# Contents

**Preface**
- Accessing Product Documentation ................................................................. v
- Conventions ........................................................................................................ vi
- Reader Comments ................................................................................................. vii

**Part I: Release Overview**

**Introduction [1]**
- Emphasis for the Cray Programming Environment 5.4 Releases .......................... 3

**Software Enhancements [2]**
- Performance Enhancements .................................................................................. 5
  - Replace if else Statements with Select Statements ............................................. 5
  - Align on Cache Line Boundary .......................................................................... 5
  - Inline C and C++ Functions ............................................................................. 5
- CrayPat Features .................................................................................................. 5
  - New API Functions .......................................................................................... 6
  - New Environment Variables PAT_RT_REGION_MAX and PAT_RT_REGION_STKSZ .................................................. 6
  - Environment Variable PAT_ROOT ...................................................................... 6
- tracebk Function Added ....................................................................................... 6
- Cray Fortran Features .......................................................................................... 7
  - Fortran 2003 Features .................................................................................... 7
- Cray C and C++ Features .................................................................................... 8
  - Tail Recursion Optimization .......................................................................... 8
  - Automatic Inlining ......................................................................................... 9
- Scientific Library Features .................................................................................. 9
  - Complex-to-complex FFTs contain butterflies for radices 7, 11, and 13 .......... 9
<table>
<thead>
<tr>
<th>Topic</th>
<th>Page</th>
</tr>
</thead>
<tbody>
<tr>
<td>LAPACK Built with Inlined BLAS Routines</td>
<td>9</td>
</tr>
<tr>
<td><strong>Compatibilities and Differences [3]</strong></td>
<td>11</td>
</tr>
<tr>
<td>C and C++ Compatibilities and Differences</td>
<td>11</td>
</tr>
<tr>
<td><strong>Limitations [4]</strong></td>
<td>13</td>
</tr>
<tr>
<td>Cray Apprentice(^2) Data Visualization Tool</td>
<td>13</td>
</tr>
<tr>
<td><strong>Documentation [5]</strong></td>
<td>15</td>
</tr>
<tr>
<td>CrayDoc Documentation Delivery System</td>
<td>15</td>
</tr>
<tr>
<td>Accessing Product Documentation</td>
<td>15</td>
</tr>
<tr>
<td>Ordering Documentation</td>
<td>16</td>
</tr>
<tr>
<td>Error Message Explanations</td>
<td>16</td>
</tr>
<tr>
<td>Books Provided with This Release</td>
<td>16</td>
</tr>
<tr>
<td>Third-party Books Provided with This Release</td>
<td>18</td>
</tr>
<tr>
<td>Changes to the Man Pages Document Set Since the Cray Programming Environment 5.3 Releases</td>
<td>18</td>
</tr>
<tr>
<td>New Cray Man Pages</td>
<td>18</td>
</tr>
<tr>
<td><strong>Release Package [6]</strong></td>
<td>19</td>
</tr>
<tr>
<td>Hardware and Software Requirements</td>
<td>19</td>
</tr>
<tr>
<td>Contents of the Release Packages</td>
<td>19</td>
</tr>
<tr>
<td>Licensing</td>
<td>21</td>
</tr>
<tr>
<td>Ordering Software</td>
<td>22</td>
</tr>
<tr>
<td><strong>Customer Services [7]</strong></td>
<td>23</td>
</tr>
<tr>
<td>Technical Assistance with Software Problems</td>
<td>23</td>
</tr>
<tr>
<td>CRInform System</td>
<td>23</td>
</tr>
<tr>
<td>Training</td>
<td>24</td>
</tr>
<tr>
<td><em>Cray Service Bulletin</em> (CRSB)</td>
<td>25</td>
</tr>
<tr>
<td>Cray Public Website</td>
<td>25</td>
</tr>
</tbody>
</table>
Part II: Installation

Installing on Cray X1 Series Systems [8] 29
Installation Requirements .............................................. 30
/opt/ctl File System Requirements ................................. 31
Product Directories ...................................................... 32
Changes the Installation Makes to Your Programming Environment ................. 33
Choosing Your Installation Tool .................................... 34
Installing with CIT ...................................................... 34
Installing with opt_install ............................................ 36
Installing, Configuring, and Accessing the Trigger Software ......................... 38
Finishing the Programming Environment Installation ................................ 42
  Returning the CPES to a Secure State ............................ 43
  Copying System Libraries and Include Files to the CPES ....................... 43
  Making a Non-default Product the Default Product .......................... 44
  Adding Site-specific Options to the Fortran Command line ..................... 45
  Updating the Modules Environment ................................ 46

Miscellaneous Pre- or Post-installation Tasks [9] 47
Granting Root Read and Write Permissions to /opt/ctl ............................. 47
Providing Access to Files that the Cray Programming Environment Does Not Include .... 48
Providing Access to the remps Command ................................ 50

Index 51

Tables
Table 1. Books Provided with This Release .................................. 17
Table 2. Separately Orderable Books ........................................ 18
Table 3. Licensing Scope for Programming Environment Products ................ 21
Table 4. Solaris Based Cray Programming Environment 5.4 Releases Disk Space Requirements . 31
Table 5. Lines to Edit in trigexecd.cfg .................................... 40
Table 6. Lines to Edit in trigsnd.cfg ..................................... 40
Table 7. Source and Destination System Libraries and Include Files . . . . . . . . . 44
The information in this preface is common to Cray documentation provided with this software release.

**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:

- **CrayDoc**, the Cray documentation delivery system that allows you to quickly access and search Cray books, man pages, and in some cases, third-party documentation—Access this HTML and PDF documentation via CrayDoc at the following URLs:
  - The local network location defined by your system administrator
  - The CrayDoc public website: docs.cray.com

- **Man pages**—Access man pages by entering 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:

  `%% man man`

- **Third-party documentation not provided through CrayDoc**—Access this documentation, if any, according to the information provided with that product.
## Conventions

These conventions are used throughout Cray documentation:

<table>
<thead>
<tr>
<th><strong>Convention</strong></th>
<th><strong>Meaning</strong></th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>command</strong></td>
<td>This fixed-space font denotes literal items, such as file names, pathnames, man page names, command names, and programming language elements.</td>
</tr>
<tr>
<td><strong>variable</strong></td>
<td>Italic typeface indicates an element that you will replace with a specific value. For instance, you may replace <em>filename</em> with the name <em>datafile</em> in your program. It also denotes a word or concept being defined.</td>
</tr>
<tr>
<td><strong>user input</strong></td>
<td>This bold, fixed-space font denotes literal items that the user enters in interactive sessions. Output is shown in nonbold, fixed-space font.</td>
</tr>
<tr>
<td><strong>[]</strong></td>
<td>Brackets enclose optional portions of a syntax representation for a command, library routine, system call, and so on.</td>
</tr>
<tr>
<td><strong>...</strong></td>
<td>Ellipses indicate that a preceding element can be repeated.</td>
</tr>
<tr>
<td><strong>name(N)</strong></td>
<td>Denotes man pages that provide system and programming reference information. Each man page is referred to by its name followed by a section number in parentheses.</td>
</tr>
</tbody>
</table>

Enter:

```
% man man
```

to see the meaning of each section number for your particular system.
Reader Comments

Contact us with any comments that will help us to improve the accuracy and usability of this document. Be sure to include the title and number of the document with your comments. We value your comments and will respond to them promptly. Contact us in any of the following ways:

E-mail:
docs@cray.com

Telephone (inside U.S., Canada):
1-800-950-2729 (Cray Customer Support Center)

Telephone (outside U.S., Canada):
+1-715-726-4993 (Cray Customer Support Center)

Mail:
Software Publications
Cray Inc.
1340 Mendota Heights Road
Mendota Heights, MN 55120-1128
USA
Part I: Release Overview
This document provides an overview of the Cray Programming Environment 5.4 releases and the installation procedures for these systems:

- Cray X1 series systems running the UNICOS/mp 2.5 release or later
- Sun systems running Solaris 8.0 or later
- Linux systems running Red Hat Enterprise Linux Version 3 or later, which contains the Linux 2.4 kernel hosting a Cray X1 series Programming Environment (Linux based) product
- Cray XD1 systems running SuSE Linux 9 and kernel version 2.6.5.

This information is intended for system administrators receiving their first release of this product or upgrading from a previous release and assumes that the administrator has a good understanding of Cray system administration. This information is also intended for application programmers to learn about software enhancement information.

