Cray Programming Environment
5.0 Releases Overview and Installation Guide
S–5212–50
# Contents

<table>
<thead>
<tr>
<th>Section</th>
<th>Page</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Introduction</strong> [1]</td>
<td>1</td>
</tr>
<tr>
<td>Emphasis for the Cray Programming Environment 5.0 Releases</td>
<td>1</td>
</tr>
<tr>
<td>Distribution of This Release Overview</td>
<td>2</td>
</tr>
<tr>
<td>Reader Comments</td>
<td>2</td>
</tr>
<tr>
<td>Typographical Conventions</td>
<td>3</td>
</tr>
<tr>
<td><strong>Software Enhancements</strong> [2]</td>
<td>5</td>
</tr>
<tr>
<td>Cray Fortran Features</td>
<td>5</td>
</tr>
<tr>
<td>Cray C and C++ Features</td>
<td>6</td>
</tr>
<tr>
<td>LibSci Features</td>
<td>7</td>
</tr>
<tr>
<td>Cray Programming Environments for Cray X1 Systems (Solaris Based)</td>
<td>8</td>
</tr>
<tr>
<td><strong>Compatibilities</strong> [3]</td>
<td>9</td>
</tr>
<tr>
<td>Trigger 2.0.0.3 Required</td>
<td>9</td>
</tr>
<tr>
<td>Cray Assembler Dependency</td>
<td>9</td>
</tr>
<tr>
<td>Optimizations for <code>std::complex&lt;float&gt;</code> Class Causes Byte Alignment Changes</td>
<td>9</td>
</tr>
<tr>
<td>Recompilation of Code Developed with Cray Programming Environment 4.2 Releases or Earlier</td>
<td>9</td>
</tr>
<tr>
<td><strong>Limitations</strong> [4]</td>
<td>11</td>
</tr>
<tr>
<td>CrayPat</td>
<td>11</td>
</tr>
<tr>
<td>ScalAPACK and BLACS Performance</td>
<td>11</td>
</tr>
<tr>
<td>Cray C and C++ Compilers</td>
<td>11</td>
</tr>
<tr>
<td>Unified Parallel C</td>
<td>11</td>
</tr>
<tr>
<td>CPU Time Computation for Programs Using SSP Mode</td>
<td>12</td>
</tr>
<tr>
<td><strong>Documentation</strong> [5]</td>
<td>13</td>
</tr>
<tr>
<td>CrayDoc Documentation System</td>
<td>13</td>
</tr>
<tr>
<td>Cray Manuals</td>
<td>13</td>
</tr>
</tbody>
</table>

S–5212–50
Cray Man Pages ........................................ 15
Error Message Explanations ....................... 16
Cray Glossary .......................................... 16
Ordering Documentation ............................. 17

**Release Package [6]** ................................. 19
Hardware and Software Requirements .......... 19
Contents of Release Package ...................... 19
Licensing ............................................. 20

**Customer Service [7]** .............................. 23
Technical Assistance with Software Problems .. 23
CRInform System .................................. 23
Training ............................................ 24
*Cray Service Bulletin* (CRSB) .................. 25
Cray Public Web Site ............................... 25

**Installation on Cray X1 Systems [8]** ........ 27
Installation Requirements ......................... 28
Product Directories ................................ 28
Installation Using CIT ............................. 29
Installation Using `opt_install` ................. 31
Installing and Configuring Trigger Software on Cray X1 Systems .................. 32
Copy System Libraries and Include Files to the CPES ............................ 34

**Installation on Solaris Systems [9]** .......... 37
Installation Requirements ......................... 37
Solaris File System Requirements ............... 37
Product Directories ................................ 39
Installation Using CIT ............................. 39
Installation Using `opt_install` ................. 41
Index

Table 1. Manuals Provided with These Releases .................................. 14
Table 2. Man Page Collections Provided with These Releases ...................... 15
Table 3. Lines to Edit in trigsnd.cfg ................................................. 33
Table 4. Solaris Based Cray Programming Environment 5.0 Releases Disk Space Requirements . 38
This document describes the Cray Programming Environment 5.0 releases and their installation on Cray X1 systems and Solaris systems.

1.1 Emphasis for the Cray Programming Environment 5.0 Releases

<table>
<thead>
<tr>
<th>Emphasis</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>SSP mode</td>
<td>SSP mode compiles Cray Fortran, C, and C++ code to produce an executable that runs on one single-streaming processor (SSP) rather than a multistream processor (MSP).</td>
</tr>
<tr>
<td>Performance enhancements</td>
<td>The Cray Fortran Compiler and associated libraries contain many performance enhancements that produce more efficient and faster compiled code.</td>
</tr>
<tr>
<td>Motif version 2.1 libraries</td>
<td>The Motif version 2.1 libraries are now available on the Cray X1 system. Motif allows you to develop X Window System applications that are portable across a variety of platforms. This product is not part of the Cray Programming Environment 5.0 releases but is shipped with it.</td>
</tr>
<tr>
<td>Cray Programming Environments for Cray X1 Systems (Solaris based)</td>
<td>The Cray Programming Environment 5.0 releases are now available on Solaris systems as cross compiler environments and are separately licensed packages.</td>
</tr>
<tr>
<td>Recompiling 4.2 code or earlier</td>
<td>As mentioned in earlier Cray Programming Environment release overviews, the Cray Programming Environment 5.0 releases require recompilation of code compiled with Programming Environment 4.2 releases or earlier. Refer to Section 3.4, page 9.</td>
</tr>
</tbody>
</table>
1.2 Distribution of This Release Overview

