

# **HP C/HP-UX Version A.10.32.30 Release Notes**

**For HP-UX 10.20 Systems**



**Manufacturing Part Number: 5969-0354**

**June 1999**

© Copyright 1999 Hewlett-Packard Company

---

## Legal Notice

*Hewlett-Packard makes no warranty of any kind with regard to this manual, including, but not limited to, the implied warranties of merchantability and fitness for a particular purpose.* Hewlett-Packard shall not be liable for errors contained herein or direct, indirect, special, incidental or consequential damages in connection with the furnishing, performance, or use of this material.

Copyright © 1999 Hewlett-Packard Company.

This document contains information which is protected by copyright. All rights are reserved. Reproduction, adaptation, or translation without prior written permission is prohibited, except as allowed under the copyright laws.

### Corporate Offices:

*Hewlett-Packard Co.  
3000 Hanover St.  
Palo Alto, CA 94304*

Use, duplication or disclosure by the U.S. Government Department of Defense is subject to restrictions as set forth in paragraph (b)(3)(ii) of the Rights in Technical Data and Software clause in FAR 52.227-7013.

Rights for non-DOD U.S. Government Departments and Agencies are as set forth in FAR 52.227-19(c)(1,2).

Use of this manual and flexible disc(s), compact disc(s), or tape cartridge(s) supplied for this pack is restricted to this product only. Additional copies of the programs may be made for security and back-up purposes only. Resale of the programs in their present form or with alterations, is expressly prohibited.

A copy of the specific warranty terms applicable to your Hewlett-Packard product and replacement parts can be obtained from your local Sales and Service Office.

OpenGL is a registered trademark of Silicon Graphics, Inc.

PostScript is a trademark of Adobe Systems Incorporated.

UNIX is a registered trademark of The Open Group.

---

# Contents

## 1. New and Changed Features

New or Changed Compiler Options .....	7
+objdebug Option .....	8
+W n1[, n2, ... nN] Option .....	8
+We n1[, n2, ...nN] Option .....	9
+Ww n1[, n2, ...nN] Option .....	9
+Olevel=name1[,name2,...] Option .....	9
+Oreusedir=directory Option .....	10
Improving Shared Library Performance .....	10
HP_NO_RELOCATION Pragma .....	11
HP_LONG_RETURN Pragma .....	12
HP_DEFINED_EXTERNAL Pragma .....	13
New Compiler Defaults .....	14
HP WDB Debugger .....	15
Summary of HP WDB 1.1 Features .....	15

## 2. Installation Information

HP C/ANSI Developer's Bundle Components .....	19
Disk Space Requirements for HP C .....	20
HP WDB Components .....	20
Disk Space Requirements for HP WDB .....	20
Filesets .....	21
Supported Configurations .....	21
Installation Instructions .....	21

## 3. Documentation Overview

C Compiler Documentation .....	23
HP C/HP-UX Release Notes .....	23

---

## Contents

Printed Documentation . . . . .	24
HP C Online Help . . . . .	24
Mailing List with Product News . . . . .	25
HP WDB Documentation . . . . .	26
Mailing List for Product Announcements . . . . .	27
Related Documentation . . . . .	27
<b>4. Problem Descriptions and Fixes</b>	
Getting Information about Problems and Fixes . . . . .	29
HP C Problems and Limitations . . . . .	30
Required Patch . . . . .	30
Nonterminating Parallel Processes . . . . .	30
Memory Allocation Routine alloca() . . . . .	31
HP WDB Problems and Limitations . . . . .	31
HP WDB Limitation on Window Resizing with TUI . . . . .	31
HP WDB Limitation with Processes Started across NFS Mounts . . . . .	32
Terminal Window Sometimes Fails to Start from WDB GUI on 10.20 . . . . .	32

---

## Preface

This document summarizes the changes to HP C/HP-UX in version A.10.32.30. This version of HP C is available on HP-UX 10.20.

Please note that product software releases and operating system releases often occur independently of each other. In general, a product software release applies to the immediately preceding or concurrent system release and to all subsequent system releases until the software is revised.

Also, the product software release number (shown above) indicates the version level of the software product at the time that the release notes were issued. Since some product updates do not require documentation changes, it is possible that there may not be a one-to-one correspondence between the version number on your software and the version number of the release notes.

The printed copy of these *Release Notes* has additional late-breaking information over the online *Release Notes* file at `/opt/ansic/newconfig/RelNotes/ansic.10.32.30`. The printed version is also available at <http://docs.hp.com> under *Development Tools and Distributed Computing*.



# 1 New and Changed Features

Version A.10.32.30 of HP C/HP-UX includes support for the following new options:

- `+objdebug` leaves debugging information in the object files rather than copying it to the executable file.

NOTE: You must install the latest linker patch (PHSS\_17903) or its successor for this option to work. Without this patch, the option is ignored.

- `+W` suppresses warnings.
- `+We` changes warnings to errors.
- `+Ww` enables warnings that would otherwise not be enabled.
- `+Olevel=name` lowers the optimization level for the named functions.
- `+Oreusedir=directory` reduces link time by saving and reusing object files in *directory* rather than recompiling them from intermediate object files.

