HP WDB 5.8 Release Notes

HP-UX 11i v1, HP-UX 11i v2, and HP-UX 11i v3

HP Part Number: 5992-0593
Published: February 2008

Legal Notices

Table of Contents

List of Tables

HP WDB 5.8 Release Notes

Announcement

HP Wildebeest Debugger (WDB) 5.8 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 HP 9000 systems running HP-UX 11i v1, HP-UX 11i v2, or HP-UX 11i v3.

This document discusses the most recent product information for HP WDB 5.8

For the latest version of HP WDB, see the HP WDB Technical Resources website at:

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

What Is New in This Version

This section describes the new features that are introduced in this version of HP WDB.

- "Supports Batch mode of thread debugging"
- "Detects various kinds of memory corruptions"
- "Displays run time type information" (page 8)

Supports Batch mode of thread debugging

HP-WDB supports batch mode of debugging threads for HP-UX 11iv2 and later, on Integrity systems and on HP-UX 11i v3 in PA-RISC systems for 64 bit applications. The debugger provides a log file with the list of thread-related errors that occur in the application.

HP WDB 5.8 detects the following thread-related conditions in batch mode:

1. The thread attempts to acquire a non-recursive mutex that it currently holds.
2. The thread attempts to unlock a mutex or a read-write lock that it has not acquired.
3. The thread waits (blocked) on a mutex or read-write lock that is held by a thread with a different scheduling policy.
4. Different threads non-concurrently wait on the same condition variable, but with different associated mutexes.
5. The threads terminate execution without unlocking the associated mutexes or read-write locks.
6. The thread waits on a condition variable for which the associated mutex is not locked.
7. The thread terminates execution and the resources associated with the terminated thread continue to exist in the application because the thread has not been joined or detached.
8. The thread uses more than the specified percentage of the stack allocated to the thread.
9. The number of threads waiting on any pthread object exceeds the specified threshold number.

The debugger reports information specific to the pthread objects and the stack frame of the executing thread at the point of error.

Detects various kinds of memory corruptions

HP WDB 5.8 enables you to detect various kinds of memory corruption at runtime without the need to have access to the application source codes.

The various features introduced are as follows:

- Detect dangling pointers and dangling blocks
- Detect in-block corruption of freed blocks
- Specify the amount of guard bytes for every block of allocated memory

The following is the update to the existing command:

```
info corruption
```

Lists the potential in-block corruptions in all the freed blocks.

The following are the new commands related to detection of memory corruption:

1. The following command enables you to detect the dangling pointers and blocks.

   ```
   info dangling
   ```

   Displays a list of all the dangling pointers and dangling blocks that are potential sources of memory corruption (may have false positives)

2. The following are the commands that enable you to change the guard byte size. These take effect only when `bounds check` is set on (`set heap-check bounds on`).

   - `set heap-check header-size<no of bytes>`

     Sets the *Header* guard for each block of the allocated memory. The default number of bytes for the footer is 16 bytes if this option is not used. If the user specifies a value less than 16 for the number of bytes, the debugger ignores it and takes the default of 16 bytes. If the user specifies more than 16 bytes, then the debugger considers the largest closest 16 byte integral of the user-specified value.

     Example:

     If the user specifies 60 bytes, the debugger takes it as 48 bytes. If the user specifies 65, the debugger considers 64 bytes.

   - `set heap-check footer-size<no of bytes>`

     Sets the *Footer* guard for each block of the allocated memory. The default number of bytes for the footer is one byte if this option is not used.

Displays run time type information

HP WDB enables you to view the run time type information for C++ polymorphic objects. The following command displays the run time information for C++ polymorphic object.

```
info rtti <address>
```

The input to this command is `address` of the C++ polymorphic object. GDB displays de-mangled type name as output.

NOTE: This command is supported only on Integrity systems.

Patches and Defect Fixes in HP WDB 5.8

The following defects are fixed in HP WDB 5.8:

- GDB depends on `objdebug_type` section alone to decide if the executable is `DOOM` executable.
- GDB cannot print global variables on `chatr +dbg enable` on a shared library.
- GDB is unable to resolve break points placed using line number in the debugging of multi-threaded applications.
- GDB does not access the environment variable LIBRTC_SERVER, if the variable is set within the gdb prompt.
- GDB does not allow info heap or info corruption when another thread locks the mutex.
- GDB is unable to access `base class` vars with multiple inheritance.
- The parser does not recognize the white spaces in the specification of batch mode configuration options in an `rtcconfig` file.
- GDB provides limited support to debug Purify Plus instrumented binaries.
- GDB does not find Line table information for nested functions or subroutines inside modules for Fortran program compiled with `+noobjdebug` executables.