1.1 Emphasis for the Cray Programming Environment 5.4 Releases

The key reasons for this release are:

- Implementation of selected Fortran 2003 language features
- Cray C and C++ compiler support of automatic, explicit and combined inlining
- Tail recursion optimization
This chapter describes the software enhancements that have been made since the last release of this software.

Note: Documentation changes since the last release of this software are described in Chapter 5, page 15.

2.1 Performance Enhancements

The Cray Programming Environment 5.4 releases provide the following new compiler enhancements, applied automatically during compilation or by compiler command options, to improve run-time performance of Fortran, C, and C++ applications, except where noted.

2.1.1 Replace if else Statements with Select Statements

The compilers may replace conditional blocks of code with equivalent select statements to avoid branch overhead. The transformation is made only if all expressions are guaranteed safe to evaluate and the transformed code is more efficient than the original code. No user action is required.

2.1.2 Align on Cache Line Boundary

The Fortran ALLOCATE statement aligns memory allocations on a cache line boundary when the size of the allocation exceeds a system-defined number of bytes. No user action is required.

2.1.3 Inline C and C++ Functions

The C and C++ compilers now support automatic inlining, explicit inlining, and combined inlining. For more information about inlining of C and C++ functions, refer to Section 2.5.2, page 9.

2.2 CrayPat Features

The CrayPat performance analysis tool was enhanced.
2.2.1 New API Functions

The following new API functions were added:

- **PAT_region_begin**
  
  \[(\text{int id, const char } * \text{label})\]

- **PAT_region_end**
  
  Defines the boundaries of a region. For each region, a summary of activity including time and hardware performance counters, if selected, is produced. The argument *id* must be greater than zero and it assigns a numerical value to the region. All *ids* must be unique across the entire program. The argument *label* assigns a character string to the region allowing for easier identification of the region in the report.

2.2.2 New Environment Variables **PAT_RT_REGION_MAX** and **PAT_RT_REGION_STKSZ**

The following two new run-time environment variables affect region processing:

- **PAT_RT_REGION_MAX**
  
  Specifies the largest numerical id that may be used as an argument to the CrayPat API functions **PAT_region_begin** and **PAT_region_end**. Values greater than this cause the API function to be ignored. The default is 64.

- **PAT_RT_REGION_STKSZ**
  
  Specifies the depth of the stack for which the CrayPat API function **PAT_region_begin** and **PAT_region_end** are maintained. It is the maximum number of consecutive **PAT_region_begin** references that can be made without an intervening **PAT_region_end**. The default is 128.

2.2.3 Environment Variable **PAT_ROOT**

The **PAT_SV2** environment variable has been renamed **PAT_ROOT**.

2.3 tracebk Function Added

The **tracebk(3c)** function was added to improve user originated tracebacks. The **tracebk** function prints a traceback beginning with its caller and ending at "Start-up."
By default, the `tracebk` signal handler is not installed when the core file size limit is greater than zero.

By default, the `tracebk` signal handler is not installed for command mode executables.

A `$TRACEBK` environment variable value greater than zero forces installation of `tracebk` handler, overriding these defaults.

You can enable inline traceback via the `tracebk()` function.

**Note:** Use of this function requires both Cray C++ Programming Environment 5.4 release or Cray Fortran Programming Environment 5.4 release and UNICOS/mp 3.0 release.

See the `tracebk(3c)` and `signal(7)` man pages provided with the UNICOS/mp 3.0 release, and the `tracebk(3f)` man page, provided with the Cray Fortran Programming Environment 5.4 release for additional information.

---

### 2.4 Cray Fortran Features

The Cray Fortran Programming Environment release includes these new features.

#### 2.4.1 Fortran 2003 Features

The Cray Fortran Compiler has been enhanced to support these Fortran 2003 features:

- The `ERRMSG` and `SOURCE` specifiers on the `ALLOCATE` statement
- The `IOMSG` specifier on `OPEN`, `CLOSE`, `INQUIRE`, `READ`, `WRITE`, `BACKSPACE`, `ENDFILE`, `REWIND`, and `FLUSH` statements
- The `SIZE` specifier on the `INQUIRE` statement
- The `ABSTRACT INTERFACE` statement
- Procedure pointers in pointer assignments and calls


---

"S–5212–54"
2.5 Cray C and C++ Features

The Cray C and Cray C++ Programming Environment releases include these new features. For more information about the new C or C++ features, refer to Cray C and C++ Reference Manual and the cc(1) man page.

2.5.1 Tail Recursion Optimization

The tail recursion optimization replaces a recursive call with a direct goto statement when there is no executable code following the call, thereby eliminating the call overhead.

The Cray C and C++ compilers automatically apply this optimization when any level of inlining is enabled (that is, when the -h ipan option is used, where $n$ is greater than 0).

For example, the compiler will apply tail recursion optimization to the call of factorial_tail_2:

```c
int factorial_tail_2( int n, int m )
{
    int factorial;
    
    if( n <= 1 ) {
        factorial = m;
    }
    else {
        factorial = factorial_tail_2( n-1, n*m );
    }
    return (factorial);
}
```

```c
int factorial_tail_1( int n )
{
    int factorial;
    factorial = factorial_tail_2( n, 1 );
    return(factorial);
}
```
2.5.2 Automatic Inlining

The C and C++ compilers now support automatic inlining, explicit inlining, and combined inlining, as follows:

- **Automatic inlining** is invoked by the `-h ipan` compiler command option. This option allows the compiler to automatically select, depending on the inlining level `n`, the functions to inline. Each `n` is a different set of heuristics. For more information, refer to *Cray C and C++ Reference Manual*. The candidates for expansion are all those functions that are present in the input file to the compile step.

- **Explicit inlining** is invoked by the `-h ipafrom=source [:source ]` compiler command option. The source arguments identify the files and/or directories that contain the functions to consider for inlining. Only those functions present in source are candidates for inlining. When a call is encountered to a function that resides in source, an attempt will be made to expand the function in place at that call site.

- **Combined inlining** is a combination of automatic inlining and explicit inlining. The only candidates for expansion are those functions that reside in source. The rules that apply to deciding whether we inline are defined by the `-h ipan` setting.

2.6 Scientific Library Features

The scientific library routines, called LibSci, contain the following new features.

2.6.1 Complex-to-complex FFTs contain butterflies for radices 7, 11, and 13

Complex-to-complex FFTs contain butterflies for radices 7, 11, and 13. Applications that previously involved complex-to-complex FFTs with FFT lengths containing products of powers of these radices will run faster than with previous releases. This also includes the convolution routine `CCNVLF`, which effects a complex convolution via the Fourier transform.

2.6.2 LAPACK Built with Inlined BLAS Routines

LAPACK was built with Level 1 BLAS and some level 2 BLAS inlined.
This chapter describes compatibility issues and functionality changes to be aware of when using this software after upgrading from the previous release of this software.

### 3.1 C and C++ Compatibilities and Differences

The size of some object files created by the Programming Environment 5.4 C or C++ compiler may be somewhat larger than those created using a previous C or C++ compiler. This is primarily due to a change in how individual functions are processed and should not be a concern.
This chapter describes limitations with the release.

4.1 Cray Apprentice² Data Visualization Tool

Because of Trigger environment and X11 forwarding issues, the Cray Apprentice² data visualization tool does not work in high-security environments where the CPES is not accessible via the customer network. This limitation is expected to be removed in a future Cray Programming Environments update package.
This chapter describes the documentation that supports the Cray Programming Environment 5.4 releases.

5.1 CrayDoc Documentation Delivery System

The CrayDoc documentation delivery system, along with product documentation, is provided with each Cray software release. The CrayDoc software runs on any operating system based on UNIX systems or systems like UNIX including Mac OS X, Linux, BSD, and anywhere else that Perl and Apache can be compiled from source code with freely available (GNU) tools. The installation and administration of the CrayDoc server software and Cray documentation are described in *CrayDoc Installation and Administration Guide*.