This chapter summarizes these new options and provides additional information on the pragmas `HP_NO_RELOCATION`, `HP_LONG_RETURN`, and `HP_DEFINED_EXTERNAL` which were introduced in an earlier version of HP C/HP-UX.

Finally, this chapter summarizes new features in the HP WDB 1.1 debugger.

---

## New or Changed Compiler Options

The following command line options are new in this release of HP C/HP-UX.

## +objdebug Option

---

**NOTE**

---

You must install the latest linker patch (PHSS\_17903) or its successor for this option to work. Without this patch, the option is ignored.

When used with `-g`, `+objdebug` leaves debug information in the object files instead of copying it to the executable file at link time, resulting in shorter link times and smaller executables. The default, `+noobjdebug`, copies the debug information to the executable file.

When you specify `-g`, the compiler places symbolic debugging information into the object files. By default, the linker calls `pxdb` which compacts this debug information & copies it to the executable file. When `+objdebug` was used at compile time, the linker leaves the debug information in the object files. To debug the executable file, the HP WDB debugger must have access to the object files. If you move the object files, use HP WDB's `objectdir` command to tell it where the object files are. (The HP DDE debugger does not support this option.) This option reduces link time and the size of the executable file by avoiding this copying of debug information.

The compile-time default is `+noobjdebug`. If the linker detects any object files that were compiled with `+objdebug`, it will leave the debug information in those files. Any object files not compiled with `+objdebug` will have their debug information copied into the executable file. You can leave debug information in some object files and not in others.

Use the `+noobjdebug` option when linking to explicitly tell the linker to copy all debug information to the executable file, even from files compiled with `+objdebug`.

For information on the HP WDB debugger, see the help file in `/opt/langtools/wdb/doc/index.html`.

## +W *n1* [, *n2*, ... *nN*] Option

This option suppresses the specified warnings, where *n1* through *nN* are valid compiler warning message numbers. See the file `/opt/ansic/lib/nls/msg/C/cc.msgs` for a partial list of compiler messages. Or use the `dumpmsg(1)` command to see compiler messages in the catalog files in `/opt/ansic/lib/nls/msg/C`.



### **+We *n1*[, *n2*, ...*nN*] Option**

This option changes the specified warnings to errors, where *n1* through *nN* are valid compiler warning messages. See the file `/opt/ansic/lib/nls/msg/C/cc.msgs` for a partial list of compiler messages. Or use the `dumpmsg(1)` command to see compiler messages in the catalog files in `/opt/ansic/lib/nls/msg/C`.

### **+Ww *n1*[, *n2*, ...*nN*] Option**

This option enables the specified warnings. Use it to enable specific warnings when all other warnings have been suppressed with `-w` or `+w3`. Or use it to enable specific warnings that would otherwise be enabled with the `+w1` option. It cannot be used to enable `+Mn` migration warnings. *n1* through *nN* are valid compiler warning messages. See the file `/opt/ansic/lib/nls/msg/C/cc.msgs` for a partial list of compiler messages. Or use the `dumpmsg(1)` command to see compiler messages in the catalog files in `/opt/ansic/lib/nls/msg/C`.

### **+O*level*=*name1*[, *name2*, ...] Option**

Optimization levels: 1, 2, 3, 4

Default: All functions are optimized at the level specified by the ordinary `+Olevel` option.

This option lowers optimization to the specified *level* for one or more named functions. *level* can be 0, 1, 2, 3, or 4. The *name* parameters are names of functions in the module being compiled. Use this option when one or more functions do not optimize well or properly. This option must be used with `-O` or an ordinary `+Olevel` option.

This option works the same as the `OPT_LEVEL` pragma described under “Optimizer Control Pragas” in the *HP C/HP-UX Programmer's Guide*. This option overrides the `OPT_LEVEL` pragma for the specified functions. As with the pragma, you can only lower the level of optimization; you cannot raise it above the level specified in the ordinary `+Olevel` or `-O` option. To avoid confusion, it is best to use either this option or the `OPT_LEVEL` pragma rather than both.

### **Examples**

The following command optimizes all functions at level 3, except for the functions `myfunc1` and `myfunc2`, which it optimizes at level 1.

## New and Changed Features

### Improving Shared Library Performance

```
$ cc +O3 +O1=myfunc1,myfunc2 funcs.c main.c
```

The following command optimizes all functions at level 2, except for the functions `myfunc1` and `myfunc2`, which it optimizes at level 0.

```
$ cc -O +O0=myfunc1,myfunc2 funcs.c main.c
```

The following command optimizes `myfunc1` at level 2, `myfunc2` at level 1, and all other functions at level 3.

```
$ cc +O3 +O1=myfunc1,myfunc2 +O2=myfunc1 funcs.c main.c
```

### **+Oreusedir=directory Option**

Optimization levels: 4 or with profile-based optimization

Default: no reuse of object files

This option specifies a directory where the linker can save object files created from intermediate object files when using `+O4` or profile-based optimization. It reduces link time by not recompiling intermediate object files when they don't need to be recompiled.