- GDB does not print the type of nested template class.
- WDB register tab does not display several registers that have a value of NaT.
- WDB does not show NaT in FP register during core dump analysis.
- Applications with custom allocator crashes when `set heap-check on`
- GDB displays error on printing floating point registers.
- GDB is unable to print formal and local variables for C++ applications that are heavy with inline functions.
- GDB is unable to print TLS variables on Integrity systems.
- GDB does not display value of `high-mem-count` for `show heap-check` command.
- GDB does not display correct message about the allocation context on `info corruption` command.
- The info leak command on a 32 bit `librtc` returns errors as pointers.
- GDB does not issue a warning message when it is unable to perform `thread-check`.
- GDB fails to retrieve debug info for some variables with correct `DWARF` emission.
- GDB crashes on info type command when wrong `DWARF` entries are emitted.
- GDB does not print a local variable correctly with a `gcc`(4.1.2 and later) compiled executable.
- GDB cannot place breakpoints in `F90` applications compiled with `+U77`.
- GDB fails to call a method on an object referenced through a `typedef` pointer in debugging a C++ code.
- GDB displays inconsistencies in evaluation of expressions.
- The `stepi` command fails to print disassembly for optimized code.
- GDB is not consistent in printing branch targets in hex for disassembled output.
- The application crashes in command line calls within GDB.
- GDB displays incorrect allocation stack trace after `dlclose` and subsequent `dlopen` in both interactive and batch mode RTC.

NOTE: This fix is not available for PA32.

Notes, Cautions, and Warnings

Following are some notes, cautions, and warnings related to WDB 5.8:

- The Run Time Checking feature (Interactive and Batch Mode) of WDB cannot be used with applications that redefine or override the default system-supplied versions of the standard library routines (under `libc.[sl|so]` and `libdld.[sl|so]`).

  Table 1 lists the dependent library routines for Run Time Checking using HP WDB.

Table 1 Dependent Library Routines for Run Time Checking using WDB

abort	atoi	chdir	clock_gettime
creat	ctime	uwx_register_callbacks	strstr
dlhook	U_STACK_TRACE	uwx_get_reg	write
execl	exit	fclose	fopen
fprintf	fscanf	getcwd	getenv
getpid	lseek	memchr	open
printf	rand	pthread_self	putenv
shl_findsym	shl_get_r	shl_load	shl_unload
sprintf	srand	sscanf	strcasecmp

Table 1 Dependent Library Routines for Run Time Checking using WDB *(continued)*

strdup	strlen	strchr	strtok_r
time	unlink	uwx_self_copyin	strchr
uwx_step	uwx_init	perror	shmctl
write	strcmp	shl_get	close
dlgetname	environ	fork	getpagesize
uwx_self_init_context	pthread_getschedparam	uwx_self_init_info	uwx_register_alloc_cb
strstr	uwx_register_callbacks	uwx_self_lookupip	

The Run Time Checking (dynamic memory, libraries, and pthreads checking) in WDB is dependent on the semantics and the standard behavior of these library routines. Run Time Checking in WDB results in unexpected and unpredictable behavior when used with applications that substitute or redefine these library routines.

Before enabling the Run Time Checking feature in WDB, use the `nm` command to determine if your application or the dependent libraries in your application redefine or substitute these library routines.

- Batch Mode RTC displays one of the following errors and causes the program to temporarily hang if the version of WDB and `librtc.[sl|so]` do not match, or if WDB is not available on the system:

  ```
  /opt/langtools/bin/gdb: unrecognized option `-brtc'
  Use `/opt/langtools/bin/gdb --help' for a complete list of options.
  ```

  Or

  ```
  execl failed. Cannot print RTC info: No such file or directory
  ```

  This error does not occur under normal usage where WDB or `librtc.[sl|so]` is used from the default location at `/opt/langtools/...`

  However, this error occurs if `GDB_SERVER` and/or `LIBRTC_SERVER` are set to a mismatched version of WDB or `librtc.[sl|so]` respectively.