5.2 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:

- CrayDoc, the Cray documentation delivery system that allows you to quickly access and search Cray books, man pages, and in some cases, third-party documentation—Access this HTML and PDF documentation via CrayDoc at the following URLs:
  - The local network location defined by your system administrator
  - The CrayDoc public website: docs.cray.com
- Man pages—Access man pages by entering 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:
  ```
  % man man
  ```
- Third-party documentation not provided through CrayDoc—Access this documentation, if any, according to the information provided with that product.
5.3 Ordering Documentation

To order Cray software documentation, contact your Cray representative or contact the Cray Software Distribution Center in any of the following ways:

E-mail:
orderdsk@cray.com

Telephone (inside U.S., Canada):
1–800–284–2729 (BUG CRAY), then 605–9100

Telephone (outside U.S., Canada):
+1–651–605–9100

Fax:
+1–651–605–9001

Mail:
Software Distribution Center
Cray Inc.
1340 Mendota Heights Road
Mendota Heights, MN 55120–1128
USA

5.4 Error Message Explanations

Access explanations of error messages by entering the `explain msgid` command, where `msgid` is the message ID string in the error message. For more information, refer to the `explain(1)` man page.

5.5 Books Provided with This Release

The books provided with this release or that are separately orderable are listed in Table 1 and Table 2, respectively, which also notes whether each book was updated and whether it is also provided in hard copy. Most books are provided in HTML and all are provided in PDF.

Note: If an errata for the release is required, it includes changes identified after the documentation for this release was packaged. If present, the errata is provided in printed hardcopy format in the release package. Also, contact your Cray representative for other possible late problems published in Field Notices (FNs).
Table 1. Books Provided with This Release

<table>
<thead>
<tr>
<th>Book Title</th>
<th>Number</th>
<th>Updated</th>
<th>Printed</th>
</tr>
</thead>
<tbody>
<tr>
<td>Cray Programming Environment Releases Overview and Installation Guide</td>
<td>S–5212–54</td>
<td>Yes</td>
<td>Yes</td>
</tr>
<tr>
<td>Cray X1 Series System Overview</td>
<td>S–2346–25</td>
<td>No</td>
<td>No</td>
</tr>
<tr>
<td>Optimizing Applications on Cray X1 Series Systems</td>
<td>S–2315–54</td>
<td>Yes</td>
<td>No</td>
</tr>
<tr>
<td>Migrating Applications to the Cray X1 Series Systems</td>
<td>S–2378–54</td>
<td>Yes</td>
<td>No</td>
</tr>
<tr>
<td>Cray X1 User Environment Differences</td>
<td>S–2310–52</td>
<td>No</td>
<td>No</td>
</tr>
<tr>
<td>Cray Fortran Compiler Commands and Directives Reference Manual</td>
<td>S–3901–54</td>
<td>Yes</td>
<td>No</td>
</tr>
<tr>
<td>Fortran Language Reference Manual, Volume 1</td>
<td>S–3692–54</td>
<td>Yes</td>
<td>No</td>
</tr>
<tr>
<td>Fortran Language Reference Manual, Volume 2</td>
<td>S–3693–54</td>
<td>Yes</td>
<td>No</td>
</tr>
<tr>
<td>Fortran Language Reference Manual, Volume 3</td>
<td>S–3694–54</td>
<td>Yes</td>
<td>No</td>
</tr>
<tr>
<td>Cray C and C++ Reference Manual</td>
<td>S–2179–54</td>
<td>Yes</td>
<td>No</td>
</tr>
<tr>
<td>Fortran Application Programmer’s I/O Reference Manual</td>
<td>S–2351–53</td>
<td>No</td>
<td>No</td>
</tr>
<tr>
<td>Dinkum C++ Library Documentation</td>
<td>S–6506–42</td>
<td>No</td>
<td>No</td>
</tr>
<tr>
<td>Cray Assembly Language (CAL) for Cray X1 Systems Reference Manual</td>
<td>S–2314–51</td>
<td>No</td>
<td>No</td>
</tr>
<tr>
<td>Common Installation Tool (CIT) Reference Card</td>
<td>S–2218–20</td>
<td>No</td>
<td>No</td>
</tr>
<tr>
<td>CrayDoc Installation and Administration Guide</td>
<td>S–2340–40</td>
<td>Yes</td>
<td>Yes</td>
</tr>
</tbody>
</table>
Table 2. Separately Orderable Books

<table>
<thead>
<tr>
<th>Book Title</th>
<th>Number</th>
<th>Updated</th>
<th>Printed</th>
</tr>
</thead>
<tbody>
<tr>
<td>Solaris Based Cray X1 Series Systems Programming Environments</td>
<td>S–2313–53</td>
<td>No</td>
<td>Yes</td>
</tr>
<tr>
<td>Differences and Installation Guide</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Linux Based Cray X1 Series Systems Programming Environments</td>
<td>S–2428–54</td>
<td>Yes</td>
<td>Yes</td>
</tr>
<tr>
<td>Differences and Installation Guide</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Cray Bioinformatics Library Release Overview and Installation</td>
<td>S–2424–23</td>
<td>Yes</td>
<td>No</td>
</tr>
<tr>
<td>Guide</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Cray Message Passing Toolkit Release Overview</td>
<td>S–3689–24</td>
<td>No</td>
<td>Yes</td>
</tr>
</tbody>
</table>

The X11 documentation is included with the X Window System X11 libraries package.

For information about Etnus TotalView, refer to TotalView Release Overview and Installation Guide.

5.6 Third-party Books Provided with This Release

The Dinkum C++ Library Documentation is provided with this release.

5.7 Changes to the Man Pages Document Set Since the Cray Programming Environment 5.3 Releases

The following subsection contains new man page information.

5.7.1 New Cray Man Pages

The following Cray man page is new with this release:

folr(3s) Sorts first-order linear recurrences
This chapter contains the following information about the Programming Environment 5.4 releases:

- Hardware and software requirements
- Contents of the release package
- Licensing
- Ordering software

6.1 Hardware and Software Requirements

The Cray Programming Environment 5.4 releases apply to:

- Cray X1 series systems running the UNICOS/mp 2.5 release or later
- Solaris systems running Solaris 8.0 or later
- Linux systems running Red Hat Enterprise Linux Version 3 or later, which contains the Linux 2.4 kernel hosting a Cray X1 series Programming Environment (Linux based) product
- Cray XD1 systems running SuSE Linux 9 and kernel version 2.6.5.

Note: Cray Programming Environment 5.4 releases running on Cray X1 series systems are dependent on the Trigger 2.0.0.5 software. Refer to Section 8.8, page 38 for information about installing and configuring the Trigger product.

6.2 Contents of the Release Packages

The release packages of the Cray Programming Environment 5.4 releases for Cray X1 series systems, Solaris based systems, and Linux based systems that contain the following products, except where noted:

- One or more of these compilers (depending on your contractual license):
  - Cray Fortran Compiler, version 5.4 (for the Cray Fortran Programming Environment release)
  - Cray C Compiler, version 8.4 and Cray C++ Compiler, version 5.4 (for the Cray C++ Programming Environment release)
• CrayLibs, version 5.4
• LibSci, version 5.4
• CrayTools, version 5.4 that contains CrayPat, Cray loader, and Cray Apprentice²

    Note: Cray Apprentice² is not included with the Linux based Cray Programming Environments.

• CrayDoc software suite and the documentation, described in Chapter 5, page 15

Also included with the Cray Programming Environment 5.4 release packages are other related products for Cray X1 series systems, Solaris based systems, and Linux based systems, except where noted:

• Cray Bioinformatics Library, version 2.3.
• Cray Assembly Language (CAL), version 1.2.
• Modules, version 3.1.6. (not included with the Linux based Cray Programming Environments)
• Message Passing Toolkit (MPT) 2.4 release.

    This is included with the Solaris based Cray Programming Environments and the Linux based Cray Programming Environments; but is a separately orderable and licensed product for Cray X1 series systems. Refer to Cray Message Passing Toolkit Release Overview for more information about the MPT release.
• Motif, version 2.1.
• X Window System X11 libraries, version 6.6.
• Trigger, version 2.0.0.5 (not included with and not needed by the Solaris based or Linux based Cray Programming Environments).
6.3 Licensing

The Cray Fortran Programming Environment and the Cray C++ Programming Environment are contractually licensed as separate products for the Cray X1 series systems, Solaris systems, and Linux systems. For Cray X1 series customers who license either programming environment for Solaris or Linux systems, a prerequisite Programming Environment license of the same type on the Cray X1 series system is required.

Table 3 shows the licensing scope for each Cray programming environment product.

