MULTI: IDE Version 6.1.6 Release Notes

Green Hills Software
30 West Sola Street
Santa Barbara, California 93101
USA
Tel: 805-965-6044
Fax: 805-965-6343
www.ghs.com

DISCLAIMER

GREEN HILLS SOFTWARE MAKES NO REPRESENTATIONS OR WARRANTIES WITH RESPECT TO THE CONTENTS HEREOF AND SPECIFICALLY DISCLAIMS ANY IMPLIED WARRANTIES OF MERCHANTABILITY OR FITNESS FOR ANY PARTICULAR PURPOSE. Further, Green Hills Software reserves the right to revise this publication and to make changes from time to time in the content hereof without obligation of Green Hills Software to notify any person of such revision or changes.

PubID: multi_release_notes-525957
Branch: http://toolsvc/branches/release-branch-61
Date: February 19, 2015

Contents

Chapter 1

Changes in MULTI 6.1.6

Contents

System Requirements

Note

If your Compiler is newer than the IDE, see the Compiler release notes for up-to-date system requirements.

Windows

Running a full MULTI 6 installation on Windows requires:

- A 1.4 GHz Pentium 4 processor, or greater (a 2.0 GHz Core 2 Duo processor is recommended).
- Windows XP (IA-32, 32-bit mode only), Windows Vista (32- or 64-bit mode), or Windows 7 (32- or 64-bit mode).
- 256 MB of RAM (1 GB is recommended).
- 2 GB of free disk space per installation. If you are installing to a drive other than your primary Windows drive, you must also have at least 50 MB available on the primary drive for temporary files and DLLs.
- A 256-color display at a resolution of 1024x768 (1920x1200 is recommended).
- A CD-ROM drive.
- An Ethernet connection.
- The ability to write to the Windows System directories. On most Windows systems, you must also be a member of the Administrators group. End users must have full access to installed files.

Linux

Running a full MULTI 6 installation on Linux requires:

- A 1.4 GHz Pentium 4 processor, or greater (a 2.0 GHz Core 2 Duo processor is recommended).
- Red Hat Enterprise Linux 5 or Ubuntu 8.04 or 10.04 (32-bit or 64-bit mode; native development of 64-bit applications is not supported). 32-bit compatibility libraries are required in 64-bit mode.
- 256 MB of RAM (1 GB is recommended).

- 2 GB of free disk space per installation.
- A 256-color display at a resolution of 1024x768 (1920x1200 is recommended).
- A mounted CD-ROM drive.
- An Ethernet connection.
- Write permissions to the installation directory.

Solaris

Running a full MULTI 6 installation on Solaris requires:

- A sun4u SPARC-based workstation.
- Solaris 8, 9, or 10 (32-bit or 64-bit mode; native development of 64-bit applications is not supported).
- 256 MB of physical memory (1 GB is recommended).
- 2 GB of free disk space per installation.
- A graphical display running the X Windowing System.
- A mounted CD-ROM drive.
- An Ethernet connection.
- Write permissions to the installation directory.

Product Compatibility

The MULTI 6.1.6 IDE is officially supported with:

- INTEGRITY versions 5.x, 10.0.2, 11.0.2, 11.0.4, and 11.2.2. For information about compatibility with newer versions of INTEGRITY, see the INTEGRITY release notes.
- Green Hills Compiler versions 2012.1 through 2014.1. If your Compiler is newer than the IDE, see the Compiler release notes for up-to-date compatibility information.

Warning

Do not install Green Hills Debug Probe software version 4.2 or earlier over your MULTI 6 installation.

New Features and Enhancements

Support for Large Virtual Function Tables

MULTI 6.1.6 adds support for large virtual function tables (**--large_vtbl_offsets**)—a feature that was enabled by default in Compiler 2014.1.

Support for Freeze-Mode Debugging of INTEGRITY

MULTI 6.1.6 adds support for freeze-mode debugging of INTEGRITY 11.2.2.

Chapter 2

Changes in MULTI 6

Contents

New Features and Enhancements

Faster Builds

Parallel builds (**Parallel Build**) are now enabled by default in the Project Manager, and there have been substantial improvements to the parallel build algorithm. Customers who were not making use of parallel builds before have observed build speedups of 4.5 times on 4-core, hyper-threaded machines. Actual speedups will vary depending on your build machine and the structure of your code base.

Ability to Retrieve More Trace Data from New SuperTrace Probes

MULTI 6 makes it easy to take advantage of the large amount of trace RAM on the SuperTrace Probe v3. As in previous versions of MULTI, you can use the **Target buffer size** option to configure the amount of trace data that the Debugger retrieves from the end of the trace buffer. However, if you later find that you need more data than was originally configured, MULTI 6 allows you to go back and retrieve it from the SuperTrace Probe v3, which always uses all available trace RAM regardless of the configured **Target buffer size**. For more information, see "Retrieving Trace Data" in Chapter 19, "Analyzing Trace Data with the TimeMachine Tool Suite" in the *MULTI: Debugging* book.

Faster Trace Processing

Trace processing performance is greatly improved in MULTI 6. In addition to general optimizations, the Debugger is now capable of making use of multiple cores to decode trace data in parallel. Parallel trace decoding is currently supported with ETMv1 targets; support for other targets will be added by future Green Hills Debug Probe releases.

