Preface

This publication documents the differences between Cray Programming Environment products running on CRAY T3E systems and the same products running on CRAY T3D systems.

Related publications

The following documents contain additional information that may be helpful:

- *Cray Standard C Reference Manual*, publication SR–2074
- *Scientific Libraries Reference Manual*, publication SR–2081
- *Intrinsic Procedures Reference Manual*, publication SR–2138
- *Application Programmer's Library Reference Manual*, publication SR–2165
- *Cray C/C++ Reference Manual*, publication SR–2179
- *Cray C/C++ Ready Reference*, publication SQ–2180
- *Message Passing Toolkit: PVM Programmer's Manual*, publication SR–2196
- *Introducing the Cray TotalView Debugger*, publication IN–2502
- *Cray Assembler for MPP (CAM) Reference Manual*, publication SR–2510
- *Introducing the MPP Apprentice Tool*, publication IN–2511
- *CF90 Ready Reference*, publication SQ–3900
- *CF90 Commands and Directives Reference Manual*, publication SR–3901
- *Fortran Language Reference Manual, Volume 1*, publication SR–3902
- *Fortran Language Reference Manual, Volume 2*, publication SR–3903
- *Fortran Language Reference Manual, Volume 3*, publication SR–3905

Ordering Cray Research publications

The *User Publications Catalog*, publication CP–0099, describes the availability and content of all Cray Research hardware and software documents that are available to customers. Cray Research customers who subscribe to the Cray

Inform (CRInform) program can access this information on the CRInform system.

To order a document, either call the Distribution Center in Mendota Heights, Minnesota, at +1–612–683–5907, or send a facsimile of your request to fax number +1–612–452–0141. Cray Research employees may send electronic mail to `orderdsk` (UNIX system users).

Customers who subscribe to the CRInform program can order software release packages electronically by using the `Order Cray Software` option.

Customers outside of the United States and Canada should contact their local service organization for ordering and documentation information.

Conventions

The following conventions are used throughout this document:

Convention	Meaning
`command`	This fixed-spaced font denotes literal items such as commands, files, routines, path names, signals, messages, and programming language text.
`manpage`(*x*)	Man page section identifiers appear in parentheses after man page names. The following list describes the identifiers:

1	User commands
1B	User commands ported from BSD
2	System calls
3	Library routines, macros, and opdefs
4	Devices (special files)
4P	Protocols
5	File formats
7	Miscellaneous topics
7D	DWB-related information

	8 Administrator commands
	Some internal routines (for example, the `ddcnt()` routine) do not have man pages associated with them.
variable	Italic typeface denotes variable entries and words or concepts being defined.
`user input`	This bold, fixed-space font denotes literal items that the user enters in interactive sessions. Output is shown in nonbold, fixed-space font.
[ ]	Brackets enclose optional portions of a command or directive line.
...	Ellipses indicate that a preceding element is repeated.

The following machine naming conventions may be used throughout this document:

Term	Definition
Cray PVP systems	All configurations of Cray parallel vector processing (PVP) systems, including the following: CRAY C90 series CRAY C90D series CRAY J90 series CRAY T90 series
Cray MPP systems	All configurations of the CRAY T3E series
All Cray Research systems	All configurations of Cray PVP and Cray MPP systems that support this release

Reader comments

If you have comments about the technical accuracy, content, or organization of this document, please tell us. You can contact us in any of the following ways:

- Send us electronic mail at the following address:

  `publications@cray.com`

- Contact your customer service representative and ask that an SPR or PV be filed. If filing an SPR, use `PUBLICATIONS` for the group name, `PUBS` for the command, and `NO-LICENSE` for the release name.

- Call our Software Publications Group in Eagan, Minnesota, through the Customer Service Call Center, using either of the following numbers:

  1–800–950–2729 (toll free from the United States and Canada)

  +1–612–683–5600

- Send a facsimile of your comments to the attention of “Software Publications Group” in Eagan, Minnesota, at fax number +1–612–683–5599.

We value your comments and will respond to them promptly.

