3
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 Use dynamic IP configuration (BOOTP/DHCP) is selected. Fedora 7 determines the host name and domain from the dhcp/bootp server.

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

9. Enter an FTP account name and password.

10. The following is displayed:

    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>, where <target IP>::<vnc session">, <vnc> 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.

Configuring YUM (if required):

This topic describes how to configure YUM on the BladeCenter QS20.

If required, configure the /etc/yum.conf file so that it points to the HTTP server.
You must change the baseurl entry:

```
[main]
cachedir=/var/cache/yum
downloadlevel=2
logfile=/var/log/yum.log
pkgpolicy=newest
distroverpkg=redhat-release
tolerant=1
exactarch=1
retries=20
obsoletes=1
gpgcheck=0
# PUT YOUR REPOS HERE OR IN separate files named file.repo
# in /etc/yum.repos.d

/etc/yum.repos.d/fedora-core.repo
```

```
[base]
name=Fedora Core $releasever - $basearch - Base
#baseurl=http://download.fedora.redhat.com/pub/fedora/linux/ \
core/$releasever/$basearch/os/  <<<< modify baseurl here
baseurl=http://10.64.0.31/
mirrorlist=http://fedora.redhat.com/download/mirrors/fedora-core-$releasever
enabled=1

gpgcheck=0
gpgkey=file:///etc/pki/rpm-gpg/RPM-GPG-KEY-fedora
```

**BladeCenter QS21 Specifics**

These are specifics about installing Linux on the BladeCenter QS21.
Creating a Linux network installation for BladeCenter QS21

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 64-bit POWER-based system with sufficient local disk space for your initial installation.

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

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](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/](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 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
This topic provides an overview of the network boot process.

Figure 4 on page 17 summarizes the flow of information during the network boot process. The file names and IP addresses in the diagram correspond to the examples used in the topics that describe the setup steps.
1. The network boot command is issued from the Slimline Open Firmware command prompt. The command includes the location of the root file system on the NFS server. For example,
   ```
   # boot net root=192.0.2.22:/nfsroot/root_for_192.0.2.10
   ```

2. The blade server broadcasts its MAC address.

3. BOOTP on the DHCP server uses the MAC address to locate the DHCP configuration for the blade server. The DHCP server returns the IP address of the blade server and the name of the zImage to be used for booting the blade server.

4. The blade 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 information from the boot command to locate the root file system on the NFS server.

**Steps for creating a network installation for BladeCenter QS21**

Perform these main steps to set up your network installation.

1. **Install Linux on a POWER-based system.**
2. **Set up a zImage with NFS support.**
3. **Set up the root file system on the network.**
4. **Boot from the network.**

**Step 1: Installing Linux on a POWER-based system:**
Install Fedora 7 as usual.

Perform the following steps to install Linux:
1. Obtain an installation CD/DVD or an ISO image of Fedora 7.
2. Perform the installation as usual. See the documentation that is provided with your distribution for details.

**Step 2: Setting up a zImage with NFS support:**

You need to create a zImage with an initial RAM disk that supports booting from NFS.

Perform the following steps to create a zImage:
1. Download the zImage from the BSC Web site. The full URL is:
   
   http://www.bsc.es/projects/deepcomputing/linuxoncell/cellsimulator/sdk3.0earlyRel/zImage.initrd-2.6.22-5.20070821bsc

2. Copy the zImage to the exported TFTP directory on the TFTP server. For example, assuming that the exported TFTP directory is called tftp_zImages and mounted as /mnt/tftp_zImages enter:
   ```sh
   cp /boot/zImage.initrd-2.6.18-28.el5 \
   /mnt/tftp_zImages/zImage.initrd-2.6.18-28.el5
   ```