Table 3. Licensing Scope for Programming Environment Products

<table>
<thead>
<tr>
<th>License for Product</th>
<th>Host platform</th>
<th>Concurrent users</th>
<th>MPT License</th>
</tr>
</thead>
<tbody>
<tr>
<td>Cray Fortran Programming Environment</td>
<td>CPES</td>
<td>Unlimited</td>
<td>Separate</td>
</tr>
<tr>
<td>Cray C++ Programming Environment</td>
<td>CPES</td>
<td>Unlimited</td>
<td>Separate</td>
</tr>
<tr>
<td>Cray Fortran Programming Environment (Solaris based)</td>
<td>Solaris</td>
<td>5 users or unlimited</td>
<td>Bundled</td>
</tr>
<tr>
<td>Cray C++ Programming Environment (Solaris based)</td>
<td>Solaris</td>
<td>5 users or unlimited</td>
<td>Bundled</td>
</tr>
<tr>
<td>Cray Fortran Programming Environment (Linux based)</td>
<td>Linux</td>
<td>Unlimited</td>
<td>Bundled</td>
</tr>
<tr>
<td>Cray C++ Programming Environment (Linux based)</td>
<td>Linux</td>
<td>Unlimited</td>
<td>Bundled</td>
</tr>
</tbody>
</table>

Only binary code licenses are available for the Cray Programming Environment releases.

Upgrades to all products are provided only when a software support agreement for Cray software is in place.

Note: The Message Passing Toolkit (MPT) and Etnus TotalView products must be licensed separately and ordered separately from the Cray Programming Environment 5.4 releases package for Cray X1 series systems. Contractual rights to Message Passing Toolkit (MPT) on the Solaris and Linux systems are included, respectively, under the licenses for the Solaris based and Linux based Cray Programming Environments. Refer to TotalView Release Overview and Installation Guide and Cray Message Passing Toolkit Release Overview.
For more information about licensing and pricing, contact your Cray sales representative, or send e-mail to crayinfo@cray.com.

Customers outside the United States and Canada must sign a Letter of Assurance before software can be shipped to them. For questions about whether you have signed this agreement, or questions about which software requires this letter, send e-mail to crayinfo@cray.com.

6.4 Ordering Software

This release package is distributed by order only to customers who have signed a license agreement for the Cray software that includes this product. The most current revision of the release package is supplied. To receive any upgrades to a given Cray product, the customer must also have a signed support agreement for this Cray software.

You can order the release package from the Cray Software Distribution Center in any of the following ways:

E-mail:
orderdsk@cray.com

CRInform (for subscribers):
crinform.cray.com

Click on the Order Cray Software link.

Telephone (inside U.S., Canada):
1–800–284–2729 (BUG CRAY), then 605–9100

Telephone (outside U.S., Canada):
+1–651–605–9100

Fax:
+1–651–605–9001

Mail:
Software Distribution Center
Cray Inc.
1340 Mendota Heights Road
Mendota Heights, MN 55120–1128
USA

Software will be shipped by ground service or 5-day international service.
This chapter describes the customer services that support this release.

7.1 Technical Assistance with Software Problems

If you experience problems with Cray software, contact your Cray service representative. Your service representative will work with you to resolve the problem. If you choose to have full- or part-time support on site, your on-site personnel are your primary contacts for service. If you have elected not to have on-site support, please call or send e-mail to the Cray Customer Support Center:

**E-mail:**
support@cray.com

**Telephone (inside U.S., Canada):**
1-800-950-2729 (CRAY)

**Telephone (outside U.S., Canada):**
+1-715-726-4993

**CRInform (for subscribers):**
crinform.cray.com

You can also create a Request for Technical Assistance (RTA) and track and search RTAs and Software Problem Reports (SPRs) online if you are a CRInform subscriber, as described in Section 7.2.

7.2 CRInform System

The CRInform system is the information and problem-reporting system for Cray customers who are CRInform subscribers.

You are a CRInform subscriber if your site has a software license agreement and software support agreement. Access CRInform at:

**crinform.cray.com**

Ask your system administrator for your password. Some of the things a subscriber to CRInform can do include:

- Report software problems (SPRs)
• Request technical assistance (RTAs)
• Communicate with other Cray system users
• Read about software problems reported at other sites
• Learn about solutions to various problems
• Order Cray software
• View Cray Service Bulletin

The CRInform program automatically logs events pertinent to your Cray system site as news items, so you do not have to search through the system for new information. The logged events include Software Problem Report (SPR) or Request for Technical Assistance (RTA) activity, new orderable software, new issues of the Cray Service Bulletin, new field notices (FNs), new software release documents, new software problem fix information, new marketing information, and new CRInform program information. You can also get automatic e-mail notification of any or all of the news items.

7.3 Training

To find out more about Cray training, contact your Cray representative or contact us in any of the following ways:

E-mail:
wwtng@cray.com

Web:
www.cray.com/training/

Fax:
+1–715–726–4991

Mail:
Technical Training
Cray Inc.
P.O. Box 6000
Chippewa Falls, WI 54729–0080
USA
7.4 Cray Service Bulletin (CRSB)

The CRInform site provides access to the online Cray Service Bulletin, also called CRSB, which contains descriptions of software problems, information about service procedures or agreements, and announcements of product upgrades and future products for the private use of Cray customers.

7.5 Cray Public Website

The Cray public website offers information about a variety of topics and is located at:

www.cray.com
Part II: Installation
This chapter describes how to install the Cray Programming Environment products listed in Section 6.2, page 19 on Cray X1 series systems running the UNICOS/mp 2.5 release or later.

**Note:** To install or configure the following items, refer to the indicated documentation:

- To install the Linux based Cray Programming Environment products, refer to the *Linux Based Cray X1 Series Systems Programming Environments Differences and Installation Guide*.
- To install the Solaris based Cray Programming Environment products, refer to the *Solaris Based Cray X1 Programming Environments Differences and Installation Guide*.
- To install the Etnus TotalView products, refer to *TotalView Release Overview and Installation Guide*.
- To configure the Cray Programming Environment Server (CPES) files systems and NFS mounts, refer to *Cray Programming Environment Server Administration*.

Before installing any of the Cray Programming Environment product, understand the following items:

- The installation requirements; refer to Section 8.1, page 30
- The location of the installed products; refer to Section 8.3, page 32
- The changes that the installation makes to your Cray Programming Environment; refer to Section 8.4, page 33

Next, to install the products:

1. Choose your installation tool; refer to Section 8.5, page 34
2. Install with the Common Installation Tool (CIT) (refer to Section 8.6, page 34) or `opt_install` (refer to Section 8.7, page 36)
3. Finish the installation process; refer to Section 8.9, page 42

Your site may require miscellaneous post-installation tasks as discussed in Chapter 9, page 47.
8.1 Installation Requirements

Before attempting to install any Cray Programming Environment products, verify that these requirements have been met:

- UNICOS/mp 2.5 release or later is installed and running on the Cray X1 series system.
- CPES is installed and running. If not, refer to the Cray Programming Environment Server Software Installation and Administration.
- NFS mount of `sv2/opt/ctl` to `/opt/ctl` is present.

To verify this, enter the following command:

```bash
 cray# df -k | grep /opt/ctl
```

If the NFS mount of `/opt/ctl` is present, the output of the commands will contain a statement similar to this line:

```
[cpes]:/sv2/opt/ctl nfs /opt/ctl
```

where `cpes` is the host name of your CPES.

If the `/opt/ctl` directory is not mounted, the command returns a blank line. If this occurs, refer to Section 9.2, page 48 or Cray Programming Environment Server Administration for information about mounting a directory.

- You have root permission on the Cray X1 system and root has read and write permissions for the `/opt/ctl` directory. Root has these permissions if root can execute this set of commands successfully:

  ```bash
  cray# cd /opt/ctl
  cray# touch file
  cray# rm /opt/ctl/file
  ```

  where `file` is the name of a nonexistent file.

  If root does not have read and write permissions, refer to Section 9.1, page 47.

- Trigger version 2.0.0.5 or later is installed. To display the version, enter:

  ```bash
  # cat /opt/ctl/trigger/trigger/version
  ```

  The output of this command contains lines with comments that show the Trigger version. If the proper version is not installed, refer to Section 8.8, page 38 to install it.
8.2 /opt/ctl File System Requirements

To successfully install the software packages of the Solaris based Cray Programming Environments, these file system requirements must be met:

- The /opt file system must have sufficient disk space for the software packages that will be installed.
- The /tmp directory must have sufficient disk space to hold temporary installation files during the installation process.