When you compile with `+I`, `+P`, or `+O4`, the compiler generates intermediate code in the object file. Otherwise, the compiler generates regular object code in the object file. When you link, the linker first compiles the intermediate object code to regular object code, then links the object code. With this option you can reduce link time on subsequent links by not recompiling intermediate object files that have already been compiled to regular object code and have not changed.

Note that when you do change a source file or command line options and recompile, a new intermediate object file will be created and compiled to regular object code in the specified directory. The previous object file in the directory will not be removed. You should periodically remove this directory or the old object files since the old object files cannot be reused and will not be automatically removed.

---

## Improving Shared Library Performance

The pragmas described here can improve performance of shared libraries by reducing the overhead of calling shared library routines. All three

pragmas should be used together, where applicable, as they depend on one another to a certain extent. You must be very careful using these pragmas because incorrect use can result in incorrect and unpredictable behavior. See the *HP-UX Linker and Libraries User's Guide* for more information on improving shared library performance.

### HP\_NO\_RELOCATION Pragma

This pragma improves performance of shared library calls by omitting floating-point parameter relocation stubs in calls to shared library functions. Put this pragma in header files of functions that take floating point parameters or return floating point data and that will be placed in shared libraries. Putting it in the header file and ensuring all calls reference the header file is one way to ensure that it is specified at the function definition and at all calls.

---

#### WARNING

**This pragma *must* be at the function definition and at *all* call sites. If the pragma is omitted from the function definition or from any call, the linker will generate parameter relocation code and the application will behave incorrectly since floating point parameters will not be in expected registers.**

---

#### Syntax

```
#pragma HP_NO_RELOCATION name1[, name2[, ...]]
```

where *name1*, *name2*, and so forth are names of functions in shared libraries.

#### Background

Parameter relocation stubs are instructions that move (relocate) floating point parameters and function return values between floating point registers and general registers. They are generated for calls to routines in shared libraries. Relocation stubs are generated when passing floating point parameters or using a floating point function return in routines in shared libraries. This pragma prevents this unnecessary relocation from being done.

---

#### NOTE

Do not use this option with functions that use the `varargs` macros. See

New and Changed Features  
Improving Shared Library Performance

the *HP C/HP-UX Reference Manual* or the *varargs(5)* man page for information on the `varargs` macros.

---

## HP\_LONG\_RETURN Pragma

This pragma improves performance of shared library calls by using a long return instruction sequence instead of an interspace branch and by omitting export stubs. An export stub is a short code segment generated by the linker for a global definition in a shared library. External calls to shared library functions go through the export stub.

Put this pragma in header files of functions that will go in shared libraries so it is specified at the function definition and at all calls. For functions with floating point parameters or returns, use the `HP_NO_RELOCATION` pragma along with this pragma.

---

### WARNING

**This pragma *must* be at the function definition and at *all* call sites. If the pragma is omitted from the function definition or from any call, the compiler will generate incompatible return code and the application will behave incorrectly.**

---

### Syntax

```
#pragma HP_LONG_RETURN name1[, name2[, ...]]
```

where *name1*, *name2*, and so forth are names of functions in shared libraries.

### Background

An export stub is generated by default for each function in a shared library. Each call to the function goes through the export stub. The export stub serves two purposes: to relocate parameters and perform an interspace return.

The `HP_LONG_RETURN` pragma generates a long return sequence in the export stub instead of an interspace branch. If you also use the `HP_NO_RELOCATION` pragma (for functions taking floating point parameters), all the code in the export stub is omitted, eliminating the export stub entirely. For functions taking non-floating-point parameters,

the `HP_LONG_RETURN` pragma by itself eliminates the need for export stubs.

---

**NOTE**

Using `HP_LONG_RETURN` without using `HP_NO_RELOCATION` with floating point parameters, could actually degrade performance by creating export stubs and relocation stubs.

These pragmas improve performance of calls to shared library functions from outside the shared library. Therefore do not use this pragma for hidden functions (see the `-h` and `+e` linker options) or for functions called only from within the same shared library linked with the `-B` symbolic linker option, otherwise this pragma may degrade performance. (See the *HP-UX Linker & Libraries User's Guide* for information on the above mentioned options.)

Do not use this pragma if you compile on PA-RISC 2.0 or later or with the `+DA2.0` option since the effect is the default. That is, if no relocations are generated, export stubs are not generated on PA-RISC 2.0 and later, and a long return instruction sequence is generated by default, so this pragma has no effect.

---

### **HP\_DEFINED\_EXTERNAL Pragma**

This pragma improves performance of shared library calls by inlining import stubs. Place this pragma at calls to shared library routines along with the `HP_NO_RELOCATION` pragma (if using floating-point parameters or return values) and the `HP_LONG_RETURN` pragma.