**Step 3: Setting 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 blade 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 and exclude /initrd, /proc, and /sys. For example, by entering a command of this form on a command prompt on the NFS server:
   ```sh
   # rsync -avp -e ssh -x --exclude /initd --exclude /proc --exclude /sys \n   root@<POWER-server>:/ /<NFS-dir>
   ```
   where `<NFS-dir>` is the directory to which you want to copy the root file system. For example, enter:
   ```sh
   # rsync -avp -e ssh -x --exclude /initd --exclude /proc --exclude /sys \n   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.
   - 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
     ```conf
     /dev/VolGroup00/LogVol00 / ext3 defaults 1 1
     ```
   to read
     ```conf
     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 /dev/VolGroup00/LogVol01 swap 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.

Step 4: Booting 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.
2. Select Network as the first device of the boot sequence for your BladeCenter. Make this change in the Management Module of the BladeCenter. Next, boot the system.

Finishing the Linux installation

At this point you have Fedora 7 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 5, “SDK components,” on page 21.

Now proceed to Chapter 6, “Installing and uninstalling the SDK,” on page 33.
Chapter 5. 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 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 which are either CBEA hardware or the Full System Simulator when available. Note that the Simulator can also be run on CBEA hardware as a debugging aid. 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 Fedora 7:

Table 2. Fedora 7 platforms

<table>
<thead>
<tr>
<th>Development platform</th>
<th>CBEA execution platform (BladeCenter QS20 or BladeCenter QS21)</th>
<th>Full System Simulator execution platform</th>
</tr>
</thead>
<tbody>
<tr>
<td>X86</td>
<td></td>
<td>✔</td>
</tr>
<tr>
<td>X86_64</td>
<td></td>
<td>✔</td>
</tr>
<tr>
<td>PPC64</td>
<td></td>
<td>✔</td>
</tr>
<tr>
<td>CBEA</td>
<td></td>
<td>✔</td>
</tr>
</tbody>
</table>

SDK directories

The SDK installs files into a number of different directories depending on the host platform and the type of file. 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 is used for level 2 and level 4 components (not prototype code) and follows the Linux file hierarchy standard. In some cases, subdirectories are used to store individual components under include, lib or lib64.
- /opt/ibmcmp contains the XL C/C++ or Fortran single-source compilers.

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.

Under the `/opt/cell/sdk` directory are various subdirectories for parts of the SDK:

• **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 it easier to identify what the RPM is used for. The following table details these conventions:

*Table 3. RPM naming conventions*

<table>
<thead>
<tr>
<th>Convention</th>
<th>Explanation</th>
</tr>
</thead>
<tbody>
<tr>
<td>RPM version number</td>
<td>For IBM-owned code the version number is always 3.0 or 3.0.0 to reflect this version of the SDK</td>
</tr>
<tr>
<td>-source suffix</td>
<td>The RPM contains source code, typically used for examples built using the SDK rather than rpmbuild which uses a SRPM (src.rpm)</td>
</tr>
<tr>
<td>-cross-devel suffix</td>
<td>The RPM contains development code for a cross-build environment (X86 or X86_64) rather than a native one</td>
</tr>
<tr>
<td>-trace suffix</td>
<td>The RPM contains libraries that have been enabled for the IBM Performance and Debugging Tool (PDT)</td>
</tr>
<tr>
<td>-hybrid suffix</td>
<td>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</td>
</tr>
<tr>
<td>cell- prefix</td>
<td>The RPM is oriented for CBEA platforms and can be used to differentiate the RPM from a standard implementation</td>
</tr>
<tr>
<td>ppu- prefix</td>
<td>The RPM contains a PPU-only library</td>
</tr>
<tr>
<td>spu- prefix</td>
<td>The RPM contains a SPU-only library</td>
</tr>
</tbody>
</table>

The SDK RPMs also use a number of different RPM targets. They are listed in the following table:
Table 4. SDK target platforms

<table>
<thead>
<tr>
<th>Architecture/Platform</th>
<th>Explanation</th>
</tr>
</thead>
<tbody>
<tr>
<td>PPC, PPC64</td>
<td>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.</td>
</tr>
<tr>
<td>i386, i686 (X86)</td>
<td>This is native code that executes on a 32-bit X86 platform.</td>
</tr>
<tr>
<td>X86_64</td>
<td>This native code only executes on a 64-bit X86 platform and is used for the hybrid programming model.</td>
</tr>
<tr>
<td>noarch</td>
<td><em>noarch</em> 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.</td>
</tr>
<tr>
<td>src</td>
<td>The source code for some SDK components are available as SRPMs (src.rpm).</td>
</tr>
</tbody>
</table>

### SDK component descriptions

The SDK can be divided into components each of which is at a particular level of development, meaning that some 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 5. Component development levels

<table>
<thead>
<tr>
<th>Development level</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>Prototype-level code where 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.</td>
</tr>
<tr>
<td>2</td>
<td>Beta-level code that is stable.</td>
</tr>
<tr>
<td>3</td>
<td>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</td>
</tr>
<tr>
<td>4</td>
<td>Product-level code that is stable and has been fully tested. This code is also warranted on certain platforms and is fully supported by IBM through standard support channels.</td>
</tr>
</tbody>
</table>

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

Table 6. SDK component list

<table>
<thead>
<tr>
<th>Component</th>
<th>Level</th>
<th>License</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>ALF</td>
<td>4</td>
<td>IBM</td>
<td>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.</td>
</tr>
<tr>
<td>ALF for Hybrid-x86</td>
<td>1</td>
<td>IBM</td>
<td>This version of ALF is directed toward a hybrid computing environment with an X86_64 host and CBEA hardware accelerators.</td>
</tr>
<tr>
<td>Component</td>
<td>Level</td>
<td>License</td>
<td>Description</td>
</tr>
<tr>
<td>-----------------------</td>
<td>-------</td>
<td>---------</td>
<td>-------------------------------------------------------------------------------------------------</td>
</tr>
<tr>
<td>BLAS</td>
<td>4</td>
<td>IBM</td>
<td>BLAS library for single and double precision linear algebra functions.</td>
</tr>
<tr>
<td>Cell Performance</td>
<td>2</td>
<td>IBM</td>
<td>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.</td>
</tr>
<tr>
<td>Counter</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Crash SPU Commands</td>
<td>*</td>
<td>GPL</td>
<td>Crash extension with specific commands for analyzing Cell Broadband Engine Architecture SPU run control state.</td>
</tr>
<tr>
<td>DaCS</td>
<td>4</td>
<td>IBM</td>
<td>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.</td>
</tr>
<tr>
<td>DaCS for Hybrid-x86</td>
<td>4</td>
<td>IBM</td>
<td>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 a X86_64 host and CBEA hardware.</td>
</tr>
<tr>
<td>Documentation</td>
<td>4</td>
<td>IBM</td>
<td>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.</td>
</tr>
<tr>
<td>Examples</td>
<td>3</td>
<td>IBM</td>
<td>This component contains example code including example libraries, demos, and a tutorial.</td>
</tr>
<tr>
<td>FDPR-Pro</td>
<td>4</td>
<td>IBM</td>
<td>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.</td>
</tr>
<tr>
<td>GCC Toolchain</td>
<td>4</td>
<td>GPL</td>
<td>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.</td>
</tr>
<tr>
<td>Hybrid Performance</td>
<td>1</td>
<td>IBM</td>
<td>These tools are designed to assist in using a number of the performance tools in a hybrid system that uses more than one processor architecture in the design. In particular the Cell Broadband Engine is used as an accelerator for a host system with a different architecture.</td>
</tr>
<tr>
<td>Tools</td>
<td>4</td>
<td>IBM</td>
<td>Eclipse-based integrated development environment for the SDK.</td>
</tr>
</tbody>
</table>

Table 6. SDK component list (continued)
<table>
<thead>
<tr>
<th>Component</th>
<th>Level</th>
<th>License</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Kernel</td>
<td>*</td>
<td>GPL</td>
<td>Operating System kernel with Cell Broadband Engine Architecture support.</td>
</tr>
<tr>
<td>LibFFT</td>
<td>1</td>
<td>IBM</td>
<td>This library handles a wide range of 1D and 2D Fast Fourier Transforms.</td>
</tr>
<tr>
<td>LibSPE/LibSPE2</td>
<td>*</td>
<td>LGPL</td>
<td>Low level library that defines the user space API to program for Cell Broadband Engine Architecture applications.</td>
</tr>
<tr>
<td>MASS Library</td>
<td>4</td>
<td>IBM</td>
<td>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 the 32-bit, 64-bit PPU, and SPU libraries are supported.</td>
</tr>
<tr>
<td>netpbm</td>
<td>*</td>
<td>GPL</td>
<td>This graphics bitmap library is used by the Julia example. A cross-devel version is provided in the SDK to facilitate use on X86 and X86_64 platforms.</td>
</tr>
<tr>
<td>numacl</td>
<td>*</td>
<td>LGPL</td>
<td>A library for tuning Non-Uniform Memory Access (NUMA) machines.</td>
</tr>
<tr>
<td>OProfile</td>
<td>*</td>
<td>GPL</td>
<td>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.</td>
</tr>
<tr>
<td>PDT</td>
<td>4</td>
<td>IBM</td>
<td>The Performance Debugging Tool (PDT) provides the ability to trace events of interest during the application execution, and record relevant data related to these events from the SPEs and PPE and the Opteron.</td>
</tr>
<tr>
<td>PDTR</td>
<td>4</td>
<td>IBM</td>
<td>pdtr is a command line tool that reads and post-processes PDT traces. See the man page for usage information.</td>
</tr>
<tr>
<td>Random Number Library</td>
<td>1</td>
<td>IBM</td>
<td>A random number generator library suitable for simulation.</td>
</tr>
<tr>
<td>SIMDMath</td>
<td>4</td>
<td>IBM</td>
<td>A math library that takes advantage of the Single Instruction, Multiple Data (SIMD) instructions in CBEA compliant hardware.</td>
</tr>
<tr>
<td>Simulator</td>
<td>2</td>
<td>IBM</td>
<td>A full system simulation infrastructure and tools for the Cell Broadband Engine™ processor.</td>
</tr>
<tr>
<td>SPU-Isolation</td>
<td>1</td>
<td>IBM</td>
<td>SPU-Isolation provides a build and runtime environment for signing and encrypting SPE applications.</td>
</tr>
<tr>
<td>SPU-Timer</td>
<td>1</td>
<td>IBM</td>
<td>The SPU timer library provides virtual clock and timer services for SPU applications.</td>
</tr>
<tr>
<td>SPU-Timing Tool</td>
<td>2</td>
<td>IBM</td>
<td>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.</td>
</tr>
<tr>
<td>Sysroot Image</td>
<td>*</td>
<td>GPL/LGPL</td>
<td>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.</td>
</tr>
<tr>
<td>Component</td>
<td>Level</td>
<td>License</td>
<td>Description</td>
</tr>
<tr>
<td>------------------</td>
<td>-------</td>
<td>---------</td>
<td>---------------------------------------------------------------------------------------------------------------------------------------------</td>
</tr>
<tr>
<td>XL C/C++</td>
<td>2</td>
<td>IBM</td>
<td>The IBM XL C/C++ Alpha Edition for Multicore Acceleration for Linux, V 0.9 is an advanced, high-performance cross compiler that is tuned for the Cell Broadband Engine Architecture and allows users to compile and link both PPU and SPU code segments with a single compiler invocation.</td>
</tr>
</tbody>
</table>

Notes about the table:

1. The following components are provided for Fedora 7 only either as a convenience or to ensure correct functionality of a dependent component:
   - Netbpm cross-devel package
   - Numactl. Fedora 7 does not have a version that works correctly on CBEA hardware.
2. 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.
3. GPL and LGPL open source components have a development level marked with an asterisk (*). These RPMs are not directly supported by IBM but have been tested with Fedora 7 and the SDK.

**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 in the file `/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 BE applications. It does not contain any development libraries, tools or example code. This group is installed, updated or uninstalled when you pass the `--runtime` option to the `cellsdk` script.

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 these groups using Pirut, or by using the cellsdk script with the --gui option.

The following is a table of each component and the YUM group that contains its RPMs. In general, components are typically only defined in one group. One exception is that a Cell Development Library that 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 7. YUM group for SDK each component

<table>
<thead>
<tr>
<th>Component</th>
<th>YUM group</th>
</tr>
</thead>
<tbody>
<tr>
<td>ALF</td>
<td>Cell Development Libraries</td>
</tr>
<tr>
<td>ALF for Hybrid-x86</td>
<td>Cell Development Libraries</td>
</tr>
<tr>
<td>BLAS</td>
<td>Cell Development Libraries</td>
</tr>
<tr>
<td>Cell Performance Counter</td>
<td>Cell Performance Tools</td>
</tr>
<tr>
<td>Crash SPU Commands</td>
<td>Cell Runtime Environment</td>
</tr>
<tr>
<td>DaCS</td>
<td>Cell Development Libraries</td>
</tr>
<tr>
<td>DaCS for Hybrid-x86</td>
<td>Cell Development Libraries</td>
</tr>
<tr>
<td>Documentation</td>
<td>Cell Development Libraries</td>
</tr>
<tr>
<td>Examples</td>
<td>Cell Programming Examples</td>
</tr>
<tr>
<td>FDPR-Pro</td>
<td>Cell Performance Tools</td>
</tr>
<tr>
<td>GCC Toolchain</td>
<td>Cell Development Tools</td>
</tr>
<tr>
<td>Hybrid Performance Tools</td>
<td>Cell Performance Tools</td>
</tr>
<tr>
<td>IDE</td>
<td>Cell Development Tools</td>
</tr>
<tr>
<td>Kernel</td>
<td>Cell Runtime Environment</td>
</tr>
<tr>
<td>LibFFT</td>
<td>Cell Development Libraries</td>
</tr>
<tr>
<td>LibSPE/LibSPE2</td>
<td>Cell Development Libraries</td>
</tr>
<tr>
<td>MASS Library</td>
<td>Cell Development Libraries</td>
</tr>
<tr>
<td>netpbm</td>
<td>Cell Development Libraries</td>
</tr>
<tr>
<td>numactl</td>
<td>Cell Development Libraries</td>
</tr>
<tr>
<td>OProfile</td>
<td>Cell Performance Tools</td>
</tr>
<tr>
<td>PDT</td>
<td>Cell Performance Tools</td>
</tr>
<tr>
<td>PDTR</td>
<td>Cell Performance Tools</td>
</tr>
<tr>
<td>Random Number Library</td>
<td>Cell Development Libraries</td>
</tr>
<tr>
<td>SIMDMath</td>
<td>Cell Development Libraries</td>
</tr>
<tr>
<td>Simulator</td>
<td>Cell Simulator</td>
</tr>
<tr>
<td>SPU-Isoiation</td>
<td>Cell Development Libraries</td>
</tr>
<tr>
<td>SPU-Timer</td>
<td>Cell Development Libraries</td>
</tr>
<tr>
<td>SPU-Timing Tool</td>
<td>Cell Performance Tools</td>
</tr>
<tr>
<td>Sysroot Image</td>
<td>Cell Simulator</td>
</tr>
<tr>
<td>XL C/C++</td>
<td>Cell Development Tools</td>
</tr>
</tbody>
</table>
After installing the SDK you might want to install, update or uninstall components or even 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.

For information about the individual RPMs per component, see “RPMs by component.”

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

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 a X86 machine, the following command installs the Random Number library:

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

  Because this component is listed as level 1 (prototype code) in “RPMs by component,” 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 21 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 the X86_64 machine, the requisite development
RPMs are needed, include the examples. Because ALF for Hybrid-x86 depends