You can use Table 4 to calculate the disk space requirements for the software packages of the Solaris based Cray Programming Environments to install. As you compute the disk space requirements for the /sv2 file system, remember that the Cray Fortran Programming Environment and the Cray C++ Programming Environment share the CrayLibs (craylibs), LibSci (libsci), and CrayTools (craytools) packages. That is, the total disk space requirement for both Programming Environments is 625 MB. Also, it is important to make the /opt file system sufficiently large to accommodate future releases and products.

Table 4. Solaris Based Cray Programming Environment 5.4 Releases Disk Space Requirements

<table>
<thead>
<tr>
<th>Package Name</th>
<th>Required Space</th>
</tr>
</thead>
<tbody>
<tr>
<td>The Cray Fortran Programming Environment product:</td>
<td>580 MB required for the product. Disk space required for each component:</td>
</tr>
<tr>
<td>- CFTN</td>
<td>23 MB</td>
</tr>
<tr>
<td>- craylibs</td>
<td>34 MB</td>
</tr>
<tr>
<td>- libsci</td>
<td>263 MB</td>
</tr>
<tr>
<td>- craytools</td>
<td>260 MB</td>
</tr>
<tr>
<td>The C++ Programming Environment product:</td>
<td>602 MB required for the product. Disk space required for each component:</td>
</tr>
<tr>
<td>- CC</td>
<td>45 MB</td>
</tr>
<tr>
<td>- craylibs</td>
<td>34 MB</td>
</tr>
<tr>
<td>- libsci</td>
<td>263 MB</td>
</tr>
<tr>
<td>- craytools</td>
<td>260 MB</td>
</tr>
</tbody>
</table>
### 8.3 Product Directories

On Cray X1 series systems, each Programming Environment product is installed in an `/opt/ctl/prod/version` directory, where `prod` is the name of the product directory (for example, `cftn` for the Cray Fortran Compiler) and `version` is the version of the product (for example, `5.4.0.0`).

Each version of the product has its own directory from which its software is executed. This way you can maintain as many versions of a product as you want to, and all versions will be available to users.

**Note:** All directory paths discussed in this chapter, unless noted, are those seen while logged in to the Cray X1 series mainframe.

Module files and driver scripts for each product are created during product installation. Module files are stored at:

```
/opt/PE/modulefiles
```

Module files are created for each of these:

- The default version of the product
- The specific product version you are installing

Driver scripts are stored at:

```
/opt/PE/bin
```
Driver scripts (such as `ftn`) are used to access the executable files for which they are named. Driver scripts also define correct paths and environment variables for compilers, loaders, and tools.

### 8.4 Changes the Installation Makes to Your Programming Environment

The installation of the new Cray Programming Environment makes the following changes to your current default Programming Environment; that is, the one defined by the `PrgEnv` module file:

- **Adds the `PrgEnv.54.first_set` master module file**

  The Cray Programming Environment 5.4 releases provide a master module file (`PrgEnv.54.first_set`) that you can use to test the new products without introducing them into your default Programming Environment.

  Like the `PrgEnv` master module file that loads your default Programming Environment, the `PrgEnv.54.first_set` module file loads each newly installed product by version number. This module file is created on your system only for the initial product in a major release and not for updates (for example, it is created for release 5.4 but not for release 5.4.0.1).

  The `PrgEnv.54.first_set` module file is created in the `/opt/PE/modulefiles` directory when you install the new CrayTools product.

- **Overwrites script drivers in the `/opt/PE/bin` directory**

  The new script drivers add needed changes for the new products and support older product versions.

- **Overwrites some module files in the `/opt/PE/modulefiles` directory**

  The affected module files are `PrgEnv` and the non-versioned product module files like `cftn`, `CC`, `craytools`, and so on.

  **Note:** The `PrgEnv` module file loads your default Programming Environment.

To use a non-default product as the default product, refer to Section 8.9.3, page 44.
8.5 Choosing Your Installation Tool

Select your installation tool based on where the Cray Programming Environment product installation files are located:

- If you are installing from a CD, you must use the Common Installation Tool (CIT).

For a quick reference guide for using CIT, refer to Common Installation Tool (CIT) Reference Card on the CD. The card can be printed from the /cdrom_mount/cdrom0/CYRIinstall/2218.ps PostScript file.

You can also find more information about CIT by selecting the Help button in CIT, entering help all at the interactive interface prompt, or checking the cit(8) man page.

Note: Installation log files are located on the workstation at:
/tmp/cit.CWS_login/*.log

- If you are installing product files copied from a CD or downloaded from CRInform and copied to the Cray X1 series system, you must use the opt_install script located in the /opt/PE/admin/bin directory.

8.6 Installing with CIT

This section tells you how to use CIT to install the products of the Cray Programming Environment releases. The release CD contains the most current version of CIT.

1. At the CWS, enter the following command to verify that the ~CWS_login/.rhosts file on the CWS allows root to send remote shell commands to the CWS from the mainframe, and to verify that the /root/.rhosts file on the Cray mainframe allows CWS_login to send remote shell commands to the mainframe from the CWS:

   cws$ rsh Cray_System_Name -1 root "rsh CWS_Name -1 CWS_login uname -a"

   If root can send remote shell commands successfully, the output of the command returns a string displaying basic information about the system; otherwise, it returns a permission-denied message.

   Note: For more information on the communications path between the CWS and the Cray mainframe, refer to Common Installation Tool (CIT) Reference Card.
2. Enter:

```
# cd /cdrom_mount/cdrom0
```

3. Use the CIT installation default settings or specify how each product is installed. You must make this determination for each product. These are the CIT installation default settings for each product:

- Make this version default (No)
- Overwrite if it exists (No)
- Install news files (No)

**Note:** If the overwrite option is not enabled and the product already exits, the installation process of that product aborts. You must click `Continue` to proceed with the installation.

After determining how you want to install the products, perform one of the following steps:

- Use the CIT installation defaults by entering:
  
  ```
  # setup -c Cray_System_Name -D USE_DEFAULTS -l root
  ```

- Indicate that you want to choose how each product is installed by entering:
  
  ```
  # setup -c Cray_System_Name -l root
  ```

CIT installs all products in the `/opt/ctl/prod(version)` directories.

4. Load the packages:

   a. Double-click on the packages you want to load in the Packages Available panel.

      The selected package(s) will move to the Order of Installation panel.

   b. Click on the Install button at the bottom of the CIT window to load the packages listed in the Order of Installation panel.

      **Note:** If there is no Programming Environment on the machine, you must make sure that each new product is installed as the default version by selecting `Make this version default` when its window comes up.
Depending on the Cray software you are installing, preinstallation scripts may be involved before CIT actually loads the software. If this is the case, provide all the information requested in the dialog boxes that appear.

After you have provided all preinstallation information, a progress window appears and the actual installation begins.

As each package is loaded, it moves from the Order of Installation panel to the Packages Installed panel on the CIT window. The progress window displays loading progress.

If an error occurs, the progress window turns red and an error dialog box is displayed. Choose the appropriate option for the error listed in the error dialog box.

**Note:** To see the installation progress, click the Display install logs button.

5. Quit CIT.

6. If desired for security purposes, remove the root entry from the ~CWS_login/.rhosts file on the workstation and the CWS_login entry from the /root/.rhosts file on the Cray X1 system.

7. Perform the next appropriate step.

To finish the installation process, you should perform the appropriate steps discussed in Section 8.9, page 42. One of these steps is to remove root’s write permission from the /opt/ctl directory.

**Warning:** Failure to remove root’s write permission from the /opt/ctl directory will compromise the security of the CPES.

### 8.7 Installing with opt_install

This section describes how to use opt_install to install the Cray Programming Environment products as default products or as non-default. Use the opt_install script to install product files that were copied from a CD or downloaded from the CRInform website and copied to the Cray X1 series system.

The opt_install script is located in the /opt/PE/admin/bin directory.

To allow you to test a new version of a product, its older version is not automatically removed and the new version is not automatically made the default version.
Note: Products loaded using the opt_install script are expanded and copied into the /tmp/Kk$$ directory. This will require sufficient disk space in /tmp. To use another directory to hold these temporary files, set the PE_TMPDIR environment variable to another directory. For example, to use the /ptmp directory:

```
setenv PE_TMPDIR /ptmp
```

This will cause all temporary files to be copied to the /ptmp/tmp/Kk$$ directory.

An error log is created during product installation. The /opt/ctl/prod/version/.install_log file is the record of the installation steps.

If an installation fails, the installation file is placed here:

```
$TMPDIR/NEW_prod/install.log
```

1. Log onto the Cray X1 series system.

2. Decide where to install the product.

   To install the product to its default location, skip this step and continue to the next.