---

**WARNING**

**Do not use this pragma at function definitions, only at function calls. Specifying it at function definitions will result in incorrect behavior.**

**On PA-RISC 1.1, use this pragma only when calling a shared library from an executable file. Using it on calls within an executable file will cause the program to abort.**

---

#### **Syntax**

```
#pragma HP_DEFINED_EXTERNAL name1[, name2[, ...]]
```

## New and Changed Features

### New Compiler Defaults

where *name1*, *name2*, and so forth are names of functions in shared libraries.

#### Background

Import stubs are code sequences generated at calls to shared library routines. The import stub queries the PLT (Procedure Linkage Table) to determine the address of the shared library function & calls it. The `HP_DEFINED_EXTERNAL` pragma inlines this import stub.

---

#### NOTE

If your function takes floating-point parameters, you should also use the `HP_NO_RELOCATION` pragma (if floating point parameters are present). You should also use the `HP_LONG_RETURN` pragma with this pragma. If you don't, the import stub may be too large to inline.

Use this pragma only on calls to functions in shared libraries. On PA-RISC 2.0, it will degrade performance of calls to any other functions.

---

---

## New Compiler Defaults

Here are the new defaults for the HP C/HP-UX compiler:

- The `+objdebug` compile-line option is off by default. That is, `+noobjdebug` is the default.
- The `+W` option is off by default. That is, warnings are enabled.
- The `+We` option is off by default. That is, warnings are not treated as errors.
- The `+Ww` option is off by default. That is, only warnings indicating that code generation might be affected are issued.
- The `+Oreusedir` option is off by default. That is, intermediate object files are not saved between links and are recompiled to object files at every link.
- The `+Olevel=name` option is off by default. That is, without this option, all functions are compiled at the level specified by the `-O` option or the ordinary `+Olevel` option.

## HP WDB Debugger

The HP WDB 1.1 debugger is an HP-supported implementation of the GDB debugger. It supports source-level debugging of object files written in HP C, HP aC++, Fortran 90, and Fortran77 on HP-UX Release 10.20 and later.

The major changes between HP WDB 1.0 and HP WDB 1.1 include:

- HP WDB 1.1 is based on GDB version 4.17.
- An optional HP WDB GUI, which is a graphical interface designed by Hewlett-Packard for the WDB 1.1 debugger. For details about WDB GUI, refer to the HP World Wide Web site, <http://www.hp.com/go/wdb/>.
- Support for user threads on HP-UX 10.20 and 11.0.

The most recent version of HP WDB is available from the World Wide Web at the URL: <http://www.hp.com/go/wdb/>.

HP WDB 1.1 is available for any PA-RISC 1.1 or PA-RISC 2.0 HP 9000 system running HP-UX Release 10.20 or 11.0.

HP WDB 1.1 does not support the debugging of optimized code (that is, code compiled with both `-g` and `-O` options), though some commands may be available. See Section 4.1 of *Debugging with GDB* for information on what to expect if you debug optimized code.

---

### NOTE

HP WDB 1.1 is not supported on releases of the HP-UX operating system prior to 10.20.

---

---

## Summary of HP WDB 1.1 Features

HP WDB 1.1 provides the following features in addition to the standard GDB 4.17 features:

- Support for debugging 32-bit and 64-bit programs.

## New and Changed Features

### Summary of HP WDB 1.1 Features

- Support for a Terminal User Interface (TUI) in addition to the GDB line-mode interface, enabled with the `-tui` option
- Support for automatic loading of debug information from object modules when an application is compiled with the `+objdebug` option. The benefits of this option are only available if you have installed the most recent linker patch. See “+objdebug Option” on page 8.
- Support for debugging code in shared libraries and archive libraries
- Support for kernel threads, including thread local storage (HP-UX 11.0 only)
- Support for user threads on HP-UX 10.20 and 11.0
- Support for hardware watchpoints (HP-UX 11.0 only)
- Support for assembly-level debugging (PA-RISC 1.x and PA-RISC 2.0)
- Support for a subset of XDB commands, enabled with the `-xdb` option

HP WDB 1.1 contains the following new GDB commands and subcommands:

`backtrace-other-thread`

Print backtrace of all stack frames for a thread with stack pointer SP and program counter PC. This command is useful in cases where the debugger does not support a user thread package fully.

`tcatch event`

Set a catchpoint that is enabled only for one stop. The catchpoint is automatically deleted after the first time the event is caught.

`set overload-resolution {on | off}`

Enable or disable overload resolution for C++ expression evaluation. The default is `on`.

`show overload-resolution`

Display current overload resolution setting for C++ expression evaluation.

`set opaque-type-resolution {on | off}`

Enable or disable resolution of opaque struct, class, or union types. The default is `on`.



`show opaque-type-resolution`

Display current opaque type resolution setting.

`set auto-solib-add threshold`

Set the autoloading size threshold (in megabytes) of shared library symbols.

`show auto-solib-add`

Display the autoloading size threshold of shared library symbols. `MAXDSIZE` defines the default threshold size.

`set threadverbose {on | off}`

Enables and disables the output of informational messages regarding thread creation. The default setting is on. You can set it to off to stop the display of messages.

