



# **Cray Application Developer's Environment Installation Guide**

**S-2465-611**

---

© 2007–2012 Cray Inc. All Rights Reserved. This document or parts thereof may not be reproduced in any form unless permitted by contract or by written permission of Cray Inc.

---

## U.S. GOVERNMENT RESTRICTED RIGHTS NOTICE

The Computer Software is delivered as "Commercial Computer Software" as defined in DFARS 48 CFR 252.227-7014.

All Computer Software and Computer Software Documentation acquired by or for the U.S. Government is provided with Restricted Rights. Use, duplication or disclosure by the U.S. Government is subject to the restrictions described in FAR 48 CFR 52.227-14 or DFARS 48 CFR 252.227-7014, as applicable.

Technical Data acquired by or for the U.S. Government, if any, is provided with Limited Rights. Use, duplication or disclosure by the U.S. Government is subject to the restrictions described in FAR 48 CFR 52.227-14 or DFARS 48 CFR 252.227-7013, as applicable.

---

Cray is a federally registered trademark and Active Manager, Cray Apprentice2, Cray Apprentice2 Desktop, Cray C++ Compiling System, Cray CX, Cray CX1, Cray CX1-iWS, Cray CX1-LC, Cray CX1000, Cray CX1000-C, Cray CX1000-G, Cray CX1000-S, Cray CX1000-SC, Cray CX1000-SM, Cray CX1000-HN, Cray Fortran Compiler, Cray Linux Environment, Cray SHMEM, Cray X1, Cray X1E, Cray X2, Cray XD1, Cray XE, Cray XEm, Cray XE5, Cray XE5m, Cray XE6, Cray XE6m, Cray XK6, Cray XK6m, Cray XMT, Cray XR1, Cray XT, Cray XTm, Cray XT3, Cray XT4, Cray XT5, Cray XT5<sub>h</sub>, Cray XT5m, Cray XT6, Cray XT6m, CrayDoc, CrayPort, CRInform, ECOphlex, LibSci, NodeKARE, RapidArray, Sonexion, The Way to Better Science, Threadstorm, uRiKA, UNICOS/lc, and YarcData are trademarks of Cray Inc.

---

AMD and Opteron are trademarks of Advanced Micro Devices, Inc. FlexNet is a trademark of Flexera Software. GNU is a trademark of The Free Software Foundation. Gemini and SeaStar are trademarks of Intel Corporation in the United States and/or other countries. ISO is a trademark of International Organization for Standardization (Organisation Internationale de Normalisation). Intel is a trademark of Intel Corporation or its subsidiaries in the United States and other countries. Java is a trademark of Oracle and/or its affiliates. Linux is a trademark of Linus Torvalds. PETSc is a trademark of Copyright (C) 1995-2004 University of Chicago. PGI is a trademark of The Portland Group Compiler Technology, STMicroelectronics, Inc. Platform is a trademark of Platform Computing Corporation. SLES and SUSE are trademarks of Novell, Inc. UNIX, the "X device," X Window System, and X/Open are trademarks of The Open Group in the United States and other countries. All other trademarks are the property of their respective owners.

---

## RECORD OF REVISION

S-2465–611 Published June 2012 Documents installation of Cray Application Developer's Environment 6.11 or later on Cray XT, Cray XE, or Cray XK systems, or on standalone Linux systems.

S-2465–60 Published July 2011 Documents installation of Cray Application Developer's Environment 6.0 or later on Cray XT and Cray XE systems, or on standalone Linux systems.

S-2465–512 Published June 2011 Documents installation of Cray Application Developer's Environment 5.12 or later on Cray XT and Cray XE systems.

5.0 Published June 2010 Documents installation of Cray Application Developer's Environment 5.0 or later on Cray XT and Cray XE systems.

4.7 Published November 2009 Documents installation of Cray Application Developer's Environment 4.7 or later on Cray XT systems.

4.6 Published October 2009 Documents installation of Cray Application Developer's Environment 4.6 on Cray XT systems.

4.5 Published September 2009 Documents installation of Cray Application Developer's Environment 4.5 on Cray XT systems.

4.4 Published August 2009 Documents installation of Cray Application Developer's Environment 4.4 on Cray XT systems.

4.3 Published July 2009 Documents installation of Cray Application Developer's Environment 4.3 on Cray XT systems.

4.2 Published June 2009 Documents installation of Cray Application Developer's Environment 4.2 on Cray XT systems.

4.1 Published May 2009 Documents installation of Cray Application Developer's Environment 4.1 on Cray XT systems.

4.0 Published April 2009 Documents installation of Cray Application Developer's Environment 4.0 on Cray XT systems.

3.3 Published March 2009 Documents installation of Cray Application Developer's Environment 3.3 and Cray Application Developer's Environment Supplement 1.0 on Cray XT systems and standalone Linux systems.