Trace processing is about 4 to 6 times faster than before on targets where parallel decoding is supported (with a 4-core PC), and 1.5 to 3 times faster on other targets. SuperTrace Probe v3 is required for maximum trace processing performance.

Separate Releases Allow Greater Flexibility When Upgrading

The Compiler and IDE products now release independently of each other, allowing you to upgrade them separately. This makes it possible to freeze on one version of the Compiler while upgrading the version of the IDE, or vice versa.

64-Bit Support in the Debugger

The standard MULTI Debugger (but not the TimeMachine Debugger) has been improved to support 64-bit targets. Specific processor support is dependent on INTEGRITY and Green Hills Probe releases.

128-Bit Hardware Registers Supported

MULTI 6 adds support for 128-bit hardware registers.

Security and Relationship Views Added to Integrate

The Integrate GUI now displays high-level relationships between AddressSpaces, allowing you to easily make sense of large, complicated INTEGRITY systems. Additionally, it helps you find security risks by showing avenues of communication between AddressSpaces. For more information, see Appendix G, “Integrate Views” in the *MULTI: Debugging* book.

$locals$ Window Updates as You Navigate the Call Stack

Local variable views (**$locals$** in the Data Explorer) now update to show local variables in the new stack frame as you navigate up or down the call stack (for example by using the **E+** or **E-** command). Local variable views that you open by specifying a stack frame (for example, with **view number:$locals$**) are tied to that stack frame and still do not update when you move up or down the stack.

Ability to Restore Breakpoints

MULTI 6 gives you the ability to view and restore breakpoints that have been removed during the current MULTI debugging session. This is especially useful if

you accidentally delete a breakpoint that took some time to set up (for example, if you created a breakpoint that is associated with a series of commands). For more information, see "Restoring Deleted Breakpoints" in Chapter 8, "Executing and Controlling Your Program from the Debugger" in the *MULTI: Debugging* book.

Greater Window Availability

The **Call Stack**, **Register View**, **Memory View**, and **Local Variables** windows can now be opened before your program is loaded to the target or while it is running.

New System Variables

MULTI 6 includes the new `_OS_DIR`, `_SETUP_SCRIPT`, `_SETUP_SCRIPT_DIR`, `_TOP_PROJECT`, and `_TOP_PROJECT_DIR` system variables, which can be used to allow for more complex functionality in setup scripts. For more information about these variables, see "System Variables" in Chapter 14, "Using Expressions, Variables, and Procedure Calls" in the *MULTI: Debugging* book.

Target List Remembers Node Expansion/Collapse

The target list now remembers node expansion/collapse for INTEGRITY applications and AddressSpaces across sessions.

Sortable Columns in the Target List

The target list can now be sorted by column. Simply click the header for the column that you want to sort by. This setting is not saved when the Debugger is closed or when new Debuggers are opened.

Improved Support for Preprocessor Macros in the Debugger

Support for evaluation of preprocessor macros in the Debugger has been rewritten to conform more closely to the C/C++ standards. The new implementation better handles expansion of nested macros and arguments containing commas, and it supports the ANSI C macro argument concatenation and stringification operators.

Browsing References Improvement

When browsing references of a C++ class's member function, you are now shown all possible call sites for the corresponding virtual functions of the class's super classes, not just the explicit references of the member function itself.

if Command Supports else if

The MULTI Debugger command **if** now supports `else if` clauses, which make it easier for you to specify a complex series of conditionally executed commands in MULTI scripts.

Licensing Improvements

A number of improvements have made to licensing in MULTI 6. Improvements include:

- A streamlined **MULTI Licensing Wizard**, which differentiates between end user and system administrator tasks
- A new **License Information** window, which gives users and system administrators an easy way to look at usage patterns and to determine who is using licenses
- Whitelisting and blacklisting capabilities, which are useful if you want to reserve licenses for certain users, or if you want to bar particular users from obtaining licenses
- Clearer and more concise documentation

Faster Scanning of Subversion Checkouts

The MULTI 6 Checkout Browser completes scans of Subversion checkouts up to 10 times faster than before.

Changes in Behavior

IDE and Compiler Installed into Separate Linking Directories

The MULTI IDE and Green Hills Compiler products are now installed into separate directories that are linked as part of the installation process. If the link between the products breaks (for example, because the Compiler directory is moved), you are prompted to relink them.

Upgrading to MULTI 6

If you are upgrading from an earlier version of MULTI, you must contact Green Hills Software to obtain MULTI 6 licenses.

License Clients No Longer Broadcast for a License Manager

In MULTI 5, license clients that could not obtain licenses would broadcast over the network for a License Manager from which they could obtain licenses. In MULTI 6, clients do not broadcast over the network. If a GUI program is unable to obtain a license, the **MULTI Licensing Wizard** opens so that you can request or install licenses or specify a License Manager. If a non-GUI program is unable to obtain a license, the program exits.

Changes to Computer- and Dongle-Locked Licensing

In MULTI 6, computer- and dongle-locked licenses are only available to one user at a time. If more than one user is connected at once, the license is only available from the console.

