



Cray Compiling Environment 8.2 Release Overview and Installation Guide

S-5212-82

© 2013 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 and Sonexion are federally registered trademarks and Active Manager, Cascade, Cray Apprentice2, Cray Apprentice2 Desktop, Cray C++ Compiling System, Cray CS300, 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 XC30, Cray XD1, Cray XE, Cray XEm, Cray XE5, Cray XE5m, Cray XE6, Cray XE6m, Cray XK6, Cray XK6m, Cray XK7, Cray XMT, Cray XR1, Cray XT, Cray XTm, Cray XT3, Cray XT4, Cray XT5, Cray XT5_h, Cray XT5m, Cray XT6, Cray XT6m, CrayDoc, CrayPort, CRInform, ECOphlex, LibSci, NodeKARE, RapidArray, The Way to Better Science, Threadstorm, Urika, UNICOS/lc, and YarcData are trademarks of Cray Inc.

FlexNet is a trademark of Flexera Software. GNU is a trademark of The Free Software Foundation. Intel is a trademark of Intel Corporation in the United States and/or other countries. Google is a trademark of Google. ISO is a trademark of International Organization for Standardization (Organisation Internationale de Normalisation). Linux is a trademark of Linus Torvalds. OpenACC and NVIDIA are trademarks of NVIDIA Corporation. OpenMP is a trademark of OpenMP Architecture Review Board. All other trademarks are the property of their respective owners.

Contents

Page

Part I: Release Overview

Introduction [1]	7
1.1 Emphasis for the Cray Compiling Environment 8.2 Release	7
1.2 Cray Compiling Environment 8.2 Release Package Description	7
Software Enhancements [2]	9
2.1 Performance Enhancements	9
2.2 Accelerator Support	9
2.2.1 OpenACC Directives	9
2.3 PGAS Support	10
2.3.1 Coarray C++	10
2.3.2 UPC	10
2.4 Technical Specification on Further Interoperability of Fortran with C	10
2.5 Compiler Command Line Changes	10
2.6 Changed Defaults	12
2.7 New Compiler Directives	12
2.8 Library Changes	12
2.9 GNU/OpenMP Interoperability	13
Compatibilities and Differences [3]	15
Documentation [4]	17
4.1 Accessing Product Documentation	17
4.2 Cray-developed Books Provided with This Release	18
4.3 Additional Documentation Resources	18
4.4 Changes to Man Pages	18
4.4.1 New Cray Man Pages	18
4.4.2 Removed Cray Man Pages	19
4.4.3 Changed Cray Man Pages	19
4.5 Other Related Documents Available	20

	<i>Page</i>
Release Package [5]	21
5.1 Hardware and Software Requirements	21
5.2 Contents of the Release Package	22
5.3 Licensing	22
 Part II: Installation	
 Installing Cray Compiling Environment 8.2 [6]	27
6.1 Installing from DVD on System Management Workstation (SMW)	27
6.2 Installing the License Key	29
6.3 Using Cray Compiling Environment 8.2	30
 Appendix A Installing and Managing FlexNet	33
A.1 License Management Utilities and Files	33
A.2 Combining License Files	34
A.3 Resolving Problems	34
 Procedures	
Procedure 1. Installing the RPM files	27
Procedure 2. Adding a new key to a license file	30
 Tables	
Table 1. Books Provided with This Release	18
Table 2. Additional Documentation Resources	18
Table 3. Required CLE	21
Table 4. Required Asynchronous Products	21
Table 5. Optional Asynchronous Products	22
Table 6. Licensing Scope for Programming Environment Products	23

Part I: Release Overview

Introduction [1]

This document provides an overview of the Cray Compiling Environment 8.2 release for the Cray XE, Cray XK, and Cray XC30 systems.

This document does **not** describe hardware, software, installation of related products, or products that Cray does not provide.

1.1 Emphasis for the Cray Compiling Environment 8.2 Release

The Cray Compiling Environment (CCE) release provides the following key enhancements:

- Enhanced support for accelerators on Cray Systems, including support for the *OpenACC Application Programming Interface, Version 2.0*.
- Enhanced support for PGAS programming models. This release introduces *Coarray C++*, a template library that implements the coarray concept in C++. Also, this release offers preliminary support for the UPC 1.3 draft specification.
- The release supports the Intel™ Xeon™ processor E5-2600 v2 product family, code named *Ivy Bridge*.
- Performance improvements for Cray XE, Cray XK, and Cray XC30 systems.