on ALF for Cell BE, YUM will install those dependencies if they are not already
installed.

Therefore, on the X86_64 machine issue 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 7, ”DaCS for Hybrid-x86
configuration,” on page 51.

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 9. RPMs by component**

<table>
<thead>
<tr>
<th>Component</th>
<th>RPM Name</th>
<th>Install Type</th>
</tr>
</thead>
<tbody>
<tr>
<td>ALF</td>
<td>alf</td>
<td>Default</td>
</tr>
<tr>
<td>ALF</td>
<td>alf-cross-devel</td>
<td>Default</td>
</tr>
<tr>
<td>ALF</td>
<td>alf-debuginfo</td>
<td>Optional</td>
</tr>
<tr>
<td>ALF</td>
<td>alf-devel</td>
<td>Default</td>
</tr>
<tr>
<td>ALF</td>
<td>alf-examples-source</td>
<td>Default</td>
</tr>
<tr>
<td>ALF</td>
<td>alf-trace</td>
<td>Optional</td>
</tr>
<tr>
<td>ALF</td>
<td>alf-trace-devel</td>
<td>Optional</td>
</tr>
<tr>
<td>ALF for Hybrid-x86</td>
<td>alf-hybrid</td>
<td>Optional</td>
</tr>
<tr>
<td>ALF for Hybrid-x86</td>
<td>alf-hybrid-cross-devel</td>
<td>Optional</td>
</tr>
<tr>
<td>ALF for Hybrid-x86</td>
<td>alf-hybrid-devel</td>
<td>Optional</td>
</tr>
<tr>
<td>ALF for Hybrid-x86</td>
<td>alf-hybrid-examples-source</td>
<td>Optional</td>
</tr>
<tr>
<td>ALF for Hybrid-x86</td>
<td>alf-hybrid-trace</td>
<td>Optional</td>
</tr>
<tr>
<td>ALF for Hybrid-x86</td>
<td>alf-hybrid-trace-devel</td>
<td>Optional</td>
</tr>
<tr>
<td>BLAS</td>
<td>blas</td>
<td>Default</td>
</tr>
<tr>
<td>BLAS</td>
<td>blas-cross-devel</td>
<td>Default</td>
</tr>
<tr>
<td>BLAS</td>
<td>blas-devel</td>
<td>Default</td>
</tr>
<tr>
<td>Cell Performance</td>
<td>cellperfctr-tools</td>
<td>Optional</td>
</tr>
<tr>
<td>Counter</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Crash SPU Commands</td>
<td>crash-spu-commands</td>
<td>Optional</td>
</tr>
<tr>
<td>Crash SPU Commands</td>
<td>crash-spu-commands-debuginfo</td>
<td>Optional</td>
</tr>
<tr>
<td>DaCS</td>
<td>dacs</td>
<td>Default</td>
</tr>
<tr>
<td>DaCS</td>
<td>dacs-cross-devel</td>
<td>Default</td>
</tr>
<tr>
<td>DaCS</td>
<td>dacs-debuginfo</td>
<td>Optional</td>
</tr>
<tr>
<td>DaCS</td>
<td>dacs-devel</td>
<td>Default</td>
</tr>
<tr>
<td>Component</td>
<td>RPM Name</td>
<td>Install Type</td>
</tr>
<tr>
<td>--------------------</td>
<td>-------------------</td>
<td>--------------</td>
</tr>
<tr>
<td>DaCS</td>
<td>dacs-trace</td>
<td>Optional</td>
</tr>
<tr>
<td>DaCS</td>
<td>dacs-trace-devel</td>
<td>Optional</td>
</tr>
<tr>
<td>DaCS for Hybrid-x86</td>
<td>dacs-hybrid</td>
<td>Optional</td>
</tr>
<tr>
<td>DaCS for Hybrid-x86</td>
<td>dacs-hybrid-cross-devel</td>
<td>Optional</td>
</tr>
<tr>
<td>DaCS for Hybrid-x86</td>
<td>dacs-hybrid-devel</td>
<td>Optional</td>
</tr>
<tr>
<td>DaCS for Hybrid-x86</td>
<td>dacs-hybrid-trace</td>
<td>Optional</td>
</tr>
<tr>
<td>DaCS for Hybrid-x86</td>
<td>dacs-hybrid-trace-devel</td>
<td>Optional</td>
</tr>
<tr>
<td>Documentation</td>
<td>alfman</td>
<td>Default</td>
</tr>
<tr>
<td>Documentation</td>
<td>cell-documentation</td>
<td>Default</td>
</tr>
<tr>
<td>Documentation</td>
<td>dacsman</td>
<td>Default</td>
</tr>
<tr>
<td>Documentation</td>
<td>libspe2man</td>
<td>Default</td>
</tr>
<tr>
<td>Documentation</td>
<td>simdman</td>
<td>Default</td>
</tr>
<tr>
<td>Examples</td>
<td>cell-buildutils</td>
<td>Default</td>
</tr>
<tr>
<td>Examples</td>
<td>cell-compliance-tests</td>
<td>Optional</td>
</tr>
<tr>
<td>Examples</td>
<td>cell-compliance-tests-cross</td>
<td>Optional</td>
</tr>
<tr>
<td>Examples</td>
<td>cell-compliance-tests-source</td>
<td>Optional</td>
</tr>
<tr>
<td>Examples</td>
<td>cell-demos</td>
<td>Default</td>
</tr>
<tr>
<td>Examples</td>
<td>cell-demos-cross</td>
<td>Default</td>
</tr>
<tr>
<td>Examples</td>
<td>cell-demos-source</td>
<td>Default</td>
</tr>
<tr>
<td>Examples</td>
<td>cell-examples</td>
<td>Default</td>
</tr>
<tr>
<td>Examples</td>
<td>cell-examples-cross</td>
<td>Default</td>
</tr>
<tr>
<td>Examples</td>
<td>cell-examples-source</td>
<td>Default</td>
</tr>
<tr>
<td>Examples</td>
<td>cell-libs</td>
<td>Default</td>
</tr>
<tr>
<td>Examples</td>
<td>cell-libs-cross</td>
<td>Default</td>
</tr>
<tr>
<td>Examples</td>
<td>cell-libs-cross-devel</td>
<td>Default</td>
</tr>
<tr>
<td>Examples</td>
<td>cell-libs-devel</td>
<td>Default</td>
</tr>
<tr>
<td>Examples</td>
<td>cell-libs-source</td>
<td>Default</td>
</tr>
<tr>
<td>Examples</td>
<td>cell-tutorial</td>
<td>Default</td>
</tr>
<tr>
<td>Examples</td>
<td>cell-tutorial-cross</td>
<td>Default</td>
</tr>
<tr>
<td>Examples</td>
<td>cell-tutorial-source</td>
<td>Default</td>
</tr>
<tr>
<td>FDPR-Pro</td>
<td>fdprpro</td>
<td>Optional</td>
</tr>
<tr>
<td>GCC Toolchain</td>
<td>ppu-binutils</td>
<td>Default</td>
</tr>
<tr>
<td>GCC Toolchain</td>
<td>ppu-binutils-debuginfo</td>
<td>Optional</td>
</tr>
<tr>
<td>GCC Toolchain</td>
<td>ppu-gcc</td>
<td>Default</td>
</tr>
<tr>
<td>GCC Toolchain</td>
<td>ppu-gcc-c++</td>
<td>Default</td>
</tr>
<tr>
<td>GCC Toolchain</td>
<td>ppu-gcc-debuginfo</td>
<td>Optional</td>
</tr>
<tr>
<td>GCC Toolchain</td>
<td>ppu-gcc-fortran</td>
<td>Optional</td>
</tr>
<tr>
<td>GCC Toolchain</td>
<td>ppu-gcc-gnat</td>
<td>Optional</td>
</tr>
<tr>
<td>GCC Toolchain</td>
<td>ppu-gdb</td>
<td>Default</td>
</tr>
<tr>
<td>GCC Toolchain</td>
<td>ppu-gdb-debuginfo</td>
<td>Optional</td>
</tr>
</tbody>
</table>
Table 9. RPMs by component (continued)

<table>
<thead>
<tr>
<th>Component</th>
<th>RPM Name</th>
<th>Install Type</th>
</tr>
</thead>
<tbody>
<tr>
<td>GCC Toolchain</td>
<td>ppu-sysroot</td>
<td>Default</td>
</tr>
<tr>
<td>GCC Toolchain</td>
<td>ppu-sysroot64</td>
<td>Default</td>
</tr>
<tr>
<td>GCC Toolchain</td>
<td>spu-binutils</td>
<td>Default</td>
</tr>
<tr>
<td>GCC Toolchain</td>
<td>spu-binutils-debuginfo</td>
<td>Optional</td>
</tr>
<tr>
<td>GCC Toolchain</td>
<td>spu-gcc</td>
<td>Default</td>
</tr>
<tr>
<td>GCC Toolchain</td>
<td>spu-gcc-c++</td>
<td>Default</td>
</tr>
<tr>
<td>GCC Toolchain</td>
<td>spu-gcc-debuginfo</td>
<td>Optional</td>
</tr>
<tr>
<td>GCC Toolchain</td>
<td>spu-gcc-fortran</td>
<td>Optional</td>
</tr>
<tr>
<td>GCC Toolchain</td>
<td>spu-gdb</td>
<td>Default</td>
</tr>
<tr>
<td>GCC Toolchain</td>
<td>spu-gdb-debuginfo</td>
<td>Optional</td>
</tr>
<tr>
<td>GCC Toolchain</td>
<td>spu-newlib</td>
<td>Default</td>
</tr>
<tr>
<td>GCC Toolchain</td>
<td>spu-newlib-debuginfo</td>
<td>Optional</td>
</tr>
<tr>
<td>GCC Toolchain</td>
<td>spu-tools</td>
<td>Optional</td>
</tr>
<tr>
<td>GCC Toolchain</td>
<td>spu-tools-debuginfo</td>
<td>Optional</td>
</tr>
<tr>
<td>Hybrid Performance Tools</td>
<td>cell-perf-hybrid-tools</td>
<td>Optional</td>
</tr>
<tr>
<td>IDE</td>
<td>alf-ide-template</td>
<td>Optional</td>
</tr>
<tr>
<td>IDE</td>
<td>cellide</td>
<td>Optional</td>
</tr>
<tr>
<td>Kernel</td>
<td>kernel</td>
<td>Mandatory</td>
</tr>
<tr>
<td>Kernel</td>
<td>kernel-debuginfo</td>
<td>Optional</td>
</tr>
<tr>
<td>LibFFT</td>
<td>libfft</td>
<td>Optional</td>
</tr>
<tr>
<td>LibFFT</td>
<td>libfft-cross-devel</td>
<td>Optional</td>
</tr>
<tr>
<td>LibFFT</td>
<td>libfft-devel</td>
<td>Optional</td>
</tr>
<tr>
<td>LibFFT</td>
<td>libfft-examples-source</td>
<td>Optional</td>
</tr>
<tr>
<td>LibSPE/LibSPE2</td>
<td>elfspe2</td>
<td>Mandatory</td>
</tr>
<tr>
<td>LibSPE/LibSPE2</td>
<td>libspe</td>
<td>Mandatory</td>
</tr>
<tr>
<td>LibSPE/LibSPE2</td>
<td>libspe2</td>
<td>Mandatory</td>
</tr>
<tr>
<td>LibSPE/LibSPE2</td>
<td>libspe2-adapter-devel</td>
<td>Optional</td>
</tr>
<tr>
<td>LibSPE/LibSPE2</td>
<td>libspe2-cross-devel</td>
<td>Default</td>
</tr>
<tr>
<td>LibSPE/LibSPE2</td>
<td>libspe2-debuginfo</td>
<td>Optional</td>
</tr>
<tr>
<td>LibSPE/LibSPE2</td>
<td>libspe2-devel</td>
<td>Default</td>
</tr>
<tr>
<td>LibSPE/LibSPE2</td>
<td>libspe-debuginfo</td>
<td>Optional</td>
</tr>
<tr>
<td>MASS Library</td>
<td>mass-cross-devel</td>
<td>Default</td>
</tr>
<tr>
<td>MASS Library</td>
<td>ppu-mass-devel</td>
<td>Default</td>
</tr>
<tr>
<td>MASS Library</td>
<td>spu-mass-devel</td>
<td>Default</td>
</tr>
<tr>
<td>netpbm</td>
<td>netpbm-cross-devel</td>
<td>Default</td>
</tr>
<tr>
<td>numactl</td>
<td>numactl</td>
<td>Default</td>
</tr>
<tr>
<td>numactl</td>
<td>numactl-cross-devel</td>
<td>Default</td>
</tr>
<tr>
<td>numactl</td>
<td>numactl-devel</td>
<td>Default</td>
</tr>
<tr>
<td>OProfile</td>
<td>oprofile</td>
<td>Default</td>
</tr>
</tbody>
</table>
Table 9. RPMs by component (continued)

<table>
<thead>
<tr>
<th>Component</th>
<th>RPM Name</th>
<th>Install Type</th>
</tr>
</thead>
<tbody>
<tr>
<td>PDT</td>
<td>pdt</td>
<td>Optional</td>
</tr>
<tr>
<td>PDT</td>
<td>pdt-cross-devel</td>
<td>Optional</td>
</tr>
<tr>
<td>PDT</td>
<td>pdt-devel</td>
<td>Optional</td>
</tr>
<tr>
<td>PDT</td>
<td>pdt-module</td>
<td>Optional</td>
</tr>
<tr>
<td>PDT</td>
<td>trace-cross-devel</td>
<td>Optional</td>
</tr>
<tr>
<td>PDT</td>
<td>trace-devel</td>
<td>Optional</td>
</tr>
<tr>
<td>PDTR</td>
<td>pdtr</td>
<td>Optional</td>
</tr>
<tr>
<td>Random Number Library</td>
<td>libmc-rand-cross-devel</td>
<td>Optional</td>
</tr>
<tr>
<td>Random Number Library</td>
<td>libmc-rand-devel</td>
<td>Optional</td>
</tr>
<tr>
<td>SIMDMath</td>
<td>ppu-simdmath</td>
<td>Default</td>
</tr>
<tr>
<td>SIMDMath</td>
<td>ppu-simdmath-devel</td>
<td>Default</td>
</tr>
<tr>
<td>SIMDMath</td>
<td>simdmath-cross-devel</td>
<td>Default</td>
</tr>
<tr>
<td>SIMDMath</td>
<td>simdmath-debuginfo</td>
<td>Optional</td>
</tr>
<tr>
<td>SIMDMath</td>
<td>spu-simdmath-devel</td>
<td>Default</td>
</tr>
<tr>
<td>Simulator</td>
<td>systemsim-cell</td>
<td>Default</td>
</tr>
<tr>
<td>SPU-Isolation</td>
<td>cell-spu-isolation-cross-devel</td>
<td>Optional</td>
</tr>
<tr>
<td>SPU-Isolation</td>
<td>cell-spu-isolation-devel</td>
<td>Optional</td>
</tr>
<tr>
<td>SPU-Isolation</td>
<td>cell-spu-isolation-emulated-samples</td>
<td>Optional</td>
</tr>
<tr>
<td>SPU-Isolation</td>
<td>cell-spu-isolation-loader</td>
<td>Optional</td>
</tr>
<tr>
<td>SPU-Isolation</td>
<td>cell-spu-isolation-loader-cross</td>
<td>Optional</td>
</tr>
<tr>
<td>SPU-Isolation</td>
<td>cell-spu-isolation-tool</td>
<td>Optional</td>
</tr>
<tr>
<td>SPU-Isolation</td>
<td>cell-spu-isolation-tool-source</td>
<td>Optional</td>
</tr>
<tr>
<td>SPU-Timer</td>
<td>spu-timer-cross-devel</td>
<td>Optional</td>
</tr>
<tr>
<td>SPU-Timer</td>
<td>spu-timer-devel</td>
<td>Optional</td>
</tr>
<tr>
<td>SPU-Timing Tool</td>
<td>cell-spu-timing</td>
<td>Optional</td>
</tr>
<tr>
<td>Sysroot Image</td>
<td>sysroot_image</td>
<td>Default</td>
</tr>
<tr>
<td>XL C/C++</td>
<td>cell-xlc-ssc-cmp</td>
<td>Optional</td>
</tr>
<tr>
<td>XL C/C++</td>
<td>cell-xlc-ssc-lib</td>
<td>Optional</td>
</tr>
<tr>
<td>XL C/C++</td>
<td>cell-xlc-ssc-omp</td>
<td>Optional</td>
</tr>
<tr>
<td>XL C/C++</td>
<td>cell-xlc-ssc-rte</td>
<td>Optional</td>
</tr>
<tr>
<td>XL C/C++</td>
<td>cell-xlc-ssc-rte-lnk</td>
<td>Optional</td>
</tr>
</tbody>
</table>
Chapter 6. Installing and uninstalling the SDK

This topic describes how to add and remove the SDK from your system.

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 35
3. “Prepare for installation” on page 35
4. “Install the SDK Installer” on page 36
5. “Start the SDK installation” on page 36
6. Do “Post-install configuration” on page 37

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.

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 10. Product set group descriptors

<table>
<thead>
<tr>
<th>Descriptor</th>
<th>Options</th>
<th>Rationale</th>
</tr>
</thead>
<tbody>
<tr>
<td>Distributor</td>
<td>IBM or BSC</td>
<td>All GPL or LGPL code is distributed by BSC and is in separate products set from the IBM owned code that is distributed using ISO images from either developerWorks or Passport Advantage.</td>
</tr>
<tr>
<td>Operating system</td>
<td>Fedora 7</td>
<td>The SDK requires different product sets for each supported Operating System.</td>
</tr>
<tr>
<td>License</td>
<td>Warranted product, unwarranted product or early release.</td>
<td>The license is used for example to distribute early release components in a separate product set from other components.</td>
</tr>
</tbody>
</table>

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-3.0.0.0.0.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 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>
<thead>
<tr>
<th>Component</th>
<th>License</th>
<th>Devel-Fedora</th>
<th>Open-Fedora</th>
</tr>
</thead>
<tbody>
<tr>
<td>ALF</td>
<td>IBM</td>
<td></td>
<td></td>
</tr>
<tr>
<td>ALF for Hybrid-x86</td>
<td>IBM</td>
<td></td>
<td></td>
</tr>
<tr>
<td>BLAS</td>
<td>IBM</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Cell Performance Counter</td>
<td>IBM</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Crash SPU Commands</td>
<td>GPL</td>
<td></td>
<td></td>
</tr>
<tr>
<td>DaCS</td>
<td>IBM</td>
<td></td>
<td></td>
</tr>
<tr>
<td>DaCS for Hybrid-x86</td>
<td>IBM</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Documentation</td>
<td>IBM</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Examples</td>
<td>IBM</td>
<td></td>
<td></td>
</tr>
<tr>
<td>FDPR-Pro</td>
<td>IBM</td>
<td></td>
<td></td>
</tr>
<tr>
<td>GCC Toolchain</td>
<td>GPL</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Hybrid Performance Tools</td>
<td>IBM</td>
<td></td>
<td></td>
</tr>
<tr>
<td>IDE</td>
<td>IBM</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Kernel</td>
<td>GPL</td>
<td></td>
<td></td>
</tr>
<tr>
<td>LibFFT</td>
<td>IBM</td>
<td></td>
<td></td>
</tr>
<tr>
<td>LibSPE/LibSPE2</td>
<td>LGPL</td>
<td></td>
<td></td>
</tr>
<tr>
<td>MASS Library</td>
<td>IBM</td>
<td></td>
<td></td>
</tr>
<tr>
<td>netpbm</td>
<td>GPL</td>
<td></td>
<td></td>
</tr>
<tr>
<td>numactl</td>
<td>LGPL</td>
<td></td>
<td></td>
</tr>
<tr>
<td>OPProfile</td>
<td>GPL</td>
<td></td>
<td></td>
</tr>
<tr>
<td>PDT</td>
<td>IBM</td>
<td></td>
<td></td>
</tr>
<tr>
<td>PDTR</td>
<td>IBM</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Random Number Library</td>
<td>IBM</td>
<td></td>
<td></td>
</tr>
<tr>
<td>SIMDMath</td>
<td>IBM</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Simulator</td>
<td>IBM</td>
<td></td>
<td></td>
</tr>
<tr>
<td>SPU-Isolation</td>
<td>IBM</td>
<td></td>
<td></td>
</tr>
<tr>
<td>SPU-Timer</td>
<td>IBM</td>
<td></td>
<td></td>
</tr>
<tr>
<td>SPU-Timing Tool</td>
<td>IBM</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Sysroot Image</td>
<td>GPL/LGPL</td>
<td></td>
<td></td>
</tr>
<tr>
<td>XL C/C++</td>
<td>IBM</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>
Download the SDK files

This topic describes how to download the SDK files needed for installation.

The Barcelona Supercomputing Center (BSC) Web site provides access to the GPL and LGPL open source components of the SDK as RPM packages. The SDK installation program automatically downloads most of these RPMs so that it is not necessary for you to manually download them to install the SDK.

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 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:

<table>
<thead>
<tr>
<th>Product set</th>
<th>ISO name</th>
<th>Location</th>
</tr>
</thead>
</table>

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 necessary, install or upgrade your operating system. See "Operating system installation," on page 7.

2. Verify that your BladeCenter QS20 or BladeCenter QS21 has the right firmware level. See "Hardware prerequisites" on page 3.

3. 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.

4. Some SDK packages have open source versions which must be removed before installing the SDK version. These packages are numactl, numactl-devel, blas, blas-debuginfo, blas-devel, oprofile and oprofile-debuginfo. The easiest way to uninstall these packages is to use RPM with the --no-deps --allmatches options. The following example removes all of these packages:

```
rpm -e --nodeps --allmatches numactl numactl-devel blas blas-debuginfo \
blas-devel oprofile oprofile-debuginfo
```

**Note:** If you previously added exclude clauses to the /etc/yum.conf file for these packages, temporarily remove the clauses to ensure that these RPMs are installed for the SDK.

5. 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 46.

---

### Install the SDK Installer

This topic shows you how to install the SDK Installer.

The SDK Installer requires the `tcl` package. If it is not installed on your system, type the following command:

```
yum install tcl
```

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.

To install the this RPM, type for example the following command:

```
rpm -ivh cell-install-3.0.0.0.0.noarch.rpm
```

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

### 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 48), 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 39.

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 the IBM license agreement for early release (ILAER). Follow the on-screen menu to agree to the license. This IBM license is installed into the /opt/cell/license file for later reference.

3. Answer the license question, then YUM will install the RPM files.

Answer 'y' to the package install question from YUM. The installation will proceed. If you do not agree to the license terms, the installation will stop.

See Appendix B, "cellsdk script SDK installation example," on page 59 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.

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
```

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 packages from regular updates, YUM will not automatically update it when new versions become available. If new versions containing security updates or bug fixes are 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/sdk3.0earlyRel/kernel-2.6.22-5.20070821bsc.ppc64.rpm

Next, install the kernel by typing for example the following command:

```
rpm -ivh --force kernel-2.6.22-5.20070821bsc.ppc64.rpm
```

You must reboot to activate the new kernel.