Changes to Trace Retrieval

MULTI 6 retrieves trace if you step or run backwards in the live Debugger or if you otherwise launch TimeMachine. Additionally, **Retrieve trace when target halts** is now disabled by default. These changes help prevent gaps in trace data when you halt and run with a large trace buffer.

Trace Decoding No Longer Defaults to Reading Unknown Opcodes

If a previous version of the MULTI Debugger encountered an instruction address during trace decoding that did not map to any loaded ELF files, it attempted to read the opcode from target memory by default. In MULTI 6, the Debugger aborts trace decoding by default. To revert to the old behavior, select **Read unknown opcodes from target (may halt target)** on the **Analysis** tab of the **Trace Options** window, and click **Apply**.

Changes to TS_Packet May Require TimeMachine API Application Updates

The `aid` and `tid` fields in `TS_Packet` structs of type `TSP_PID` have been changed from `uint32` to `uint64`, and the `supervisor` field has been removed. As a result, any TimeMachine API applications that inspect `TS_Packet` structs of type `TSP_PID` must be updated as follows:

- C and C++ TimeMachine API applications — Rebuild these with the updated version of ***ide_install_dir*\timemachine_api\ts_packet.h**.
- Python TimeMachine API applications — Update these to import the updated version of ***ide_install_dir*\timemachine_api\example_python\timemachine_api.py**.

Behavior of Primary and Secondary Debugger Windows

The target list is no longer available in Debugger windows that have been launched from another Debugger window. Additionally, the original Debugger window functions as a control window; when it is closed, any secondary Debugger windows that were launched from it are also closed.

Large Numbers No Longer Broken Up by Default

In MULTI 5.x and earlier, large numbers were displayed with underscores or dots to break up the numbers and ease reading. MULTI no longer breaks up large numbers by default; however, you can revert to the old behavior by using the **numberSeparator** configuration option. To use an underscore to break up large numbers in MULTI 6, enter the following command in the Debugger command pane:

```
> configure numberSeparator _
```

Default Number of Scroll Back Lines Reduced

The default number of scroll back lines available in the **Cmd**, **Trg**, **I/O**, **Srl**, **Py**, and **Tfc** panes has been decreased from `524288` to `10240` to reduce memory usage. Additionally, the minimum number of scroll back lines that can be set has changed from `1024` to `10`. For more information, see the description of **cTextSize** in "Other Debugger Configuration Options" in Chapter 8, "Configuration Options" in the *MULTI: Managing Projects and Configuring the IDE* book.

Storage of PID and TID Debugger Aliases

The Debugger component previously stored 32-bit process ID (PID) and task ID (TID) aliases as signed integers and hex numbers, but now stores 64-bit PID and TID aliases as unsigned integers and hex numbers. As a consequence, previously valid PID aliases such as `debugger.pid.-1` are no longer used. Aliases are commonly specified as arguments to the **route** command. For information about the **route** command, see "Command Manipulation and Macro Commands" in Chapter 15, "Scripting Command Reference" in the *MULTI: Debugging Command Reference* book.

Deprecated Features

The following features are deprecated in MULTI 6:

- **grun** utility
- **versCtrl_RcsAutoCheckout** configuration option
- The **-filter** option to **lic_lm** and the equivalent **IP Addr Filter** field of the **License Manager Settings** dialog box (superseded by new whitelisting and blacklisting capabilities)
- Certain MULTI 5 licensing programs have been deprecated in favor of improved MULTI 6 programs. To prevent accidental use, the executables have been renamed as stated.

Deprecated Program	Discontinued Executable	Renamed Executable	Superseded by
Green Hills License Manager (GHSlm)	**lm**	**lm_compat**	**lic_lm**
glicusers utility	**glicusers**	**glicusers_compat**	**lic_userlist**
find_ghs_licenses utility	**find_ghs_licenses**	**find_ghs_licenses_compat**	**lic_find_licenses**

Removed Features

The following features are not available in MULTI 6:

- Legacy licensing technology (aka Elan)
- License Manager Web interface (superseded by a more complete license information display)
- `LMSHOW` and `LMHIDE` licensing configuration variables
- Legacy **.bld** project files
- The Project Manager menu entry **Connect → Task Manager**
- **ctags**-style tags files

- The following MULTI Editor commands and equivalent MULTI Editor menu entries:
  - **Column** command/**View → Column**
  - **Make** command/**Tools → Make**
  - **Notepad** command/**Tools → Notepad**
  - **WGUtils** command/**Tools → Launch Utility Programs**
- Hex Editor
- The **Checkout Browser** menu entry **File → Local Rescan**
- PVCS and RCS integration
- MULTI Version Control (MVC)
- Custom version control integration
- **-dotciscxx** command line option (removed from the Debugger only)
- The following MULTI Debugger commands and menu entries:
  - **C**, **Cb**, **Cu**, and **CU** commands
  - **note** command/**Tools → Notepad**
  - **Debug → Load Symbols** (Use the **loadsym** command instead. See Chapter 2, "General Debugger Command Reference" in the *MULTI: Debugging Command Reference* book.)
  - **Debug → Unload Symbols** (Use the **unloadsym** command instead.)
- **Find Function** window
- `$ent` and `$ret` special operators
- FORTRAN debugger
- Hardware Registry
- Memory filters
- RoseRT integration
- HP-UX host support
- Windows 2000 host support

If you depend on any of the preceding features, please contact your Green Hills sales representative.