A printed copy of this release overview is provided with each Cray Programming Environment 5.0 release package. You can also access this document at any of the following sites:

- The network location specified for Cray documentation by your system administrator
- The Cray public web site:
  http://www.cray.com/
- CRInform (for subscribers only):
  http://crinform.cray.com
  Click on the Release Documents link.
- The Cray internal web site (Cray personnel only):
  http://swpubs.us.cray.com/craydoc/
- Network directory for Cray service employees:
  /home/craypark/release_docs/

If you cannot get electronic copies of the release overview files using any of these methods, contact your Cray representative.

1.3 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:
swpubs@cray.com

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

Telephone (outside U.S., Canada):
Contact your Cray representative, or call +1–715–726–4993 (Cray Customer Support Center)
1.4 Typographical Conventions

The following conventions are used throughout this document:

<table>
<thead>
<tr>
<th>Definition</th>
<th>Meaning</th>
</tr>
</thead>
<tbody>
<tr>
<td>command</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>variable</td>
<td>Italic typeface indicates an element that you will replace with a specific value. For instance, you may replace filename with the name datafile in your program. It also denotes a word or concept being defined.</td>
</tr>
<tr>
<td>user input</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>
</tbody>
</table>
This chapter highlights the features available through the Cray Programming Environment 5.0 releases that are new or enhanced since the Cray Programming Environment 4.3 releases.

2.1 Cray Fortran Features

The Cray Fortran Programming Environment 5.0 release includes these new features:

SSP mode SSP mode (option -O ssp) compiles Cray Fortran code to produce an executable that runs on one single-streaming processor (SSP) rather than a multistream processor (MSP). This allows applications, using MPI or other distributed programming models, to scale by using multiple SSPs.

While the SSP mode feature is functionally available in the Cray Programming Environment 5.0 releases, the computation of "CPU time" for applications using SSP mode can be confusing (see Section 4.5, page 12 for more information) and there are additional optimizations we would like to make for SSP mode libraries. We plan to address these issues in upcoming releases.

While the SSP mode feature is functionally available in the Cray Programming Environment 5.0 releases, the time charged against applications using SSP mode is inflated (see Section 4.5, page 12 for more information) and SSP mode code is not optimized. We plan to fix the inflated charges and optimize code using SSP mode in the next release of the Cray Programming Environment.

SSP mode is supported and documented but marked as deferred in the ftn(1) man page and Cray Fortran Compiler Commands and Directives Reference Manual.
Faster Fortran formatted I/O
Library enhancements improve the efficiency and performance of Fortran formatted I/O.

Options for auto aprun
The CRAY_AUTO_APRUN_OPTIONS environment variable specifies options for the aprun command when automatically invoked (auto aprun).

Vector update improvements
Compiler enhancements improve vector processing of indirect references that result in memory access conflicts (that is, vector updates where the indirect references refer to the same memory location).

Fortran makefile generator
The Cray Fortran Programming Environment now supports the ftnmgen command.

Include files expand in source listings
The ftn and ftnlx commands have a new \texttt{-r e} option that expands included files into the source listing. This option is off by default.

Additional memory pages available
The new X1_PRIVATE_STACK_GAP environment variable consolidates, when used with the X1_PRIVATE_STACK_SIZE environment variable, the four private stacks within an multistreaming processor (MSP) into one segment, which frees up nontext pages for application use. This feature is available under UNICOS/mp 2.2.

For more information about new features, see the Cray Fortran Compiler Commands and Directives Reference Manual and the appropriate man pages.

### 2.2 Cray C and C++ Features

The Cray C and Cray C++ Programming Environment 5.0 release includes these new features:

<table>
<thead>
<tr>
<th>Feature</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>SSP mode</td>
<td>SSP mode (option \texttt{-h ssp}) compiles Cray C or C++ code to produce an executable that runs on one single-streaming processor (SSP) rather than a multistream processor (MSP). This allows applications, using MPI or other distributed programming models, to scale by using multiple SSPs.</td>
</tr>
</tbody>
</table>
While the SSP mode feature is functionally available in the Cray Programming Environment 5.0 releases, the time charged against applications using SSP mode is inflated (see Section 4.5, page 12 for more information) and SSP mode code is not optimized. We plan to fix the inflated charges and optimize code using SSP mode in the next release of the Cray Programming Environment.

SSP mode is supported and documented but marked as deferred in the CC man page and Cray C and C++ Reference Manual.

**Loopmark information**
The `-h list=opt` compiler option creates source listings that includes loopmark information.

**Options for auto aprun**
The `CRAY_AUTO_APRUN_OPTIONS` environment variable specifies options for the aprun command when automatically invoked (auto aprun).

**Vector update improvements**
Compiler enhancements improve vector processing of indirect references that result in memory access conflicts (that is, vector updates where the indirect references refer to the same memory location).

**Additional memory pages available**
The new `X1_PRIVATE_STACK_GAP` environment variable consolidates, when used with the `X1_PRIVATE_STACK_SIZE` environment variable, the four private stacks within an multistreaming processor (MSP) into one segment, which frees up nontext pages for application use. This feature is available under UNICOS/mp 2.2.