- If the application does not link in the `libpthread` library (`lipbpthread.so|libpthread.sl`) or the `libpthread` tracer library (`libpthread_tr_so.1|libpthread_tr.1`), the following error message appears:

  ```
  Tracing version of libpthread is not loaded: set thread_check before starting the program, link your
  program with libpthread_tr or set LD_LIBRARY_PATH to contain /opt/langtools/wdb/lib/hpux32
  ```

  HP-WDB Debugger does not display backtrace under `Backtrace at the point of event occurrence` for thread conditions detected at thread-exit, if there is an implicit call to `pthread_exit()`.

Known Problems and Workarounds

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

- **Debugging an attached process that is not compiled for debugging may generate warnings**

  On attaching the debugger to a program that is not compiled for debugging, the process may stop in a system call and the following warning message about the various registers is displayed:

  ```
  No data warning: reading 'r3' register: No data warning: reading 'r4' register: No data
  warning: reading 'r5' register: No data warning: reading 'r6' register: No data warning
  ```

  This warning occurs when executing the `step` command, the `backtrace` command, or when attempting to view the register information. To avoid this warning message, use the `finish` command to execute the process until the system call returns. This warning message is generated when the process stops at a system call, and the registers cannot be read by the debugger. When the debugger calls the routine `ttrace`, it returns this warning.

- **Error attaching WDB to a process that is traced by tools using `ttrace`**

  HP WDB cannot attach to a process that is traced by tools that use `ttrace`, such as Caliper, adb, and tusc.

  The debugger displays the following error message while attempting to attach to such a process:

  ```
  Attaching to process <pid> failed.
  Hint: Check if this process is already being traced by another gdb or other ttrace tools like caliper and
   tusc.
  Hint: 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.
  ```

Known Issues

This section describes known issues in this release of HP WDB.

Batch Mode Thread Debugging

- During the execution of advanced thread checking for applications that fork, in the interactive mode, the following message appears if the GDB follows the child:

  ```
  Pthread analysis file missing!
  ```

  This error message appears because the thread-error information for the forked process is not available. However, if the forked process `exec()` s another binary, the thread-error information is available for the `exec`-ed binary.

- In both interactive and batch modes, if the applications exceed their thread stack utilization, the following error message appears:

  ```
  Error accessing memory address!
  ```

  This occurs when GDB attempts a command line call on an already overflowing thread stack.

Known Limitations

This section describes known limitations in this release of HP WDB.

Batch Mode Thread Debugging

- The feature does not obtain the thread-error information in batch mode for forked process in a multiprocessing application. However, if the `librtc.sl` library is pre-loaded, the debugger obtains the thread-error information in the batch mode for `exec`-ed application.
- You cannot specify an alternate output directory for the thread-error log. The thread-error log file is output into the current working directory only.
- HP WDB cannot execute both batch mode thread check and batch mode heap check together. If the `rtcconfig` file has both entries, then batch heap check overrides the batch thread check.

Installation Requirements and Compatibility Information

This section discusses the installation information for HP WDB.

Table 2 lists the `swinstall` products for HP WDB.

Table 2 HP WDB Installation

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

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

Compatibility

HP WDB is not supported on releases of the HP-UX operating system prior to 11i v1.

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 operating system.
- Installing 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

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: `DEBUG-PRG` and `SENTINEL` are co-requisite filesets for HP WDB and are 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 HP WDB product. If HP WDB is not already installed, it is automatically selected during installation.

Disk Space Requirements

For information on the disk space requirements to install HP WDB 5.8, see the Downloads website at:

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

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.

For specific installation instructions, see the HP WDB Technical Resources website at:

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

For more information on installation procedures and related issues, see the *Managing HP-UX Software with SD-UX* and other `README`, installation, and upgrade documentation provided with the HP-UX 11.x operating system package.

NOTE: If you install a compiler product that includes a version of HP WDB earlier than this version, `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”

To retain the later version of HP WDB, ignore this message. The new products are installed, and the latest version of HP WDB continues to be available.

Features Supported in Previous Versions of HP WDB

For information on previous HP WDB releases, see the HP WDB Technical Resources website at:

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

Related Documentation

HP WDB documentation is available at the following location:

`/opt/langtools/wdb/doc`

Table 4 lists the documents available for HP WDB.

Table 3 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	HTML	`/opt/langtools/wdb/doc/index.html`
GDB manpage	*gdb*(1)	

Software Availability in Native Languages

HP WDB 5.8 is available in the English language only.

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 subject of the message:

```
subscribe wdb-announce
```

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

```
unsubscribe wdb-announce
```