1.2 Cray Compiling Environment 8.2 Release Package Description

The Cray Compiling Environment 8.2 consists of:

- Cray Fortran Compiler, version 8.2
- Cray C and C++ Compiler, version 8.2
- CrayLibs (libraries and utilities), version 8.2
- Cray documentation, described in [Chapter 4, Documentation on page 17](#)

All software is installed by means of scripts and RPM Package Manager (RPM) files. For more detail about Cray Compiling Environment 8.2 release package, refer to [Chapter 5, Release Package on page 21](#).

Software Enhancements [2]

This chapter describes the software enhancements provided with the Cray Compiling Environment 8.2 release.

2.1 Performance Enhancements

- Improvements to register allocation benefit loops with large bodies containing many independent computations.
- The performance of certain performance-critical math intrinsics was enhanced for Cray XC30 systems at default optimization. In some cases, additional speedups are available with the `-hfp3` and `-hfp4` options.
- Autothreaded loops and OpenMP loops with no `schedule` clause or a `schedule(auto)` clause will now use aligned scheduling when the loop is vectorized and would benefit from using aligned memory accesses, yielding improved performance. Aligned scheduling is a modified form of static scheduling that allows each participating thread to use aligned memory accesses safely.
- Enhancements to improve restructuring capability for very large code bodies.
- Improved automatic restructuring of transpose nests that run on the accelerator.
- Performance improvements for code using the Fortran `ASYNCHRONOUS` attribute.

2.2 Accelerator Support

2.2.1 OpenACC Directives

This release supports the *OpenACC Application Programming Interface, Version 2.0* specification. Refer to the OpenACC home page at <http://www.openacc-standard.org/>. Under the Downloads link, select the *OpenACC 2.0 Specification*.

See the `intro_openacc(7)` man page for the most current information regarding this implementation of OpenACC.

2.3 PGAS Support

2.3.1 Coarray C++

This release introduces *Coarray C++*, a template library that implements the coarray concept in C++. The template library specifications are contained on a set of *.html pages that the CCE installation copies to `/opt/cray/cce/version/doc/html/` on the Cray platform; they may be copied to any location which provides HTML web content for your site, or any location that can be accessed by site local web browsers. Also, see the *Cray C and C++ Reference Manual*.

2.3.2 UPC

This release offers preliminary support for the UPC 1.3 draft specification. The proposed UPC 1.3 standard is discussed on the UPC specification website, <http://code.google.com/p/upc-specification>. Predefined macros have been updated for the proposed standard. See the `intro_pgas(7)` man page for more information regarding support for PGAS programming models.

2.4 Technical Specification on Further Interoperability of Fortran with C

The CCE 8.2 Fortran compiler supports the technical specification [TS 29113 Further Interoperability of Fortran with C](#). A new C include file, `ISO_Fortran_binding.h`, defines a C descriptor structure that is designed to interoperate with a Fortran interface that has objects such as pointers and allocatable arrays. New C callable library functions are provided to access information and create these C descriptors. Please see the [TS 29113 Further Interoperability of Fortran with C](#) specification.

2.5 Compiler Command Line Changes

The following command line changes apply to CCE 8.2.

- `-h acc_model=option[:option]` . . . explicitly controls execution and memory model utilized by the accelerator support system.
- `-dp` and `-ep` options to control double precision type when default data size is 64 bits.

- `-h gasp[=opt[:opt]]` requests GASP (Global Address Space Performance Analysis) instrumentation. With no options specified, remote data accesses are profiled. `#pragma pupc [on|off]` has no effect in the current GASP implementation. When *opt* is specified, the compiler provides additional instrumentation as follows:
 - `local`
Enables instrumentation of events generated by shared local accesses. Instrumenting these events can add runtime overhead to the application.
 - `functions`
Enables function instrumentation. Sets IPA to 0.
- The `-hvector1` option no longer restricts vectorization to innermost loops.