For more information about the new features, see the Cray C and C++ Reference Manual and the CC(1) man page.

### 2.3 LibSci Features

These are the new LibSci features:

- Performance improvements for level 2 and 3 BLACS routines, fast Fourier transforms (FFTs) routines, and a few LAPACK routines.
- Cray man pages for the BLACS library.
2.4 Cray Programming Environments for Cray X1 Systems (Solaris Based)

Beginning with Cray Programming Environment 5.0 releases, you can use the Cray Fortran Programming Environment and the Cray C++ Programming Environment on a Solaris system as cross compilers to build most applications for Cray X1 systems without logging into a Cray X1 system.

Note: Cray Inc. is reviewing long-term plans for its Cray Programming Environments, which will likely include cross compiler environments. Due to customer surveys and technical assessments, we are considering the offering of Linux based cross compilers in a future release. You should be aware that while we now offer full support for Solaris based cross compilers, it is possible that this support may be dropped soon after the release of Linux based cross compilers. You should contact your Cray representative for additional information.

The Solaris based cross compiler environments are offered in the Cray Programming Environment for Cray X1 Systems (Solaris based) products, licensed separately from the Cray X1 Programming Environments, and run on a Solaris system that is separate from the system that hosts the Cray Programming Environment Server (CPES). From here on, Cray Programming Environments for Cray X1 Systems (Solaris based) are referred to as the Solaris based Cray Programming Environments.

The Solaris based Cray Programming Environments produce the same binary as the UNICOS/mp based Programming Environments, as long as the versions of both environments are the same. The binary is executed on the UNICOS/mp system, after making it accessible to the UNICOS/mp system through a file transfer mechanism or cross mounted file systems.

The advantages of using the Solaris based Cray Programming Environments include faster compile time and access to the Cray Programming Environments when the Cray X1 system is not available.

For more information about which products are included with the Solaris based Cray Programming Environments, see Section 6.2, page 19. For further information about the Solaris based Cray Programming Environments, see Section 6.3, page 20; Chapter 9, page 37; and Differences for Solaris Based Cray X1 Programming Environments.
This chapter provides information about compatibility issues for the Cray Programming Environment 5.0 releases.

3.1 Trigger 2.0.0.3 Required

The Cray Programming Environment 5.0 releases require Trigger version 2.0.0.3 or later. Ensure that this version is installed before installing the Cray Programming Environment 5.0 release package.

3.2 Cray Assembler Dependency

The Cray Assembler version 1.1 can only be used with the loader of Cray Programming Environment 5.0 releases and later. It cannot be used with earlier versions of the loader.

3.3 Optimizations for std::complex<float> Class Causes Byte Alignment Changes

To improve optimization of C++ code using the std::complex<float> class in structures or classes, we changed the byte alignment for members within these structures or classes from 4 bytes to 8 bytes. This change requires any code using the std::complex<float> class in structures or classes be recompiled before it can be used with the Cray C++ Programming Environment 5.0 release.

3.4 Recompilation of Code Developed with Cray Programming Environment 4.2 Releases or Earlier

As mentioned in earlier Cray Programming Environment releases, the Cray Programming Environment 5.0 releases require recompilation of code compiled with Programming Environment 4.2 releases or earlier. Binary files and libraries produced by compilers from Programming Environment 4.2 releases or earlier are not compatible with Programming Environment 5.0 releases or with future UNICOS/mp releases. The Message Passing Toolkit 2.1.1.0 package and the Cray Assembler 1.0.0.4 package are compatible with the Programming Environment 4.3 releases and later. After Programming Environment 5.0, we expect no more forced recompilations.
This chapter describes the limitations of the Cray Programming Environment 5.0 releases with respect to expected functionality.

4.1 CrayPat

The following features of CrayPat are not yet supported:
- Running CrayPat in graphical user interface (GUI) mode
- Running CrayPat in PerfShell mode

4.2 ScaLAPACK and BLACS Performance

The ScaLAPACK and BLACS routines are not yet optimized. Performance improvements for these routines will be provided in the future.

4.3 Cray C and C++ Compilers

The pointer and array bounds checking option (-h bounds) is not fully supported. We expect to provide full support in a later release.

When including the Cray operating system and intrinsic.h header files, include the system file headers before the intrinsic.h file. Including the intrinsic.h file before the system header files causes the compiler to generate error messages about previously declared objects. The fix will be provided in a later update.

4.4 Unified Parallel C

Unified Parallel C (UPC) of the Cray C++ Programming Environment 5.0 release is not fully operational at this time because of a bug. A fix will be provided in the next update for the Cray C++ Programming Environment 5.0 release.
4.5 CPU Time Computation for Programs Using SSP Mode

Until UNICOS/mp 2.2 is released, there will be an anomaly in the computation of CPU time usage for processes running on MSPs versus SSPs. Currently, for applications using SSP mode, the system computes CPU usage time at 4 times the rate of applications using MSP mode. For example, a CPU-bound program compiled in MSP mode running for 1 second on one MSP uses 1 CPU second. The same program compiled in SSP mode and running on four SSPs for one second uses 1 CPU second for each SSP used, for a total of 4 CPU seconds. We believe this approach will be confusing, and plan to change the UNICOS/mp 2.2 release so programs running in SSP mode will be charged 1/4th of a CPU second for every second that an SSP is used. That is, if 4 SSPs are used for 1 second, the program will be charged 1 CPU second.
This chapter describes the documentation for the Cray Programming Environment 5.0 releases.

