

HP WDB 5.9 Release Notes

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



© Copyright 2008 Hewlett-Packard Development Company, L.P

Legal Notices

Confidential computer software. Valid license from HP required for possession, use or copying. Consistent with FAR 12.211 and 12.212, Commercial Computer Software, Computer Software Documentation, and Technical Data for Commercial Items are licensed to the U.S. Government under vendor's standard commercial license.

The information contained in this document is subject to change without notice. The only warranties for HP products and services are set forth in the express warranty statements accompanying such products and services. Nothing herein should be construed as constituting an additional warranty. HP shall not be liable for technical or editorial errors or omissions contained herein.

UNIX is a registered trademark of The Open Group.

Intel and Itanium are trademarks or registered trademarks of Intel Corporation or its subsidiaries in the United States and other countries.

Table of Contents

| | |
|--|----|
| HP WDB 5.9 Release Notes..... | 7 |
| Announcement..... | 7 |
| What Is New in This Version..... | 7 |
| Debugger Support for Decimal Floating Point data type..... | 7 |
| Print Decimal Floating point data types..... | 7 |
| Printing Decimal floating point constant..... | 7 |
| Printing Decimal floating point variable..... | 8 |
| Evaluate Decimal floating point data types..... | 8 |
| Printing Type of DFP variable..... | 8 |
| Additional Support for binary floating point data type..... | 9 |
| Support of Binary Floating Point constants – f, l :..... | 9 |
| Support Binary Floating Point variables with format specifier <fmt>..... | 9 |
| Automatic Preload of librtc.[sl so]..... | 9 |
| Simplified Steps for RTC Using the +rtc option for the chatr command:..... | 10 |
| Simplified steps for Run Time Checking in the Attach Mode..... | 11 |
| Availability of a new compiler runtime check option +check=thread..... | 12 |
| Using Batch RTC to Process the .gdbinit File..... | 12 |
| Patches and Defect Fixes in HP WDB 5.9..... | 12 |
| Notes, Cautions, and Warnings..... | 13 |
| Known Problems and Workarounds..... | 15 |
| Known Issues..... | 15 |
| Batch Mode Thread Debugging..... | 15 |
| Known Limitations..... | 16 |
| Batch Mode Thread Debugging..... | 16 |
| Debugger Support for Optimized Code..... | 16 |
| Installation Requirements and Compatibility Information..... | 16 |
| Compatibility..... | 17 |
| Supported Configurations..... | 17 |
| Filesets..... | 17 |
| Disk Space Requirements | 17 |
| Installation Instructions..... | 17 |
| Features Supported in Previous Versions of HP WDB..... | 18 |
| Related Documentation..... | 18 |
| Software Availability in Native Languages..... | 19 |
| WDB Mailing Lists..... | 19 |

List of Tables

| | | |
|---|---|----|
| 1 | Dependent Library Routines for Run Time Checking using WDB..... | 13 |
| 2 | HP WDB Installation..... | 16 |
| 3 | HP WDB Documentation..... | 18 |

List of Examples

| | | |
|---|-------------------------------------|---|
| 1 | Binary Floating point constant..... | 9 |
|---|-------------------------------------|---|

HP WDB 5.9 Release Notes

Announcement

HP Wildebeest Debugger (WDB) 5.9 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 5.9

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.

- Debugger Support for Decimal Floating Point data type
- Additional Support for binary floating point data type
- Automatic Preload of `librtc.[sl|so]` with new `+rtc` option for `chatr` command
- Availability of a new compiler runtime check option `+check=thread`
- Batch RTC now enables processing of the `.gdbinit` file

Debugger Support for Decimal Floating Point data type

HP WDB 5.9 enables you to print and evaluate decimal floating point data types for programs running on the September 2008 release of HP-UX 11iv3, Integrity systems. This feature is available only for compiler versions A.06.20 and later.

The various features supported for decimal floating point data types are as follows:

- Print Decimal Floating point data types
- Evaluate Decimal floating point data types

Print Decimal Floating point data types

HP WDB 5.9 enables you to print a decimal floating point data type constant or variable. It handles and prints decimal floating point constant or variable when you use common GDB commands such as stack trace and commands line calls.