Chapter 3

Changes in MULTI 5.0.6

Contents

New Features and Enhancements

MULTI Project Manager

MULTI 5 introduces the MULTI Project Manager, a tool providing both the power of the traditional project builder along with new features designed to get the user up and running faster and make changes to existing projects safely and quickly.

MULTI's Project Manager gets you up and running with its new Project Wizard. Using this tool, you can create a stand-alone or INTEGRITY-based system quickly by answering straightforward questions.

Once your project has been created, the Project Manager continues to help you maintain your project as it grows and changes, allowing you to add and reconfigure components as varied as source files and INTEGRITY file systems.

To help you visually understand different aspects of your system, the Project Manager provides several different views of the project including a high level whiteboard view which can help you understand the structure of very large or complicated projects quickly.

Flexible Target Connections

Flexible target connections allow you to connect to a target once to perform a wider variety of debugging tasks. The power of a flexible connection saves you time as you explore hardware-software interaction. Using a flexible target connection, you can inspect or modify your bare target, download code or flash code to ROM, or debug code already present on the target. Using MULTI's Prepare Target Dialog, you specify what kind of access to the board you will need or how you need to modify your target's connection to the MULTI Debugger.

Improved Target Management from the Debugger

The new target list greatly improves visibility of items connected to the Debugger including bare hardware targets, multi-core systems, stand-alone programs, INTEGRITY systems, and many combinations thereof.

This list, which appears in the top pane of the Debugger window, allows you to see groupings between different tasks, address spaces, and applications for more powerful debugging of INTEGRITY systems. Each entry in the list can be selected, allowing you to debug or view that item in the source pane. This greatly reduces window clutter by allowing you to reuse a single Debugger window. Debugging multiple items side by side is as simple as double-clicking an entry in the list.

For stand-alone or bare board systems, you can now more easily connect to a board that is not yet running an application to perform board exploration or setup tasks.

Target Traffic Pane

For developers who need a behind-the-scenes look at how MULTI is interacting with their hardware, MULTI 5 introduces the Target Traffic Pane. This console, which is integrated into the tabbed pane at the bottom of the debugger window, shows all messages sent between MULTI and your hardware. This can help diagnose and/or rule out problems caused by a sensitive target's response to being probed by a debugger.

TimeMachine Enhancements

TimeMachine is more integrated into the debug environment than ever before. If your target supports TimeMachine, trace data is collected automatically in the background, allowing you to step back through code or use any other TimeMachine feature at any point in your debugging session. TimeMachine can handle many gigabytes of trace data with improved performance.

TimeMachine provides many tools which allow you to see what your system is doing. PathAnalyzer now features an array of visual aids to make understanding the path analysis information more intuitive.

You can also create your own custom TimeMachine tools using the TimeMachine API. The TimeMachine API can be accessed by your C code or Python script, allowing you to build your own analysis tools.

Improved Kernel Object Awareness

With the release of INTEGRITY 10, MULTI users will experience new levels of operating system debugging excellence. Paired with INTEGRITY 10, MULTI 5 will allow users to view INTEGRITY objects, explore relationships between INTEGRITY objects, and communicate directly with the kernel.

Global Profiling

Global Profiling provides continuously updated profiling information including CPU usage for individual address spaces and tasks. It also provides up-to-the-millisecond profiling information via MULTI's standard profiling tools.

Global Profiling is available only when debugging INTEGRITY 10 systems.

Integrated Python Engine

MULTI 5 comes equipped with a fully functional integrated python engine. Using this powerful new tool users can use MULTI in new and exciting ways.

Using a GUI building tool (TCL/TK, for example) you can build your own customized debugging windows. These windows can be viewers for special data structures, system emulators, or specialized program controllers.

Using the integrated Python engine you can control and customize the behavior of the MULTI IDE. You can specify how the MULTI IDE is launched down to window size and location.

Using the MULTI Launcher, users can execute any series of actions in the MULTI IDE. The integrated python engine expands this capability to allow the user to create programmed actions which can be executed based on dynamically available information.

Python is a complete scripting language. This flexibility and power provides all of the capabilities you might need for complete scripting including string manipulation, complete math libraries, and the power of a fully functional programming language.

Extended Version Control Support

MULTI 5 expands its version control integration to include support for the popular Subversion version control system (versions 1.3.x through 1.5.x). Additionally, support for ClearCase has been tuned to support dynamic view checkouts.

Windows Taskbar Organization

The MULTI Taskbar Organizer groups all windows from the MULTI IDE into a single entry in the Windows taskbar or tray, providing quick access to all windows. This feature is only available on Windows.

Improved Help Navigation

MULTI's new Help Viewer allows users to navigate and search the Green Hills library of documentation with greater ease, speed, and accuracy. This provides a more pleasant and powerful documentation browsing experience.

Changes in Behavior

Use of Dongle-Locked Licenses over Remote Desktop

In earlier versions of MULTI, dongle-locked licenses were unintentionally available to machines connecting via Remote Desktop to a dongle-licensed PC. In MULTI 5, dongle-locked licenses are no longer available over Remote Desktop sessions if the dongle-licensed machine allows more than one remote user connection. Dongle-locked licenses are available over Remote Desktop sessions if the remote machine allows only a single user connection.