5.1 CrayDoc Documentation System

Cray provides a documentation system called CrayDoc, a collection of open-source software components. CrayDoc gives you fast, easy access and the ability to search all Cray manuals, man pages, and glossary in PDF and/or HTML format from a web browser. CrayDoc runs on any operating system based on UNIX or Linux system.

The installation and administration of the CrayDoc server software and Cray documentation are described in the CrayDoc Installation and Administration Guide.

5.2 Cray Manuals

Using a web browser, you can access Cray manuals at any of the following locations:

- Locally, using the network path defined by your system administrator
- On the Cray public web site at:
  http://www.cray.com/craydoc/
- On the CRInform site (for subscribers only):
  http://crinform.cray.com
  Click on the Release Documents link.
- On the Cray internal web site (Cray personnel only):
  http://swpubs.us.cray.com/craydoc/

Table 1 lists the manuals that are provided with this release. Most are provided as HTML files, and all are provided as PDF files. Those that are also provided in printed form are indicated in the table by an asterisk. Sites may install the CrayDoc system and allow users to view the HTML and PDF files through that interface or move the files to a location of their own choosing.
Table 1. Manuals Provided with These Releases

<table>
<thead>
<tr>
<th>Manual title</th>
<th>Publication</th>
<th>Updated</th>
</tr>
</thead>
<tbody>
<tr>
<td><em>Cray Programming Environment Releases Overview and Installation Guide</em> (this document)*</td>
<td>S–5212–50</td>
<td>Yes</td>
</tr>
<tr>
<td>Cray X1 System Overview</td>
<td>S–2346–22</td>
<td>Yes</td>
</tr>
<tr>
<td>Cray X1 User Environment Differences</td>
<td>S–2310–50</td>
<td>Yes</td>
</tr>
<tr>
<td>Migrating Applications to the Cray X1 System</td>
<td>S–2378–50</td>
<td>Yes</td>
</tr>
<tr>
<td>Optimizing Applications on the Cray X1 System</td>
<td>S–2315–50</td>
<td>Yes</td>
</tr>
<tr>
<td>Cray Fortran Compiler Commands and Directives Reference Manual</td>
<td>S–3901–50</td>
<td>Yes</td>
</tr>
<tr>
<td>Fortran Language Reference Manual, Volume 1</td>
<td>S–3692–50</td>
<td>Yes</td>
</tr>
<tr>
<td>Cray C and C++ Reference Manual</td>
<td>S–2179–50</td>
<td>Yes</td>
</tr>
<tr>
<td>Dinkum C++ Library Documentation</td>
<td>S–6506–42</td>
<td>No</td>
</tr>
<tr>
<td>Cray Assembly Language (CAL) for Cray X1 Systems Reference Manual</td>
<td>S–2314–50</td>
<td>Yes</td>
</tr>
<tr>
<td>Differences for Solaris Based Cray X1 Programming Environments</td>
<td>S-2313-50</td>
<td>New</td>
</tr>
</tbody>
</table>

* This document is also provided in printed form with this release.

Note: The *Cray Fortran Co-array Programming Manual* was incorporated into *Fortran Language Reference Manual, Volume 3*.

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

5.3 Cray Man Pages

Cray man pages are now provided for the BLACS library and can be accessed like other man pages. These man pages are included in Man Page Collection: Scientific Library Routines.

Table 2 lists the man page collections that are included with the releases. Man page collections are formatted man pages collected by topic and provided in PDF files for viewing and printing.

<table>
<thead>
<tr>
<th>Title</th>
<th>Publication</th>
<th>Updated</th>
</tr>
</thead>
<tbody>
<tr>
<td>Man Page Collection: Fortran Application Programmer’s I/O</td>
<td>S–2379–50</td>
<td>New</td>
</tr>
<tr>
<td>Man Page Collection: C/C++ Library Functions</td>
<td>S–2382–50</td>
<td>New</td>
</tr>
<tr>
<td>Man Page Collection: Fortran and C/C++ Intrinsic Procedures</td>
<td>S–2385–50</td>
<td>New</td>
</tr>
<tr>
<td>Man Page Collection: Scientific Library Routines</td>
<td>S–2386–50</td>
<td>New</td>
</tr>
</tbody>
</table>

The documentation for each of these related products is accessible when the corresponding product is installed:

- as(1) man page, included with the Cray Assembly Language package
- Modules man pages, included with the Modules package
- Motif man pages, included with the Motif package
- Trigger man pages and Man Page Collection: Trigger Commands (S–2387–50), included in the Trigger package

Man pages provide system and programming reference information. Each man page is referred to by its name followed by a number in parentheses:

`manpagename (n)`

where `n` is the man page section identifier:

1 User commands
2 System calls
3 Library routines
Access man pages in any of these ways:

- Enter the man command to view individual man pages in ASCII format; for example:
  
  ```
  man ftn
  ```

  To print individual man pages in ASCII format, enter, for example:
  
  ```
  man ftn | col -b | lpr
  ```

- Use a web browser with the CrayDoc system to view, search, and print individual man pages in HTML format.