Installing OProfile

If you are installing on IBM BladeCenter hardware, the SDK version of OProfile must be manually installed. First download the version of OProfile enhanced for the SDK from the Barcelona Supercomputing Center Web site. The OProfile URL is
Install this version of OProfile by typing the following commands as root:

```
rpm -e --nodeps oprofile
rpm -ivh oprofile-0.9.3-4bsc.ppc.rpm
```

## Installing BLAS

If you do not want to use the BLAS (Basic Linear Algebra Subprograms) libraries, you can disregard this prerequisite. If you are installing on the IBM BladeCenter QS20 or BladeCenter QS21, you must manually install the BLAS library that is optimized for Cell BE.

Next, install the BLAS libraries optimized for the SDK as follows.

1. Mount the ISO image. The following command is an example. Substitute the name of the ISO image that corresponds to the product set you are installing.
   ```
   mount -o loop CellSDK-Devel-Fedora_3.0.0.0.iso /mnt
cd /mnt/rpms
   ```

2. Install the RPMs that correspond to the product set you are installing, for example.
   ```
rpm -ivh blas-3.0-6.ppc.rpm
rpm -ivh blas-3.0-6.ppc64.rpm
   ```

3. For development on POWER™ architecture machines, including the BladeCenter QS20 and BladeCenter QS21, install the following RPMs:
   ```
rpm -ivh blas-devel-3.0-6.ppc.rpm
rpm -ivh blas-devel-3.0-6.ppc64.rpm
   ```

4. For development on X86 and X86_64 architecture machines, install the following RPM:
   ```
rpm -ivh blas-cross-devel-3.0-6.ppc.rpm
   ```

5. Unmount the ISO image typing the following commands:
   ```
   cd /
   umount /mnt
   ```

## Adding SDK components

After the SDK is installed, you can install optional packages. Type the following command as root to see a list of packages that are already installed or are available for you to install:

```
/opt/cell/cellsdk verify
```

See Chapter 5, “SDK components,” on page 21 for more information about the contents of the SDK. See “RPMs by component” on page 28 for a list of RPMs that can be installed.

### DaCS for Hybrid-x86

If you installed the DaCS for Hybrid-x86 or the ALF for Hybrid-x86 component, see Chapter 7, “DaCS for Hybrid-x86 configuration,” on page 51.

### The SPU-Isolation RPMs

If you have installed the SPU-Isolation RPMs then you should build the isolation tool, See “Building the SPU-Isolation component” on page 47 for more details.
The Eclipse IDE

If you have installed the Eclipse IDE RPM, see "Configuring the Eclipse IDE" on page 47 for how to complete the install for the IDE.

Restarting automatic updates

Finally, start the YUM updates daemon by typing the following command as root:
/etc/init.d/yum-updatesd start

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 36, 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 Fedora 7 X86 system.

1. This is the first screen you see after starting Pirut. It shows some of the YUM groups defined for the SDK:

2. Tick the checkbox for Cell Development Libraries to have the default Cell Development Libraries installed:
3. Click on Optional Packages and make your selection to add or remove individual RPMs:

4. Tick the checkbox for Cell Development Tools to have the default Cell Development Tools installed:
5. Tick the checkbox for *Cell Performance Tools* to have the default Cell Performance Tools installed:

6. Tick the checkbox for *Cell Programming Examples* to have the default Cell Programming Examples installed:
7. Click on Optional Packages and make your selection to add or remove RPMs:

8. Tick the checkbox for Cell Simulator to install the Full System Simulator and the Fedora 7 sysroot image:
9. Click the *Apply* button. The *Package Selections* window appears:

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

10. 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.

11. Pirut displays the Downloading packages window:

12. After Pirut finishes downloading necessary files, it displays the Updating software window:
13. When the installation finishes, Pirut displays the *Installation success* window:

Click OK.

After you have installed the SDK, close Pirut.

The **cellsdk** script

This topics explains the cellsdk script in detail and its available options.
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:


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

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)

The cellsdk script uses the YUM-based tools as follows:
- no flags starts yum using groupinstall
- --gui start up pirut (install, uninstall) or pup (update). The gui tools take no arguments, so --gui prohibits using --runtime or --auto.
- --runtime only uses the YUM group Cell Runtime Environment
- --auto starts yum using the -y (yes to everything) flag

**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 67](#).

**Uninstalling the SDK**

The following topics describe how to uninstall specific versions of the SDK.

**Uninstalling SDK version 2.1**

This topic describes how to uninstall SDK version 2.1.

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

This topic describes how to uninstall 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. Then, uninstall the SDK by following these steps:

1. Uninstall the SDK RPMs using YUM. Use the uninstall option of the cellsdk script for example ./cellsdk uninstall.
   For an example, see Appendix D, “cellsdk script SDK uninstallation example.” on page 71. 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. Uninstall the SDK Installer RPM using RPM.
   a. Type the following command:
      rpm -e cell-install

3. Clean up the YUM cache.
   a. Type the following commands:
      yum clean all
      rm -rf /var/cache/yum/*

4. 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 37.

5. 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

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

2. Build the samples by typing the following commands as the user root:
   ```shell
   # 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:
   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/](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/ctd/downloads.php](http://www.eclipse.org/ctd/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 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-f7.repo file might contain the following:

```
baseurl1=file:///opt/cell/yum-repos/F7-Trial/x86 ftp://w3.myserver.com/sdk30
baseurl1=file:///opt/cell/yum-repos/F7-Open/x86 ftp://w3.myserver.com/sdk30
```

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 command:

```
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
```

To see more information about YUM, set the options:

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

in /etc/yum.conf. You can 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/cellsdk.log`. Looking at these files might provide helpful information about what went wrong.
Chapter 7. 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 53

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:

/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 on 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 Cell BE BladeCenters. 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>
  </hardware>
  <topology>
    <canreserve he="OC1" ae="CBI"/>
    <canreserve he="OC2" ae="CBI"/>
    <canreserve he="OC3" ae="CBI"/>
    <canreserve he="OC4" ae="CBI"/>
  </topology>
</DaCS_Topology>
```

The <hardware> section identifies the host system with its four cores (OC1-OC4) and the Cell BE BladeCenter (CBI) 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 (CBI) 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="CBI" 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, 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 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>
  </hardware>
  <topology>
    <canreserve he="OC1" ae="CBI"/>
    <canreserve he="OC2" ae="CBI"/>
    <canreserve he="OC3" ae="CBI"/>
    <canreserve he="OC4" ae="CBI"/>
    <canreserve he="OC4" ae="CBE12"/>
  </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:

```
# taskset -p 0x00000001 $$
```