Introduction [1]

This document describes the differences between CRAY T3E and CRAY T3D programming environments. The goal in implementing CRAY T3E software was that users should have to do no more than recompile, relink, and run applications without change. The differences described in this document are the exceptions for which achieving the goal was not possible.

The chapters in this book summarize the differences between use of the following products on CRAY T3E systems and CRAY T3Dsystems:

- Fortran and related libraries (see Chapter 2, page 3).
- The Cray Standard C and Cray C++ program languages, and the C and C++ libraries (see Chapter 3, page 5).
- The Cray Assembler for MPP (CAM) (see Chapter 4, page 11).
- Scientific libraries (see Chapter 5, page 13).
- System calls (see Chapter 6, page 15).
- Parallel Virtual Machine (PVM) routines (see Chapter 7, page 17).
- Shared memory (SHMEM) routines (see Chapter 8, page 19).
- The CLD linker (see Chapter 9, page 21). This chapter compares CLD, which is a new product, to the CRAY T3D loader, MPPLD.
- Cray TotalView debugger (see Chapter 10, page 33).

The following products have not changed between the CRAY T3E and CRAY T3D systems. They behave the same on both systems:

- MPP Apprentice tool
- The Program browser, xbrowse
- Math libraries

The list of differences for some of these products will change as subsequent versions are released.

The following differences either cover more than one product or refer to the overall programming environment:

- The `mppexec`(1) command is not implemented on the CRAY T3E system. The `mpprun`(1) command takes its place.
- Plastic `a.out`(5) files are not supported on CRAY T3E systems. A plastic `a.out` file (supported only on CRAY T3D systems) is a Fortran executable that can be reconfigured with respect to the number of PEs assigned to a job. For example, a static array that is dimensioned by a function of `N$PES` changes size given different values of `N$PES`. A malleable `a.out` file (supported only on CRAY T3E systems) is an executable that has no dependence on the value of `N$PES`. If you specify the `-Xm` option on your `f90`(1), `cc`(1), `CC`(1), or `cld`(1) command line, you can choose the number of PEs when you execute a program by including the `-n` option on your `mpprun`(1) command line. All arrays in this type of executable are allocated dynamically.
- The CRAY T3E system is self-hosted, which means that the UNICOS/mk operating system runs on the same processing elements (PEs) on which the applications are executing.
- The power-of-two restriction on the number of PEs, which existed for CRAY T3D systems, has been removed for CRAY T3E systems
- The Message Passing Toolkit (MPT) is supported on CRAY T3E systems. MPT includes optimized versions of PVM and the Message Passing Interface (MPI). On CRAY T3D systems, an optimized version of PVM is included in CrayLibs, and an optimized version of MPI is supported by Edinburgh Parallel Computing Centre (EPCC). MPI will not be supported in the initial release of MPT for CRAY T3E systems (MPT 1.1). The current plan is to support it in MPT 1.2.

Fortran [2]

The differences between Fortran on CRAY T3E systems and Fortran on CRAY T3D systems are as follows:

- The Cray Research Fortran CFT77 compiler executes only on CRAY T3D systems. The Cray Research CF90 compiler is the only Fortran compiler on CRAY T3E systems. If you will be migrating your program from a CFT77 compiler to a CF90 compiler, the *Fortran Language Reference Manual*, publication SR–3905, lists the differences between the CF90 language and the CFT77 language.
- For CRAY T3D systems, the CF77 executables and libraries are stored in `/opt/ctl/cf77_m/`*version-number*`/bin` and `/opt/ctl/cf77_m/`*version-number*`/lib`. On CRAY T3E systems, the CF90 executables and libraries are stored in `/opt/ctl/cf90/`*version-number*`/bin` and `/opt/ctl/cf90/`*version-number*`/lib`.
- The following intrinsic routines are not available on CRAY T3D systems but are available on CRAY T3E systems. They are part of the implementation of the *IEEE Standard for Binary Floating-Point Arithmetic*, ANSI/IEEE standard 754–1985:

  `clear_ieee_exception`(3)

  `disable_ieee_interrupt`(3)

  `enable_ieee_interrupt`(3)

  `get_ieee_exceptions`(3)

  `get_ieee_interrupts`(3)