Fortran Only

- `-dX` and `-eX` options control runtime initialization of module variables. If a module variable has initializers, implicit or explicit, and the variable has greater than 10,000 elements to be initialized, optionally create a new module procedure to do the initialization at runtime before MAIN is called. Enabling this option may significantly reduce compile time and reduce the size of the executable for some code, while increasing execution time. By default, this option is enabled. If performance is the only issue, disable this option.
- Fortran provides `-h` equivalents of `-O` options for compatibility with the C/C++ option set.
- The `-hcontiguous` option declares that every assumed shape array and array pointer target is contiguous, whether or not they have a CONTIGUOUS keyword, potentially increasing the range of permitted compiler optimizations.
- Fixed- and free-format source lines may be up to 1023 columns with `-N col`.

C Only

- Numeric values replace alphabetic values in the *level* specification of the `-G` option. `-G0` replaces `-Gn`, `-G1` replaces `-Gp`, `-G2` replaces `-Gf`, and `-G3` replaces `-Gfast`. The usage of alphabetic values results in a WARNING message and support will be removed in a future release.

2.6 Changed Defaults

- `-hpgas_runtime` is now default. The `-hpgas_runtime` option directs the compiler driver to link with the runtime libraries required when linking programs that use UPC, or coarrays.
- Full DWARF information is no longer generated by default. By default, only DWARF source line information is generated to support traceback analysis. Full DWARF information is still generated whenever a debugging command line option is enabled.
- `-ew` is now default for CCE Fortran compiler. This option enables support for automatic memory allocation for allocatable variables and arrays that are on the left hand side of intrinsic assignment statements. This may have a performance impact for certain codes. For best overall performance, use `-dw`, especially if also using `-hflex_mp=intolerant`.

Also, note that if a program is written assuming that it will be compiled using `-ew`, subsequently disabling automatic memory allocation (`-dw`) may cause runtime errors, such as segmentation faults, or it may result in wrong answers. If you suspect that a code may need changes in order to correctly use the `-dw` option, compile it using the `-RC` option, which adds runtime conformance checking.

2.7 New Compiler Directives

- When a `loop_info` directive/pragma has an `est_trips` clause with value 1, the compiler generates code specific to a single trip loop, subject to a runtime test.
- The `OPTIMIZE` directive overrides the optimization level set via Fortran command line. See the `OPTIMIZE(7)` man page.
- The `SAME_TBS` directive allows the Fortran compiler to generate more efficient code for assumed size arrays when two or more assumed shape dummy arguments to a given routine have the same rank, element type and sequences of low-bounds, extents, and strides. See the `SAME_TBS(7)` man page.
- The `VECTOR ALIGNED` and `VECTOR UNALIGNED` directives are available for CCE. See the `VECTOR(7)` man page.

2.8 Library Changes

- The Fortran `OPEN` statement now supports the `CONVERT` specifier which converts unformatted data between `BIG-` and `LITTLE-ENDIAN` representation.

2.9 GNU/OpenMP Interoperability

OpenMP object files and libraries compiled with `gcc` may now be linked with CCE-compiled applications. The linking must be done while the `PrgEnv-cray` module is loaded to ensure that CCE OpenMP runtime libraries are used.

Compatibilities and Differences [3]

This chapter describes compatibility issues and functionality changes to be aware of when upgrading from earlier releases of this software.

- UPC 1.3 requires a non-backwards compatible API change to the UPC runtime library. PGAS files compiled with CCE 8.2 are not binary compatible with those compiled with earlier versions of CCE. Attempting to link a pre-CCE 8.2 object file in the 8.2 CCE runtime environment will result in a **runtime** error message containing the text:

```
"... source file compiled with compiler expecting PGAS
runtime API version 1.1, but linked with PGAS runtime
using API version 2.0 ..."
```

If you see this error, recompile the indicated source file(s) with CCE 8.2 and relink your application.

A number of former extensions to UPC 1.2 have been standardized in UPC 1.3, including non-blocking bulk copies (`upc_nb.h`), privatizability (`upc_castable.h`) and timing (`upc_tick.h`) interfaces. These interfaces have been removed from the `upc_cray.h` header and moved into new headers as required by the UPC 1.3 specification. Additionally, some of the semantics and interfaces have been changed slightly, so existing users of these interfaces may need to update their applications. See the `intro_pgas(7)` man page.