Halt Issued Before Setup Script in Pre-5.0.6 Versions of MULTI

In versions of MULTI 5.x prior to 5.0.6, the MULTI Debugger issues a **halt** command to the target prior to running the setup script (if any). This adversely affects some hardware targets, where run control via the Debugger is not reliable after the target has been powered on, but before it has been reset. Unless you

manually reset the target, the Debugger typically reports errors when trying to download a program to such a target, and the download fails.

MULTI 5.0.6 does not issue the preliminary **halt** command. If your target has a setup script, it should use the MULTI **halt** command to halt the target before reading or writing any registers or memory. This is especially important if your setup script uses the MULTI expression evaluator to write registers (for example, `$r1 = 0x12345`) instead of using debug server scripting commands (for example, `target rw r1 0x12345`).

Unbundled Features

Unbundled features are no longer included in MULTI, but are still available for separate purchase.

The following features have been unbundled from MULTI 5:

- FORTRAN debugger
- Hex Editor
- MULTI Version Control (MVC)
- Custom version control integration
- Hardware Registry Server
- Script Debugger
- PCI Devices window
- Memory filters
- OSE integration
- RoseRT integration
- Deprecated MATLAB integration interfaces. MATLAB integration is still supported through MULTI's Python interface
- Embedded Linux debugging
- HP-UX host support

Deprecated Features

The following features are deprecated in MULTI 5:

- Legacy licensing technology (aka Elan)
- PVCS, VSS, CVS, and RCS integration
- **-dotciscxx** command line option (deprecated in the Debugger only)

Removed Features

The following features are not available in MULTI 5:

- Support for using Elan dongle-locked licenses over Remote Desktop
- Function Flow tool
- The Profile window's Block Detailed report is not available with TimeMachine profiling
- **rtnserv** debug server (use multiple **rtserv** or **rtserv2** connections instead)

Chapter 4

Known Issues

Contents

MULTI Installation

For known issues regarding INTEGRITY installation, see “Compatibility with INTEGRITY” on page 31.

Name of Install Directory Cannot Contain Spaces

The MULTI IDE cannot be run from a directory whose name contains spaces.

Host Support

Use of TimeMachine API on Solaris Requires Python Installation

On Solaris, use of the TimeMachine API requires an installation of Python 2.5 or Python 2.3 with the ctypes package installed. On all other supported hosts, Python 2.3 with the ctypes package is installed as part of the MULTI IDE installation.

MULTI-Python Integration on Linux May Require Compatibility Library

To use the MULTI-Python integration with some newer Linux distributions, you will need to install a C++ compatibility library. On Debian/Ubuntu, this can be installed with the command:

```
sudo apt-get install libstdc++5
```

On Fedora Core/Red Hat Enterprise Linux, this can be installed with:

```
yum install compat-libstdc++33
```

There might be some variance in the name of the package depending on the specific version you are using. When using a version of Linux beyond those mentioned here, you will need to install a version of the GNU libstdc++v3 that is used with gcc-3.3.

Project Manager Issues

Project Manager Performance on Very Large Project Files

Individual project files containing more than 50,000 files may be slow to load and require significant memory resources when opened in the MULTI Project Manager on Windows hosts. If insufficient resources are available, the Project Manager may exit unexpectedly. This issue is most likely to occur on Windows XP hosts.

Workaround: To improve performance, try breaking your project file into several Subproject files. These files will only be loaded as needed, reducing the resources needed by the Project Manager.

Undefined __OS_DIR__ Macro When Opening Some ThreadX Projects

If you open a ThreadX project that does not include a **-os_dir** setting in the Project Manager, you may see a dialog containing the error message: "macro not defined (__OS_DIR__)". If you see this message, adding new examples and executables to your project via the Project Manager may produce unexpected results.

Workaround: To resolve this problem, open your project, click **OK** in the error message dialog, select **Edit → Set Build Target**, and enter the path to your ThreadX installation in the **Target Selector** dialog that appears.

Editor Issues

Limitation of Brace Matching with Preprocessor Directives

In the MULTI Editor, placing an opening brace inside a conditional preprocessor block may cause the closing brace to match the incorrect opening brace. This is most common when there are multiple conditional blocks and only one is expected to be compiled, for instance with `#if`/`#else`/`#endif`.

File Permissions Under Cygwin

If you use Cygwin's **chmod** to change a file's permissions to read only, you will not be able to overwrite the file using the MULTI Editor. Cygwin does not update its version of the file system when a change is made to the Windows file system permissions.

Version Control Issues

Subversion Authentication on Windows

The MULTI Subversion integration will not properly handle password prompts on Windows when using some versions of the Subversion client if you do not already have cached authentication information. If your repository uses authentication for read or write access, and you have not yet entered a password for any Subversion operation, you are likely to encounter this problem, during which you will notice that MULTI or the Checkout Browser appears to hang.

Workaround: Perform your first password-requiring Subversion operation from the command line to ensure authentication is set up properly.

If you encounter this problem, you can recover your session by taking the following steps:

1. Kill the running **svn.exe** process. This should recover MULTI.
2. Run **svn cleanup** from the command line inside your checkout.
3. Perform the operation (probably a checkout, update, or commit) from the command line. Please consult the documentation for your specific version of the Subversion client for exact syntax.
4. Once you have successfully entered your password through Subversion's password prompt, the MULTI version control integration should work correctly.