  `get_ieee_rounding_mode`(3)

  `get_ieee_status`(3)

  `ieee_binary_scale`(3)

  `ieee_class`(3)

  `ieee_copy_sign`(3)

  `ieee_exponent`(3)

  `ieee_finite`(3)

  `ieee_int`(3)

`ieee_is_nan`(3)

`ieee_next_after`(3)

`ieee_real`(3)

`ieee_remainder`(3)

`ieee_unordered`(3)

`int_mult_upper`(3)

`set_ieee_exception`(3)

`set_ieee_exceptions`(3)

`set_ieee_interrupts`(3)

`set_ieee_rounding_mode`(3)

`set_ieee_status`(3)

`test_ieee_exception`(3)

`test_ieee_interrupt`(3)

The following differences are not related to the IEEE standard:

– The `write_memory_barrier`(3) routine is available only on CRAY T3E systems. For more information about this routine, see the man page.

– There are currently no plans to support the FFIO ER90 layer on CRAY T3E systems. This layer is part of CrayLibs on CRAY T3D systems. It is selected by using `assign -F er90` or `asgcmd -F er90`. On CRAY T3D system, users can access tapes with the ER90 layer.

- The `!DIR$ SPLIT` source directive and the `-O split` command-line option are available only on CRAY T3E systems. For more information, see the *CF90 Commands and Directives Reference Manual*, publication SR–3901, and *CRAY T3E Fortran Optimization Guide*, publication SG–2518.
- The `!DIR$ USES_EREGS` source directive is available only on CRAY T3E systems. For more information, see the `f90`(1) man page.
- Task common data is available only on CRAY T3E systems.
- Bottom loading, an optimization technique that involves retrieving data before it is needed, is available only on CRAY T3E systems.

C and C++ [3]

The Cray Standard C and Cray C++ compilers are documented in the *Cray C/C++ Reference Manual*, publication SR–2179. See Table 1, page 5 for clarification of the appropriate compiler release level for your system. For additional, up-to-date information on the C and C++ compilers, see the `CC`(1) man page.

Table 1. Compiler release levels

Compiler	Release level for Programming Environment Release 3.0.1	
	CRAY T3D	CRAY T3E
Standard C	5.0	6.0.1
Cray C++	2.0	3.0.1

3.1 PE 2.0.1 differences

The two compilers share the following differences between the CRAY T3D system and the CRAY T3E system:

- Intrinsic functions. The `_write_memory_barrier`(3) function is available only on the CRAY T3E system.
- Include files. The `/usr/include` directory is searched by default for include files on CRAY T3E systems, instead of the `/usr/include/mpp` directory on CRAY T3D systems.
- Predefined macros. The `_CRAYT3E` macro is predefined on the CRAY T3E system; the `_CRAYT3D` macro is predefined on the CRAY T3D system.
- `#pragma` directives. The following directives are available only on CRAY T3E systems: `#pragma _CRI split` and `#pragma _CRI uses_eregs`.
- Command-line options. The `-h jump` option is the default on the CRAY T3E system; `-h nojump` is the default option on the CRAY T3D system. The `-h split` and `-h unroll` options are specific to the CRAY T3E system. The `-Xm` option creates a malleable `a.out`(5) file only on CRAY T3E systems. (For more information on malleable `a.out` files, see Chapter 1, page 1.)