`show threadverbose`

Displays whether set `threadverbose` is on or off.

`objectdir path`

Specify a colon (:) separated list of directories in which HP WDB searches for object files. These directories are added to the front of the already existing `objectdir` path. If a directory already existing in the `objectdir` path is specified, it is moved up in the `objectdir` path so that it is searched earlier.

HP WDB recognizes two special directory names: `$cdir`, which refers to the compilation directory (if available) and `$cwd` which tracks the HP WDB current working directory.

`objectload file.c`

Causes HP WDB to load the debug information for `file.c` immediately. The default is to load debug information from object modules on demand.

New and Changed Features  
**Summary of HP WDB 1.1 Features**

`objectretry file.c`

Forces HP WDB to retry loading an object file if HP WDB encounters a file error while reading an object module. File errors that might cause this include incorrect permissions, file not found, or if the `objectdir` path changes. By default, HP WDB does not try to read an object file after an error.

For more information about these commands, see *Debugging with GDB* or use the GDB `help` command.

## 2 Installation Information

This chapter contains installation information for HP C/ANSI C Developer's Bundle, HP WDB 1.1 and HP WDB GUI 1.1. HP WDB is bundled with the HP C/HP-UX, HP ANSI C++, HP Fortran 90 and HP Fortran77 compilers.

Read this entire document, and any other Release Notes or READMEs you may have, before you begin an installation.

---

### HP C/ANSI Developer's Bundle Components

The HP C/ANSI C Developer's Bundle for HP-UX contains the following packages:

C-ANSI-C	HP C/ANSI C compiler (8.9 Mb)
C-Analysis-Tools	C language analysis tools (337 Kb) Includes <i>cb(1)</i> , <i>cflow(1)</i> , <i>cxref(1)</i> , <i>lint(1)</i> , and <i>protogen(1)</i> .
C-Dev-Tools	C language development tools (1.1 Mb) Includes <i>cpp(1)</i> , <i>lex(1)</i> , and <i>yacc(1)</i>
WDB	HP WDB debugger (4.5 Mb)
WDB-GUI	HP WDB debugger graphical user interface (9 Mb)
DDE	Distributed Debugging Environment (24 Mb)
XDB	HP XDB debugger (1.1 Mb)
DebugPrg	Debugging support tools (150 Kb)
HPPAK	HP Programmer's Analysis Kit (4.2 Mb)
BLINKLINK	Incremental linking tool (1.2 Mb)
Auxiliary-Opt	Auxiliary optimizer for HP languages (7.8 Mb) Includes the <i>HP-UX Linker and Libraries Online User Guide</i> .
X11MotifDevKit	X11 and Motif developer's kit (22.7 Mb)

Installation Information  
**HP WDB Components**

VUEHelpDevKit HP Vue online help developer's kit (2.7 Mb)  
CDEDevKit CDE developer's kit (8.2 Mb)  
AudioDevKit Audio subsystem developer's kit (512 Kb)  
ImagingDevKit X1 bit-mapped imaging developer's kit (2.2 Mb)  
TechPrintSvcDK Technical Print Service Development Kit (690 Kb)  
GraphicsPEX5DK PEX 5.1 version 3 graphics developer's kit (Series 700 only) (11.8 Mb)  
GraphicsSBaseDK Starbase 3D graphics developer's kit (Series 700 only) (14.9 Mb)  
OpenGLDevKit OpenGL 3D graphics developer's kit (5.8 Mb)  
OpenGLDebugger OpenGL 3D graphics debugger (2.2 Mb)  
DigitalVideoDK Video subsystem developer's kit (6.5 Mb)

### **Disk Space Requirements for HP C**

If you install all the packages of the HP C/ANSI Developer's Bundle, they occupy approximately 126 megabytes of disk space.

---

## **HP WDB Components**

The HP WDB debugger consists of the following swinstall products:

HP WDB 1.1 HP WDB Debugger  
HP WDB-GUI Optional graphical user interface component for the WDB Debugger

If you install WDB-GUI on a system where HP WDB 1.1 has not yet been installed, HP WDB 1.1 will be installed automatically.

### **Disk Space Requirements for HP WDB**

The approximate disk space required for HP WDB 1.1 is 4.5 megabytes.  
The HP WDB GUI 1.1 requires 9 megabytes of disk space.

## Filesets

HP WDB is the product name for the HP WDB 1.1 debugger.

The HP WDB product contains the subproducts:

- Runtime contains one fileset: WDB
- Documentation contains one fileset: WDB-DOC
- Manuals contains one fileset: WDB-MAN

---

### NOTE

DEBUG-PRG is a co-requisite fileset for HP WDB and is automatically selected during installation.

---

The WDB-GUI product contains the subproducts:

- Runtime contains one fileset: WDB-GUI-RUN
- Help contains one fileset: WDB-GUI-HELP
- Manuals contain one fileset: WDB-GUI-MAN

HP WDB-GUI requires the WDB product. If WDB is not already installed, it will be automatically selected during installation.

