HP WDB 5.5.2 Release Notes

HP 9000 and HP Integrity Systems

Legal Notices

1 HP WDB 5.5.2 Release Notes

HP Wildebeest Debugger (WDB) 5.5.2 is an HP-supported implementation of the Open Source Debugger GDB. It supports source-level debugging of programs written in HP C, HP aC++, and Fortran 90 on Itanium®-based systems running HP-UX 11i v2 or HP-UX 11i v3, and PA-RISC systems running HP-UX 11.0, HP-UX 11i v1, HP-UX 11i v2, and HP-UX 11i v3. This version of WDB supports programs compiled for Integrity systems (Itanium®) and HP 9000 (PA-RISC) systems.

The latest version of HP WDB is available at:

http://www.hp.com/go/wdb

This document discusses the following topics:

- "What is New in This Release"
- "Patches, Defects, and Fixes"
- "Known Problems and Workarounds"
- "Compatibility Information and Installation Requirements"
- "Features Supported in Previous Versions of HP WDB"
- "Related Documentation"
- "WDB Mailing Lists"

What is New in This Release

Following features are introduced in this version of HP WDB

- **Enhanced thread debugging support**

  The thread debugging enhancement is meant for developers of POSIX multithreaded applications and provides extended information on the state of following `pthread` primitives:

  — `threads`

  — `mutexes`

  — `read-write locks`

  — `condition variables`

  The following new commands are supported:

  — `set thread-check on/off`

    The `set thread-check on/off` command can be used to enable or disable extended thread debugging functionality. The `set thread-check on/off` command must be run prior to debugging the application under HP WDB, to force the underlying runtime system to start collection of information on `pthread` primitives. The `set thread check` feature is `off` by default.

— `info thread [thread-id]`

The `info thread [thread-id]` command displays a list of known threads. If a `thread-id` is provided, the command displays extended information on the thread.

— `info mutex [mutex-id]`

The info mutex [mutex-id] command displays a list of known mutexes. If a mutex-id is provided, the command displays extended information on the mutex.

— `info condvar [condvar-id]`

The `info condvar [condvar-id]` command displays a list of known condition variables. If `condvar-id` is provided, the command displays extended information on the condition variables.

— `info rwlock [rwlock-id]`

The `info rwlock [rwlock-id]` command displays a list of known read-write lock. If `rwlock-id` is provided, the command displays extended information on the read-write lock.

NOTE

— The commands, `info mutex`, `info condvar`, `info thread`, and `info rwlock` work only if thread-check is set to `on`.

— The enhanced thread functionality is supported only on HP-UX 11i v2 and later on both PA-RISC and Integrity systems.

— For PA-RISC 32-bit applications, the enhanced thread functionality does not work if the application is built using default linker options.

To enable this functionality, you must enable the dynamic library lookup through the `LD_LIBRARY_PATH` environment variable using the following command on the HP-UX prompt:

```
# chatr +s enable   <PA32-bitApp>
```

• **Support for memory checking analysis for user defined memory management routines**

Many user applications have their own memory management routines. These custom allocator routines are either user defined or sometimes wrappers to the default memory management routines. Memory checking features has been extended for user defined memory management routines. Memory leak, profile of heap memory or memory errors can be obtained for user defined memory routines.

Limitations of the memory checking analysis feature:

— This feature doesnot work in batch mode.

— This feature routes to default memory routines instead of calling user defined memory routines to obtain memory analysis.

• **Enhanced heap checking**

The high water mark records the number of times the `brk()` value changes. The following commands are supported:

— `info heap high-mem`: Displays the number of times `brk()` value changes for a given run.

— `info heap high-mem X_number`: Stops when `brk()` value has moved `X_number` of times.

Limitation

— High-mem commands are not supported in batch mode.

— This feature assumes that an application has a deterministic memory allocation pattern from one run to another.

- **New commands info heap process and info heap arena**

  The following new commands are supported:

  — info heap process: This command enables the user to view a high level memory usage report of a process.

  — info heap arena: This command enables the user to view high level memory usage details of each arena.

  Limitation

  — The `info heap arena` command is not supported in batch mode.

  — The `info heap process` and `info heap arena` are available only on HP-UX 11i v3.

- **New command line options**

  — `min-heap-size`

    The `min-heap size` option, sets the minimum block size to use, for heap reporting.

    Example:

    ```
    set heap-check min-heap-size 100
    ```

    When the option min-heap-size is set to 100 info heap command only reports, blocks of size larger than or equal to 100 bytes.