3.0 Published May 2008 Documents installation of Cray Programming Environments on Cray XT systems, including Cray XT5<sub>h</sub> (Cray X2) systems.

2.0 Published October 2007 Documents installation of Cray Programming Environments on Cray XT systems.

---



# Contents

---

	<i>Page</i>
<b>Introduction [1]</b>	<b>7</b>
1.1 Description of the Cray Application Developer's Environment . . . . .	7
1.2 Description of the Cray Application Developer's Environment Supplement . . . . .	9
1.2.1 Linux System Hardware Requirements . . . . .	9
1.2.2 Linux System Software Requirements . . . . .	10
1.2.3 Expectations . . . . .	10
1.3 Software Dependencies . . . . .	10
<b>Installing on Cray Systems [2]</b>	<b>13</b>
2.1 Caveats . . . . .	13
2.1.1 About Intel Composer and PathScale Compiler Suite Support . . . . .	13
2.2 Standard Installation Procedure . . . . .	13
2.2.1 Installing Intel Composer or PathScale Compiler Support . . . . .	17
2.3 Deleting and Reinstalling CADE Components . . . . .	18
2.4 Changes to the Default Programming Environment . . . . .	20
2.4.1 Configuring the <code>/etc/*rc.local</code> files on a Cray Linux Environment (CLE) 4.0 System . . . . .	20
<b>Installing on Standalone Linux Systems [3]</b>	<b>23</b>
3.1 Standard Installation Procedure . . . . .	23
3.2 Configuring <code>/etc/*rc.local</code> files . . . . .	24
3.2.1 Cray Linux Environment (CLE) 2.2 and 3.1 . . . . .	25
3.2.2 Cray Linux Environment (CLE) 4.0 . . . . .	25
3.3 Using the Cray Application Developer's Environment . . . . .	26
<b>Procedures</b>	
Procedure 1. Installing CADE on Cray systems . . . . .	13
Procedure 2. Installing Intel or PathScale compiler support on Cray systems . . . . .	17
Procedure 3. Deleting and reinstalling CADE components . . . . .	18
Procedure 4. Installing CADE and CADES on Linux Systems (Scripted) . . . . .	23
Procedure 5. Using the Cray Application Developer's Environment . . . . .	26



# Introduction [1]

---

This document describes how to install the Cray Application Developer's Environment (CADE) on Cray XT, Cray XE, and Cray XK systems running the Cray Linux Environment (CLE) operating system.

This document also describes how to install CADE and the Cray Application Developer's Environment Supplement (CADES) on a standalone Linux system. CADES is a collection of supplemental components required in order to use CADE on a standalone Linux system, when the Linux system is being used as a development and compilation workstation in support of a Cray supercomputer system.

The information in this guide is intended for system administrators receiving their first release of this product or upgrading from a previous release. This guide assumes the administrator has a good understanding of Cray and Linux system administration.

**Note:** Beginning with the CADE 6.10 release, Cray is changing the prefixes of rpm files, module file names, and selected environment variables from the system-specific `xt-` to the more generic `cray-` or `craype-`, depending on the module, environment variable, or rpm file type. For example the MPICH2 module, `xt-mpich2`, is being replaced by `cray-mpich2`. The current names will be maintained for a transitional period to ensure compatibility with existing make files and informational messages will be issued when obsolete names are used.

During the transitional period, the examples provided in this book may reflect either module naming convention.

For information about using the Cray Application Developer's Environment, see the *Cray Application Developer's Environment User's Guide* and *Workload Management and Application Placement for the Cray Linux Environment*.

## 1.1 Description of the Cray Application Developer's Environment

The Cray Application Developer's Environment (CADE) consists of the basic libraries and components needed to develop and compile code on Cray systems, including the GNU Fortran, C, and C++ compilers. This package does **not** include the Cray Compiling Environment (CCE), Portland Group (PGI), Intel Composer, or PathScale Compiler Suite compilers. All compilers other than the GNU compilers are sold, installed, and licensed separately.

The exact contents of the CADE package vary from release to release. The release package typically includes the latest versions of the following software components. In some cases, multiple versions are included for compatibility purposes. See the release announcement or the *README* file in the CADE package for the specific version numbers of the products included in the current CADE package.