Printing Decimal floating point constant

```
(gdb) print <num> <df/dd/dl/DF/DD/DL>
df, DF - _Decimal32
dd, DD - _Decimal64
dl, DL - _Decimal128
```

This prints the decimal floating point constant based on the data type

Printing Decimal floating point variable

```
(gdb) print/<fmt> <var-name>  
<fmt> = df, dd, dl  
<var-name> - variable name
```

This prints the decimal floating point variable. If you specify format `fmt` then it prints the variable based on its data type.

Evaluate Decimal floating point data types

HP WDB 5.9 enables you to evaluate the decimal floating point variable and display the output. Use the commonly used `gdb` commands for evaluating and displaying expressions such as `print` to evaluate the decimal floating point variable.

HP WDB supports:

- Evaluation of expressions with decimal floating point constants and variables.
- Setting or assignment of decimal floating point constant or variable.
- Arithmetic operations such as addition, subtraction, multiplication, division, and negation with decimal floating point constants or variables.
- Comparison operations such as `==`, `!=`, `>`, `>=`, `<`, `<=` with decimal floating point constants or variables.
- Conversion between data types during assignment, arithmetic and comparison operations and while printing values in specified format.

Printing Type of DFP variable

```
(gdb) ptype <dfp-const>/<dfp-var>
```

It prints the type of the DFP variable or constant.

HP WDB handles exceptions such as overflow, infinity and division by zero for decimal floating point data type.

HP WDB handles finite, infinite and NaN (Not a Number) values of decimal floating point data type.



NOTE: HP WDB does not support decimal floating point data type for the following:

- Command line calls of intrinsic functions (Mathematical functions such as cos, sin, and log)
 - Decimal floating point data type support for FORTRAN and c++
 - HP-UX 11iv2 Integrity and HP 9000 systems
-

Additional Support for binary floating point data type

HP WDB 5.9 provides the following support for binary floating point data type:

- Support of Binary Floating Point constants – f, l :
- Support Binary Floating Point variables with format specifier <fmt >

Support of Binary Floating Point constants – f, l :

If the binary floating point constant contains the letter `f` or `l` then HP WDB recognizes it as float or long double binary floating point constant. Generally a floating point constant without `f` or `l` is considered as double binary floating point constant.

Example 1 Binary Floating point constant

```
(gdb) p <num><f/l>
```

This prints the binary floating point constant based on its data type.

Support Binary Floating Point variables with format specifier <fmt>

HP WDB 5.9 prints the variable in the specified binary floating point format. The following are the new format specifiers for binary floating point variables:

```
(gdb) p/f <var-name>
```

This prints the binary floating point value as float.

```
(gdb) p/db <var-name>
```

This prints the binary floating point value as double.

```
(gdb) p/l <var-name>
```

This prints the binary floating point value as long double.

Automatic Preload of `librt.[sl|so]`

The new `+rtc` option for the `chatr` command enables you to automatically preload the `librt.[sl|so]` runtime library. In addition to automatically preloading the `librt.[sl|so]` library, it also maps the shared library as *private*. This enhancement simplifies the current method of explicitly preloading the appropriate `librt.[sl|so]` library using the `LD_PRELOAD` environment variable.

The `+rtc<enable | disable>` option is available for dynamic linker versions B.11.66 and later on HP 9000 systems, and dynamic linker versions B.12.51 and later on Integrity systems.

To set the target application to preload `librtc.[sl|so]`, enter the following command at the HP-UX prompt:

```
$ chatr +rtc enable <executable>
```

The `+rtc` option for the `chatr` simplifies the steps for batch mode and attach mode of memory and thread debugging. Run Time Checking (Memory and Thread debugging) is made available using this option.

Simplified Steps for RTC Using the `+rtc` option for the `chatr` command:

To use batch mode memory leak detection, complete the following steps:

1. Compile the source files.
2. Create an `rtcconfig` file in the current directory with the required batch mode commands.
3. Define the environment variable `BATCH_RTC` at the HP-UX prompt, as follows:

```
export BATCH_RTC=on
```

4. Enter the following command at the HP-UX prompt to set the target application to preload `librtc.[sl|so]`:

```
$ chatr +rtc enable <executable>
```

At the end of the run, output data file is created in `output_dir` (if defined in `rtcconfig`), or the current directory.