Problems Observed with Subversion 1.4.x

Using Subversion 1.4.x is not recommended. Subversion 1.4.2 on Linux has been observed to occasionally crash while attempting to access a remote repository. Additionally, a bug in Subversion 1.4.x sometimes makes it impossible to determine

whether files in a deleted directory have been committed, potentially resulting in incorrect status listings in the Checkout Browser.

Missing Version Information for libxml2.so.2

You may encounter the following error message when trying to use version control on a Linux host:

```
[path]/libxml2.so.2: no version information available
(required by ...)
```

Workaround: Remove the **libxml2.so** files that ship with MULTI and replace them with a symbolic link to the system-installed **libxml2.so.2**.

Target Connection Issues

Serial Ports Supported for Use with rtserv

On Windows, **rtserv** can only connect to serial ports with single-digit COM numbers (that is, serial ports **COM1** through **COM9**). Serial ports with two or more digits (**COM10**, for example) are not supported.

Debugging Issues

Source Pane Displays Unexpected File

If your project nests identical source trees and you attempt to navigate to or launch a particular file from the inner source tree (for example, `(1)` in the following project), the Debugger's source pane may erroneously display a file from the outer source tree (for example, `(2)` in the following project).

```
\_ nested/
| \_ src/
| | \_ file.c (1)
\_ src/
| \_ file.c (2)
```

Workaround:

1. Navigate to:
   - Windows 7/Vista — ***user_dir*\AppData\Roaming\GHS\Workspaces**
   - Windows XP — ***user_dir*\Application Data\GHS\Workspaces**
   - Linux/Solaris — ***user_dir*/.ghs/Workspaces/**
2. Delete the file that corresponds to the program you are debugging when this problem occurs. Alternatively, delete all the files in the **Workspaces** directory. (Note that this does not delete any project workspaces, only saved history information that the Debugger uses when executing history commands, jumping to previous or subsequent source pane views, etc.)

Only users of a previous version of MULTI may encounter this problem.

Maximum Number of Attached Tasks

The number of tasks that the Debugger can attach to at one time is limited by the number of file descriptors that can be opened by a process in the operating system. On most Linux systems, the limit is approximately 1000. On Windows, the limit is approximately 500.

Up to 65534 Macros Supported

If a program defines more than 65534 distinct macros, MULTI may fail to evaluate expressions containing those macros.

Limitation of Browsing Traced Memory Accesses

You can use the **tracebrowse** command in MULTI to view all reads and writes to a particular variable. However the **tracebrowse** command only displays the memory accesses of the first four bytes of the data type. Therefore any structure, array, or class that is bigger than four bytes will only have the reads and writes to the first four bytes displayed.

Limitation of Browsing Traced Function Calls

The Trace Call Browser allows you to view all calls of a particular function and view the minimum, maximum, mean, and total duration of calls to that function. When trace data is discarded to free disk space for new data, the function calls contained in the discarded trace data are removed from the list of calls in the Trace Call Browser. The Min, Max, Mean, and Total statistical values listed in the Trace Call Browser, however, are cumulative and take account of the discarded calls until you clear the trace data. You can clear the trace data by entering the `trace clear` command in the Debugger command pane or by selecting **TimeMachine → Clear Data** from the Debugger.

Cannot Retrieve Trace During a Blocking Run or Step

If trace retrieval is initiated while a blocking run or step operation is in progress in the Debugger, the retrieval is delayed until the blocking run or step completes. If trace retrieval is initiated while a blocking run or step operation is in progress in the TimeMachine Debugger, the retrieval is aborted. For information about blocking, see the description of the **b** parameter in "Single-Stepping Commands" in Chapter 13, "Program Execution Command Reference" in the *MULTI: Debugging Command Reference* book.

Incompatibility Between MULTI 4.x and MULTI 6+ Trace Configuration Files

The trace configuration file **trace.cfg** created by MULTI 6.0 and later is not compatible with MULTI 4.x.

Workaround: To use both MULTI 4.x and MULTI 6.0 or later on the same system:

1. Delete the **trace.cfg** file (if it exists) from:
   - Windows 7/Vista — ***user_dir*\AppData\Roaming\GHS**
   - Windows XP — ***user_dir*\Application Data\GHS**
   - Linux/Solaris — ***user_dir*/.ghs**
2. In the MULTI 4.x Debugger, select **Tools → Trace → Trace Options**.
3. In the **Trace Options** window that appears, click the **Save as Default** button.

You will only need to do this once.

Limitations of UNC Paths on Windows

Debugging projects from an INTEGRITY distribution loaded via a UNC path is not supported. Mapping the UNC path to a network drive will allow you to debug the project.

Limitations for Debugging Large Enumeration Types

The Debugger incorrectly truncates the value of an enumeration constant or of a variable of an enumeration type if the value requires more than 32 bits for representation.