- Compilers
  - GNU Fortran, C, and C++
- Libraries
  - Cray Message Passing Toolkit (MPT)
  - ACML
  - Cray LibSci
  - PETSc
  - FFTW
  - hdf5 and netcdf
- Tools and Support Packages
  - Java Development Kit
  - Cray Debugger Tools
  - xt-asyncpe
  - xt-pathscalesup (support for the PathScale Compiler Suite)
  - xt-intelsup (support for the Intel Composer)
  - mpfr (support for GCC 4.3.2 and later)
  - cray-modules (required)<sup>1</sup>

---

<sup>1</sup> The Cray Application Developer's Environment requires the use of the Modules application to manage multiple versions of products and libraries, and includes an updated, ready-to-use copy of Modules. Alternatively, you can download the Modules source from this link <http://prdownloads.sourceforge.net/modules> and make it yourself. However, if you do so, be advised that some Cray-specific features will not be available.

## 1.2 Description of the Cray Application Developer's Environment Supplement

The Cray Application Developer's Environment Supplement (CADES) contains the additional components required in order to install and use CADE on standalone Linux systems. Most of these components are specific to a particular compiler or Cray Linux Environment (CLE) version. Some are specific to particular Cray hardware platforms.

The files on the installation media are organized by directory:

- The files in the root directory are common to all Cray hardware types and operating system versions. These files must be installed on all Linux cross-compiler systems.
- Additionally, the files in the subdirectories are specific to particular Cray hardware types and operating system versions. Along with the files in the root directory, install only the files that are appropriate for your Cray hardware platform and operating system version. For example, if your site has a Cray XT6 (SeaStar) system and is running CLE release 3.1.20, install only the files in the `/3.1.20-ss` subdirectory.

The CADES release package typically includes the latest versions of released software components. In some cases, multiple versions may be included for compatibility purposes. See the release announcement or the *README* file in the CADES package for the specific version numbers of the products included in the current CADES package.

### 1.2.1 Linux System Hardware Requirements

Cray recommends that any Linux system being used as a standalone code development and compilation workstation meet the following minimum requirements.

**Table 1. Hardware Requirements**

Component	Requirement
Processor	Any AMD 64-bit processor; e.g., Opteron, Athlon, or Turion 64
Operating System	SUSE Linux Enterprise Server (SLES) release 10 (x64) or release 11 (x64)
Memory	At least 1GB/core minimum; 2GB/core recommended
Disk Space	At least 200GB minimum; 500GB or larger recommended

## 1.2.2 Linux System Software Requirements

The operating system used on the standalone Linux system **must** match the version of the CLE operating system used on the Cray system. If your Cray system runs 2.2, the standalone Linux system must run SLES 10. If your Cray system runs CLE 3.0 or later, the standalone Linux system must run SLES 11. If you attempt to cross versions, users will be able to write and compile code on the Linux system but the resulting executables may not run correctly on the Cray system.

Further, all SLES distributed products installed on the Cray system must also be installed on the standalone Linux system. The CLE distributions provide lists of installed RPMs, and these should be compared with the products installed on the standalone Linux system. Any missing products can be installed from the CLE distribution media.

## 1.2.3 Expectations

When installed on a standalone Linux system, CADE produces binaries that will run on Cray systems. There may be differences between binaries built on Cray systems and those built on standalone Linux systems, for the following reasons:

- Date and time stamps are compiled into the binary.
- Path names may be different. This is especially apparent when compiling code that uses dynamically linked objects.
- Compilers may have hard-coded paths to specific files such as core runtime libraries.
- There may be minor differences in the software installed on the different systems, including security patches.
- Differences in the modules that are loaded at compilation time.

## 1.3 Software Dependencies

The following known dependencies affect the CADE and CADES installation processes.

- If you use the PathScale Compiler Suite, you must install the PathScale Compiler Suite **before** installing the Cray support package for PathScale, `xt-pathscalesup-version.x86_64.rpm`.
- If you use the Intel Composer, you must install the Intel Composer **before** installing the Cray support package for Intel, `cray-intelsup-version.x86_64.rpm`.
- There are no known order dependencies between CADE, CADES, CCE, and PGI.

- There is a correlation between the version of the CLE operating system being used on the Cray system for which you are developing code and the CADES RPM files to be installed. For example, if the Cray system for which you are developing code runs the CLE 2.2.31 operating system, install only these RPM files on the standalone Linux system:

```
xt-prgenv-xc-2.2.31-14317.x86_64.rpm
xt-xcpe-2.2.31-14317.x86_64.rpm
```

Beginning with the CLE 3.0 release, the installation files are sorted into subdirectories by hardware type and OS version, as described in [Description of the Cray Application Developer's Environment Supplement on page 9](#). On CLE 3.0 and later systems, install only the files in the subdirectory corresponding to your Cray system hardware type and OS version.

- After CADES is installed, there is a correlation between the version of the CLE operating system being used on the Cray system for which you are developing code and the programming environment module (PrgEnv) you must load before compiling. For example, if the Cray system for which you are developing code runs the CLE 3.1.71 operating system and you are using the Cray (CCE) compilers, you must load this module file on the standalone Linux system:

```
PrgEnv-cray/3.1.71
```

Likewise, if you are using the PGI compilers, you must load this module file:

```
PrgEnv-pgi/3.1.71
```

Ideally, the version of the module that matches the version of your Cray OS should be the default version of the compiler on your Linux system.