The following example launches a DaCS application using taskset:

```
# taskset 0x00000001 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 8. Getting support


This version of the SDK supersedes all versions of the SDK that were available from alphaWorks.
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 tactiely 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/](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.

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, Linux Kernel, netpbm, numactl, oprofile, crash-spu-command, spu_tools, and SYSROOT image.

Lesser GNU Public License (LGPL) - see http://www.gnu.org/copyleft/lesser.html.
   LGPL licensed components include LibSPE, LibSPE2 and SYSROOT image.

Installation and use of this software requires you to certify you have read the licenses above, and accept their 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-2001: license accepted
cellsdk INFO-2023: Trying to install/update cell-early-license
Loading "installonlyn" plugin
Setting up Install Process
Parsing package install arguments
file:///tmp/sdk/CellSDK-Devel-Fedora/ppc64/repodata/repomd.xml: [Errno 5] Resolving Dependencies
   --> Running transaction check
      ---  Package cell-early-license.noarch 0:3.0.0-0.0 set to be updated

Dependencies Resolved

Package Arch Version Repository Size

-------------------------------
Installing:
   cell-early-license noarch 3.0.0-0.0 CellSDK-Devel-Fedora-ppc64 238 k

Transaction Summary

Install     1 Package(s)
Update      0 Package(s)
Remove      0 Package(s)

Total download size: 238 k
Downloading Packages:
Running Transaction Test
Finished Transaction Test
Transaction Test Succeeded

© Copyright IBM Corp. 2006, 2007 (DRAFT)
Running Transaction
Installing: cell-early-license

Installed: cell-early-license.noarch 0:3.0.0-0.0
Complete!
cellsdk INFO-2024: cell-early-license is installed
International License Agreement for Early Release of Programs

Part 1 - General Terms

THIS INTERNATIONAL LICENSE AGREEMENT FOR EARLY RELEASE OF PROGRAMS ("AGREEMENT") IS A LEGAL AGREEMENT BETWEEN YOU AND IBM. 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.

"Early Release" is a release of a Program that (1) may still be under development (and therefore, is potentially unreliable) or (2) may no longer be under development but has not been made commercially available to users.

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

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" to print, "4" to read non-IBM terms.

cellsdk INFO-2014: License accepted.
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
Setting up Group Process
Package cell-early-license - 3.0.0-0.0.noarch 0:3.0.0-0.0
Package systemsim-cell - 3.0-14.ppc64 already installed and latest version
Resolving Dependencies
--> Running transaction check
--> Package numacl-1-devel.ppc64 0:0.9.10-1 set to be updated
--> Package alman.noarch 0:3.0-4 set to be updated
--> Package cell-utilities.noarch 0:3.0-10 set to be updated
--> Package ppu-simmath.ppc64 0:3.0-3 set to be updated
--> Package libspe2-devel.ppc 0:2.2.0-87 set to be updated
--> Package cell-demos.ppc64 0:3.0-8 set to be updated
--> Package dacs-devel.ppc64 0:3.0-4 set to be updated
--> Package dacsman.noarch 0:3.0-4-18 set to be updated
--> Package cell-tutorial-source.noarch 0:3.0-4 set to be updated
--> Package libspe2.ppc64 0:2.2.0-87 set to be updated
--> Package ppu-simmath-devel.ppc 0:3.0-3 set to be updated
--> Package numacl-devel.ppc 0:0.9.10-1 set to be updated
--> Package numacl.ppc 0:0.9.10-1 set to be updated
--> Package libspe2man.noarch 0:2.2.0-4 set to be updated
--> Package blas-devel.ppc 0:3.1.1-1.fc7 set to be updated
--> Package cell-libs.ppc64 0:3.0-15 set to be updated
--> Package cell-libs-devel.ppc 0:3.0-15 set to be updated
--> Package numactl-devel.ppc 0:3.0-4 set to be updated
--> Package numacl.ppc 0:0.9.10-1 set to be updated
--> Package cell-demos-source.noarch 0:3.0-8 set to be updated
--> Package sysroot_image.noarch 0:3.0-7 set to be updated
--> Package blas.ppc64 0:3.1.1-1.fc7 set to be updated
--> Package blas.ppc 0:3.1.1-1.fc7 set to be updated
--> Package cell-libs.ppc 0:3.0-15 set to be updated
--> Package libspe.ppc 0:1.2.2-2 set to be updated
--> Package ppu-mass-devel.ppc 0:4.5.0-9 set to be updated
Installing:

--- Package dacs.ppc64 0:3.0.0-18 set to be updated
--- Package spu-simdmath-devel.ppc 0:3.0-3 set to be updated
--- Package spu-gcc-c++.ppc 0:4.1.1-100 set to be updated
--- Package alf-devel.ppc64 0:3.0.0-8 set to be updated
--- Package ppu-binutils.ppc 0:2.17.50-31 set to be updated
--- Package ppu-gdb.ppc 0:6.6.50-23 set to be updated
--- Package spu-gcc.ppc 0:4.1.1-4 set to be updated
--- Package spu-binutils.ppc 0:2.17.50-31 set to be updated
--- Package cell-libs-source.noarch 0:3.0-15 set to be updated
--- Package cell-examples.ppc64 0:3.0.0-8 set to be updated
--- Package cell-libs-devel.ppc64 0:4.5.0-9 set to be updated
--- Package libspe2man.ppc 0:1.2.2-2 set to be updated
--- Package ppu-gcc.ppc 0:4.1.1-54 set to be updated
--- Package spu-mass-devel.ppc 0:4.1.1-100 set to be updated
--- Package spu-gcc-c++.ppc 0:4.1.1-100 set to be updated
--- Package ppu-mass-devel.ppc64 0:4.5.0-9 set to be updated
--- Package libspe.ppc64 0:1.2.2-2 set to be updated
--- Package libspe2.ppc 0:1.2.2-2 set to be updated

Dependencies Resolved

<table>
<thead>
<tr>
<th>Package</th>
<th>Arch</th>
<th>Version</th>
<th>Repository</th>
<th>Size</th>
</tr>
</thead>
<tbody>
<tr>
<td>dacs</td>
<td>ppc64</td>
<td>0:3.0.0-18</td>
<td>CellSDK-Devel-Fedora-ppc64</td>
<td>33 k</td>
</tr>
<tr>
<td>alf</td>
<td>ppc</td>
<td>3.0.0-8</td>
<td>CellSDK-Devel-Fedora-ppc64</td>
<td>33 k</td>
</tr>
<tr>
<td>alf</td>
<td>ppc64</td>
<td>3.0.0-8</td>
<td>CellSDK-Devel-Fedora-ppc64</td>
<td>33 k</td>
</tr>
<tr>
<td>alf-devel</td>
<td>ppc</td>
<td>3.0.0-8</td>
<td>CellSDK-Devel-Fedora-ppc64</td>
<td>33 k</td>
</tr>
<tr>
<td>alf-devel</td>
<td>ppc</td>
<td>3.0.0-8</td>
<td>CellSDK-Devel-Fedora-ppc64</td>
<td>33 k</td>
</tr>
<tr>
<td>alf</td>
<td>noarch</td>
<td>3.0.0-6</td>
<td>CellSDK-Devel-Fedora-ppc64</td>
<td>33 k</td>
</tr>
<tr>
<td>alfman</td>
<td>ppc</td>
<td>3.0.0-8</td>
<td>CellSDK-Devel-Fedora-ppc64</td>
<td>33 k</td>
</tr>
<tr>
<td>alf</td>
<td>noarch</td>
<td>3.0.0-6</td>
<td>CellSDK-Devel-Fedora-ppc64</td>
<td>33 k</td>
</tr>
<tr>
<td>blas</td>
<td>ppc64</td>
<td>3.1.1-1.fc7</td>
<td>updates</td>
<td>345 k</td>
</tr>
<tr>
<td>blas</td>
<td>ppc</td>
<td>3.1.1-1.fc7</td>
<td>updates</td>
<td>338 k</td>
</tr>
<tr>
<td>blas-devel</td>
<td>ppc64</td>
<td>3.1.1-1.fc7</td>
<td>updates</td>
<td>173 k</td>
</tr>
<tr>
<td>blas-devel</td>
<td>ppc</td>
<td>3.1.1-1.fc7</td>
<td>updates</td>
<td>165 k</td>
</tr>
<tr>
<td>cell-buildutils</td>
<td>noarch</td>
<td>3.0-10</td>
<td>CellSDK-Devel-Fedora-ppc64</td>
<td>19 k</td>
</tr>
<tr>
<td>cell-demos</td>
<td>ppc64</td>
<td>3.0-8</td>
<td>CellSDK-Devel-Fedora-ppc64</td>
<td>12 M</td>
</tr>
<tr>
<td>cell-demos-source</td>
<td>noarch</td>
<td>3.0-8</td>
<td>CellSDK-Devel-Fedora-ppc64</td>
<td>12 M</td>
</tr>
<tr>
<td>cell-documentation</td>
<td>noarch</td>
<td>3.0-3</td>
<td>CellSDK-Devel-Fedora-ppc64</td>
<td>31 M</td>
</tr>
<tr>
<td>cell-examples</td>
<td>ppc64</td>
<td>3.0-9</td>
<td>CellSDK-Devel-Fedora-ppc64</td>
<td>365 k</td>
</tr>
<tr>
<td>cell-examples-source</td>
<td>noarch</td>
<td>3.0-9</td>
<td>CellSDK-Devel-Fedora-ppc64</td>
<td>325 k</td>
</tr>
<tr>
<td>cell-libs</td>
<td>ppc64</td>
<td>3.0-15</td>
<td>CellSDK-Devel-Fedora-ppc64</td>
<td>83 k</td>
</tr>
<tr>
<td>cell-libs</td>
<td>ppc</td>
<td>3.0-15</td>
<td>CellSDK-Devel-Fedora-ppc64</td>
<td>83 k</td>
</tr>
<tr>
<td>cell-libs-devel</td>
<td>ppc64</td>
<td>3.0-15</td>
<td>CellSDK-Devel-Fedora-ppc64</td>
<td>668 k</td>
</tr>
<tr>
<td>cell-libs-devel</td>
<td>ppc</td>
<td>3.0-15</td>
<td>CellSDK-Devel-Fedora-ppc64</td>
<td>668 k</td>
</tr>
<tr>
<td>cell-libs-source</td>
<td>noarch</td>
<td>3.0-15</td>
<td>CellSDK-Devel-Fedora-ppc64</td>
<td>210 k</td>
</tr>
<tr>
<td>cell-tutorial</td>
<td>ppc</td>
<td>3.0-4</td>
<td>CellSDK-Devel-Fedora-ppc64</td>
<td>48 k</td>
</tr>
<tr>
<td>cell-tutorial-source</td>
<td>noarch</td>
<td>3.0-4</td>
<td>CellSDK-Devel-Fedora-ppc64</td>
<td>13 k</td>
</tr>
<tr>
<td>dacs</td>
<td>ppc64</td>
<td>3.0-0-18</td>
<td>CellSDK-Devel-Fedora-ppc64</td>
<td>24 k</td>
</tr>
<tr>
<td>dacs-devel</td>
<td>ppc64</td>
<td>3.0-0-18</td>
<td>CellSDK-Devel-Fedora-ppc64</td>
<td>24 k</td>
</tr>
<tr>
<td>dacsman</td>
<td>noarch</td>
<td>3.0-4</td>
<td>CellSDK-Devel-Fedora-ppc64</td>
<td>59 k</td>
</tr>
<tr>
<td>libspe</td>
<td>ppc</td>
<td>1.2.2-2</td>
<td>CellSDK-Open-Fedora-ppc64</td>
<td>31 k</td>
</tr>
<tr>
<td>libspe</td>
<td>ppc64</td>
<td>1.2.2-2</td>
<td>CellSDK-Open-Fedora-ppc64</td>
<td>31 k</td>
</tr>
<tr>
<td>libspe2</td>
<td>ppc64</td>
<td>2.2.0-87</td>
<td>CellSDK-Open-Fedora-ppc64</td>
<td>35 k</td>
</tr>
<tr>
<td>libspe2</td>
<td>ppc</td>
<td>2.2.0-87</td>
<td>CellSDK-Open-Fedora-ppc64</td>
<td>35 k</td>
</tr>
<tr>
<td>libspe2-devel</td>
<td>ppc64</td>
<td>2.2.0-87</td>
<td>CellSDK-Open-Fedora-ppc64</td>
<td>7.3 k</td>
</tr>
<tr>
<td>libspe2-devel</td>
<td>ppc</td>
<td>2.2.0-87</td>
<td>CellSDK-Open-Fedora-ppc64</td>
<td>7.3 k</td>
</tr>
<tr>
<td>libspe2-man</td>
<td>noarch</td>
<td>2.2.0-4</td>
<td>CellSDK-Devel-Fedora-ppc64</td>
<td>52 k</td>
</tr>
</tbody>
</table>
numactl ppc 0.9.10-1 CellSDK-Open-Fedora-ppc64 82 k
numactl ppc64 0.9.10-1 CellSDK-Open-Fedora-ppc64 70 k
numactl-devel ppc 0.9.10-1 CellSDK-Open-Fedora-ppc64 15 k
numactl-devel ppc64 0.9.10-1 CellSDK-Open-Fedora-ppc64 15 k
ppu-binutils ppc 2.17.50-31 CellSDK-Open-Fedora-ppc64 8.2 M
ppu-gcc ppc 4.1.1-54 CellSDK-Open-Fedora-ppc64 4.2 M
ppu-gcc-c++ ppc 4.1.1-54 CellSDK-Open-Fedora-ppc64 17 M
ppu-gdb ppc 6.6.50-23 CellSDK-Open-Fedora-ppc64 3.1 M
ppu-mass-devel ppc 4.5.0-9 CellSDK-Devel-Fedora-ppc64 257 k
ppu-mass-devel ppc64 4.5.0-9 CellSDK-Devel-Fedora-ppc64 265 k
ppu-simdmath ppc64 3.0-3 CellSDK-Devel-Fedora-ppc64 31 k
ppu-simdmath ppc 3.0-3 CellSDK-Devel-Fedora-ppc64 26 k
ppu-simdmath-devel ppc 3.0-3 CellSDK-Devel-Fedora-ppc64 93 k
ppu-simdmath-devel ppc64 3.0-3 CellSDK-Devel-Fedora-ppc64 98 k
simdman noarch 3.0-4 CellSDK-Devel-Fedora-ppc64 148 k
spu-binutils ppc 2.17.50-31 CellSDK-Open-Fedora-ppc64 3.2 M
spu-gcc ppc 4.1.1-100 CellSDK-Open-Fedora-ppc64 2.7 M
spu-gcc-c++ ppc 4.1.1-100 CellSDK-Open-Fedora-ppc64 13 M
spu-gdb ppc 6.6.50-10 CellSDK-Open-Fedora-ppc64 2.3 M
spu-mass-devel ppc 4.5.0-9 CellSDK-Devel-Fedora-ppc64 96 k
spu-newlib ppc 1.15.0-76 CellSDK-Open-Fedora-ppc64 1.3 M
spu-simdmath-devel ppc 3.0-3 CellSDK-Devel-Fedora-ppc64 179 k
sysroot_image noarch 3.0-7 CellSDK-Open-Fedora-ppc64 271 M

Transaction Summary
******************************************************************************
Install 56 Package(s)
Update 0 Package(s)
Remove 0 Package(s)

Total download size: 387 M
Is this ok [y/N]: Downloading Packages:
Running Transaction Test
Finished Transaction Test
Transaction Test Succeeded
Running Transaction
Installing: libspe2
Installing: libspe2
Installing: alf
Installing: ppu-simdmath
Installing: blas
Installing: numacl
Installing: numacl
Installing: cell-buildutils
Installing: cell-libs
Installing: cell-libs
Installing: spu-binutils
Installing: spu-gcc
Installing: ppu-gcc
Installing: dacs
Installing: blas
Installing: numacl-devel
Installing: alfman
Installing: ppu-simdmath
Installing: libspe2-devel
Installing: cell-demos
Installing: dacsman
Installing: dacs-devel
Installing: cell-tutorial-source
Installing: ppu-simdmath-devel
Installing: blas-devel
Installing: cell-libs-devel
Installing: numacl-devel
Installing: libspe2man
Installing: blas-devel
Installing: cell-libs-devel
Installing: cell-documentation

62  SDK 3.0 Installation Guide
Installing: ppu-simdmath-devel #-------------------------- [32/56]
Installing: libspe2-devel #-------------------------- [33/56]
Installing: cell-demos-source #-------------------------- [34/56]
Installing: sysroot_image #-------------------------- [35/56]
Installing: libspe #-------------------------- [36/56]
Installing: ppu-mass-devel #-------------------------- [37/56]
Installing: spu-simdmath-devel #-------------------------- [38/56]
Installing: spu-gcc-c++ #-------------------------- [39/56]
Installing: elf-devel #-------------------------- [40/56]
Installing: ppu-binutils #-------------------------- [41/56]
Installing: ppu-gdb #-------------------------- [42/56]
Installing: simdman #-------------------------- [43/56]
Installing: spu-newlib #-------------------------- [44/56]
Installing: ppu-mass-devel #-------------------------- [45/56]
Installing: libspe #-------------------------- [46/56]
Installing: spu-gdb #-------------------------- [47/56]
Installing: cell-libs-source #-------------------------- [48/56]
Installing: cell-libs #-------------------------- [49/56]
Installing: spu-mass-devel #-------------------------- [50/56]
Installing: alf-examples-source #-------------------------- [51/56]
Installing: elf-devel #-------------------------- [52/56]
Installing: ppu-gcc-c++ #-------------------------- [53/56]
Installing: cell-tutorial #-------------------------- [54/56]
Installing: elf #-------------------------- [55/56]
Installing: cell-examples-source #-------------------------- [56/56]

Installed: alf.ppc 0:3.0.0-8 alf.ppc64 0:3.0.0-8 alf-devel.ppc64 0:3.0.0-8
alf-devel.ppc 0:3.0.0-8 alf-examples-source.noarch 0:3.0.0-6 alfman.noarch
0:3.0-4 blas.ppc64 0:3.1.1-1.fc7 blas.ppc 0:3.1.1-1.fc7
blas-devel.ppc64 0:3.1.1-1.fc7 blas-devel.ppc 0:3.1.1-1.fc7
cell-buildutils.noarch 0:3.0-10 cell-demos.ppc64
0:3.0-8 cell-demos-source.noarch 0:3.0-8 cell-documentation.noarch 0:3.0-3
cell-examples.ppc64 0:3.0-9 cell-examples-source.noarch 0:3.0-9 cell-libs.ppc64
0:3.0-15 cell-libs.ppc 0:3.0-15 cell-libs-devel.ppc 0:3.0-15 cell-libs-devel.ppc64
0:3.0-15 cell-libs-source.noarch 0:3.0-15 cell-tutorial.ppc 0:3.0-4
cell-tutorial-source.noarch 0:3.0-4 dacs.ppc64 0:3.0.0-18
dacs-devel.ppc64 0:3.0.0-18 dacsman.noarch 0:3.0-4
libspe.ppc 0:1.2.2-2 libspe.ppc64 0:1.2.2-2
libspe2.ppc64 0:2.2.0-87 libspe2.ppc 0:2.2.0-87
libspe2-devel.ppc64 0:2.2.0-87 libspe2-devel.noarch 0:2.2.0-4 numactl.ppc
0:0.9.10-1 numactl.ppc64 0:0.9.10-1 numactl-devel.ppc 0:0.9.10-1
numactl-devel.ppc 0:0.9.10-1 ppu-binutils.ppc 0:2.17.50-31
ppu-gcc.ppc 0:4.1.1-54 ppu-gcc-c++.ppc 0:4.1.1-54 ppu-gdb.ppc 0:6.6.50-23
ppu-mass-devel.ppc 0:4.5.0-9 ppu-mass-devel.ppc64 0:4.5.0-9 ppu-simdmath.ppc64
0:3.0-3 ppu-simdmath.ppc 0:3.0-3 ppu-simdmath-devel.ppc 0:3.0-3
ppu-simdmath-devel.ppc64 0:3.0-3 simdman.noarch 0:3.0-4
spu-binutils.ppc 0:2.17.50-31 spu-gcc.ppc 0:4.1.1-100
spu-gcc-c++.ppc 0:4.1.1-100 spu-gdb.ppc 0:6.6.50-10
spu-mass-devel.ppc 0:4.5.0-9 spu-newlib.ppc 0:1.15.0-76
spu-simdmath-devel.ppc 0:3.0-3 sysroot_image.noarch 0:3.0-7
Complete!

cellsdk INFO-2022: Copying rpms to install in the simulator sysroot

< verbose output from wget and file copies as cellsdk tries to find the
ppc versions of files and copy them into /tmp/cellsdk/rpms >

...  

< more verbose output from wget and file copies as cellsdk tries to find the
cellsdk versions of blas, numactl and oprofile and copy them into
/tmp/cellsdk/openSrc >

cellsdk INFO-2027: Installing cellsdk versions of openSource rpms

Loading "installonlyn" plugin
Setting up Local Package Process
Examining /tmp/cellsdk/openSrc/blas-3.0-6.ppc64.rpm: blas - 3.0-6.ppc64
Examining /tmp/cellsdk/openSrc/numactl-0.9.10-1.ppc64.rpm:
  numactl - 0.9.10-1.ppc64
Examining /tmp/cellsdk/openSrc/numactl-devel-0.9.10-1.ppc64.rpm:
  numactl-devel - 0.9.10-1.ppc64
Examining /tmp/cellsdk/openSrc/blas-devel-3.0-6.ppc64.rpm: blas-devel - 3.0-6.ppc64
Marking /tmp/cellsdk/openSrc/blas-3.0-6.ppc64.rpm to be installed
Marking /tmp/cellsdk/openSrc/numactl-0.9.10-1.ppc64.rpm to be installed
Marking /tmp/cellsdk/openSrc/blas-devel-3.0-6.ppc64.rpm to be installed
Marking /tmp/cellsdk/openSrc/blas-3.0-6.ppc.rpm to be installed
Marking /tmp/cellsdk/openSrc/numactl-devel-0.9.10-1.ppc64.rpm to be installed
Marking /tmp/cellsdk/openSrc/blas-devel-3.0-6.ppc.rpm to be installed
Marking /tmp/cellsdk/openSrc/blas-3.0-6.ppc.rpm to be installed
Marking /tmp/cellsdk/openSrc/numactl-0.9.10-1.ppc.rpm to be installed
Marking /tmp/cellsdk/openSrc/blas-devel-3.0-6.ppc64.rpm to be installed
Marking /tmp/cellsdk/openSrc/blas-devel-3.0-6.ppc.rpm to be installed
Marking /tmp/cellsdk/openSrc/numactl-devel-0.9.10-1.ppc.rpm to be installed
Marking /tmp/cellsdk/openSrc/blas.devel-3.0-6.ppc.rpm to be installed
Marking /tmp/cellsdk/openSrc/blas.devel-3.0-6.ppc64.rpm to be installed
Marking /tmp/cellsdk/openSrc/blas.devel-3.0-6.ppc.rpm to be installed
Marking /tmp/cellsdk/openSrc/blas.devel-3.0-6.ppc64.rpm to be installed

Resolving Dependencies
--- Running transaction check
---- 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.ppc64 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

Dependencies Resolved

=============================================================================  
<table>
<thead>
<tr>
<th>Package</th>
<th>Arch</th>
<th>Version</th>
<th>Repository</th>
<th>Size</th>
</tr>
</thead>
<tbody>
<tr>
<td>Installing:</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>blas</td>
<td>ppc</td>
<td>3.0-6</td>
<td>/tmp/cellsdk/openSrc/blas-3.0-6.ppc.rpm</td>
<td>1.3  M</td>
</tr>
<tr>
<td>blas</td>
<td>ppc64</td>
<td>3.0-6</td>
<td>/tmp/cellsdk/openSrc/blas-3.0-6.ppc64.rpm</td>
<td>1.4  M</td>
</tr>
<tr>
<td>blas-devel</td>
<td>ppc64</td>
<td>3.0-6</td>
<td>/tmp/cellsdk/openSrc/blas-devel-3.0-6.ppc64.rpm</td>
<td>0.0  k</td>
</tr>
<tr>
<td>blas-devel</td>
<td>pcc</td>
<td>3.0-6</td>
<td>/tmp/cellsdk/openSrc/blas-devel-3.0-6.ppc.rpm</td>
<td>741  k</td>
</tr>
<tr>
<td>numactl</td>
<td>ppc64</td>
<td>0.9.10-1</td>
<td>/tmp/cellsdk/openSrc/numactl-0.9.10-1.ppc64.rpm</td>
<td>180  k</td>
</tr>
<tr>
<td>numactl</td>
<td>ppc</td>
<td>0.9.10-1</td>
<td>/tmp/cellsdk/openSrc/numactl-0.9.10-1.ppc.rpm</td>
<td>186  k</td>
</tr>
<tr>
<td>numactl-devel</td>
<td>ppc64</td>
<td>0.9.10-1</td>
<td>/tmp/cellsdk/openSrc/numactl-devel-0.9.10-1.ppc64.rpm</td>
<td>12   k</td>
</tr>
<tr>
<td>numactl-devel</td>
<td>ppp</td>
<td>0.9.10-1</td>
<td>/tmp/cellsdk/openSrc/numactl-devel-0.9.10-1.ppc.rpm</td>
<td>12   k</td>
</tr>
</tbody>
</table>

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

Total download size: 3.8 M
Running Transaction Test
Finished Transaction Test
Transaction Test Succeeded
Running Transaction
Installing: blas
Installing: numactl
Installing: numactl
Installing: numactl-devel
Installing: blas-devel
Installing: blas
Installing: blas-devel
Installing: numactl-devel

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!
cellsdk INFO-2007: Please run '/opt/cell/cellsdk_sync_simulator install'
cellsdk INFO-2007: to install IBM-licensed rpms into the simulator.
Appendix C. cellsdk script SDK verify example

The following text is an example of the output produced on a PPC64 architecture Fedora 7 system:

The following output is a result of typing the command:

```
./cellsdk verify
```

Here is the output:

cellsdk logs to /var/log/cellsdk.log

```
repository=CellSDK-Devel-Fedora-ppc64
  default CellDevelopmentLibraries alf-devel-3.0.0-8.ppc
  default CellDevelopmentLibraries alf-devel-3.0.0-8.ppc64
  default CellDevelopmentLibraries alfman-3.0-4.noarch
  default CellDevelopmentLibraries blas-devel-3.0-6.ppc
  default CellDevelopmentLibraries blas-devel-3.0-6.ppc64
  default CellDevelopmentLibraries dacs-devel-3.0-0-18.ppc64
  default CellDevelopmentLibraries dacsman-3-0-4.noarch
  default CellDevelopmentLibraries libspe2man-2.2-0-4.noarch
  default CellDevelopmentLibraries ppu-mass-devel-4.5-0-9.ppc
  default CellDevelopmentLibraries ppu-mass-devel-4.5-0-9.ppc64
  default CellDevelopmentLibraries ppu-simdmath-devel-3.0-3.ppc
  default CellDevelopmentLibraries ppu-simdmath-devel-3.0-3.ppc64
  default CellDevelopmentLibraries simdman-3.0-4.noarch
  default CellDevelopmentLibraries spu-mass-devel-4.5-0-9.ppc
  default CellDevelopmentLibraries spu-mass-devel-4.5-0-9.ppc64
  default CellDevelopmentLibraries spu-simdmath-devel-3.0-3.ppc
  default CellDevelopmentLibraries spu-simdmath-devel-3.0-3.ppc64
  default CellProgrammingExamples alf-examples-source-3.0-0-6.noarch
  default CellProgrammingExamples cell-buildutils-3-0-10.noarch
  default CellProgrammingExamples cell-demos-3.0-8.ppc64
  default CellProgrammingExamples cell-demos-source-3.0-8.noarch
  default CellProgrammingExamples cell-examples-3.0-9.ppc64
  default CellProgrammingExamples cell-examples-source-3.0-9.noarch
  default CellProgrammingExamples cell-libs-3-0-15.ppc
  default CellProgrammingExamples cell-libs-3-0-15.ppc64
  default CellProgrammingExamples cell-libs-devel-3.0-15.ppc
  default CellProgrammingExamples cell-libs-devel-3.0-15.ppc64
  default CellProgrammingExamples cell-libs-source-3.0-15.noarch
  default CellProgrammingExamples cell-tutorial-3.0-4.ppc
  default CellProgrammingExamples cell-tutorial-source-3.0-4.noarch
  default CellRuntimeEnvironment alf-3.0-0-8.ppc
  default CellRuntimeEnvironment alf-3.0-0-8.ppc64
  default CellRuntimeEnvironment blas-3-0-6.ppc
  default CellRuntimeEnvironment blas-3-0-6.ppc64
  default CellRuntimeEnvironment dacs-3-0-0-18.ppc64
  default CellRuntimeEnvironment ppu-simdmath-3.0-3.ppc
  default CellRuntimeEnvironment ppu-simdmath-3.0-3.ppc64
  default CellSimulator systemsim-cell-3.0-14.ppc64
  mandatory CellDevelopmentLibraries cell-documentation-3.0-3.noarch
  mandatory CellRuntimeEnvironment cell-early-license-3.0-0-0.noarch
  optional CellDevelopmentLibraries alf-hybrid-devel not installed
  optional CellDevelopmentLibraries alf-hybrid-trace-devel not installed
  optional CellDevelopmentLibraries alf-trace-devel not installed
  optional CellDevelopmentLibraries dacs-hybrid-devel not installed
  optional CellDevelopmentLibraries dacs-hybrid-trace-devel not installed
  optional CellDevelopmentLibraries dacs-trace-devel not installed
  optional CellDevelopmentLibraries libfft-devel not installed
  optional CellDevelopmentLibraries libmc-rand-devel not installed
  optional CellDevelopmentLibraries pdt-devel not installed
  optional CellDevelopmentLibraries spu-timer-devel not installed
```

© Copyright IBM Corp. 2006, 2007 (DRAFT) 67
optional CellDevelopmentLibraries trace-devel not installed
optional CellDevelopmentLibraries alf-ide-template not installed
optional CellDevelopmentLibraries cellide not installed
optional CellDevelopmentLibraries cell-spu-isolation-tool not installed

optional CellDevelopmentTools cell-spu-isolation-tool-source not installed
optional CellDevelopmentTools cell-xlc-ssc-cmp not installed
optional CellDevelopmentTools cell-xlc-ssc-help not installed
optional CellDevelopmentTools cell-xlc-ssc-lib not installed
optional CellDevelopmentTools cell-xlc-ssc-omp not installed
optional CellDevelopmentTools cell-xlc-ssc-rte not installed
optional CellDevelopmentTools cell-xlc-ssc-rte-lnk not installed
optional CellDevelopmentTools cell-spu-isolation-loader not installed
optional CellDevelopmentTools cell-spu-isolation-tool not installed
optional CellDevelopmentTools fdprop not installed
optional CellDevelopmentTools pdt not installed
optional CellDevelopmentTools pdtr not installed
optional CellProgrammingExamples alf-hybrid-examples-source not installed
optional CellProgrammingExamples cell-compliance-tests not installed
optional CellProgrammingExamples cell-compliance-tests-source not installed

optional CellProgrammingExamples cell-spu-isolation-emulated-samples not installed
optional CellProgrammingExamples libfft-examples-source not installed
optional CellRuntimeEnvironment alf-debuginfo not installed
optional CellRuntimeEnvironment alf-hybrid not installed
optional CellRuntimeEnvironment alf-hybrid-trace not installed
optional CellRuntimeEnvironment alf-trace not installed
optional CellRuntimeEnvironment cell-spu-isolation-loader not installed
optional CellRuntimeEnvironment dacs-debuginfo not installed
optional CellRuntimeEnvironment dacs-hybrid not installed
optional CellRuntimeEnvironment dacs-hybrid-trace not installed
optional CellRuntimeEnvironment dacs-trace not installed
optional CellRuntimeEnvironment libfft not installed
optional CellRuntimeEnvironment simdmath-debuginfo not installed

repository=CellSDK-Open-Fedora-ppc64

default CellDevelopmentLibraries libspe2-devel-2.2.0-87.ppc

default CellDevelopmentLibraries libspe2-devel-2.2.0-87.ppc64

default CellDevelopmentLibraries numactl-devel-0.9.10-1.ppc

default CellDevelopmentLibraries numactl-devel-0.9.10-1.ppc64

default CellDevelopmentTools ppu-binutils-2.17.50-31.ppc

default CellDevelopmentTools ppu-gcc-4.1.1-54.ppc

default CellDevelopmentTools ppu-gcc-c++-4.1.1-54.ppc

default CellDevelopmentTools ppu-gcc-6.6.50-23.ppc

default CellDevelopmentTools ppu-gcc-6.6.50-10.ppc

default CellDevelopmentTools ppu-gcc-4.1.1-100.ppc

default CellDevelopmentTools ppu-gcc-c++-4.1.1-100.ppc

default CellDevelopmentTools ppu-gdb-6.6.50-10.ppc

default CellDevelopmentTools numactl-0.9.10-1.ppc

default CellRuntimeEnvironment numactl-0.9.10-1.ppc64

default CellRuntimeEnvironment numactl-0.9.10-1.ppc64

default CellSimulator sysroot_image-3.0-7.noarch

mandatory CellRuntimeEnvironment libspe-1.2.2-2.ppc

mandatory CellRuntimeEnvironment libspe-1.2.2-2.ppc64

mandatory CellRuntimeEnvironment libspe2-1.2.2-2.ppc64

mandatory CellRuntimeEnvironment libspe2-2.2.0-87.ppc

mandatory CellRuntimeEnvironment libspe2-2.2.0-87.ppc64

optional CellDevelopmentTools libspe2-2.2.0-adabinding-devel not installed

optional CellDevelopmentTools libspe2-2.2.0-2.2.0-87.ppc

optional CellDevelopmentTools libspe2-2.2.0-2.2.0-87.ppc64

optional CellDevelopmentTools libspe2-2.2.0-2.2.0-87.ppc

optional CellDevelopmentTools libspe2-2.2.0-2.2.0-87.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 CellDevelopmentTools spu-tools-debuginfo not installed
<table>
<thead>
<tr>
<th>optional</th>
<th>CellDevelopmentTools</th>
<th>spu-tools not installed</th>
</tr>
</thead>
<tbody>
<tr>
<td>optional</td>
<td>CellRuntimeEnvironment</td>
<td>libspe2-debuginfo not installed</td>
</tr>
<tr>
<td>optional</td>
<td>CellRuntimeEnvironment</td>
<td>libspe-debuginfo not installed</td>
</tr>
</tbody>
</table>
Appendix D. cellsdk script SDK uninstallation example

This is an example of using the cellsdk script to uninstall the SDK.

The following output is the result of typing the commands:

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

Here is the output:
cellsdk logs to /var/log/cellsdk.log
cellsdk INFO-2019: yum groupproceed 'Cell Runtime Environment'
 'Cell Development Tools' 'Cell Development Libraries'
 'Cell Programming Examples' 'Cell Simulator' 'Cell Performance Tools'
Setting up Group Process
Resolving Dependencies
--> Running transaction check
---> Package systemsim-cell.ppc64 0:3.0-14 set to be erased
---> Package alfanman.noarch 0:3.0-4 set to be erased
---> Package cell-buildutils.noarch 0:3.0-10 set to be erased
---> Package ppu-simdmath.ppc64 0:3.0-3 set to be erased
---> Package cell-early-license.noarch 0:3.0.0-0.0 set to be erased
---> Package libspe2-devel.ppc 0:2.2.0-87 set to be erased
---> Package cell-demos.ppc64 0:3.0-8 set to be erased
---> Package dacs-devel.ppc64 0:3.0.0-18 set to be erased
---> Package cell-tutorial-source.noarch 0:3.0-4 set to be erased
---> Package libspe2.ppc64 0:2.2.0-87 set to be erased
---> Package ppu-simdmath-devel.ppc 0:3.0-3 set to be erased
---> Package cell-libs-devel.ppc 0:3.0-15 set to be erased
---> Package dacsman.noarch 0:3.0-4 set to be erased
---> Package libspe2man.noarch 0:2.2.0-4 set to be erased
---> Package cell-libs.ppc64 0:3.0-15 set to be erased
---> Package cell-libs-devel.ppc64 0:3.0-15 set to be erased
---> Package cell-documentation.noarch 0:3.0-3 set to be erased
---> Package ppu-simdmath-devel.ppc64 0:3.0-3 set to be erased
---> Package libspe2-devel.ppc64 0:2.2.0-87 set to be erased
---> Package cell-demos-source.noarch 0:3.0-8 set to be erased
---> Package sysroot_image.noarch 0:3.0-7 set to be erased
---> Package blas.ppc 0:3.1.1-1.fc7 set to be erased
---> Package cell-libs.ppc 0:3.0-15 set to be erased
---> Package libspe.ppc 0:1.2.2-2 set to be erased
---> Package ppu-mass-devel.ppc 0:4.5.0-9 set to be erased
---> Package dacs.ppc64 0:3.0.0-18 set to be erased
---> Package spu-simdmath-devel.ppc 0:3.0-3 set to be erased
---> Package spu-gcc-c++.ppc 0:4.1.1-100 set to be erased
---> Package libspe.ppc64 0:1.2.2-2 set to be erased
---> Package ppu-gcc.ppc 0:4.1.1-54 set to be erased
---> Package spu-gdb.ppc 0:6.6.50-10 set to be erased
---> Package spu-gcc.ppc 0:4.1.1-100 set to be erased
---> Package ppu-gcc.ppc 0:4.1.1-54 set to be erased
---> Package ppu-gdb.ppc 0:6.6.50-23 set to be erased
---> Package simdman.noarch 0:3.0-4 set to be erased
---> Package spu-newlib.ppc 0:1.15.0-76 set to be erased
---> Package ppu-mass-devel.ppc64 0:4.5.0-9 set to be erased
---> Package libspe.ppc64 0:1.2.2-2 set to be erased
---> Package ppu-gcc.ppc 0:4.1.1-54 set to be erased
---> Package spu-gdb.ppc 0:6.6.50-10 set to be erased
---> Package spu-gcc.ppc 0:4.1.1-100 set to be erased
---> Package spu-gcc.ppc 0:4.1.1-54 set to be erased
---> Package spu-binutils.ppc 0:2.17.50-31 set to be erased
---> Package cell-libs-source.noarch 0:3.0-15 set to be erased
---> Package cell-examples.ppc64 0:3.0-9 set to be erased
---> Package spu-mass-devel.ppc 0:4.5.0-9 set to be erased
---> Package alf-examples-source.noarch 0:3.0.0-6 set to be erased
---> Package alf-devel.ppc 0:3.0.0-8 set to be erased
---> Package ppu-gcc-c++.ppc 0:4.1.1-54 set to be erased
Removing:

- Package ppu-simdmath.ppc 0:3.0-3 set to be erased
- Package cell-tutorial.ppc 0:3.0-4 set to be erased
- Package alf.ppc 0:3.0.0-8 set to be erased
- Package libspe2.ppc 0:2.2.0-0.8 set to be erased
- Package alf.ppc64 0:3.0.0-8 set to be erased
- Package cell-examples-source.noarch 0:3.0-9 set to be erased

Dependencies Resolved

<table>
<thead>
<tr>
<th>Package</th>
<th>Arch</th>
<th>Version</th>
<th>Repository</th>
<th>Size</th>
</tr>
</thead>
<tbody>
<tr>
<td>alf</td>
<td>ppc</td>
<td>3.0.0-8</td>
<td>installed</td>
<td>71 k</td>
</tr>
<tr>
<td>alf</td>
<td>ppc64</td>
<td>3.0.0-8</td>
<td>installed</td>
<td>96 k</td>
</tr>
<tr>
<td>alf-devel</td>
<td>ppc64</td>
<td>3.0.0-8</td>
<td>installed</td>
<td>136 k</td>
</tr>
<tr>
<td>alf-devel</td>
<td>ppc</td>
<td>3.0.0-8</td>
<td>installed</td>
<td>366 k</td>
</tr>
<tr>
<td>alf-examples-source</td>
<td>noarch</td>
<td>3.0.0-6</td>
<td>installed</td>
<td>1.4 M</td>
</tr>
<tr>
<td>alffman</td>
<td>noarch</td>
<td>3.0-4</td>
<td>installed</td>
<td>32 k</td>
</tr>
<tr>
<td>blas</td>
<td>ppc</td>
<td>3.1.1-1.fc7</td>
<td>installed</td>
<td>673 k</td>
</tr>
<tr>
<td>cell-buildutils</td>
<td>noarch</td>
<td>3.0-10</td>
<td>installed</td>
<td>76 k</td>
</tr>
<tr>
<td>cell-demos</td>
<td>ppc64</td>
<td>3.0-8</td>
<td>installed</td>
<td>19 M</td>
</tr>
<tr>
<td>cell-demos-source</td>
<td>noarch</td>
<td>3.0-8</td>
<td>installed</td>
<td>19 M</td>
</tr>
<tr>
<td>cell-documentation</td>
<td>noarch</td>
<td>3.0-3</td>
<td>installed</td>
<td>46 M</td>
</tr>
<tr>
<td>cell-early-license</td>
<td>noarch</td>
<td>3.0-0.0</td>
<td>installed</td>
<td>949 k</td>
</tr>
<tr>
<td>cell-examples</td>
<td>ppc64</td>
<td>3.0-9</td>
<td>installed</td>
<td>1.4 M</td>
</tr>
<tr>
<td>cell-examples-source</td>
<td>noarch</td>
<td>3.0-9</td>
<td>installed</td>
<td>1.1 M</td>
</tr>
<tr>
<td>cell-libs</td>
<td>ppc64</td>
<td>3.0-15</td>
<td>installed</td>
<td>20 k</td>
</tr>
<tr>
<td>cell-libs</td>
<td>ppc</td>
<td>3.0-15</td>
<td>installed</td>
<td>172 k</td>
</tr>
<tr>
<td>cell-libs-devel</td>
<td>ppc</td>
<td>3.0-15</td>
<td>installed</td>
<td>2.5 M</td>
</tr>
<tr>
<td>cell-libs-devel</td>
<td>ppc64</td>
<td>3.0-15</td>
<td>installed</td>
<td>540 k</td>
</tr>
<tr>
<td>cell-libs-source</td>
<td>noarch</td>
<td>3.0-15</td>
<td>installed</td>
<td>1.5 M</td>
</tr>
<tr>
<td>cell-tutorial</td>
<td>ppc</td>
<td>3.0-4</td>
<td>installed</td>
<td>145 k</td>
</tr>
<tr>
<td>cell-tutorial-source</td>
<td>noarch</td>
<td>3.0-4</td>