- Support for an early draft of OpenMP accelerator directives was deprecated in CCE 8.1 and is removed in CCE 8.2. Please use the OpenACC accelerator directives instead. Support for OpenMP 4.0 accelerator directives is planned for a future CCE release.
- Some system headers were removed from `upc.h`, including `stdlib.h` which includes `sys/types.h`. Do not rely on `upc.h` to include other system headers. If your code requires `sys/types.h`, Include `stdint.h` if you were not previously doing so.
- The `SHORTLOOP` and `SHORTLOOP128` directives have been deprecated will be removed in the next major release. `LOOP_INFO mintrips(1) max_trips(64)` replaces `SHORTLOOP`. `LOOP_INFO mintrips(1) max_trips(128)` replaces `SHORTLOOP128`.

This chapter describes the documentation that supports the Cray Compiling Environment 8.2 release.

4.1 Accessing Product Documentation

With each software release, Cray provides books and man pages, and in some cases, third-party documentation. These documents are provided in the following ways:

- | | |
|---------------------------|--|
| CrayPort | CrayPort is the external Cray website for registered users that offers documentation for each product. CrayPort has portal pages for each product that contains links to all of the documents that are associated to that product. CrayPort enables you to quickly access and search Cray books, man pages, and in some cases, third-party documentation. You access CrayPort by using the following URL:

http://crayport.cray.com |
| CrayDoc | CrayDoc is the Cray documentation delivery system. CrayDoc enables you to quickly access and search Cray books, man pages, and in some cases, third-party documentation. Access the HTML and PDF documentation via CrayDoc at the following locations. <ul style="list-style-type: none">• The local network location defined by your system administrator• The CrayDoc public website: http://docs.cray.com |
| Man pages | Man pages are textual help files available from the command line on Cray machines. To access man pages, enter the man command followed by the name of the man page. For more information about man pages, see the man(1) man page by entering:

<code>% man man</code> |
| Third-party documentation | Third-party documentation that is not provided through CrayPort or CrayDoc is included with the third-party product. |

4.2 Cray-developed Books Provided with This Release

The books provided with this release are listed in [Table 1](#), which also indicates whether each book was updated. Books are provided in HTML and PDF formats.

Table 1. Books Provided with This Release

Book Title	Number	Updated
<i>Cray Compiling Environment Release Overview and Installation Guide</i> (this document)	S-5212-82	Yes
<i>Cray C and C++ Reference Manual</i>	S-2179-82	Yes
<i>Cray Fortran Reference Manual</i>	S-3901-82	Yes

4.3 Additional Documentation Resources

[Table 2](#) lists additional resources for obtaining documentation not included with this release package.

Table 2. Additional Documentation Resources

Product	Documentation Source
Example:	
GNU compilers	Documentation for the GNU C and Fortran compilers is available at http://gcc.gnu.org/onlinedocs/
glibc	glibc documentation is available at http://gcc.gnu.org/onlinedocs
GLIB	GLIB documentation is available at http://developer.gnome.org/glib/stable
RPM	RPM documentation is available at http://www.rpm.org

4.4 Changes to Man Pages

4.4.1 New Cray Man Pages

These man pages are new with this release:

- `fseek(3i)`
- `optimize(7)`
- `nofission(7)`
- `vector(7)`
- `same_tbs(7)`

- `upc_all_broadcast(3c)`
- `upc_all_exchange(3c)`
- `upc_all_gather(3c)`
- `upc_all_gather_all(3c)`
- `upc_all_permute(3c)`
- `upc_all_reduce(3c)`
- `upc_all_scatter(3c)`
- `upc_sync(3c)`
- `upc_flag_t(3c)`
- `upc_op_t(3c)`

4.4.2 Removed Cray Man Pages

The following Cray man pages were removed with this release:

- `upc_sync_nb(3c)`

4.4.3 Changed Cray Man Pages

The following Cray man pages were changed with this release:

- `craycc(1)`
- `crayCC(1)`
- `crayftn(1)`
- `intro_directives(7)`
- `intro_openacc(7)`
- `openacc.examples(7)`
- `intro_pgas(7)`
- `clonealways(7)`
- `loop_info(7)`
- `shortloop(7)`
- `abs(3i)`
- `c_loc(3i)`

4.5 Other Related Documents Available

The following publications contain additional information that may be helpful in setting up your Cray Compiling Environment 8.2 environment; they are not provided with this release but are supplied with other products purchased from Cray:

- *Cray Programming Environments Installation Guide* (S-2372)
- *Cray Programming Environment User's Guide* (S-2529)
- *Managing System Software for the Cray Linux Environment* (S-2393)

Release Package [5]

5.1 Hardware and Software Requirements

The CCE 8.2 release is supported on the following:

Table 3. Required CLE

System	CLE
Cray XE	4.0.UP01 and later
Cray XK	4.0.UP01 and later
Cray XC30	5.0.UP00 and later

The Cray Compiling Environment 8.2 release requires the following supporting asynchronous software products. Minimum required versions are listed below:

Table 4. Required Asynchronous Products

Product	Cray XE	Cray XK	Cray XC30
craype	N/A	N/A	1.02
xt-asyncpe	5.22	5.22	N/A
PMI	4.0.0	4.0.0	4.0.0
LibSci	12.0	12.0	12.0
FlexNet	11.10.0	11.10.0	11.10.0
gcc	4.4.4	4.4.4	4.4.4

Note: GNU GCC 4.4.4 must be installed, but must **not** be the default GCC.

The Cray Compiling Environment 8.2 release requires the following minimum versions of these **optional** products:

Table 5. Optional Asynchronous Products

Product	Cray XE, Cray XK, and Cray XC30 Systems
hdf5-netcdf	1.8.9/4.2.1
MPT	5.6
Perftools	6.1.1 (if MPT 6.0.x installed), 6.1.0 (if MPT 5.6.x installed)

5.2 Contents of the Release Package

The release package includes:

- Cray Fortran Compiler, version 8.2
- Cray C and C++ Compiler, version 8.2
- CrayLibs (libraries and utilities), version 8.2
- CrayDoc documentation, described in [Chapter 4, Documentation on page 17](#)

5.3 Licensing

The Cray Compiling Environment 8.2 is licensed under a software license agreement which is specific to the Cray Compiling Environment software. Upgrades to this product are provided only when a software support agreement for this Cray software is in place.

The software license agreement is enforced by FlexNet license manager software. [Table 6](#) shows the licensing scope for Cray Compiling Environment 8.2. A new license key is required for initially installing Cray Compiling Environment software and when upgrading from a previous release to a new major release. For information on installing the FlexNet Server Software, see [Appendix A, Installing and Managing FlexNet on page 33](#).

To request new FlexNet license manager keys for Cray Compiling Environment, contact license_keys@cray.com.

Table 6. Licensing Scope for Programming Environment Products

License for Product	Host Platform	Concurrent Users
Cray Compiling Environment 8.2	Cray XE, Cray XK, Cray XC30 systems	Unlimited or 5

For more information about contractual licensing and pricing, contact your Cray sales representative, or send e-mail to crayinfo@cray.com.

Part II: Installation

Installing Cray Compiling Environment 8.2 [6]

Cray Compiling Environment 8.2 is installed on the shared root. You must have root permissions in order to install this software. Cray Compiling Environment 8.2 requires that asynchronous products be installed on your system. See [Table 4](#).

The CCE software release consists of the downloadable rpm file:

- `cce-8.2.0-N.x86_64.rpm` or later

Please refer to the most recent version of the *Cray Programming Environments Installation Guide* (S-2372) available at <http://docs.cray.com/>. For installations on Cray esLogin external linux systems also refer to *Installing CLE Support Package on a Cray Development and Login (CDL) Node* (S-2528).

Note: To set CCE to be the default compiling environment use the `/opt/cray/admin-pe/set_default_files/set_default_cce_version` script.

This release introduces *Coarray C++*, a template library that implements the coarray concept in C++. The template library specifications are contained on a set of *.html pages that the CCE installation copies to `/opt/cray/cce/version/doc/html/` on the Cray platform; they may be copied to any location which provides HTML web content for your site, or any location that can be accessed by site local web browsers.