- There is a correlation between the version of the PGI and Intel compilers and the Cray support package to be installed. For example, if you are using PGI version 11.5.0, install this RPM file on the standalone Linux system:

```
cray-pgisup-11.5.0-x.x86_64.rpm
```

- By default, PGI compilers are installed in `/opt/pgi`. For correct operation with `cray-pgisup`, they must instead be installed in `/opt/pgi/version`; for example: `/opt/pgi/11.5.0`.
- By default, the PGI installation scripts asks if you want to install MPICH1. Cray recommends answering no, as MPICH2 is already incorporated into Cray MPT (Message Passing Toolkit).
- Many of the optional compilers require the FlexNet license manager. If you do not already have this software installed on your system or network, it is supplied in the following file.

```
pgroupd-lin64-v1161.tar.gz
```

This software is not installed automatically. If you require it, you must unzip it, untar it, and install it using the instructions provided in the FlexNet package.



# Installing on Cray Systems [2]

---

Use the System Management Workstation (SMW) to install the Cray Application Developer's Environment (CADE) on the shared root of your Cray system. You must have administrative privileges in order to install software on Cray systems.

The Cray Application Developer's Environment packages are distributed on DVD-ROM. In some cases they may also be made available as one or more downloadable files. The instructions in this chapter assume that you are working with a DVD-ROM. If you are working with either a single `iso` file or a `tar` file composed of individual `rpm` files, adjust the instructions accordingly.

## 2.1 Caveats

### 2.1.1 About Intel Composer and PathScale Compiler Suite Support

The CADE installation script attempts to install two support packages that are required by the Intel Composer and PathScale Compiler Suite, respectively. This leads to one of the following conditions.

- If you install the Intel Composer or PathScale Compiler Suite **before** installing CADE, the Cray support package for the respective compiler installs successfully.
- If you have **not** installed the Intel Composer or PathScale Compiler Suite and do not intend to do so, the support package portion of the installation script ends in a dependency error. This error can be ignored. The remainder of the CADE installation script runs to completion.
- If you install the Intel Composer or PathScale Compiler Suite **after** installing CADE, you must rerun the installation script in order to successfully install the support package(s). For more information, see [Installing Intel Composer or PathScale Compiler Support on page 17](#).

## 2.2 Standard Installation Procedure

### Procedure 1. Installing CADE on Cray systems

1. As `crayadm`, log on to the SMW.

```
% ssh crayadm@smw
```

2. If necessary, load and mount the PE distribution media.

```
smw:~# mount /dev/cdrom /media/cdrom
```

3. Make a temporary directory on the boot node for the installation files.

```
smw:~# ssh boot mkdir /tmp/install.pe
```

4. Copy the installation files from the distribution media to the boot node.

```
smw:~# scp -pr /media/cdrom/* boot:/tmp/install.pe
```

5. Unmount and remove the PE distribution media.

```
smw:~# umount /media/cdrom
```

6. As root, log on to the boot node.

```
smw:~# ssh root@boot
```

7. Change to your temporary directory.

```
boot001:~# cd /tmp/install.pe
```

8. Execute the script CADEinstall.sh to determine which packages are already installed.

```
boot001:/tmp/install.pe # ./CADEinstall.sh -m `pwd` -v -G -V -s /rr/current
```

If this is a new installation, the script returns a list of packages similar to the following example. There may be multiple versions of some packages and the exact list and number of RPMs varies depending on the packages included in the current CADE release.

```
ROOTDIR=/rr/current
```

RPM Name and Current Version	Status	Installed Version(s)
-----	-----	-----
acml-<version>	NOT INSTALLED	
cray-mpt-<version>	NOT INSTALLED	
fftw-<version>	NOT INSTALLED	
fftw-<version>	NOT INSTALLED	
java-<version>	NOT INSTALLED	
lgdb-<version>	NOT INSTALLED	
hdf5_netcdf-<version>	NOT INSTALLED	
petsc-<version>	NOT INSTALLED	
xt-asyncpe-<version>	NOT INSTALLED	
xt-gcc-<version>	NOT INSTALLED	
xt-gcc-<version>	NOT INSTALLED	
xt-gcc-<version>	NOT INSTALLED	
xt-libsci-<version>	NOT INSTALLED	
xt-pathscalesup-<version>	NOT INSTALLED	

```
Number of CURRENT rpms = 0
```

```
Number of MISMATCH rpms = 0
```

```
Number of NOT INSTALLED rpms = 15
```

If this is an upgrade installation, the script should return a similar list, but with status and installed version information.