## Supported Configurations

The following configurations are supported by HP WDB:

- Installing and running locally from an HP-UX 10.20 or 11.x system.
- Installing on an HP-UX 10.20 or 11.x system, with the display redirected to a remote HP-UX 9.x, 10.x, or 11.x node.

---

## Installation Instructions

To install your software, run the SD-UX `swinstall` command. It invokes a user interface that leads you through the installation. It also gives you information about disk space requirements, version numbers, product descriptions, and dependencies.

Installation Information  
Installation Instructions

For more information about installation procedures and related issues, refer to *Managing HP-UX Software with SD-UX* and other README, installation, and upgrade documentation provided or described in your HP-UX operating system package. See the *swinstall(1M)* and *sd(4)* man pages. Most of this information is also available on the web at <http://docs.hp.com>.

---

**NOTE** After installing the HP C/ANSI C Developer's Bundle, install the latest linker patch (PHSS\_17903) or its successor. This patch is required by the `+objdebug` option. Without this patch, the option is ignored.

---

---

**NOTE** Most files related to the HP C compiler are installed in the directories `/opt/ansic` and `/opt/langtools`. The installation scripts add `/opt/ansic/bin` and `/opt/langtools/bin` to the login file `/etc/PATH`. They also add `/opt/ansic/share/man/%L:/opt/ansic/share/man` and `/opt/langtools/share/man/%L:/opt/langtools/share/man` to the login file `/etc/MANPATH`. (%L is replaced by the value of the `LC_MESSAGES` environment variable when the `man` command is executed. It determines the language used for manpage searches. If `LC_MESSAGES` is not set, %L defaults to null. See *environ(5)*.)

---

---

**NOTE** The HP C/ANSI C compiler installation package provides the capability to create and remove transition links from HP-UX release 9.x locations to HP-UX release 10.x locations. The HP C/ANSI C product installs the ISU transition link table specification files on the system.

The Software Distribution update tool `tlinstall` uses these files to install transition links from previous HP-UX file and directory names to the corresponding HP-UX 10.x file and directory names. To remove these transition links, use the update tool `tlremove`. For more detail, read the update tools manpages. These tools are installed in `/opt/upgrade/bin`.

---

## 3 Documentation Overview

This chapter summarizes the C compiler documentation and various other documents that may be of interest to the C programmer. Some of this documentation is online, and some may be ordered in printed versions, as described below. Most of this documentation is available on the web site <http://docs.hp.com>. For the latest information, visit the HP C/HP-UX web page at <http://www.hp.com/go/hpc>.

The HP C documentation consists of:

- *HP C/HP-UX Release Notes* (this document)
- *HP C/HP-UX Programmer's Guide*
- *HP C/HP-UX Reference Manual*
- *HP-UX Floating Point Guide*
- *HP C/HP-UX Online Help*

The HP WDB documentation consists of:

- *Getting Started with HP WDB*
- *Debugging with GDB*
- *GDB Quick Reference*
- *Using the HP WDB Terminal User Interface*
- *XDB to WDB Transition Guide*

The following sections summarize these and other, related documents.

---

### C Compiler Documentation

#### HP C/HP-UX Release Notes

The *HP C/HP-UX Release Notes*, which you are reading now, provide release-specific information such as new feature summaries, installation

instructions, and known defects. In addition, they contain this documentation overview to help you orient yourself regarding available documentation. The release notes are also available online in the text file `/opt/ansic/newconfig/RelNotes/ansic.10.32.30`.

## Printed Documentation

You can order printed versions of a wide variety of Hewlett-Packard documents. Use the `man manuals` command for details on available documents and ordering information. See also the HP documentation web site <http://docs.hp.com> and the HP C/HP-UX web site at <http://www.hp.com/go/C>. Listed below are some of the documents most closely related to the C compiler.

- *HP C/HP-UX Reference Manual* (92453-90024 for 10.20 and 92453-90087 for 11.0)

Provides reference material for HP C as implemented on HP 9000 systems. This document is based on the ANSI C standard 9899-1990, and documents implementations and extensions unique to HP C on HP-UX. It does not replicate the ANSI C standard and you are referred to the standard for any fine points not covered.

- *HP C/HP-UX Programmer's Guide* (92434-90002 for 10.20 and 92434-900013 for 11.0)

Contains a detailed discussion about selected C topics for HP 9000 systems. Included are a discussion of data type sizes and alignment modes, some comparisons between HP C and other languages, and information on 64-bit programming, optimization, threads, and parallel processing.

- *HP-UX Floating-Point Guide* (B3906-90004 for 10.20 and B3906-90006 for 11.0)

Describes how floating-point arithmetic is implemented on HP 9000 systems and discusses how floating-point behavior affects the programmer. Provides information about the C and Fortran math libraries.

## HP C Online Help

The C compiler online help includes examples, tutorial information, error message discussions, and reference information.



To access the online help, on a system where the HP C compiler is installed, enter the following:

```
cc +help
```

or

```
/usr/dt/bin/dthelpview -h c
```

and the online help window appears on your display. (Note that this requires that your `$DISPLAY` variable is set correctly. If you do not know how to do this, contact your system administrator.)