- File system. For CRAY T3D systems, the Standard C executables and libraries are stored in `/opt/ctl/scc_m/scc_m/bin` and `/opt/ctl/scc_m/scc_m/lib`, and the C++ executables and libraries are stored in `/mpp/bin` and `/mpp/lib`. On CRAY T3E systems, the Standard C and C++ executables are stored in `/opt/ctl/CC/CC/bin` and the C++ libraries in `/opt/ctl/CC/CC/lib`. The `libc.a` functions are stored in `/mpp/lib` for CRAY T3D systems and `/lib` on CRAY T3E systems.
- New libraries. Almost all of the usual UNICOS libraries are built for the UNICOS/mk operating system on CRAY T3E systems, many more than for the UNICOS MAX operating system on the CRAY T3D system. The libraries available only on CRAY T3E systems include the following:
  - `libcrypt`
  - `libcurses`
  - `libgen`
  - `libl`
  - `libshare`
  - `libuex`
  - `liby`
- New library routines. The following routines are not supported on CRAY T3D systems but are supported on CRAY T3E systems:
  - The `getfsent`(3) routines.
  - The `getmntent`(3) routines.
  - The `ia_failure`(3), `ia_mlsuser`(3), `ia_success`(3), and `ia_user`(3) routines.
  - The `shutdsav`(3) routine.
- Library routines removed. The following functions are supported on the CRAY T3D system but not on the CRAY T3E system:
  - Functions dealing with the block transfer engine, such as `blt_copy`
  - Functions dealing with user protocol threads, such as `upt_create`, `upt_resume`, `upt_suspend`, and `upt_terminate`
  - Lock functions from CRAFT

  - The FFIO ER90 layer on CRAY T3E systems. This layer is part of CrayLibs on CRAY T3D systems. It is selected by using `assign -F er90` or `asgcmd -F er90`.

- The `malloc_brk`(3) routine (see the `malloc`(3) man page) need not be called on CRAY T3E systems to make an arbitrary private heap address remotely accessible.

3.2 New Features for PE 3.0.1

This section describes new Cray Standard C and Cray C++ features for CRAY T3E and Cray PVP systems. The features that are supported on CRAY T3E systems only or on Cray PVP systems only are specified in the text. For more information on these features, see the *Cray C/C++ Reference Manual*, publication SR–2179. The PE 3.0.1 release is not supported on CRAY T3D systems.

- Default compilers. Beginning with the Cray C/C++ 3.0 Programming Environment, Standard Cray C (SCC) 6.0 has become the default C compiler. This compiler provides a common front end for C and C++ language support. The default in the Cray C/C++ 2.0 Programming Environment was SCC 4.0 on all Cray PVP systems except for CRAY T90 IEEE systems. SCC 5.0 was the default for CRAY T90 IEEE systems and CRAY T3E systems. SCC 5.0 is provided as a transitional compiler in the 3.0 Programming Environment.

- Standard Template Library (STL). The Cray C++ compiler now supports the Standard Template Library (STL), which is a library of container classes, algorithms, and iterators. It provides many of the basic algorithms and data structures of computer science. A snapshot of STL has been placed on the documentation CD in the Cray C++ Programming Environment 3.0 release package. More information is available at the following URL: `http://www.sgi.com/Technology/STL/index.html`

- Support for new C++ draft standard. The Cray C++ compiler provides support for the new C++ draft standard language features. Supported features include run-time type identification (RTTI), name spaces, wide characters, member templates, partial specialization, booltype, plain `char` as a separate type, and other compatibility features.

- Detection of taskcommon errors. The Cray Standard C and Cray C++ compilers now detect such errors as specifying a taskcommon variable as `const`, specifying a static class variable as `taskcommon`, and so on.

- Enhancement of error checks for tasking directives. Error checks for tasking directives have been enhanced to support detection of branches into or out of tasking regions, which are regions of code under the control of `parallel`/`endparallel`, `guard`/`endguard`, `taskloop`/`endloop`, and `case`/`endcase` directives.
- Improved vector performance. The vector performance of many loops is enhanced by an improved chaining method between vector addition and multiplication operations. This optimization is also available on CF90 compilers.
- Support for 32-bit HCOSS on CRAY T3E systems. Calls to `sinf` and `cosf` with identical float (32-bit) operands are now merged into a single call to `HCOSS` on CRAY T3E systems. This is already being done for 64-bit types. This optimization is performed by default along with all other scalar optimizations. This optimization is also available on CF90 compilers.
- New template command-line options. New template command-line options for the `CC`(1) command are as follows:

Option	Description
`-h prelink_local_only`	Indicates that only local files (that is, those in the current directory) are candidates for assignment of instantiations.
`-h prelink_copy_if_nonlocal`	Indicates that the assignment of an instantiation to a nonlocal object file should result in the object file being recompiled in the current directory.
`-h remove_instantiation_flags`	Indicates that the prelinker must recompile all of the sources in such a way as to remove all instantiation flags.