Limitation of Mixing PIC or PID Code with ABS Sections

When evaluating addresses of code compiled with **-pic** or **-pid**, the Debugger first subtracts the `_TEXT` or `_DATA` offset and then determines the text or data symbol at the resulting address. If an `ABS` section happens to exist at the same address, the Debugger may incorrectly choose the symbol from the `ABS` section rather than from the PIC or PID section. This address-to-symbol mapping most frequently occurs when the Data Explorer displays addresses that point to global data such as constant strings. In this case, the Data Explorer may incorrectly name the wrong symbol as pointing to the data.

Debugging AltiVec-Specific Data Types

Power Architecture only

From the MULTI Debugger, you cannot modify the values of AltiVec-specific data types (such as vector) or cast variables to or from these types to another type. You can, however, display the values of AltiVec data types in expressions and in the Data Explorer.

Viewing Long Doubles on x86 Systems

x86 only

When viewing long doubles on native or embedded x86 systems, be advised that the Debugger rounds the value to double size before displaying it, so some precision is lost.

Some Tasks Do Not Have Associated AddressSpaces in the EventAnalyzer

Versions of **rtserv2** prior to v2.0.32 may not provide the MULTI EventAnalyzer with an AddressSpace name for certain tasks when the combined task name length and AddressSpace name length exceed 32 bytes. As a result, the MULTI EventAnalyzer may display these tasks without an associated AddressSpace.

Workaround:

- Shorten the name of your task or AddressSpace.
- Give your tasks unique names so that the AddressSpace association is unnecessary.
- Upgrade **rtserv2**.

Compatibility with INTEGRITY

INTEGRITY 10 Installer Screen Ignored by MULTI 6

The INTEGRITY installer's **Identify Tools Distribution** screen is used to provide the MULTI IDE with the location of the installed INTEGRITY distribution. If you are using INTEGRITY 10 and MULTI 6, the setting on this screen has no effect.

Workaround: Ignore the INTEGRITY installer's **Identify Tools Distribution** screen and the corresponding INTEGRITY documentation, and instead follow the instructions at “Configuring MULTI for Use with INTEGRITY or u-velOSity” in Chapter 2, “MULTI Tutorial” in the *MULTI: Getting Started* book.

Limitations for Loading and Unloading Applications on INTEGRITY 5.x

INTEGRITY 5.x does not support loading or unloading multiple applications at the same time, even though this can be performed through the MULTI Debugger's target list. If you attempt to load or unload multiple applications, the operation will only succeed on one of the applications.

You must wait for the previous load or unload to finish before another can be performed successfully. A load or unload has been completed successfully when the application's status changes in the target list to **Loaded** (for a load) or **Not Loaded** (for an unload).

Note that operations such as preparing your target, running, stepping, or restarting may also cause a module to become loaded. When this occurs, these operations are subject to the same restrictions.

Collecting Memory Leak Information on INTEGRITY 5.x

Collecting memory leak information when halted in the INTEGRITY kernel's `IdleTask()` is not supported and may appear to run without finishing. Pressing the **Esc** key will abort collection.

Workaround: To collect memory leak information for the kernel space `malloc` heap, you must halt your target in a C-library-enabled kernel task.

RunTask May Not Have an Effect on Dynamically Downloaded Tasks

When MULTI does a dynamic download, MULTI may automatically attach to all statically defined tasks within the downloaded application as soon as they have been loaded. INTEGRITY version 5.2.1 and INTEGRITY version 5.0.7 and earlier did not allow a task to do a `RunTask` on a halted task that MULTI was attached to. This means that downloading an INTEGRITY application may result in that application being unable to start some of its own tasks.

Workaround: Run halted tasks from the Debugger.

MULTI RTSERV Auto-Connect and High Priority Tasks

If you are debugging an INTEGRITY 5.0 application with high priority tasks such that the `Idle` function will never run, MULTI will not be able to automatically start an **rtserv** connection, even if one has been requested with the Run-Mode Partner dialog or with the **set_runmode_partner** command.

Workaround: Manually establish an **rtserv** connection. It will automatically be partnered with the corresponding stop-mode connection.

Run-Mode Partnering Not Supported with -checksum

Making an automatic run-mode partner connection involves setting a breakpoint in the target kernel, which can disrupt the proper functioning of the **-checksum** build option.

Halting or Attaching to Task Cancels a System Call with rtserv

If a task is in the middle of a host I/O system call with **rtserv** (for example, printing to the MULTI Debugger's **I/O** pane via `printf()`) and the task is halted either directly or by virtue of attaching the Debugger to the task, the system call will not complete successfully. This is most likely to occur when the task is blocked in a host I/O call, reading from standard input or from a file on the host. If you are using **rtserv2** (INTEGRITY 10 or later), halting a task does not interfere with host I/O.

System Halt with Host I/O May Make Target Unresponsive

When connected with **rtserv** to a target running INTEGRITY version 5.2.1 or INTEGRITY version 5.0.8 or earlier, and opening the OSA Explorer or otherwise invoking a System Halt, and a Host I/O operation completes after the system is halted, the target can become unresponsive to **rtserv**. This can most commonly be triggered by a target program printing to standard output immediately before the system is halted. This problem is resolved in INTEGRITY versions after 5.0.8.