9. (Optional) If you want to set the packages you are about to install to be the default versions, set the `CRAY_INSTALL_DEFAULT` environment variable to 1.
10. Execute the script `CADEinstall.sh` again to synchronize the system with the packages in the temporary directory.

```
boot001:/tmp/install.pe # ./CADEinstall.sh -m `pwd` -v -K -V -s /rr/current
```

You are asked to confirm that you want to continue. Enter **y**.

The script first determines which type of Cray system your site is using and which version of the CLE operating system is in use on your system. The script then installs only those packages that are appropriate for your Cray hardware and operating system.

If this is an update installation, the script installs any packages that are not already installed. It also installs packages if the versions in the temporary directory are newer than the versions in `/rr/current`.

**Note:** Numerous messages are generated to `stdout`. Watch for error messages.

In particular, if the PathScale Compiler Suite is not installed, the file `xt-pathscalesup-version.x86_64.rpm` fails with a dependency error. Likewise, if the Intel Composer is not installed, the file `cray-intelsup-version.x86_64.rpm` fails with a dependency error. These errors can be ignored.

11. When the process completes and the script exits, execute `CADEinstall.sh` again to verify that the installation was performed correctly.

```
boot001:/tmp/install.pe # ./CADEinstall.sh -m `pwd` -v -G -V -s /rr/current
```

The script returns a list of packages similar to the following example. Again, the exact names and number of RPMs shown varies depending on the packages included in the current CADE release.

```
ROOTDIR=/rr/current
```

RPM Name and Current Version -----	Status -----	Installed Version(s) -----
acml-<version>	CURRENT	
cray-mpt-<version>	CURRENT	
fftw-<version>	CURRENT	
fftw-<version>	CURRENT	
java-<version>	CURRENT	
lgdb-<version>	CURRENT	
hdf5_netcdf-<version>	CURRENT	
petsc-<version>	CURRENT	
xt-asyncpe-<version>	CURRENT	
xt-gcc-<version>	CURRENT	
xt-gcc-<version>	CURRENT	
xt-gcc-<version>	CURRENT	
xt-libsci-<version>	CURRENT	
xt-pathscalesup-<version>	CURRENT	

```
Number of CURRENT rpms = 15
Number of MISMATCH rpms = 0
Number of NOT INSTALLED rpms = 0
```

Alternately, use this command to check the package build and installation dates.

```
boot001:/tmp/install.pe # ./CADEinstall.sh -m `pwd` -v -V -s /rr/current
```

The script returns a list of packages with the Build and Installation dates for each package.

12. (Optional) If you set the CRAY\_INSTALL\_DEFAULT environment variable previously, unset it now.
13. After verifying the installation, log out of the boot node.

```
boot001:/tmp/install.pe # exit
logout
Connection to boot closed.
smw:~#
```

14. If you are done performing system administration work, log out of the SMW.

```
smw:~# exit
logout
%
```

If you want to install the Intel Composer or PathScale Compiler Suite **after** installing CADE, see [Installing Intel Composer or PathScale Compiler Support on page 17](#) for more information.

## 2.2.1 Installing Intel Composer or PathScale Compiler Support

The Intel Composer and PathScale Compiler Suite are licensed separately. To use either or both compilers on a Cray system, you must also install support packages which supply additional software components required by the compilers.

If you obtain the Intel or PathScale software and install it **before** installing CADE, the Cray support package for respective compiler is installed automatically and no further installation or configuration is needed.

If you install the Intel or PathScale software **after** installing CADE, you must reinstall the Cray support package for the compiler in order to use it successfully. The easiest way to do so is by rerunning the PE installation process described in [Standard Installation Procedure on page 13](#).

Alternatively, follow the steps in [Procedure 2](#) to install only the Cray support package for the compiler.

### Procedure 2. Installing Intel or PathScale compiler support on Cray systems

1. As `crayadm`, log on to the System Management Workstation (SMW).

```
% ssh crayadm@smw
```

2. If necessary, load and mount the PE distribution media.

```
smw:~# mount /dev/cdrom /media/cdrom
```

3. If it does not already exist, make a temporary directory on the boot node for the installation files.

```
smw:~# ssh boot mkdir /tmp/install.pe
```

4. Copy the installation file from the distribution media to the boot node.

```
smw:~# scp -pr /media/cdrom/craydist/pe-packages/cray-compilersup-version.x86_64.rpm \
boot:/tmp/install.pe
```

Where *compiler* is either `intel` or `pathscale`.