- Use Adobe Acrobat Reader with the CrayDoc system to view, search, and print from collections of formatted man pages provided in PDF format.

If more than one topic appears on a page, the man page has one primary name (`grep`, for example) and one or more secondary names (`egrep`, for example). Access the ASCII or HTML man page using either name; for example:

- Enter the command `man grep` or `man egrep`
- Search in the CrayDoc system for `grep` or `egrep`

### 5.4 Error Message Explanations

You can 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, see the `explain(1)` man page.

### 5.5 Cray Glossary

A glossary of Cray terms that relate to the Cray system with which this release is associated is also included with CrayDoc.
5.6 Ordering Documentation

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

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

**Web:**
http://www.cray.com/craydoc/

Click on the Cray Publications Order Form link.

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

**Telephone (outside U.S., Canada):**
Contact your Cray representative, or call +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
This chapter contains information about hardware and software requirements, contents of the release package, and licensing.

6.1 Hardware and Software Requirements

The Cray Programming Environment 5.0 releases apply to Cray X1 systems running the UNICOS/mp 2.1 release and to Solaris systems running Solaris 8.0 or later.

**Note:** Cray Programming Environment 5.0 releases running on Cray X1 systems are dependent on the Trigger 2.0.0.3 software. Refer to Section 8.5, page 32 for information about installing and configuring the trigger product.

**Caution:** The Cray Programming Environment Server (CPES) 2.0 or later software cannot be used with a Sun Fire 4800 system serving as a CPES. When using a Sun Fire 4800 Server, use the CPES 1.0 software and documentation.

6.2 Contents of Release Package

The release packages of the Cray Programming Environment 5.0 releases for Cray X1 systems and Solaris systems contain these products:

- One or both of the following compilers (depending on your contractual license):
  - Cray Fortran Compiler version 5.0
  - Cray C++ Programming Environment containing:
    - Cray C compiler version 8.0
    - Cray C++ compiler version 5.0
  - CrayLibs version 5.0
  - LibSci version 5.0
  - CrayTools version 5.0 containing CrayPat and Cray loader
  - Documentation and the CrayDoc server software (described in Chapter 5, page 13)
Also included with the Cray Programming Environment 5.0 release packages are other related packages for Cray X1 systems and Solaris systems, except where noted:

- Cray Assembly Language (CAL) version 1.1.
- Trigger version 2.0.0.3 or later (not included with and not needed by the Solaris based Cray Programming Environments).
- Message Passing Toolkit (MPT) 2.2 release (included with the Solaris based Cray Programming Environments but a separately orderable and licensed product for Cray X1 systems). Refer to Cray Message Passing Toolkit Release Overview (S–3689–22) for more information.
- Modules version 3.1.6
- Motif version 2.1
- X Window System X11 libraries version 6.3

### 6.3 Licensing

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

The Programming Environment 5.0 releases for Cray X1 systems are licensed and supported for use only on the Cray Programming Environment Server (CPES). The contractual license for each Cray Programming Environment for Cray X1 Systems (Solaris based) limits customers to one copy and five concurrent users.

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.

For more information about licensing and pricing, contact your Cray representative, or contact Jenny Gross in any of the following ways:

E-mail:
jennyg@cray.com

Telephone:
+1–651–605–8982

Fax:
+1–651–605–9001

Mail:
Jenny Gross
Software Licensing
Cray Inc.
1340 Mendota Heights Road
Mendota Heights, MN 55120–1128
USA
This chapter describes the customer services that support this release running on Cray X1 systems and Solaris systems.

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):
Contact your Cray representative, or call +1–715–726–4993

CRInform (for subscribers):
http://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 Cray mainframe and has a software license agreement and software support agreement. Access CRInform at:

http://crinform.cray.com
Subscribers to CRInform can do any of the following activities:

- Report software problems
- Request technical assistance
- 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:**
wwwtng@cray.com

**Web:**
http://www.cray.com/training/training.html

**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 Web Site

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

http://www.cray.com/
This chapter tells you how to install the Cray Programming Environment 5.0 release products and related products specified in Section 6.2, page 19 on Cray X1 systems running the UNICOS/mp 2.1 release or later. This chapter also explains how to copy the necessary UNICOS/mp system files to the Cray Programming Environment Server (CPES).

Note: Chapter 9, page 37 tells you how to the Cray Programming Environment 5.0 release products and related products specified in Section 6.2, page 19 on Solaris systems.

You can use these tools to install the products:

- The Common Installation Tool (CIT) with product files on a CD.

For more information about using CIT, see the Common Installation Tool (CIT) Reference Card on the CD. The card can be printed from the /cdrom_mount/CYRIinstall/2218.ps PostScript file. You may also select the Help button in CIT, enter help all at the interactive interface prompt, or check the cit(8) man page.

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

- The opt_install script with product files that are either copied from a CD or downloaded and copied to the Cray X1 system.

Note: The installation of the Cray Programming Environment release packages does not install the Motif, Cray Assembly Language, Message Passing Toolkit (MPT), X Windows System X11 libraries, Trigger, or Modules packages. However, because the installation process for these products is identical to the process for installing the Programming Environment release products, the installation information described in this chapter also apply to these Programming Environment related products.