You can also click on the ? icon on the HP CDE front panel on a system where HP C is installed and select HP C Online Reference.

If you do not have a graphics display or prefer to work with a character-based help interface, enter the command:

```
charhelp
```

and you will get a usage description for this command:

```
charhelp: Usage: charhelp {cc|CC|aCC|f77|ld|dde| -helpVolume  
file}
```

For help with the HP C compiler, for example, enter:

```
charhelp cc
```

and follow the menus for further direction. The `charhelp` command is in the directory `/opt/langtools/bin`. For more information, see the man page for `charhelp(1)` (`/opt/langtools/share/man` must be in your `$MANPATH` environment variable).

## Mailing List with Product News

You can subscribe to a mailing list for announcements, discussion, questions and answers about HP C and other HP-UX software development tools. Send an electronic mail message to:

```
majordomo@cxx.cup.hp.com
```

To add yourself to the list, in the body of the message type the following:

```
subscribe hpux-devtools
```

To remove yourself from the list, in the body of the message type the following: `unsubscribe hpux-devtools`

For more information, see the HP C/HP-UX web page at

<http://www.hp.com/go/hpc>.

---

## HP WDB Documentation

All of the HP WDB documentation is available online in the following directory:

`/opt/langtools/wdb/doc`

For the latest information and updates, visit the HP WDB web page, <http://www.hp.com/go/wdb/>

You can print the following PostScript documents, which constitute the official documentation for GDB version 4.17 (on which HP WDB 1.1 is based):

- *Debugging with GDB*, `/opt/langtools/wdb/doc/gdb.ps`
- *GDB Quick Reference*, `/opt/langtools/wdb/doc/refcard.ps`

The PDF version of *Debugging with GDB* is available at `/opt/langtools/wdb/doc/gdb.pdf`, which you can read with the Acrobat Reader, `acroread`. If you do not already have `acroread`, it is available for free from Adobe Systems Inc. (<http://www.adobe.com>).

In addition, all of the HP WDB documentation is available through your Web browser from `/opt/langtools/wdb/doc/index.html`.

In addition to the standard GDB documentation, HP provides the following HTML documents:

- *Getting Started with HP WDB*,  
`/opt/langtools/wdb/doc/GDBtutorial.html` (if your browser does not support frames, view `/opt/langtools/wdb/doc/tutmain.html`)
- *XDB to WDB Transition Guide*,  
`/opt/langtools/wdb/doc/xdb_trans.html`
- *Using the HP WDB Terminal User Interface*,  
`/opt/langtools/wdb/doc/TUI.html`

Use EMACS to view the following info files, an online version of *Debugging with GDB*:

*Debugging with GDB*, /opt/langtools/wdb/doc/gdb.info\*

(Copy the files to your usual “info” directory first.)

The man page is *gdb(1)*.

## Mailing List for Product Announcements

To receive an electronic mail message when HP releases a new version of HP WDB, join the HP WDB product news mailing list.

Send an electronic mail message to:

majordomo@cxx.cup.hp.com

To add yourself to the list, in the body of the message type the following:

subscribe wdb-announce

To remove yourself from the list, in the body of the message type the following: unsubscribe wdb-announce

For more information, see the HP WDB web page at <http://www.hp.com/go/wdb>.

---

## Related Documentation

This documentation is available on the HP-UX 10.20 Instant Information CD-ROM and on the web site <http://docs.hp.com>.

- *HP-UX Linker and Libraries Online User Guide* (B2355-90653 on 10.20 and B2355-90655 on 11.0)

Replaces the manual *Programming on HP-UX*. To access the *HP Linker and Libraries Online User Guide*, use the `ld +help` command.

- *HP PA-RISC Compiler Optimization Technology White Paper*

Describes the benefits of using optimization. This white paper is available online in the PostScript file `/opt/langtools/newconfig/white_papers/optimize.ps`.

- *HP Assembler Reference Manual* (92432-90009 on 10.20 and 92432-90012 on 11.0)

## Documentation Overview

### Related Documentation

Describes the use of the Precision Architecture RISC (PA-RISC) Assembler on HP 9000 computers. Describes PA-RISC Assembler directives, pseudo-operations, and how to run the Assembler on HP-UX.

- *HP-UX Reference Manual*

The reference manual pages, or man pages, are available online (use the command `man man` for more information), and are also available on the CD-ROM.

- HP-UX Software Transition Kit (STK)

Enables the application developer to easily transition software from HP-UX 10.x to either the 32-bit or the 64-bit version of HP-UX 11.0. The kit is available free of charge on the HP-UX 11.0 Application Release CD-ROM, or from the web at the following URL:

<http://www.software.hp.com/STK/index.html>

## 4 Problem Descriptions and Fixes

This chapter tells you where to look for information on known problems and fixes for HP C/HP-UX and HP WDB. It also describes the known problems and limitations.