To use batch mode thread debugging, complete the following steps:

1. Compile the source files.
2. Set the `LD_LIBRARY_PATH` environment variable to appropriate directory based on the platform.
3. Create an `rtcconfig` file in the current directory with the required thread conditions.
4. Define the environment variable `BATCH_RTC` at the HP-UX prompt, as follows:

```
export BATCH_RTC=on
```

5. Enter the following command at the HP-UX prompt to set the target application to preload `librtc.[sl|so]`:

```
$ chatr +rtc enable <executable>
```

If HP WDB detects any thread error condition during the application run, the error log is output to a file in the current working directory. The output file has the following naming convention:

```
<executable name>.<pid>.threads
```

where:

pid is the process id.

Simplified steps for Run Time Checking in the Attach Mode

This section describes the simplified steps for Run Time checking in the Attach mode using the `+rtc` option for the `chartr` command.

For debugging memory after attaching GDB to a running process, complete the following steps:

1. Enter the following command at the HP-UX prompt to set the target application to preload the `librtc.[sl|so]` runtime library.

```
$ chartr +rtc enable <executable>
```

2. Run the program.
3. Start the debugging session, as follows:

```
gdb -leaks <executable-name> <process-id>
```

For debugging threads after attaching GDB to a running process, complete the following steps:

1. Set the `LD_LIBRARY_PATH` environment variable to appropriate directory based on the platform.
2. Enter the following command at the HP-UX prompt to set the target application to preload the `librtc.[sl|so]` runtime library:

```
$ chartr +rtc enable <executable>
```

3. Run the program.
4. Start the debugging session, as follows:

- Attach the debugger to the required process and enable thread debugging, as follows:

```
gdb -thread -p <process-id>
```

Or

```
gdb -thread <executable> <process-id>
```

- Alternately, you can attach the process to the debugger and consequently invoke thread debugging, as follows:

```
$ gdb <executable> <pid>  
(gdb)set thread-check on
```



NOTE: `+mem_check` to `chatr` command can be used to automatically preload the library for memory debugging. The `+mem_check` option to `chatr` command doesn't support automatic preloading of libraries needed for thread debugging.

Availability of a new compiler runtime check option `+check=thread`

The new compiler runtime check option `+check=thread` enables batch mode thread debugging features of HP WDB, without setting any other environment at runtime. `+check=thread` must only be used with multi-threaded programs. It is not enabled by `+check=all`. This feature is available only for compiler versions A.06.20 and later.

Using Batch RTC to Process the `.gdbinit` File

Batch RTC now enables processing of the `.gdbinit` file. You can use the `.gdbinit` file to specify path settings such as `dir`, `objectdir`, and `pathmap` to set the path of the source and object files in case the source or object paths are different than the current directory, so that the generated RTC reports display the symbol names and line numbers correctly. This feature is optionally enabled only when the `RTC_PROCESS_GDBINIT` variable is set to 1. This feature has some limitations on what commands in the `.gdbinit` file. For more information on this feature, see the *Debugging with GDB* user manual and the *Debugging Dynamic Memory Usage Errors Using HP WDB* white paper at:

<http://www.docs.hp.com/en/dev.html>

Patches and Defect Fixes in HP WDB 5.9

The following defects are fixed in HP WDB 5.9:

- The `disable` command produces misleading information for multi-threaded programs.
- The `pxdb` fails with an out of memory error message while reading files of size 2 GB.
- WDB incorrectly reports heap object registered with different size when an application is compiled with `+check=bounds:pointer` to perform out of bounds check.
- WDB incorrectly reports value of thread-local variable.
- The `next` command fails to step into a forked child process.
- WDB does not recognize pack core filenames with extensions other than `.tar.Z`. If the application is compiled with `+check=memory`, it crashes while WDB attempts to print the memory information.
- Batch RTC fails even when the `output_dir` command is set and the current working directory does not have write permission.
- WDB is unable to perform Java debugging.