To install the Etnus TotalView release package on a Cray X1 system, see the TotalView Release Overview, Installation Guide, and User’s Guide Addendum for Cray X1 Systems (S-2370–61).
8.1 Installation Requirements

Before installing Cray Programming Environment 5.0 products, verify that these requirements have been met:

1. UNICOS/mp 2.1 release is installed and running on the Cray X1 system.
2. CPES is installed and running.
3. You have root permission on the Cray X1 system.
4. Root has write permission in the /opt/ctl directory.
5. Trigger version 2.0.0.3 or later is installed. See Section 8.5, page 32 to install the Trigger product.

8.2 Product Directories

On Cray X1 systems, each of the Programming Environment products is installed in the /opt/ctl/prod/version directory, where prod is the name of the product directory (for example, cftn) and version is the version of the product (for example, 5.0.0.0).

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 document, unless noted, are those seen while logged in to the Cray X1 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 allow compilers, loaders, and tools to access the correct include files and libraries.

### 8.3 Installation Using CIT

This section tells you how to install the Cray Programming Environment 5.0 release packages and related packages using CIT. The release CD contains the most current version of CIT.

**Note:** The Cray Programming Environment 5.0 releases require Trigger version 2.0.0.3 or later. To install the Trigger product, refer to Section 8.5, page 32 before continuing.

To install Cray Programming Environment 5.0 release products, complete these steps.

1. Enter the following commands to verify that the `~crayadm/.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 `crayadm` to send remote shell commands to the mainframe from the CWS:

   ```
cws$ rsh CrayHostName -l root "rsh CWS -l crayadm uname -a"
   ```

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

2. Enter:

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

3. Perform one of the following setup and installation processes for each CD:
   - To set up and install products without using defaults, enter:
     ```
     # setup -c Cray_System_Name -l root
     ```

     A Pre-install Queries window will appear for each product you are installing. After you have answered all the questions in the window, click the Done button. Installation will not begin until all questions have been answered for every product.
To set up and install products using defaults, enter:

```
#setup -c Cray_System_Name -D USE_DEFAULTS -l root
```

The Pre-install Queries window will not be displayed because this option tells CIT to use these defaults for all products during the installation:

- `overwrite=no`
- `default=no`
- `newfiles=no`

All products are installed in the `/opt/ctl/prod/version` directories. If the product already exists, the installation will abort if `overwrite` is `no`.

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.

      Depending on the Cray software you are installing, pre-installation 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 pre-installation information, CIT’s progress window appears and actual installation begins.

      As packages are loaded, they move 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.

5. Quit CIT.

6. If desired for security purposes, remove the `root` entry from the `~workstation_login/.rhosts` file on the workstation and the `workstation_login` entry from the `/root/.rhosts` file on the Cray X1 system.
8.4 Installation Using opt\_install

Use the opt\_install script to install product files that were either copied from a CD or, downloaded from the CRInform web site.

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

To install the Cray Programming Environment 5.0 release packages and related packages using opt\_install, complete these steps.

1. Log onto the Cray X1 system.

2. Execute the opt\_install script in this working directory by entering:

   ```
   #/opt/PE/admin/bin/opt\_install filename.cpio
   ```

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

   This example shows the command line for installing Cray C++ version 5.0:

   ```
   #/opt/PE/admin/bin/opt\_install CC_5000_cray-x1.cpio
   ```

   You can install a product in a nondefault location by using opt\_install and setting the environment variable ROOT to the alternate directory.

To allow testing to be performed on a new version of a product, the old 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 (see Section 9.2, page 37 for more information). 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 enter this on the command line:

```
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

8.5 Installing and Configuring Trigger Software on Cray X1 Systems

The trigger files install like other Cray Programming Environment products—either via CIT from the CD or via the /opt/PE/admin/bin/opt_install script using the downloaded or copied product file.

Obtain the install package and load it onto the Cray X1 system. Directories will be created at /opt/PE/trigger/version and /opt/ctl/trigger/version as viewed from the Cray X1 system (version is the version of the trigger product being installed. For instructions on using CIT to install the trigger product, see Section 8.3, page 29; for instructions on using opt_install, see Section 8.4, page 31.

During this part of the installation process, you will be prompted about overwriting the same version, if it exists, as this example shows:

```
trigger 2.0.0.0 already installed ... overwrite [n/y]?
```

**Note:** The trigger product is always installed as nondefault 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 install location of trigger, since it cannot be installed from the install package as default.

After you have completed the CIT or /opt_install steps, you must do the configuration and daemon restart as described in the following steps.

Starting the new trigger daemon will interrupt the connection between the Cray X1 system and CPES. Any processes started by a triggered product will be interrupted, so starting a new trigger daemon should not be done while triggered processes are running on the CPES. Operation of the Cray X1 system will not be affected.
To configure the trigger environment:

1. Log onto the Cray X1 system. Edit the file
   /opt/ctl/trigger/version/bin/trigexecd.cfg to define
   PASS and CLIENT. PASS is the site-selected password that is also defined in
   /opt/PE/trigger/version/bin/trigsnd.cfg. CLIENT is the Cray X1
   machine name.

   PORT=9930
   PASS=
   CLIENT=

2. Set the default link for trigexecd (this will not cause any interruption):

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

   where version is the name of the directory containing the version of trigger to
   use.

3. Still logged on to the Cray X1 system, edit the lines shown in the
   /opt/PE/trigger/version/bin/trigsnd.cfg file:

   Table 3. Lines to Edit in trigsnd.cfg

<table>
<thead>
<tr>
<th>Line</th>
<th>Comments</th>
</tr>
</thead>
<tbody>
<tr>
<td>TRIGROOT=</td>
<td>The /sv2 file system on the CPES. Depending on how your CPES is configured, you will set TRIGROOT= to either /sv2 or /sv2/$RCVHOST/</td>
</tr>
<tr>
<td>RCVHOST=</td>
<td>The CPES machine name</td>
</tr>
<tr>
<td>PASS=</td>
<td>The site-selected password that is also defined in /opt/ctl/trigger/version/bin/trigexecd.cfg</td>
</tr>
<tr>
<td>PORT=9930</td>
<td>We use port 9930 in the example; however, the port number can be site-selected as long as the values in trigsnd.cfg and trigexecd.cfg match.</td>
</tr>
</tbody>
</table>
4. Set the default link for trigsnd. This step and step 5 will cause interruption to the user as the daemon is restarted.