5. Unmount and remove the PE distribution media.

```
smw:~# umount /media/cdrom
```

6. As `root`, log on to the boot node.

```
smw:~# ssh root@boot
```

7. Change to your temporary directory.

```
boot001:~# cd /tmp/install.pe
```

8. Create a target directory on the shared root and copy the support package from your temporary directory to the shared root.

```
boot001:~# mkdir -p /rr/current/software/rpms
boot001:~# cp -p cray-compilersup*.x86_64.rpm
/rr/current/software/rpms
```

9. Open an xtopview session using the default view.

```
boot001:~# xtopview
```

10. Change to the location of the rpm file on the shared root.

```
default:/:/# cd /software/rpms
```

11. Use the rpm command to install the support package.

```
default:/software/rpms# rpm -ivh cray-compilersup-version.x86_64.rpm
```

12. Exit from the xtopview session.

```
default:/software/rpms# exit
```

```
boot001:~#
```

13. Log out of the boot node.

```
boot001:/tmp/install.pe # exit
logout
Connection to boot closed.
smw:~#
```

14. If you are done performing system administration work, log out of the SMW.

```
smw:~# exit
logout
%
```

Your new compiler is now ready for use.

## 2.3 Deleting and Reinstalling CADE Components

Under some circumstances it may be necessary to delete and reinstall an application developers' environment component manually. To do so, follow the steps in [Procedure 3](#).

### Procedure 3. Deleting and reinstalling CADE components

1. As crayadm, log on to the System Management Workstation (SMW).

```
% ssh crayadm@smw
```

2. If necessary, load and mount the CADE distribution media.

```
smw:~# mount /dev/cdrom /media/cdrom
```

3. If it does not already exist, make a temporary directory on the boot node for the installation files.

```
smw:~# ssh boot mkdir /tmp/install.pe
```

4. Copy the installation file(s) from the distribution media to the boot node.

```
smw:~# scp -pr /media/cdrom/craydist/pe-packages/package-version.rpm \
boot:/tmp/install.pe
```

5. Unmount and remove the CADE distribution media.

```
smw:~# umount /media/cdrom
```

6. As root, log on to the boot node.

```
smw:~# ssh root@boot
```

7. Change to your temporary directory.

```
boot001:~# cd /tmp/install.pe
```

8. Create a target directory on the shared root and copy the RPM files from your temporary directory to the shared root.

```
boot001:~# mkdir -p /rr/current/software/rpms
boot001:~# cp -p *.rpm /rr/current/software/rpms
```

9. Open an xtopview session using the default view.

```
boot001:~# xtopview
```

10. Change to the location of the rpm file on the shared root.

```
default:/:/# cd /software/rpms
```

11. Use the rpm command to remove the desired package.

```
default:/software/rpms# rpm -e package-version.rpm
```

If you need to remove more than one package, use the `--allmatches` option and wildcards. For example, this command removes all PGI compilers currently installed on the system.

```
default:/software/rpms# rpm -e --allmatches pgi
```

12. Use the rpm command to reinstall the desired package.

```
default:/software/rpms# rpm -ivh package-version.rpm
```

Repeat [step 11](#) and [step 12](#) to remove and reinstall multiple packages. If you want to make any specific package the default version of that PE component, set the `CRAY_INSTALL_DEFAULT` environment variable to 1 before installing the package and unset it after installing the package.

13. When you are finished, exit from the `xtopview` session.

```
default/:software/rpms# exit

boot001:~#
```

14. Log out of the boot node.

```
boot001:/tmp/install.pe # exit
logout
Connection to boot closed.
smw:~#
```

15. If you are done performing system administration work, log out of the SMW.

```
smw:~# exit
logout
%
```

## 2.4 Changes to the Default Programming Environment

Installing the `cade-prgenv` RPM (during the OS installation) creates a section in the `/etc/bash.bashrc.local` and `/etc/csh.cshrc.local` scripts called the **PE-set-up** block.

To change the default PE or add more PE user defaults, the site administrator may add appropriate instructions by editing the **SITE-set-up** blocks within the `/etc/*rc.local` files. See *Managing System Software for Cray XE and Cray XT Systems*, for more information.

**In CLE 3.0, 3.1, and 4.0, the programming environment requires that a default CPU target be specified in the **SITE-set-up** block of the `/etc/*rc.local` files.**

The Cray Linux Environment (CLE) 4.0 release changes defaults in the programming environment as follows:

- The module, `cray-mpich2`, is no longer loaded as part of the default `PrgEnv-*` product list. This is to accommodate site preferences for `cray-shmem`, or `cray-mpich2` as the site default.
- The CLE 4.0 environment supports only network target `gemin`.

### 2.4.1 Configuring the `/etc/*rc.local` files on a Cray Linux Environment (CLE) 4.0 System

The network type is set correctly by the operating system installation and does not need to be reset.