---

### Getting Information about Problems and Fixes

For a list of problems and their fixes, see the Technical Knowledge Database on the HP Electronic Support Center web site at:

<http://us-support.external.hp.com/>

<http://europe-support.external.hp.com/>

The Technical Knowledge Database is available to customers with support contracts.

You can find reports listing Service Requests by using the following product numbers in the **Search String** field:

- WDBDEBUGGERA for HP WDB.
- 92453-01 for entries relating to C and the programming tools.
- B3910 for entries relating to HP aC++ and the programming tools.
- HPCPLUSPLUS for entries relating to HP C++ (cfront) and the programming tools.

Use the following steps to look up information about problems and their fixes.

- Step 1.** Once at the web site, enter your User ID and Password, then click **Okay**.
- Step 2.** On the next page, click **Technical Knowledge Database**.
- Step 3.** Verify that your **Search Options** include HPUX under **System Type** and Service Request under **Document Type**.

- Step 4.** Enter the product number in the **Search String** field, then click **Search** to show a list of problems and any fixes.

---

## HP C Problems and Limitations

Note since HP-UX 10.10 is the last supported operating system for PA-RISC 1.0 architecture systems, the 10.20 compilers no longer support the compiling of code for PA-RISC 1.0.

### Required Patch

After installing the HP C/ANSI C Developer's Bundle, install the latest linker patch (PHSS\_17903) or its successor. This patch is required by the `+objdebug` option. Without this patch, the option is ignored.

### Nonterminating Parallel Processes

A program compiled with `+Oparallel` for parallel execution uses shared memory segments and additional processes when it executes. Except in rare circumstances, the program will terminate the processes and free the shared memory whether it completes normally or terminates because of a signal.

However, if enough parallel programs do not terminate their additional processes over a period of time, you may get messages indicating that the operating system refuses to start another process when you attempt to run any program. `fork()` returns the error value [EAGAIN] (resource temporarily unavailable), which in this case means that the system's limits on the number of system-wide or individual-user processes would be exceeded.

If this should happen, you can use the `ps -f` command to search for the processes that didn't terminate and the `kill` command to terminate them (see `ps(1)` and `kill(1)`). Note that the `ps -f` command displays the same name for the processes as it does for their associated main programs.

Likewise, if enough parallel programs happen not to free their shared memory segments over a period of time, programs may start failing because of insufficient memory, either shared or normal. Terminating the

additional processes will almost certainly free the shared memory segments as well. If not, you can use the `ipcs -m` command to search for shared memory segments and the `ipcrm` command to remove them (see *ipcs(1)* and *ipcrm(1)*). Note that the `ipcs -m` command lists the same owner and group for the shared memory as the associated parallel program had and that most systems normally have active shared memory segments owned by `root` and `daemon`.

### Memory Allocation Routine `alloca()`

The implementation of this `alloca()` routine can be very system dependent and its use is not encouraged.

`alloca()` is a memory allocation routine similar to `malloc()` (see *malloc(3C)*). The syntax is:

```
void *alloca(size_t size);
```

`alloca()` allocates space from the stack of the caller for a block of at least `size` bytes, but does not initialize the space. The space is automatically freed when the calling routine exits. This routine is implemented as an inlined function by the HP C compiler product.

To use this function, you may use the `<alloca.h>` header file or you may specify your own prototype and use the `+Olibcalls` option.

---

## HP WDB Problems and Limitations

### HP WDB Limitation on Window Resizing with TUI

We recommend that you not resize an `hpterm` or `xterm` terminal window while using the Terminal User Interface (TUI). If you do, the following problems occur:

- The command window contents and (gdb) prompt are erased. Issue a `refresh (ref)` command and then press the Return key to recover the prompt.
- It is no longer possible to scroll in windows by using the Page Up, Page Down, and arrow keys. You must use the `+`, `-`, `>`, and `<` commands instead.

## HP WDB Limitation with Processes Started across NFS Mounts

When HP WDB attaches to a running program you might get this message:

```
(gdb) attach 12852
Attaching to process 12852
```

```
Attaching to process 12852 failed. Check whether
program is on an NFS-mounted file-system. If so,
you will need to mount the file system with the
"nointr" option with mount(1) or make a local copy
of the program to resolve this problem.
```

The most likely cause for this message is that you have attached to a process that was started across an NFS mount.

The HP-UX kernel has a restriction that prevents a debugger from attaching to a process started from an NFS mount, unless the mount was made non-interruptible with the `-nointr` flag, see `mount(1)`.

## Terminal Window Sometimes Fails to Start from WDB GUI on 10.20

The WDB GUI uses a terminal window to display the target program's output and to allow user input to the target. The terminal program starts when you choose the `View:Program Console` menu command or when the target program generates output. Some versions of `dtterm` on 10.20 cannot be started from the WDB GUI. If this happens you will see the error:

```
Unable to start terminal
```

This is caused by a problem in `dtterm`, which is fixed in patch PHSS\_17566.

If you don't have the patch available, you can work around the problem by setting the environment variable `TERM` to `hpterm` and restarting the WDB GUI.

You can also work around the problem by starting the target program from a shell window and then attaching to the program from the WDB GUI.