6.1 Installing from DVD on System Management Workstation (SMW)

Procedure 1. Installing the RPM files

1. Log on to the SMW as root.

```
% ssh root@smw
```

2. Load and mount the distribution media, if necessary.

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

3. Create a temporary directory on the boot node for the installation files, if one does not already exist.

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

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

```
smw:~# scp -pr /media/cdrom/cce-version.x86_64.rpm \  
boot:/tmp/install.cce
```

5. (Optional) If you need to install asynchronous products, copy these files as needed.

```
smw:~# scp -pr /media/cdrom/xt-gcc-version.x86_64.rpm \  
/media/cdrom/xt-asyncpe-version.i386.rpm \  
/media/cdrom/xt-libsci-version.x86_64.rpm \  
/media/cdrom/cray-mpt-version.x86_64.rpm \  
boot:/tmp/install.cce
```

6. Unmount and remove the distribution media.

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

7. Log into the boot node as root.

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

8. Change to your temporary directory.

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

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

```
boot001:/tmp/install.cce # mkdir -p /rr/current/software/install.cce  
boot001:/tmp/install.cce # cp -p cce-version.x86_64.rpm \  
/rr/current/software/install.cce
```

10. (Optional) If you are installing asynchronous products, copy these files as needed.

```
boot001:/tmp/install.cce # cp -p xt-gcc-version.x86_64.rpm \  
xt-asyncpe-version.i386.rpm \  
xt-libsci-version.x86_64.rpm \  
cray-mpt-version.x86_64.rpm \  
/rr/current/software/install.cce
```

11. Open an xtopview session.

```
boot001:/tmp/install.cce # xtopview
```

12. Change to the temporary directory you created on the shared root.

```
default:/# cd /software/install.cce
```

13. Use the rpm command to install the files.

Note: When running rpm from within xtopview, the rpm utility issues a warning that it cannot find /rr/current. This warning may safely be ignored.

Install CCE:

```
# rpm -ivh cce-8.2.0-version.x86_64.rpm
```

Set newly installed CCE to be default:

```
# /opt/cray/admin-pe/set_default_files/set_default_cce_8.2.0
```

(Optional) If installing asynchronous products, use these commands as needed:

```
# rpm -ivh xt-gcc-version.x86_64.rpm
# rpm -ivh xt-asyncpe-version.i386.rpm
# rpm -iv xt-libsci-version.x86_64.rpm
```

Set newly installed *product* to be default:

```
# /opt/cray/admin-pe/set_default_files/set_default_product_version
```

14. Exit from the `xtopview` session:

```
# exit
```

15. Log out of the boot node:

```
# exit
```

16. Log out of the SMW.

```
# exit
```

6.2 Installing the License Key

To activate your software license, you must insert the FlexNet software license key information provided by Cray into a FlexNet license file on your system. The FlexNet license file contains data that is used to determine whether a licensed software product is allowed to run.

The license file contains the following information:

- Your FlexNet software license key for your Cray product
- Initial installation instructions
- Update instructions
- License manager utilities
- Technical Support information

Cray recommends that you name your license file `/opt/cray/cce/cce.lic`. These instructions assume that the FlexNet license manager is already running, that your license file is located in the directory `/opt/cray/cce`, and that the file is named `cce.lic`.

If you do not have FlexNet license manager already installed on your network, go to [Appendix A, Installing and Managing FlexNet on page 33](#).

Procedure 2. Adding a new key to a license file

1. Log in to your license server as `admin` or `root`.

2. Locate your existing license file, if any.

```
# ls /opt/cray/cce
```

If the directory does not exist, make it.

```
# mkdir -p /opt/cray/cce
```

3. In `/opt/cray/cce`, create the plain text file `cce.lic`. Copy your FlexNet license key you received from Cray (typically in an e-mail message) to `cce.lic`.

4. Set the file access permissions to 644.

```
# chmod 644 /opt/cray/cce/cce.lic
```

5. Update your FlexNet license server to use the new key. Verify that the license server is running.

```
# lmstat
```

If the server is not running, go to [Appendix A, Installing and Managing FlexNet on page 33](#).

Assuming the server is running, reread the license file.

```
# lmread
```

Your license is now ready to use.

6.3 Using Cray Compiling Environment 8.2

After the Cray Compiling Environment 8.2 rpm files are installed and the license is activated, load the `PrgEnv-cray` module to use CCE.

Load the `craype-accel-nvidiaversion` module to set the necessary compiler options and targets to use the accelerator.

Use either the `ftn` or `cc` command to compile. The module environment forces dynamic linking.