  — `set display-full-path`

    The option set display-full-path when set on, displays the full pathname of the source file name while printing the frame info.

  — `set dereference [on|off]`

    The option `set dereference [on |off]` when off, GDB does not dereference `char *` variables by default.

    Limitation:

    This feature does not work in batch mode.

NOTE The Run Time Checking feature (Interactive and Batch Mode) of WDB cannot be used with applications that re-define or over-ride the default system-supplied versions of the following library routines(under libc.so and libdld.so):

Table 1-1 Dependent library routines for Run Time Checking using WDB

abort()	atoi()	chdir()	clock_gettime()
creat()	ctime()	uwx_register_c allbacks()	strstr()
dlhook()	U_STACK_TRACE()	uwx_get_reg()	write()
execl()	exit()	fclose()	fopen()

Table 1-1   **Dependent library routines for Run Time Checking using WDB**

fprintf()	fscanf()	getcwd()	getenv()
getpid()	lseek()	memchr()	open()
printf()	rand()	pthread_self()	putenv()
shl_findsym()	shl_get_r()	uwx_self_looku pip()	strrchr()
sprintf()	srand()	sscanf()	strcasecmp()
strdup()	strlen()	strchr()	strtok_r()
time()	unlink()	uwx_self_copyi n()	strcat()
uwx_step()	uwx_init()	perror()	shmctl()
write()	strcmp()	shl_get()	close()
dlgetname()	environ()	fork()	getpagesize()
uwx_self_init _context()	pthread_getsched param()	uwx_self_init_ info()	uwx_register_all oc_cb()

The runtime checking (dynamic memory,libraries, and pthreads checking) in WDB relies on the semantic and standard behavior of these library routines. Run Time Checking in WDB results in unexpected and unpredictable behavior when used with applications that substitute or re-define these library routines.

Before enabling the Run Time Checking feature in WDB, use the nm(1) command to determine if your application or the dependent libraries in your application re-define or substitute these vital library routines.

Patches, Defects, and Fixes

This chapter describes the defects fixed in this version of HP WDB.

- When set follow-fork-mode is set to serial, the debugger may show spurious SIGTRAP signals after re-attaching to the parent.

- During runtime heap-checking, mmap() calls fail on PA-RISC 32-bit systems for WDB 5.4.

- On stepping out of a constructor, GDB displays the following error message: Error accessing memory and exits.

- When debugging the compiler, a data member can be accessed at one point but not at a later point.

- HP-UX extension signals not implemented.

- Casting to function pointer type fails in GDB.

- Wrong source file and line number displayed in backtraces.

- Implemented `longjmp` debugging on PA-RISC and Integrity systems.

- Unable to call base class function on a derived class object.

- GDB should print the actual path of the source file that was picked-up using the `dir` command when printing the frame information.

- Incorrect breakpoint location for assembly language functions.

- Help text not updated for the info symbol command.

- WDB places global classes in local scope.

- Command line completion does not work in some scenarious.

- Native PA-RISC registers cannot be referenced by their architected name.

- PA-RISC 32-bit GDB core dumps while running in a core-file debugging session.

- GDB must restore the `arg` registers across frames.

- GDB prints floating value incorrectly when output radix is 8 or 16.

- Fortran failures detected on PA-RISC system when inline debugging is switched on.

- On PA-RISC systems GDB core dumps on heap corruption.

- The `pxdb` command does not run with `q4p` enabled.

- The `info func` shows the disassembly of the millicode function, but does not list the name.

- The `packcore` command fails if the recent `libc` patch is installed

- Unexpected warning, for `signals.exp` on PA-RISC 64-bit systems for `+objdebug` option.

- The `add-symbol-file` cannot load shared library or object files individually.

- Output files are generated during batch RTC in the wrong format.

- WDB failure due to incorrect function arguments.

- User friendly warning message to be issued when user tries to put breakpoint on routines in shared libraries.

- GDB does not step into a signal handler.

- Issue a meaningful message to user when `info heap` command is issued after the program exits.

- Message must be displayed for `RTC_NO_ABORT` before crashing due to corrupted memory.

- GDB does not accept command-line calls to overloaded function with incorrect argument type.

- PA GDB crashes on command-line calls

- `pxdb64`: warning: corrupted PD table

- More meaningful warning message required for `pxdb` on shared libraries.

- No demangled name associated with `msymbol`

- High water mark does not work on PA Status.

- Change `infrtc.c` to exit gracefully if detect 3rd party `malloc`.