- GDB crashes while printing the type and value of the structure member of array type with array of size `MAX_INT_32`.
- GDB crashes while reading the elf large section details of the application that contains large number of sections.
- GDB crashes while printing the type of the member function of the class whose function name is similar to that of function name in standard libraries.
- Occasional gdb crash with batch mode thread check when application aborts. On running batch mode thread check on applications that crash or if **Ctrl-C** is executed while the application is running, gdb crashes.
- Occasional hang with batch mode thread check, when the application has an incorrect usage of `pthread_cond_wait()` call.

Notes, Cautions, and Warnings

Following are some notes, cautions, and warnings related to WDB 5.9:

- 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 `libdl.[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

| | | | |
|--------------------------|----------------------------|-------------------------------------|----------------------------|
| <code>abort</code> | <code>atoi</code> | <code>chdir</code> | <code>clock_gettime</code> |
| <code>creat</code> | <code>ctime</code> | <code>uwx_register_callbacks</code> | <code>strstr</code> |
| <code>dlhook</code> | <code>U_STACK_TRACE</code> | <code>uwx_get_reg</code> | <code>write</code> |
| <code>execl</code> | <code>exit</code> | <code>fclose</code> | <code>fopen</code> |
| <code>fprintf</code> | <code>fscanf</code> | <code>getcwd</code> | <code>getenv</code> |
| <code>getpid</code> | <code>lseek</code> | <code>memchr</code> | <code>open</code> |
| <code>printf</code> | <code>rand</code> | <code>pthread_self</code> | <code>putenv</code> |
| <code>shl_findsym</code> | <code>shl_get_r</code> | <code>shl_load</code> | <code>shl_unload</code> |
| <code>sprintf</code> | <code>srand</code> | <code>sscanf</code> | <code>strcasemp</code> |
| <code>strdup</code> | <code>strlen</code> | <code>strchr</code> | <code>strtok_r</code> |
| <code>time</code> | <code>unlink</code> | <code>uwx_self_copyin</code> | <code>strchr</code> |
| <code>uwx_step</code> | <code>uwx_init</code> | <code>perror</code> | <code>shmctl</code> |
| <code>write</code> | <code>strcmp</code> | <code>shl_get</code> | <code>close</code> |
| <code>dlgetname</code> | <code>environ</code> | <code>fork</code> | <code>getpagesize</code> |

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

| | | | |
|------------------------------------|-------------------------------------|---------------------------------|------------------------------------|
| <code>uwx_self_init_context</code> | <code>pthread_getschedparam</code> | <code>uwx_self_init_info</code> | <code>uwx_register_alloc_cb</code> |
| <code>strstr</code> | <code>uwx_register_callbacks</code> | <code>uwx_self_lookupip</code> | |

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 (`libpthread.so|libpthread.sl`) or the `libpthread` tracer library (`libpthread_tr.so.l|libpthread_tr.l`), 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 `librttc.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 `rttconfig` file has both entries, then batch heap check overrides the batch thread check.

Debugger Support for Optimized Code

On PA-RISC systems for 32-bit and 64-bit applications, printing of local variable values may be inaccurate for +O1 and higher optimizations.



NOTE: HP WDB 5.9 contains limited debugging support for optimized PA-RISC 32-bit applications and even more limited support for debugging optimized code (DOC) on PA-RISC 64-bit applications. Additionally, the debugging support becomes limited as higher levels of compiler optimization are used.

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 5.9, 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 |
|---|---------------------|---|
| <i>Debugging with GDB</i> | PDF | <code>/opt/langtools/wdb/doc/gdb.pdf</code> Emacs: <code>/opt/langtools/wdb/doc/gdb.info</code> (Copy the files to your info directory first.) |
| <i>GDB Quick Reference Card</i> | PDF | <code>/opt/langtools/wdb/doc</code> |
| <i>Getting Started with HP WDB</i> | HTML | <code>/opt/langtools/wdb/doc/html/wdb/C/GDBtutorial.html</code> |
| <i>XDB to WDB Transition Guide</i> | HTML | <code>/opt/langtools/wdb/doc/index.html</code> |
| <i>Using the HP WDB Terminal User Interface</i> | HTML | <code>/opt/langtools/wdb/doc/index.html</code> |
| GDB manpage | <code>gdb(1)</code> | |

Software Availability in Native Languages

HP WDB 5.9 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