- `#pragma` directives. The `#pragma ivdep` directive is now supported on CRAY T3E systems, allowing loops to be processed by loop splitting, intrinsic vectorization, `#pragma cache_bypass`, and other optimizations. A corresponding directive is also available on CF90 compilers.

The #pragma cache_bypass directive provides a semi-automatic method for programmers to run local memory references for arrays through E-registers, as in the following example for the arr array: #pragma _CRI cache_bypass arr. Loops preceded with this directive can obtain run-time reductions of up to 50 percent. A corresponding directive is also available on CF90 compilers.

The #pragma _CRI nointerchange directive inhibits the compiler's ability to interchange the loop that follows the directive with another inner or outer loop.

The #pragma _CRI concurrent directive indicates that no dependencies exist between different array references.

- Software pipelining on CRAY T3E systems. Software pipelining is now available through the use of the -h pipeline*n* compiler command-line option. *n* specifies the level of pipelining, as follows:

0	Specifies no pipelining (default).
1	Specifies safe pipelining.
2	Specifies level 1 plus operator reassociation (deferred, same as 1).
3	Specifies level 2 plus speculative execution (deferred, same as 1).

- New command-line option for vectorization on CRAY T3E systems. Cray Standard C and Cray C++ compiler command-line option, -h vector3, is now supported on CRAY T3E systems. This option directs the compiler to use vector versions of some standard math and intrinsic functions when they are found in vectorizable loops.

- Support for vectorization of generalized min / maxconstructs on Cray PVP systems. Support is now provided to vectorize loops with minimum or maximum values being calculated in a conditionally executed block that contains additional, arbitrary code. This feature provides improved functionality and performance.

CAM [4]

The Cray Assembler for MPP (CAM) version 2.2 includes support for the CRAY T3E system. CAM version 2.1 supports only the CRAY T3D system. Many of the differences between the two versions reflect the change in hardware: most significantly, CRAY T3D systems use the Alpha EV4 microprocessor from Digital Equipment Corporation (DEC), and CRAY T3E systems use DEC's Alpha EV5 microprocessor. The differences between CAM running on a CRAY T3E system versus a CRAY T3D system are as follows:

- The calling sequence has been modified on the CRAY T3E system to support new hardware. The largest change involves conventions for E register use.
- New user-level instructions associated with the EV5 processor, such as `wmb` and `excb`, have been added to CRAY T3E systems.
- New privilege assignment list (PAL) instructions for the EV5 microprocessor differ from those of the EV4 microprocessor.
- Endian behavior and how it relates to the assembly language programmer has changed. The `EXT`*xx*, `MSK`*xx*, and `INS`*xx* instructions perform byte operations on registers. On CRAY T3D EV4 systems, these instructions all behave in little endian fashion, with byte 0 in the rightmost position. On CRAY T3E EV5 systems, these instructions exhibit big endian behavior, with byte 0 in the leftmost position. A CAM program that uses these instructions will exhibit different behavior.
- The `cache` directive now equates to the size of the cache line (8 words) in the secondary cache for CRAY T3E systems, not the CRAY T3D system size of 4 words, which was for the primary cache.
- The virtual addresses format changed between the CRAY T3D system and the CRAY T3E system.
- The location of `mpp/asdef.h` changed from `/usr/include/mpp/mpp/asdef.h` on CRAY T3D systems to `/usr/include/mpp/asdef.h` on CRAY T3E systems.

For more information, see *Cray Assembler for MPP (CAM) Reference Manual*, publication SR–2510, or the `cam`(1) man page.

Cray TotalView Debugger [10]

The following differences exist between the CRAY T3E version and the CRAY T3D version of the Cray TotalView debugger:

- Core files under the UNICOS/mk operating system on the CRAY T3E system are named `core`, as opposed to `mppcore` on the CRAY T3D system under the UNICOS MAX operating system.
- When debugging core files on a CRAY T3E system, you must specify the name of the executable file to get symbolic, source-level debugging.

For more information, see *Introducing the Cray TotalView Debugger*, publication IN–2502, or the `totalview`(1) man page.