```bash
# cd /opt/PE/trigger

# rm trigger

# ln -s version trigger
```

5. Start or restart the trigger daemon on the CPES. Log onto the CPES.

- If this is an initial installation, edit /etc/inittab and add this line:
  ```
  tr:234:respawn:/sv2/opt/ctl/trigger/trigger/bin/trigexecd
  ```
  As root, kill the current trigger daemon process.

- If this is an upgrade, get the process ID and kill the trigger daemon process. For example:
  ```
  # ps -ef | grep trig
  root 9190 1 0 Dec 17 ? 3:40 /sv2/opt/ctl/trigger/trigger/bin/trigexecd
  
  # kill -15 9190
  ```
  Since trigexecd is part of inittab, it will respawn when killed, using the new trigexecd.

6. Test the setup by logging onto the Cray X1 system with module PrgEnv loaded.

```bash
% module load PrgEnv

% ftn -V
```

Cray Fortran: Version 4.3.0.0 Tue Dec 31, 2002 13:11:04

### 8.6 Copy System Libraries and Include Files to the CPES

If the UNICOS/mp system libraries and include files were not copied to the Cray Programming Environment Server (CPES) the last time the OS was upgraded, you must do so before using Cray Programming Environments.
As root on the Cray X1 system, execute the /opt/PE/admin/bin/cp_2_cpes utility to copy the Cray X1 mainframe library files from the /usr/lib/nonshared directory and the include files from the /usr/include directory to these directories on the CPES:

/opt/ctl/include_{OS_LEVEL}
/opt/ctl.libs_{OS_LEVEL}

where OS_LEVEL is the version of the UNICOS/mp operating system files you are copying.

For example, the cp_2_cpes utility creates the following directories when copying the files from UNICOS/mp 2.1 operating system:

/opt/ctl/include_2.1
/opt/ctl/libs_2.1
This chapter tells you how to install the Cray Programming Environment 5.0 release products and related products specified in Section 6.2, page 19 on Solaris systems running the Solaris operating system 8.0 release or later.

You can use these tools to install the products:

- The Common Installation Tool (CIT) with product files on a CD.
  
  For more information about using CIT, see the *Common Installation Tool (CIT) Reference Card* on the CD. The card can be printed from the `/cdrom_mount/CYRIinstall/2218.ps` PostScript file. You may also select the Help button in CIT, enter `help all` at the interactive interface prompt, or check the `cit(8)` man page.

  **Note:** Installation log files are located on the workstation at:

  `/tmp/cit.workstation_login/* .log`

- The `opt_install` script with product files that are either copied from a CD or downloaded to the Solaris system.

### 9.1 Installation Requirements

Before installing the Cray Programming Environment 5.0 release products, verify that these requirements have been met:

- A `/sv2` file system exists and is sufficiently large enough to contain the packages you want to install. The `/sv2` mount point must be in the `root` file system.

- You have root permission on the Solaris system.

- Root has write permission in to `/sv2`.

### 9.2 Solaris File System Requirements

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

- The `/sv2` file system must have sufficient disk space for the software packages that will be installed.
• The temporary 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 (craylibsci), and CrayTools (craytools) packages. That is, the total disk space requirement for both Programming Environments is 355 MB. Also, it is important to make the /sv2 file system sufficiently large to accommodate future releases and products.

Table 4. Solaris Based Cray Programming Environment 5.0 Releases Disk Space Requirements

<table>
<thead>
<tr>
<th>Package Name</th>
<th>Required Space</th>
</tr>
</thead>
<tbody>
<tr>
<td>Packages of the Cray Fortran</td>
<td>Total disk space for the product is 332 MB. Disk space required for each package:</td>
</tr>
<tr>
<td>Programming Environment product:</td>
<td></td>
</tr>
<tr>
<td>• CPTN</td>
<td>• 20 MB</td>
</tr>
<tr>
<td>• craylibs</td>
<td>• 38 MB</td>
</tr>
<tr>
<td>• craylibsci</td>
<td>• 181 MB</td>
</tr>
<tr>
<td>• craytools</td>
<td>• 93 MB</td>
</tr>
<tr>
<td>Packages of the C++ Programming</td>
<td>Total disk space for the product is 335 MB. Disk space required for each package:</td>
</tr>
<tr>
<td>Environment product:</td>
<td></td>
</tr>
<tr>
<td>• CC</td>
<td>• 23 MB</td>
</tr>
<tr>
<td>• craylibs</td>
<td>• 38 MB</td>
</tr>
<tr>
<td>• craylibsci</td>
<td>• 181 MB</td>
</tr>
<tr>
<td>• craytools</td>
<td>• 93 MB</td>
</tr>
<tr>
<td>cal</td>
<td>4 MB</td>
</tr>
<tr>
<td>X11</td>
<td>63 MB</td>
</tr>
<tr>
<td>Motif</td>
<td>20 MB</td>
</tr>
<tr>
<td>mpt</td>
<td>37 MB</td>
</tr>
</tbody>
</table>
The disk space requirement for the temporary directory holding the temporary installation files is the same space needed by the packages to be installed; that is, if you are installing the Cray Fortran Programming Environment, the disk space requirement for the temporary directory is 332 MB. For more information about the location of the temporary directory, see Section 9.5, page 41.

### 9.3 Product Directories

On Solaris systems, each of the Cray Programming Environment 5.0 release products is installed in the `/sv2/opt/ctl/prod/version` directory, where `prod` is the name of the product directory (for example, CFTN) and `version` is the version of the product (for example, 5.0.0.0).

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.

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