   To install the product to a non-default location, set the ROOT environment variable to the alternate directory.

3. Install the product by executing the opt_install script as follows:

```
cray# /opt/PE/admin/bin/opt_install filename.cpio
```

where filename is the name of the product .cpio file to install. If filename is not in the current working directory, it should contain the path to the file.

   Note: All product files on the Cray Programming Environment CD are named async.cpio. If a file was downloaded from CRInform, its name is specific to the product. For example, CC_5400_x1.cpio for the Cray C++ compiler version 5.4.

This example shows the command line for installing Cray C++ compiler version 5.4.0.0:

```
#/opt/PE/admin/bin/opt_install async.cpio
```
The output for the previous example is shown below:

```
cray# /opt/PE/bin/opt_install async.cpio
Creating directory /tmp/Kk2430180/NEW_product...
Unwrapping async.cpio.....
in directory /tmp/Kk2430180/NEW_product
17152 blocks
Installing CC for cray.
Unzipping CC.cpio.gz.....done.
Reading CC 5.4.0.0 information file.
# Installing cray-xl-targeted CC for cray-xl hardware.
# This product will install at:/opt/ctl/CC/5.4.0.0
Writing generation files into /opt/ctl/CC/5.4.0.0...done.
Current default CC is version: 5.4.0.0.48.
Make 5.4.0.0 the default CC on this system [y/n]? n
Updating /opt/PE/bin drivers...done.
Modules software is already installed.
Updating modulefiles ...done.
```

Load of package CC 5.4.0.0 complete.

To finish the installation process, perform the steps discussed in Section 8.9, page 42. One of these steps is to remove root’s write permission from the /opt/ctl directory.

**Warning:** Failure to remove root’s write permission from the /opt/ctl directory compromises the security of the CPES.

### 8.8 Installing, Configuring, and Accessing the Trigger Software

The Trigger product installs like other Cray Programming Environment products from a CD using CIT or from downloaded or copied files using the /opt/PE/admin/bin/opt_install script.

**Note:** Trigger only needs to be installed and configured at initial (factory) installation or if Trigger is updated, which usually occurs once a year.
Use the following procedure to install and configure the Trigger software:

1. Take steps to avoid disruption to users currently using the Cray Programming Environment by having your users log off or shutting down the Trigger daemon if the following warning applies to you:

   **Warning:** During the installation process of the Trigger software, you are prompted about overwriting the Trigger software. Overwriting the same version of the software will cause disruption of the Trigger daemon service if the daemon of the same version is currently running.

   **Warning:** During the configuration process you will start or restart the new Trigger daemon. This interrupts the connection between the Cray X1 series system and CPES and interrupts any processes started by a triggered product. So starting or restarting a new Trigger daemon should not be done while triggered processes are running on the CPES. Operation of the Cray X1 series system is not affected.

2. Install the Trigger software.

   During the installation process, these directories are created (as viewed from the Cray X1 series system): `/opt/PE/trigger/version` and `/opt/ctl/trigger/version`, where `version` is the version of the Trigger product being installed.

   **Note:** The Trigger product is always installed as non-default because it requires editing of configuration files and restarting of the Trigger daemon before a new version can become the running default. There is no reason to change the installation location of the Trigger product, since they cannot be installed from the release package as default.

   For instructions on installing the Trigger software using CIT, refer to Section 8.6, page 34, using `opt_install`, refer to Section 8.7, page 36.

   After you are done installing the Trigger software, return here and continue with step 3 to configure the software.

3. Log into the Cray X1 series system and open the `/opt/ctl/trigger/version/bin/trigexecd.cfg` to edit the lines indicated in Table 5.
Table 5. Lines to Edit in `trigexecd.cfg`

<table>
<thead>
<tr>
<th>Line</th>
<th>Comments</th>
</tr>
</thead>
<tbody>
<tr>
<td>CLIENT=</td>
<td>Set to the Cray X1 series machine name.</td>
</tr>
<tr>
<td>PASS=</td>
<td>Set to the site-selected password that will also be defined in /opt/PE/trigger/_version/bin/trigsnd.cfg.</td>
</tr>
<tr>
<td>PORT=9930</td>
<td>Set to a site-selected value as long as the values in the trigsnd.cfg and trigexecd.cfg files match.</td>
</tr>
</tbody>
</table>

4. Set the default symbolic link to the new Trigger version as indicated below (this will not cause any interruption):

```
cray# cd /opt/ctl/trigger
cray# rm trigger
cray# ln -s version trigger
```

where `version` is the name of the directory containing the version of Trigger to use.

5. Edit the lines, as indicated in Table 6, that are in the following file:

```
/opt/PE/trigger/\_version/bin/trigsnd.cfg
```

Table 6. Lines to Edit in `trigsnd.cfg`

<table>
<thead>
<tr>
<th>Line</th>
<th>Comments</th>
</tr>
</thead>
<tbody>
<tr>
<td>TRIGROOT=</td>
<td>Set to <code>/sv2</code> or to <code>/sv2/\$CLIENT</code> if there is more than one Cray X1 series system or partition attached to one CPES.</td>
</tr>
<tr>
<td></td>
<td>TRIGROOT must be defined to create a unique network connection from each Cray X1 series system or partition to the CPES.</td>
</tr>
<tr>
<td></td>
<td>Refer to step 3 for more information about CLIENT.</td>
</tr>
<tr>
<td>RCVHOST=</td>
<td>Set to the CPES machine name</td>
</tr>
</tbody>
</table>
Line | Comments
--- | ---
PASS= | The site-selected password that is also defined in /opt/ctl/trigger/version/bin/trigexecd.cfg
PORT=9930 | The value assigned to PORT can be site-selected as long as the values in the trigsnd.cfg and trigexecd.cfg files match.

6. Define the default symbolic link for the new Trigger version:

**Warning:** This step and step 7 will interrupt the user when the daemon is restarted.

    cray# cd /opt/PE/trigger
    cray# rm trigger
    cray# ln -s version trigger

7. Begin the daemon process as shown below:

**Warning:** This step will interrupt the user when the daemon is started or restarted.

a. Log into the CPES.

b. Perform only one of these steps to start or restart the daemon process:

   - If this is an initial installation, edit file /etc/inittab to add this line:

   ```
   tr:234:respawn:/sv2/opt/ctl/trigger/trigger/bin/trigexecd
   ```

   The above change to the file causes the Trigger daemon to start automatically.

   - If this is not an initial install, kill the running Trigger daemon process (trigexecd), as this example shows:

   ```
   cpes# ps -ef | grep trig
   root 9190 1 0 Mar 01 ? 3:40 /sv2/opt/ctl/trigger/trigger/bin/trigexecd
   cpes# kill -15 9190
   ```

   Since trigexecd is part of inittab, it will respawn when killed, using the new trigexecd.
8. Verify that your configuration is valid by doing these steps:
   a. Log into the Cray X1 series system.
   b. Load the PrgEnv module:
      
      cray% module load PrgEnv
   c. Find the version of the ftn command:
      
      cray% ftn -V

      If your configuration is valid, the ftn command returns a line similar to this:

      Cray Fortran: Version 5.4.0.0 Tue Mar 01, 2005 13:11:04

      The version number should reflect the default version of the installed Cray Fortran Compiler.

      If your site needs to provide user access to the remps command, refer to Section 9.3, page 50.

8.9 Finishing the Programming Environment Installation

After installing the Cray Programming Environment release products through CIT or the opt_install command, you must perform one or more of the following steps:

- (Required) Return the CPES to a secure state.
- Copy the UNICOS/mp system libraries and include files to the CPES.
  
  This step is required if the system libraries and files were not copied to the CPES the last time the OS was upgraded.
- Make a non-default product the default product
- Add site-specific options to the Fortran command line
- Update the modules environment
8.9.1 Returning the CPES to a Secure State

After installing the Cray Programming Environment products through CIT or the `opt_install` command, you must remove root’s write permission to the `/opt/ctl` directory on the CPES to return the CPES to a secure state. Failure to do this will compromise the security of the CPES.

To return the CPES to a secure state, follow these steps:

**Note:** Replace `system_name-cpes_name` with the name of the connection, as known by the Cray X1 series system, that is between the Cray X1 series system and the CPES on the shared FC/IP network; for example, `cray_x1-cpes0`.