- GDB `ENOMEM` without useful directives to user.

- 11.31 PA64 `info heap arenas` is incorrect.

Known Problems and Workarounds

This section describes known problems and suggested workarounds in this release of HP WDB.

• HP WDB 5.5.2 has no reported known problems.

Compatibility Information and Installation Requirements

This section contains installation information for HP WDB.

The HP WDB consists of the following `swinstall` products:

Table 1-2 **HP WDB Products**

Product Name	Description
HP WDB	HP implementation of the open source debugger GDB
HP WDB-GUI	Optional graphical user interface component for the WDB Debugger

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

Compatibility

HP WDB is not supported on releases of the HP-UX operating system prior to 11i. See the HP WDB web page, `http://www.hp.com/go/wdb`, for a version of HP WDB compatible with Itanium-based systems.

Supported Configurations

The following configurations are supported by HP WDB:

• Installing and running locally on an HP-UX 11i v1, HP-UX 11i v2 or HP-UX 11i v3 system

• Installing and running locally on an HP-UX 11i v1, HP-UX 11i v2 or HP-UX 11i v3 system with the display redirected to a remote HP-UX 11.x node

Filesets

HP WDB is the product name for the HP WDB debugger. The HP WDB product contains the following components:

• WDB: Runtime contains one fileset

• WDB-DOC: Documentation contains one fileset

• WDB-MAN: Manuals contains one fileset

NOTE	On HP-UX 11.0 and later, `DEBUG-PRG` is a co-requisite fileset for HP WDB and is automatically selected during installation.

The WDB GUI product contains the following sub-products:

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

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

Disk Space Requirements

The approximate disk space required for HP WDB is 20 MB.

The HP WDB GUI requires an additional 8 MB of disk space.

Installation Instructions

To install HP WDB, 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.

Specific installation instructions are available on the WDB download page at:

`http://www.hp.com/go/wdb/.`

For more information on 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 11.x operating system package.

NOTE	If you install a compiler product that includes a version of HP WDB earlier than this release, swinstall generates the following message:
	
	`ERROR: A later revision (one with a higher revision number) of fileset`
	`"WDB-GUI.WDB-GUI-HELP,r=B.11.31" has already been installed.`
	
	`Either remove this fileset or change the "allow_downdate"`

If you want to keep the later version, you can ignore the message. The new products will be installed, and the latest version of HP WDB will remain unchanged.

Debugging Core Files from a Different System

Debugging a core file on system other than the one on which it was originally produced is supported under the following conditions:

The correct system and user shared libraries are copied with the executable and core file to the other system, and the location of the shared libraries is defined by setting `GDB_SHLIB_PATH` or `GDB_SHLIB_ROOT` before debugging the core file.

For details about these variables, refer to Debugging with GDB , Section 12.2.

Table 1-3   Supported Systems for Debugging Core Files

Core files produced by	Can be debugged on
32 bit executables	Any PA-RISC 1.1 or PA-RISC 2.0 system with an HP-UX version greater than or equal to the HP-UX version on the machine where the core file was produced.
64 bit executables	Other PA-RISC 2.0 systems with HP-UX versions greater than or equal to the HP-UX version on the machine where the core file was produced.

Features Supported in Previous Versions of HP WDB

For information on previous HP WDB releases see:

`http://www.hp.com/go/wdb`

Related Documentation

HP WDB documentation is available at the following location: `/opt/langtools/wdb/doc`

Table 1-4 lists the documents available for HP WDB:

Table 1-4   HP WDB Documentation

Document	Format	Location

Debugging with GDB	PDF	`/opt/langtools/wdb/doc/gdb.pdf`
		 Emacs:
		 `/opt/langtools/wdb/doc/gdb.info`
		 (Copy the files to your info directory first.)

GDB Quick Reference Card	PDF	`/opt/langtools/wdb/doc`
Getting Started with HP WDB	HTML	`/opt/langtools/wdb/doc/html/wdb/C/GDBtutorial.html`
XDB to WDB Transition Guide	HTML	`/opt/langtools/wdb/doc/index.html`
Using the HP WDB Terminal User Interface		`/opt/langtools/wdb/doc/index.html`

Table 1-4   **HP WDB Documentation (Continued)**

Document	Format	Location
GDB manpage	*gdb(1)*	

WDB Mailing Lists

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

Send an electronic mail message to `majordomo@cxx.cup.hp.com`

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

```
subscribe wdb-announce
```

To remove yourself from the list, type the following in the body of the message:

```
unsubscribe wdb-announce
```