Because of the multiple compiling environments potentially available on Cray systems, the `ftn(1)`, `cc(1)`, and `CC(1)` man pages provide basic introductions to the compiler environment. For information about the Cray compiler command-line options, see the `crayftn(1)`, `craycc(1)`, and `crayCC(1)` man pages.

For more detailed information about the Cray compiler options, directives, pragmas, and optimizations, see *Cray Fortran Reference Manual* and *Cray C and C++ Reference Manual*.

The Cray Compiling Environment 8.2 compilation targets are `istanbul`, `mc8`, `mc12`, `interlagos`, `interlagos-cu`, `abudhabi`, `abudhabi-cu`, `ivybridge` and `sandybridge`.

The targeting modules (`craype-ivybridge`, or `craype-interlagos`, for example) set *target_system*. If the *target_system* is set during compilation of any source file, the same *target_system* must also be specified during linking and loading. For example, if users are compiling code for use on `ivybridge` nodes, they must load the `craype-ivybridge` module and use the same module at link and load time.

Installing and Managing FlexNet [A]

The Cray Compiling Environment requires FlexNet license manager.

If you do not already have this software installed on your system, download these packages provided in the Cray Compiling Environment 8.2 package:

- `cray-flexnet-installation-instructions.txt`
- `cray-flexnet-daemon-11.10.0-1.0000.3631.4.1.gem.x86_64.rpm`
- `cray-flexnet-manager-11.10.0-1.0000.3631.4.1.gem.x86_64.rpm`
- `cray-flexnet-publisher-switch-11.10.0-1.0000.3631.4.1.gem.x86_64.rpm`
- `cray-flexnet-utils-11.10.0-1.0000.3631.4.1.gem.x86_64.rpm`

Note: You do not need new licenses for CCE or PerfTools when upgrading the FlexNet Server Software only.

Follow the instructions in the `cray-flexnet-installation-instructions.txt` file.

A.1 License Management Utilities and Files

Use the following commands to administer the license manager software:

<code>lmdown</code>	Shut down the licensing daemons
<code>lmgrd</code>	Invoke the licensing daemon
<code>lmhostid</code>	Display the host ID of a system
<code>lmremove</code>	Return a license to the license pool
<code>lmreread</code>	Update license daemons with new license data
<code>lmstat</code>	Report current status of the license daemon
<code>lmver</code>	Display the FlexNet version being used

Note: These commands are actually arguments passed to `lmutil`. If any do not exist on your license server, you can create them by making a symbolic link to `lmutil`. For example:

```
ln -s lmutil <command name>
```

A.2 Combining License Files

If you are using three-server redundancy, you must use separate license files with different TCP/IP ports for the licensed software products from each vendor. (Licensed software products from the same vendor can share license files.)

If you are using a single server, you can combine license files. To do so, edit all license files that have matching `SERVER` lines (the port number does not have to match) into one file, and delete the extra `SERVER` lines. Place all `DAEMON` lines after the `SERVER` line.

A.3 Resolving Problems

If you are having problems with your license key, check your installation first. Then examine the log file:

- Verify that the license file is `/opt/cray/cce/cce.lic`. If this is not the location or name of your license file, each user must have `CRAYLMD_LICENSE_FILE` set to the path and name that you are using.
- Verify that the license file has a `FEATURE` line for the product you are trying to use.
- Use the `ps` command and the `lmstat` utility to verify that the servers specified in your license file are running.

When you examine the log file, look for the following messages:

- `Inconsistent encryption code for <name>` — The information encoded in the encryption code for the specified feature, server, or daemon is inconsistent with the information provided in the license file. Recheck the associated `FEATURE`, `SERVER`, or `DAEMON` line in your license file.
- `license daemon: execl failed` — The `lmgrd` path specified on the `DAEMON` line in the license file is not valid. Verify that the license file contains the correct path.
- `Retrying socket bind` — Either the TCP port number is already in use by another process, two `lmgrd` daemons were started with license files that specify the same TCP port, or the port is waiting to timeout after a recently executed `lmgrd` command. This message typically appears when you stop and then immediately try to restart `lmgrd`. In this case, wait a few minutes, to allow TCP time to relinquish the port.

If this fails to correct the problem, check the `SERVER` line in the license file. If another process needs the specified port, edit the license file to specify a different port. If no port is specified, the default port is 27000.