1. Open the `/etc/dfs/dfstab` file for editing.
2. Find this line:
   ```
   share -F nfs -o rw,root=[system_name-cpes_name] -d "/sv2/ctl" /sv2/opt/ctl
   ```
3. Replace `rw,root=[system_name-cpes_name]` with `ro`.
   After the change, the line should look similar to this:
   ```
   share -F nfs -o ro -d "/sv2/ctl" /sv2/opt/ctl
   ```
4. Export the `/sv2/opt/ctl` disk partition:
   ```
   cpes# /usr/sbin/share /sv2/opt/ctl
   ```
5. Verify that the `/sv2/opt/ctl` disk partition was exported:
   ```
   cpes# /usr/sbin/share | grep /opt/ctl
   ```
   If the disk partition was exported properly, the command returns this line:
   ```
   share -F nfs -o ro -d "/sv2/ctl" /sv2/opt/ctl
   ```
   If the command returns a blank line, the disk partition was not properly exported. Refer to the `share(1M)` Solaris man page for more information to resolve this problem.

8.9.2 Copying System Libraries and Include Files to the CPES

The UNICOS/mp system libraries and include files must be copied to the CPES. This copy occurs automatically during the installation of a new OS version. If for some reason the files were not copied, the Cray Programming Environment
releases are installed after the OS is installed, or you are reinstalling an OS update, you must recopy the system libraries and include files to the CPES by using the cp_2_cpes command. You can use this command even if the files were previously copied because it compares the timestamp of the files in the source and destination directories and copies only the newer files to the CPES.

The cp_2_cpes command copies the files from the correct location, creates the necessary directories, and copies the files to the correct location on the CPES.

As root on the Cray X1 series system, execute the /opt/PE/admin/bin/cp_2_cpes command to copy the libraries and include files from and to the directories listed in Table 7.

<table>
<thead>
<tr>
<th>Source UNICOS/mp directories</th>
<th>Destination CPES directories</th>
</tr>
</thead>
<tbody>
<tr>
<td>/usr/lib/nonshared</td>
<td>/opt/ctl/ libs_OS_LEVEL^{1}</td>
</tr>
<tr>
<td>/usr/include</td>
<td>/opt/ctl/include_OS_LEVEL^{1}</td>
</tr>
</tbody>
</table>

For example, the cp_2_cpes utility creates the following directories when copying the files from UNICOS/mp 2.5 operating system:

- /opt/ctl/include 2.5
- /opt/ctl/libs 2.5

### 8.9.3 Making a Non-default Product the Default Product

After installing the new Programming Environment products as non-default products, you can introduce selected new products into your default Programming Environment by setting a soft link in each product directory. To change the default product, perform these steps:

1. Go to the product directory as this example shows for the Cray Fortran Compiler:

   ```
   cd /opt/ctl/cftn
   ```

2. Remove the soft link to the product as this example shows for the Cray Fortran Compiler:

   ```
   rm cftn
   ```

^{1} OS_LEVEL is the version of the UNICOS/mp operating system files you are copying.
3. Recreate a soft link of the same name to the new Programming Environment product as this example shows for the new Cray Fortran Compiler:

\[ \text{ln } -s \ 5.4.0.0 \ cftn \]

### 8.9.4 Adding Site-specific Options to the Fortran Command line

This new feature allows administrators to set site-specific compiler options for the Cray Fortran compiler. These options are saved in the `/opt/PE/bin/ftnopts` configuration file and are invoked unconditionally every time a user invokes the `ftn` compiler script.

Adding command-line options to the `ftnopts` file will cause these options to be inserted into the command line for every usage of the `/opt/PE/bin/ftn` script driver. For the individual user, there is no way to make the compiler "ignore" these command-line options.

This will affect the CFTN compiler in the following ways:

- The printed and on-line reference manuals may not describe the behavior of CFTN as you have configured it for your system.
- Some combinations of command-line options are incompatible. This may cause unexpected aborts or other unusual behavior.
- Users of the compiler may be unaware of which command-line options are being used to invoke the compiler.

There may be other implications of using non-default command-line options. Please use this feature with caution.

To use this feature system-wide, edit the last line in the file, `/opt/PE/admin/bin/ftnopts`, adding your command-line options into the quoted string. Uncomment the line and copy this file to `/opt/PE/bin/ftnopts`. For example:

```bash
#_FTN_OPTS="-O2 -sdefault64 -L/local/library/path -Wl"-V"
```

Remove the "#" comment symbol from the above line.

```bash
cp /opt/PE/admin/bin/ftnopts /opt/PE/bin/ftnopts
chmod 644 /opt/PE/bin/ftnopts
```

Once this file exists at `/opt/PE/bin`, it will be evaluated by every invocation of the command, `ftn`, and its contents will be inserted into the user’s command line.
Using the edited `ftnopts` file at `/opt/PE/admin/bin`, the admin may test the new command line before copying it to `/opt/PE/bin`. To test the `ftnopts` file, set the environment variable, `FTN_OPT_DIR` to `/opt/PE/admin/bin`. With this environment variable set, the administrator can execute `ftn` to see the resulting command line and to test for command-line incompatibilities and unintended command-line interactions.

### 8.9.5 Updating the Modules Environment

The Cray Programming Environments require the use of Modules version 3.1.6 or later. The Modules product allows you to manage multiple versions of products and sets of products. Cray provides the master module file, `PrgEnv`, as the default user environment and provides `PrgEnv.template` as a pattern to follow to construct your own module files.

The master module file, `PrgEnv`, loads the set of generically named module files (for example, `CC` and `craytools`) which in turn define the default versions of each product to use when `PrgEnv` is loaded. For example, the `CC` module file loads the default C compiler `/opt/ctl/CC/CC`, where `/opt/ctl/CC` is the product directory, and `CC` is a link to the version chosen as default.

**Note:** After loading `PrgEnv`, you can list the product versions by entering `module help PrgEnv`.

You can also create your own set of product versions by defining your own module file and placing it into the `/opt/PE/modulefiles` directory. Any module file placed into this directory is automatically available for use.

To define your own module file, perform these steps:

1. Open the `/opt/PE/admin/bin/PrgEnv.template` file and edit the version string in the `prodlist` environment variable.

2. Save the file by renaming it with any name you want. This name must be unique among the file names in the `/opt/PE/modulefiles` directory.

3. Copy your renamed file to the `/opt/PE/modulefiles` directory.

You can create a module file with any `prodlist`, as long as all the products listed have their own unique module files at `/opt/PE/modulefiles`. 
If your site has the following needs, refer to the indicated information to meet those needs:

- Providing read and write permissions to the `/opt/ctl` directory. Refer to Section 9.1, page 47.
- Making files not provided by the Cray Programming Environment releases accessible to the Cray Programming Environment. Refer to Section 9.2, page 48.
- Providing user access to the `remps` command. Refer to Section 9.3, page 50.

### 9.1 Granting Root Read and Write Permissions to `/opt/ctl`

If root does not have read and write permissions in the `/opt/ctl` directory or if you do not know that root has these permissions for this directory, use this procedure to check or enable them.

**Note:** Replace `system_name-cpes_name` with the name of the connection, as known by the Cray X1 series system, that is between the Cray X1 series system and the CPES on the shared Fiber Channel (FC)/IP network; for example, `cray_x1-cpes0`.

1. Log into the CPES and verify that it exports the `/sv2/opt/ctl` disk partition by using this on the CPES command line:

   ```
cpes# grep /opt/ctl /etc/dfs/dfstab
   ```

   If the CPES exports the disk partition, the command returns a line similar to this:

   ```
   share -F nfs -o ro -d "/sv2 ctl" /sv2/opt/ctl
   ```

   If the disk partition is not exported, contact your on-site personnel or the Cray Customer Support Center. This must be corrected before continuing to step 2.

2. Examine the line returned in step 1 to see if read and write permissions are set (look for the string `-o rw,root=[system_name-cpes_name]`).
If the string exists, root has read and write permission; continue with step 4.
If the line does not, continue to step 3.

3. Modify the line so it looks like this (bolded text highlights the changes):

   ```
   share -F nfs -o rw,root=[system_name-cpes_name] -d "/sv2 ctl" /sv2/opt/ctl
   ```

4. Export the `/sv2/opt/ctl` disk partition:

   ```
   cpes# /usr/sbin/share /sv2/opt/ctl
   ```

   After installing the products, remove root’s write permission from the `/opt/ctl` directory. Section 8.9.1, page 43 explains how to do this.