The default link type is static and the default programming environment (PE) is `PrgEnv-pgi`, unless set differently in the **SITE-set-up** block of the `/etc/*rc.local` files.

For example, on the Cray XE system, the following configuration will load the `cray-mpmich2` module, ensure that code is generated for a `mc12` target, and linked statically by default.

```
##BEGIN SITE-set-up ADD SITE DEFAULTS HERE
# Site specific set up in this section.
module load cray-mpich2
module load craype-mc12
module load pbs
##END SITE-set-up
```

If you want to change the default PE settings to `PrgEnv-cray`, `SHMEM`, CPU target `xyz`, and dynamic linking, add the following to the `SITE-set-up` block in `bash.bashrc.local` and `csh.cshrc.local`:

```
##BEGIN SITE-set-up ADD SITE DEFAULTS HERE
# Site specific set up in this section.
module swap PrgEnv-pgi PrgEnv-cray
module load cray-shmem
module load craype-xyz
setenv CRAYPE_LINK_TYPE dynamic
# OR: export CRAYPE_LINK_TYPE=dynamic
##END SITE-set-up
```

The instructions in the `SITE-set-up` block are not altered by operating system installations. The `SITE-set-up` block is evaluated after the `PE-set-up` block, so make sure new instructions do not conflict with the ones in the `PE-set-up` block.



# Installing on Standalone Linux Systems [3]

---

To use a standalone Linux system as a code development and compilation workstation, you must install the following packages.

- The Cray Application Developer's Environment (CADE).
- The Cray Application Developer's Environment Supplement (CADES).
- Optionally, your choice of compilers: the Cray Compiling Environment (CCE), the Portland Group (PGI) compilers, the PathScale Compiler Suite, or the Intel Composer.

The CADE package may include several versions of the GNU Fortran, C, and C++ compilers. All other compilers are licensed and installed separately, using the instructions supplied with the compilers.

## 3.1 Standard Installation Procedure

**Note:** If installing additional SLES or third-party distributed products, you **must** delete the existing `xt-sysroot` from the Linux system before installing the upgrade. If you do not do so, the existing `xt-sysroot` may provide false satisfaction of dependencies.

### Procedure 4. Installing CADE and CADES on Linux Systems (Scripted)

You must have administrative or root privileges in order to install this software.

1. If you plan to use the Cray, Intel, PathScale, or PGI compilers, install them as described in their respective documentation.
2. If you received the packages on CD or DVD media, mount the Cray Application Developer's Environment Supplement (CADES) media and copy the required files from the media to your temporary installation directory. Note the PGI and CLE version dependencies as described in [Software Dependencies on page 10](#). Copy only those RPM files you intend to install. The RPM files in the root directory must be installed on all Linux systems. RPM files specific to different Cray hardware platforms and operating system versions are organized into separate subdirectories. In addition to the files in the root directory, install only the files in the subdirectory appropriate for your site.
3. Unmount the CADES media.

4. Mount the Cray Application Developer's Environment (CADE) media and copy the required files from the media to your temporary installation directory. Note the PathScale and Intel dependencies described in [Software Dependencies on page 10](#). If you are not using these compilers, do not copy their supported RPM files. If these compilers are not installed on your system but these RPM files are present in the temporary installation directory, the installation script will generate a dependency error message. This error message can safely be ignored.
5. Unmount the CADE media.
6. If you want to set the packages you are about to install to be the default versions, set the `CRAY_INSTALL_DEFAULT` environment variable to 1.
7. In the temporary installation directory, execute the installation script.

```
% ./CADESinstall.pl
```

The installation script finds the RPM files in the temporary installation directory and installs the associated packages in the locations specified in the RPM files.

8. When the installation script completes, unset the `CRAY_INSTALL_DEFAULT` environment variable.
9. Initialize the Modules application. Assuming that `${MPATH}` is the path in which you installed the Modules application (typically `/opt/modules` or `/usr/share/modules`).

For `csh` or `tcsh` shells, add the following line to the system or user `.cshrc` file:

```
source ${MPATH}/init/modules.csh
```

For `bash` shells, add the following line to the system or user `.bashrc` file:

```
. ${MPATH}/init/sh
```

10. Edit the `${MPATH}/init/.modulespath` file. Add the following lines if they do not already exist:

```
/opt/modulefiles          # General modulefiles directory
/opt/cray/modulefiles      # Cray modulefiles directory
```

See the *Cray Application Developer's Environment User's Guide* for more information about using the tools and processes in the environment to create code for execution on Cray systems.

## 3.2 Configuring `/etc/*rc.local` files

Site-specific settings can be added to the `/etc/*rc.local` files.

### 3.2.1 Cray Linux Environment (CLE) 2.2 and 3.1

The install of the `cade-prgenv` RPM adds the following line to the `*rc.local` files.

```
module use /opt/cray/xt-asyncpe/default/modulefiles
```

Administrators may add additional settings after this line.