<td>installed</td>
<td>80 k</td>
</tr>
<tr>
<td>dacs</td>
<td>ppc64</td>
<td>3.0-0-18</td>
<td>installed</td>
<td>67 k</td>
</tr>
<tr>
<td>dacs-devel</td>
<td>ppc64</td>
<td>3.0-0-18</td>
<td>installed</td>
<td>597 k</td>
</tr>
<tr>
<td>dacsman</td>
<td>noarch</td>
<td>3.0-4</td>
<td>installed</td>
<td>47 k</td>
</tr>
<tr>
<td>libspe</td>
<td>pcc</td>
<td>1.2.2-2</td>
<td>installed</td>
<td>78 k</td>
</tr>
<tr>
<td>libspe</td>
<td>ppc64</td>
<td>1.2.2-2</td>
<td>installed</td>
<td>96 k</td>
</tr>
<tr>
<td>libspe2</td>
<td>ppc64</td>
<td>2.2.0-87</td>
<td>installed</td>
<td>109 k</td>
</tr>
<tr>
<td>libspe2</td>
<td>ppc</td>
<td>2.2.0-87</td>
<td>installed</td>
<td>86 k</td>
</tr>
<tr>
<td>libspe2-devel</td>
<td>ppc</td>
<td>2.2.0-87</td>
<td>installed</td>
<td>18 k</td>
</tr>
<tr>
<td>libspe2-devel</td>
<td>ppc64</td>
<td>2.2.0-87</td>
<td>installed</td>
<td>18 k</td>
</tr>
<tr>
<td>libspe2-man</td>
<td>noarch</td>
<td>2.2.0-4</td>
<td>installed</td>
<td>43 k</td>
</tr>
<tr>
<td>ppu-binutils</td>
<td>ppc</td>
<td>2.17.50-31</td>
<td>installed</td>
<td>19 M</td>
</tr>
<tr>
<td>ppu-gcc</td>
<td>pcc</td>
<td>4.1.1-54</td>
<td>installed</td>
<td>12 M</td>
</tr>
<tr>
<td>ppu-gcc-c++</td>
<td>pcc</td>
<td>4.1.1-54</td>
<td>installed</td>
<td>97 M</td>
</tr>
<tr>
<td>ppu-gdb</td>
<td>ppc</td>
<td>6.6.50-23</td>
<td>installed</td>
<td>7.9 M</td>
</tr>
<tr>
<td>ppu-mass-devel</td>
<td>ppc</td>
<td>4.5-0-9</td>
<td>installed</td>
<td>556 k</td>
</tr>
<tr>
<td>ppu-mass-devel</td>
<td>ppc64</td>
<td>4.5-0-9</td>
<td>installed</td>
<td>669 k</td>
</tr>
<tr>
<td>ppu-simdmath</td>
<td>ppc64</td>
<td>3.0-3</td>
<td>installed</td>
<td>190 k</td>
</tr>
<tr>
<td>ppu-simdmath</td>
<td>pcc</td>
<td>3.0-3</td>
<td>installed</td>
<td>156 k</td>
</tr>
<tr>
<td>ppu-simdmath-devel</td>
<td>ppc64</td>
<td>3.0-3</td>
<td>installed</td>
<td>623 k</td>
</tr>
<tr>
<td>ppu-simdmath-devel</td>
<td>ppc64</td>
<td>3.0-3</td>
<td>installed</td>
<td>707 k</td>
</tr>
<tr>
<td>simdman</td>
<td>noarch</td>
<td>3.0-4</td>
<td>installed</td>
<td>125 k</td>
</tr>
<tr>
<td>spu-binutils</td>
<td>ppc</td>
<td>2.17.50-31</td>
<td>installed</td>
<td>6.9 M</td>
</tr>
<tr>
<td>spu-gcc</td>
<td>pppc</td>
<td>4.1.1-100</td>
<td>installed</td>
<td>7.0 M</td>
</tr>
<tr>
<td>spu-gcc-c++</td>
<td>pppc</td>
<td>4.1.1-100</td>
<td>installed</td>
<td>60 M</td>
</tr>
<tr>
<td>spu-gdb</td>
<td>ppc</td>
<td>6.6.50-10</td>
<td>installed</td>
<td>5.1 M</td>
</tr>
<tr>
<td>spu-mass-devel</td>
<td>ppc</td>
<td>4.5-0-9</td>
<td>installed</td>
<td>412 k</td>
</tr>
<tr>
<td>spu-newlib</td>
<td>pppc</td>
<td>1.15.0-76</td>
<td>installed</td>
<td>4.8 M</td>
</tr>
<tr>
<td>spu-simdmath-devel</td>
<td>ppc</td>
<td>3.0-3</td>
<td>installed</td>
<td>1.0 M</td>
</tr>
<tr>
<td>sysroot_image</td>
<td>noarch</td>
<td>3.0-7</td>
<td>installed</td>
<td>1.8 G</td>
</tr>
<tr>
<td>systemsim-cell</td>
<td>ppc64</td>
<td>3.0-14</td>
<td>installed</td>
<td>13 M</td>
</tr>
</tbody>
</table>

Transaction Summary

---

SDK 3.0 Installation Guide

72
Install 0 Package(s)
Update 0 Package(s)
Remove 51 Package(s)

Is this ok [y/N]:
Downloading Packages:
Running rpm_check_debug
   --> Populating transaction set with selected packages. Please wait.
   --> Package systemsim-cell.ppc64 0:3.0-14 set to be erased
   --> Package alfman.noarch 0:3.0-4 set to be erased
   --> Package cell-buildutils.noarch 0:3.0-10 set to be erased
   --> Package ppu-simdmath.ppc64 0:3.0-3 set to be erased
   --> Package cell-early-license.noarch 0:3.0.0-0.0 set to be erased
   --> Package libspe2-devel.ppc 0:2.2.0-87 set to be erased
   --> Package cell-demos.ppc64 0:3.0-8 set to be erased
   --> Package dacs-devel.ppc64 0:3.0.0-18 set to be erased
   --> Package cell-tutorial-source.noarch 0:3.0-4 set to be erased
   --> Package libspe2-devel.ppc64 0:2.2.0-87 set to be erased
   --> Package ppu-simdmath-devel.ppc 0:3.0-3 set to be erased
   --> Package cell-libs-devel.ppc 0:3.0-15 set to be erased
   --> Package dacsman.noarch 0:3.0-4 set to be erased
   --> Package libspe2man.noarch 0:2.2.0-4 set to be erased
   --> Package cell-libs.ppc64 0:3.0-15 set to be erased
   --> Package cell-libs-devel.ppc64 0:3.0-15 set to be erased
   --> Package cell-documentation.noarch 0:3.0-3 set to be erased
   --> Package ppu-simdmath-devel.ppc64 0:3.0-3 set to be erased
   --> Package libspe2-devel.ppc64 0:2.2.0-87 set to be erased
   --> Package cell-demos-source.noarch 0:3.0-8 set to be erased
   --> Package sysroot_image.noarch 0:3.0-7 set to be erased
   --> Package blas.ppc 0:3.1.1-1.fc7 set to be erased
   --> Package libspe.ppc 0:1.2.2-2 set to be erased
   --> Package ppu-mass-devel.ppc 0:4.5.0-9 set to be erased
   --> Package dacs.ppc64 0:3.0.0-18 set to be erased
   --> Package spu-simdmath-devel.ppc 0:3.0-3 set to be erased
   --> Package spu-gcc-c++.ppc 0:4.1.1-100 set to be erased
   --> Package alf-devel.ppc64 0:3.0.0-8 set to be erased
   --> Package ppu-binutils.ppc 0:2.17.50-31 set to be erased
   --> Package ppu-gdb.ppc 0:6.6.50-23 set to be erased
   --> Package simdman.noarch 0:3.0-4 set to be erased
   --> Package spu-newlib.ppc 0:1.15.0-76 set to be erased
   --> Package ppu-mass-devel.ppc64 0:4.5.0-9 set to be erased
   --> Package libspe.ppc64 0:1.2.2-2 set to be erased
   --> Package ppu-gcc.ppc 0:4.1.1-54 set to be erased
   --> Package spu-gdb.ppc 0:6.6.50-10 set to be erased
   --> Package spu-gcc.ppc 0:4.1.1-100 set to be erased
   --> Package spu-binutils.ppc 0:2.17.50-31 set to be erased
   --> Package cell-libs-source.noarch 0:3.0-15 set to be erased
   --> Package cell-examples.ppc64 0:3.0-9 set to be erased
   --> Package spu-mass-devel.ppc 0:4.5.0-9 set to be erased
   --> Package alf-examples-source.noarch 0:3.0.0-6 set to be erased
   --> Package alf-devel.ppc 0:3.0.0-8 set to be erased
   --> Package ppu-gcc-c++.ppc 0:4.1.1-54 set to be erased
   --> Package spu-simdmath.ppc 0:3.0-3 set to be erased
   --> Package cell-tutorial.ppc 0:3.0-4 set to be erased
   --> Package alf.ppc 0:3.0.0-8 set to be erased
   --> Package libspe2.ppc 0:2.2.0-87 set to be erased
   --> Package alf.ppc64 0:3.0.0-8 set to be erased
   --> Package cell-examples-source.noarch 0:3.0-9 set to be erased

Running Transaction Test
Finished Transaction Test
Transaction Test Succeeded
Running Transaction
rm: cannot remove `/etc/udev/rules.d/99-systemsim-cell.rules':
   No such file or directory
error: %preun(systemsim-cell-3.0-14.ppc64) scriptlet failed, exit status 1

Removed: alf.ppc 0:3.0.0-8 alf.ppc64 0:3.0.0-8 alf-devel.ppc64 0:3.0.0-8
cellsdk INFO-2009: looking for still-installed cellsdk rpms
installedCount=1
There are 1 cellsdk rpms still installed:
  default CellSimulator systemsim-cell-3.0-14.ppc64
Uninstall them [y/n]? Setting up Group Process
Resolving Dependencies
--> Running transaction check
----> Package systemsim-cell.ppc64 0:3.0-14 set to be erased

Dependencies Resolved

+---------------------------------------------------------------------------------------------------+
<table>
<thead>
<tr>
<th>Package</th>
<th>Arch</th>
<th>Version</th>
<th>Repository</th>
<th>Size</th>
</tr>
</thead>
<tbody>
<tr>
<td>Removing:</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>systemsim-cell</td>
<td>ppc64</td>
<td>3.0-14</td>
<td>installed</td>
<td>13 M</td>
</tr>
</tbody>
</table>
+---------------------------------------------------------------------------------------------------+

Transaction Summary
+---------------------------------------------------------------------------------------------------+
| Install | 0 Package(s) |
| Update  | 0 Package(s) |
| Remove  | 1 Package(s) |
+---------------------------------------------------------------------------------------------------+

Downloading Packages:
Running rpm_check_debug
--> Populating transaction set with selected packages. Please wait.
----> Package systemsim-cell.ppc64 0:3.0-14 set to be erased
Running Transaction Test
Finished Transaction Test
Transaction Test Succeeded
Running Transaction
rm: cannot remove '/etc/udev/rules.d/99-systemsim-cell.rules':
    No such file or directory
error: %preun(systemsim-cell-3.0-14.ppc64) scriptlet failed, exit status 1

Removed: systemsim-cell.ppc64 0:3.0-14
Complete!

Completely remove cellsdk from the system [y/n]?
If you have files in the /opt/cell you want to keep, answer no.
Completely remove /opt/cell [y/n]?
If you have files in the /opt/ibm/systemsim you want to keep, answer no.
Completely remove /opt/ibm/systemsim [y/n]?
If you have files in the /tmp/cellsdk/rpms you want to keep, answer no.
Completely remove /tmp/cellsdk/rpms [y/n]?
If you have files in the /tmp/cellsdk/openSrc you want to keep, answer no.
Completely remove /tmp/cellsdk/openSrc [y/n]?
Appendix E. Known limitations

There are a number of known limitations with this early release of SDK 3.0. These are documented in the following installed file: /opt/cell/sdk/doc/README
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.

Edition notices

© Copyright International Business Machines Corporation 2007. All rights reserved.

U.S. Government Users Restricted Rights — Use, duplication, or disclosure restricted by GSA ADP Schedule Contract with IBM Corp.

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®
PowerPC
PowerPC Architecture™

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-128.ibm.com/developerworks/power/cell/

Click on 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
- SPU Assembly Language Specification
- SPU Application Binary Interface Specification
- SIMD Math Library Specification for Cell Broadband Engine Architecture
- Cell Broadband Engine Linux Reference Implementation Application Binary Interface Specification

Programming
- Programming Tutorial
- SDK for Multicore Acceleration Version 3.0 Programmer’s Guide

Library
- SPE Runtime Management library
- SPE Runtime Management library Version 1.2 to Version 2.0 Migration Guide
- Accelerated Library Framework for Hybrid-x86 Programmer’s Guide and API Reference
- 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
- SIMD Math Library Specification
- Monte Carlo Library API Reference Manual (Prototype)

Installation
- SDK for Multicore Acceleration Version 3.0 Installation Guide
IBM XL C/C++ Compiler and IBM XL Fortran Compiler

Detail about documentation for the compilers is available on the developerWorks Web site.

IBM Full-System Simulator and debugging documentation

Detail about documentation for the simulator and debugging tools is available on the developerWorks Web site.

PowerPC Base

- PowerPC Architecture Book, Version 2.02
  - Book I: PowerPC User Instruction Set Architecture
  - Book II: PowerPC Virtual Environment Architecture
  - Book III: PowerPC Operating Environment Architecture
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.

© Copyright IBM Corp. 2006, 2007 (DRAFT)
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**


**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

A
affinity 53
Anacoda 10
automatic updates 37, 39

B
BladeCenter QS20 7, 9, 10, 14
bad firmware boot 9
booting 8
installing Fedora 7 9
managing 7
shutdown 9
BladeCenter QS21
installation specifics 15
installing Linux 15
network boot 19
BLAS 38
booting
after installation 14
BOOTP 15
bootstrap protocol 15

cell-install 6
cellsdk install 59, 71
cellsdk script 46
install 36
options 46
verify 46, 67
cellsdk verify 38
component descriptions 23
configuration 6, 37

dc
DaCS 51
affinity 53
daemon 51
topology 51
DHCP 15
directory structure 21
documentation 83
download 5
downloading
SDK 35
Dynamic Host Configuration Protocol 15

eclipse 47
Eclipse IDE 39
expat 3

F
Fedora 7
installation 7
installing on BladeCenter QS20 9
file system 21
firewall 48
firmware
checking which version 7
firmware bank 8
firmware boot 8

H
Hybrid DaCS 38
Hybrid-x86 51

I
IDE 47
installing 47
InfiniBand 9
init process 9
initrd
for NFS boot 18
installation 36
configuration 37
default 33
Fedora 7 7
finishing 19
installation steps 33
network 10, 15
operating system 7
overview 9
Pirut 39
preparation 35
SDK 33
software 33
starting 11
isolation kit 47

K
kernel 37
kickstart 10

L
license 36
licenses 2, 33
Linux kernel 37

N
net boot overview 16
netboot
setting up installation environment 11

O
operating system 15
installation 7
installing on a POWER-based system 18
OPIT 37

P
package groups 26
Pirut 39
post-install 6
postinstall 37
prerequisites
hard disk space 3
hardware 3
RAM for the simulator 3
RAM on host 3
SDK 3
software 3
product sets 33

Q
quickstart 5

R
root file system 18
RPMs 22
by component 28
rsync 4

S
SDK
components 21
installation 36
installation files 35
prerequisites 3
SDK documentation 83
security updates 37
sed 4
SELinux 3
spu-isolated-app 47
SPU-Isolation 38, 47
support 55
supported platforms 2

© Copyright IBM Corp. 2006, 2007 (DRAFT)
T
target platform 21
TCL 4, 36
TFTP 15
trademarks 81
Trivial File Transfer Protocol 15
troubleshooting 49

U
uninstalling 33, 46
   Eclipse IDE 47
   SDK 46, 47
updating 33

V
verify 46, 67

W
wget 4
what’s new 1

Y
YUM
   configuration 14
   exclude 37
   server 48
   yum-updatesd 37, 39

Z
zImage 18