   **Warning:** Failure to remove root’s write permission from the `/opt/ctl` directory will compromise the security of the CPES.

5. Verify that the `/sv2/opt/ctl` disk partition was exported:

   ```
   cpes# /usr/sbin/share | grep /opt/ctl
   ```

   If the disk partition was exported properly, the command returns this line:

   ```
   share -F nfs -o rw,root=[system_name-cpes_name] -d "/sv2 ctl" /sv2/opt/ctl
   ```

   If the command returns a blank line, the disk partition was not properly exported. Refer to the `share(1M)` Solaris man page for more information to resolve this problem.

### 9.2 Providing Access to Files that the Cray Programming Environment Does Not Include

You must cross-mount, on the Cray Programming Environment Server (CPES), files not provided by the Cray Programming Environment releases (for example, the Cray Open Software (COS) libraries). This allows the Cray Programming Environment to access them during compilation and/or linking.
Since the process for cross-mounting any file on the CPES is the same, we will describe this process through an example that cross-mounts the COS libraries. If you have multiple Cray X1 series systems hosted by a single CPES, you must repeat this process for each system and replace each of these names with the values associated with each system: `system_name-cpes_name`, `system_id`, and `cray_mount`.

**Note:** Throughout this procedure, you must replace the following names with the indicated values:

- `system_name-cpes_name`, replace with the name of the connection, as known by the Cray X1 series system, that is between the Cray X1 series system and the CPES on the shared FC/IP network.
- `system_id`, replace with the directory name under the `/sv2` directory on the CPES associated with `system_name-cpes_name`.
- `cray_mount`, replace with the name of the file containing the objects the CPES will automount.

1. Check whether the Cray X1 series system exports the directory containing the files. Since for COS this is the `/opt/open` directory, you execute the following command:

   ```
cray# grep /opt/open /etc/xtab
   ```

   If `/opt/open` is exported, the command returns a similar statement to this:

   ```
   /opt/open -ro,access=`system_name-cpes_name`
   ```

   If it is exported, go to step 4.

   If `/opt/open` is not exported, the command returns with a blank line. If this occurs, continue to step 2.

2. On the Cray X1 series system, add the following text, if it is missing, to the `/etc/exports` file:

   ```
   /opt/open /opt/open -ro,access=`system_name-cpes_name`
   ```

3. Export `/opt/open` from the Cray X1 series system:

   ```
cray# /usr/sbin/exportfs /opt/open
   ```

   If an error occurs, refer to the `exportfs(1B)` Solaris man page.
4. Check whether the CPES is automounting /opt/open:

```
cpes# grep /opt/open /etc/mnttab
```

If the CPES is automounting /opt/open, the command returns this line:

```
system_name-cpes_name:/opt/open /sv2/system_id/opt/open ....
```

If the line is returned, you are done. If a blank line is returned, continue to step 5.

5. Check if /opt/open is already in the /etc/automount/cray_mount file:

```
cpes# grep system_name-cpes_name:/opt/open
/etc/automount/cray_mount
```

If the entry exists, the command returns this line:

```
/open system_name-cpes_name:/opt/open
```

If the entry exists, go to step 7. If not, continue to step 6.

6. On the CPES, add the following text to the /etc/automount/cray_mount file:

```
/open system_name-cpes_name:/opt/open
```

7. Force the CPES to mount /opt/open from the Cray X1 series system:

```
cpes# mount -F nfs /sv2/system_name-cpes_name/opt/open
```

If an error occurs, refer to the `mount(1M)` Solaris man page.

### 9.3 Providing Access to the `remps` Command

If users want to use the triggered command `remps(1)` to see their processes that are running on a CPES, they need an account on that CPES. These accounts are added to the CPES in a locked state. This means that the user cannot log in directly on the CPES, but has access to the functionality of the triggers, including compiling and using the `remps` command. For more information about adding accounts on the CPES, refer to *Cray Programming Environment Server Administration*. 

---

50 S–5212–54
Index

A
Access
   to files not included with Cray Programming Environment, 48
   to remps command, 50
Accessing Cray documentation, 15

B
BioLib
   installing, 29
Books, 16
   accessing, 15

C
C and C++ compiler, 8
C and C++ features, 8
CAL, 20
CIT
   documentation, 16
Compatibilities and differences, 11
Compiler
   C, 8
   C++, 8
   Fortran, 7
Contact information
   Customer Support Center, 23
   Software Distribution Center, 16
   Training, 24
CPES
   coping libraries and files, 43
   returning to secure state, 43
Cray Assembly Language
   installing, 29
Cray Assembly Language version, 20
Cray Bioinformatics library
   installing, 29
Cray Bioinformatics library version, 20
Cray Service Bulletin, 24–25
Cray websites, 25
   CRInform, 23
   publications, 15
   support, 23
   training, 24
CrayLibs version, 20
CrayPat
   features, 5
CrayTools version, 20
CRInform, 23
Cross-compiler, 29
CRSB, 25
Customer services, 23
Customer Support Center, 23
Customs, 22

D
Differences, 11
Disk space requirements
   temporary directory, 32
Distribution Center, 16, 22
Documentation, 15–16
   accessing, 15

E
Enhancements, 5
Errata, 16
Error message explanations, 16
Export license, 22

F
Features
   C and C++, 8
   tail recursion optimization, 8
   CrayPat, 5
   Fortran, 7
Field notices (FNs), 24
Finishing the installation, 42
Fortran 2003 features, 7
Fortran compiler, 7
Fortran features, 7
Function
  tracebk, 6

H
Hard copy, 16
Hardware and software requirements, 19
HTML, 16

I
Installation
  choosing installation tool, 34
  finishing, 42
  with CIT, 34
  with opt_install, 36
Installation and set-up
  Trigger software, 38
Installation requirements, 30
  /opt/ctl, 31
  product directories, 32
Installing
  Cray Assembly Language, 29
  Cray Programming Environment (Linux based), 29
  Cray Programming Environment (Solaris based), 29
  Cray X1 series Programming Environment (Solaris based), 19
  Modules, 29
  Motif, 29
  MPT, 29
  Triggers, 29
  X Window System X11, 29

L
Letter of assurance, 22
LibSci version, 20
Licensing, 22
Limitations, Cray Apprentice², 13
Linux systems, 21
installing, 29

M
Man pages
  accessing, 15
  new Cray, 18
Modules, 20
  environment, 46
  installing, 29
  version, 20
Motif, 20
  version, 20
MPT, 20
  installing, 29
  licensing and ordering, 21
  version, 20

O
/opt/ctl
  root permissions, 47
Ordering
  documentation, 16
  software, 22

P
PDF, 16
Performance enhancements, 5–6
Pricing, 22
Problems, 23
Publications, 15–16
  accessing, 15

R
Release package, 19
remps command, 50
Request for Technical Assistance (RTA), 23–24
Requirements, hardware and software, 19

S
Scientific library features, 9
Shipping, 22
Site-specific options, 45
<table>
<thead>
<tr>
<th>Software enhancements</th>
<th>5</th>
</tr>
</thead>
<tbody>
<tr>
<td>Software license agreement</td>
<td>21</td>
</tr>
<tr>
<td>Software Problem Report (SPR)</td>
<td>23–24</td>
</tr>
<tr>
<td>Solaris systems</td>
<td>19, 21</td>
</tr>
<tr>
<td>installing</td>
<td>19, 29</td>
</tr>
<tr>
<td>Subscriber, CRInform</td>
<td>23</td>
</tr>
<tr>
<td>Support agreement</td>
<td>21–23</td>
</tr>
<tr>
<td>Support Center</td>
<td>23</td>
</tr>
<tr>
<td>Tracebk function</td>
<td>6</td>
</tr>
<tr>
<td>Training</td>
<td>24</td>
</tr>
<tr>
<td>Triggers</td>
<td>20</td>
</tr>
<tr>
<td>installing</td>
<td>29</td>
</tr>
<tr>
<td>version</td>
<td>20</td>
</tr>
<tr>
<td>U</td>
<td></td>
</tr>
<tr>
<td>Upgrades</td>
<td>21</td>
</tr>
<tr>
<td>X Windows System</td>
<td>20</td>
</tr>
<tr>
<td>X Windows System X11 libraries</td>
<td></td>
</tr>
<tr>
<td>installing</td>
<td>29</td>
</tr>
<tr>
<td>X11 version</td>
<td>20</td>
</tr>
<tr>
<td>X11 libraries</td>
<td>20</td>
</tr>
</tbody>
</table>