In the absence of a loaded network module, the default is `seastar`.

A CPU target must be defined. The choice of processor module depends on what system the user intends to use as a runtime system for the application they build.

To set the default programming environment to `PrgEnv-cray`, `link=dynamic`, `network=gemini`, and `cpu=mc12`, add the following lines to `/etc/bash.bashrc.local` and `/etc/csh.cshrc.local`.

```
module use /opt/cray/xt-asyncpe/default/modulefiles
module load xtpe-network-gemini
module load PrgEnv-cray
module load xtpe-mc12
setenv XTPE_LINK_TYPE dynamic
# OR: export XTPE_LINK_TYPE=dynamic
```

### 3.2.2 Cray Linux Environment (CLE) 4.0

Installing the `cade-prgenv` RPM adds the following line to the `*rc.local` files.

```
module use /opt/cray/xt-asyncpe/default/modulefiles;export XTOS_VERSION=4.0.version
```

The `XTOS_VERSION` is updated with each `cade-prgenv-4.0*` update.

Administrators can add additional settings after this line.

The default link type is `static`.

A CPU target must be defined. The choice of processor module depends on what system the user intends to use as a runtime system for the application they build. Systems that are used for cross-compiling for several different processor targets should set the default processor type to "barcelona," so that code defaults to being optimized for the lowest common denominator. Users on these systems should choose and load the target module for the actual processors they intend to use, to override the default and produce best performance.

The MPI module, `cray-mpich2`, is no longer loaded as part of the default `PrgEnv-*` product list. This is to accommodate site preferences for `cray-shmem` or `cray-mpich2` as the site default.

The CLE 4.0 environment supports only network target `gemini`.

To set the default programming environment to `PrgEnv-cray`, `link=dynamic`, `SHMEM`, and `cpu=mc12`, add the following lines to `/etc/bash.bashrc.local` and `/etc/csh.cshrc.local`:

```
module use /opt/cray/xt-asyncpe/default/modulefiles
module load PrgEnv-cray
module load craype-mc12
module load cray-shmem
setenv CRAYPE_LINK_TYPE dynamic
# OR: export CRAYPE_LINK_TYPE=dynamic
```

### 3.3 Using the Cray Application Developer's Environment

After you have installed the Cray Application Developer's Environment and Cray Application Developer's Environment Supplement on your standalone Linux system, follow these steps to use the environment.

#### Procedure 5. Using the Cray Application Developer's Environment

1. Initialize modules. Assuming `$MPATH` is the path in which you installed modules (typically `/opt/modules` or `/usr/share/modules`).

For `csh` or `tcsh` shells, add the following line to the system or user `.cshrc` file:

```
source $MPATH/init/modules.csh
```

For `bash` shells, add the following line to the system or user `.bashrc` file:

```
source $MPATH/init/sh
```

2. Load a network module, depending on desired target, if not already loaded by the `*rc.local` file. See [Configuring /etc/\\*rc.local files on page 24](#).

```
% module load craype-network-seastar
```

```
% module load craype-network-gemini
```

**Note:** In the absence a loaded network module, the default is `seastar` in the programming environment on CLE 3.1 systems and `gemini` in the programming environment on CLE 4.0 or later systems.

3. Load a CPU targeting module, depending on the types of CPUs used in the compute nodes on the Cray system for which you are developing code and if the CPU target is not already specified in the `*rc.local` file. For example, if your Cray system has compute nodes with 8-core Magny-Cours processors, type the following command.

```
% module load craype-mc8
```

4. Load the programming environment. The Cray Application Developer's Environment can provide several suites of C, C++, and Fortran compilers, depending on your licenses. Remember to specify the version of the programming environment corresponding to your Cray system CLE version, unless this is already set by default.

To load the Cray Compiling Environment (CCE), type the following command.

```
% module load PrgEnv-cray[ /version ]
```

To load the Portland Group (PGI) compilers, type the following command.

```
% module load PrgEnv-pgi[ /version ]
```

To load the PathScale Compiler Suite, type the following command.

```
% module load PrgEnv-pathscales[ /version ]
```

To load the Intel Composer, type the following command.

```
% module load PrgEnv-intel[ /version ]
```

To load the GNU (GCC) compilers, type the following command.

```
% module load PrgEnv-gnu[ /version ]
```

5. Compile and load your code.

The `CC`, `cc`, and `ftn` script drivers set up the correct libraries, header files, and options for each environment. After you choose a compiler by loading the module, no additional library or include paths are required.

For more information about using the Cray Application Developer's Environment to develop and compile code for use on Cray systems, see the *Cray Application Developer's Environment User's Guide*.