MULTI-Python Integration Issues

mpythonrun Unable to Output to stdout/stderr in Cygwin

In a Cygwin environment, **mpythonrun** is unable to output data to `stdout` or `stderr`. If it attempts to do so, a dialog box appears with an error message similar to the following:

```
WriteConsole(handle=0x17 for STD_OUT_HANDLE) failed: ...
```

Clicking **OK** in the dialog box continues the execution of **mpythonrun**.

Workaround: To work around this problem, which is caused by a bug in Cygwin, redirect output to another file, or use the Windows command console or KornShell.

File System Issues

Tilde Expansion in Cygwin

Cygwin expands the tilde symbol (~) to the value of the `HOME` environment variable as set in Cygwin. If `HOME` is set to a Cygwin path, attempts to launch non-Cygwin programs (including MULTI applications) on files whose path includes a tilde will fail.

Workaround: To work around this limitation, set the `HOME` environment variable in Cygwin to a Windows path.

Two Dots (..) Not Supported After Symbolic Links in Paths

MULTI and other Green Hills tools do not support paths that use two dots (`..`) to indicate the parent directory of a symbolic link. For example, if `symlink` is a symbolic link, the following paths are not supported:

```
/projects/symlink/..
/projects/symlink/foo/../../bar
```

NFS May Cause Poor Performance

This release note applies to Linux/Solaris hosts only.

MULTI tools may hang if they are accessing files over an NFS server that is slow or is not responding. In certain cases, it may not be possible to terminate them, even when using `kill -9`. These problems are caused by a fundamental limitation in the design and implementation of NFS.

Workaround: To help prevent these problems, eliminate references to broken file system mount points in your user configuration directory (**~/.ghs/***) — in particular in the **integrity.dist** and/or **uvelosity.dist** files located there.

Desktop Environment Issues

Numerous Instances of MULTI Can Exhaust the Windows Desktop Heap

If you are using Windows and if many graphical processes are open simultaneously, the desktop heap can be exhausted, preventing additional graphical applications from being launched. This problem is more likely to occur on Windows XP than on Windows Vista, which defaults to a larger desktop heap size.

Workaround: If you encounter this problem, try reusing existing MULTI IDE graphical processes instead of launching new ones. For example, to reuse existing MULTI Editor processes, open new Editors by issuing the MULTI Debugger **edit** command or by selecting **File → New Editor** from the MULTI Editor menu bar.

If you continue to encounter problems, see the Microsoft Knowledge Base for information about increasing the desktop heap size.

Locking Assertion Failure May Occur within libxcb

When running the MULTI IDE, you may encounter the following message at the console, possibly accompanied by part of the IDE exiting abnormally:

```
Locking assertion failure. Backtrace:
#0 /usr/lib/libxcb-xlib.so.0 [0xb7cfc767]
#1 /usr/lib/libxcb-xlib.so.0(xcb_xlib_lock+0x2e) [0xb7cfc81e]
...
```

This problem is caused by a known bug within the libxcb X transport binding used in some newer Linux distributions. GHS encountered this problem very infrequently during testing. If you observe it often, please contact Green Hills Support.

Taskbar Organizer and Windows 7 with Aero Peek

The Taskbar Organizer in taskbar mode is not compatible with Windows 7's Aero Peek feature: moving the mouse over the MULTI entry in the taskbar displays a white bar instead of showing window thumbnails. For information about how to disable the Taskbar Organizer, see "Configuring Taskbar Organization" in Chapter 7, "Configuring and Customizing MULTI" in the *MULTI: Managing Projects and Configuring the IDE* book.

Taskbar Organizer and nVidia Desktop Manager

The Taskbar Organizer setting to display only the Taskbar Organizer in the **Alt**+**Tab** list may not work correctly when using nVidia Desktop Management software.

Workaround: If all windows are displayed, including the Taskbar Organizer, perform the following steps:

1. Right-click the desktop and select nView Properties.
2. Click the **User Interface** tab.
3. Clear **Enable nView task switcher**.
4. Click **OK** to save your changes.

nVidia X Server With Compiz Window Manager May Crash

When using MULTI with the Compiz window manager on Linux, opening windows may cause the X server to crash. This appears to be caused by a bug which is present in version 169.12 of the nVidia driver.

To see the current version of the nVidia drivers being used, run `cat /proc/driver/nvidia/version` from the command line.

Workaround: Upgrade to version 173.14.12 or newer of the nVidia driver. For information about obtaining and installing new drivers, consult your distribution documentation or visit nVidia's Web site.

Alternatively, you can use a different window manager, such as Metacity.

Drawing problems with Exceed 2008 (13.0)

You may experience problems in the MULTI Editor and Debugger command pane where text may not draw correctly when used within Exceed 2008 (version 13.0).

Workaround: Disable Exceed's **Batch Requests** option. This can be found in Exceed's configuration window under **Other Server Settings**.

SELinux Restricts Dynamically Loaded Code

If SELinux is set to `enforcing`, it restricts dynamically loaded code to trusted modules. In enforcement mode, you may see a dialog box with contents similar to the following:

```
MULTI failed to load cpudetect.so:
cannot restore segment prot after reloc: Permission denied
```

Workaround: Temporarily disable enforcement with the following command:

```
/usr/sbin/setenforce 0
```

Alternatively, try registering the library:

```
chcon -t  texrel_shlib_t svc_cpu_ppc_so.so
chcon -t  texrel_shlib_t cpudetect.so
...
```