```
/sv2/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:

```
/sv2/opt/PE/bin
```

Driver scripts (such as `ftn`) are used to access the executable files for which they are named. Driver scripts also allow compilers, loaders, and tools to access the correct include files and libraries.

### 9.4 Installation Using CIT

This section tells you how to install the Cray Programming Environment 5.0 release packages and related packages using CIT. The release CD contains the most current version of CIT.
To install Cray Programming Environment 5.0 release products, complete these steps:

1. If you are not installing across a network, skip to step 2. Otherwise, perform the following instructions. If you are installing the products onto a system that is across the network from the system that mounts the release CD, verify that both systems can communicate with each other. To verify network communications, log onto the system that mounts the release CD and enter this statement on the command line:

   \texttt{admin\$ rsh \textit{Solaris\_system\_name} \ -l root "rsh \textit{Solaris\_admin\_system} \ -l admin \texttt{uname -a}}

   where \textit{Solaris\_system\_name} is the system to install the products onto and \textit{Solaris\_admin\_system} is the system that reads the release CD.

   If the command fails, verify that the \texttt{.rhost} file on both systems are properly configured.

2. Enter:

   \texttt{# cd /cdrom\_mount/cdrom0}

3. Perform one of the following setup and installation processes for each CD:

   \begin{itemize}
   \item To set up and install products without using defaults, enter:
     \texttt{# setup -c \textit{Solaris\_System\_Name} \ -l root}
   \item To set up and install products using defaults, enter:
     \texttt{# setup -c \textit{Solaris\_System\_Name} \ -D USE\_DEFAULTS \ -l root}
   \end{itemize}

   The Pre-install Queries window will not be displayed because this option tells CIT to use these defaults for all products during the installation:

   \begin{verbatim}
   overwrite=no
default=no
newfiles=no
   \end{verbatim}

   All products are installed in the /sv2/opt/ctl/prod/\textit{version} directories. If the product already exists, the installation will abort if overwrite is no.

4. Load the packages:

   \begin{itemize}
   \item 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.
   \end{itemize}
b. Click on the Install button at the bottom of the CIT window to load the packages listed in the Order of Installation panel.

An Installation Options window will appear for each product you are installing. After you have answered all the questions in the window, click the Done button. Installation will not begin until all questions have been answered for every product.

As packages are loaded, they move 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.

5. Quit CIT.

9.5 Installation Using opt_install

Use the opt_install script to install product files that were either copied from a CD or, if you a subscriber of CRInform, downloaded from the CRInform web site.

The opt_install script is located in the /sv2/opt/PE/admin/bin directory after the CrayTools product has been installed once.

To install the Cray Programming Environment 5.0 release packages and related packages using opt_install, complete these steps.

1. Execute the opt_install script by entering:

   
   ```
   # /sv2/opt/PE/admin/bin/opt_install filename.cpio
   ```

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

   This example shows the command line for installing Cray C++ version 5.0:

   ```
   # /sv2/opt/PE/admin/bin/opt_install CC_5000_cray-xl.cpio
   ```

   You can install a product in a nondefault location by using opt_install and setting the environment variable ROOT to the alternate directory.
To allow testing to be performed on a new version of a product, the old 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` (see Section 9.2, page 37 for more information). 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 enter this on the command line:

```
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 `/sv2/opt/ctl/prod/version/.install_log` file is the record of the installation steps.

If an installation fails, the installation file is placed in the `$TMPDIR/NEW_prod/install.log` directory.
Publications, 13

R
Reader comments, 2
Recompiling, 9
Release package, 19
Request for Technical Assistance (RTA), 23–24
Requirements, hardware and software, 19

S
Software license agreement, 20
Software Problem Report (SPR), 23–24
Software Publications contact information, 2
Subscriber
   CRInform, 23
Support agreement, 23
Support Center, 23

T
Technical support, 23
Temporary directory defining another, 31, 42
Temporary directory space requirements, 39
TotalView, 27
Training, 24
Trigger Environment, 32
Triggers, 20

W
Web sites
   See Cray web sites

X
X Windows System, 20
X11, 20