

MULTI: IDE Version 7 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

LEGAL NOTICES AND DISCLAIMERS

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.

Copyright © 1983-2016 by Green Hills Software. All rights reserved. No part of this publication may be reproduced, stored in a retrieval system, or transmitted, in any form or by any means, electronic, mechanical, photocopying, recording, or otherwise, without prior written permission from Green Hills Software.

Green Hills, the Green Hills logo, CodeBalance, GMART, GSTART, INTEGRITY, MULTI, and Slingshot are registered trademarks of Green Hills Software. AdaMULTI, Built with INTEGRITY, EventAnalyzer, G-Cover, GHnet, GHnetLite, Green Hills Probe, Integrate, ISIM, u-velOSity, PathAnalyzer, Quick Start, ResourceAnalyzer, Safety Critical Products, SuperTrace Probe, TimeMachine, TotalDeveloper, DoubleCheck, and velOSity are trademarks of Green Hills Software.

All other company, product, or service names mentioned in this book may be trademarks or service marks of their respective owners.

For a listing of Green Hills Software's periodically updated patent marking information, please visit http://www.ghs.com/copyright_patent.html.

PubID: multi_release_notes-656015

Branch: <http://toolsvc/branches/release-branch-70>

Date: August 16, 2019

Contents

1. Changes in MULTI 7	1
System Requirements	2
Windows	2
Linux	3
Solaris	4
Product Compatibility	5
New Features and Enhancements	5
History	5
INTEGRITY TimeMachine	6
Debug Snapshot	6
Synchronous Run Control	6
Profile Window	7
Project Build Configurations	7
INTEGRITY Monolith Debugging	8
Simplified Multi-Core Configuration	8
Automatic Setup of Certain Multi-Core Targets	8
Choose Which Target List Entries are Displayed	8
Choose Which Data Explorer Entries are Displayed	9
Run to the Next/Previous Modification of a Variable	9
Set Watchpoints from the Data Explorer	9
Quickly Navigate Complex Data Structures in the Data Explorer	10
Data Explorer Column Settings Remembered Within and Across Sessions	10
Commands to Time Target Operations	10
Command to Control Breakpoint Installation After Target Setup	10
Command to List and Optionally Attach to OSA Tasks	10
Command to Clean up OSA Information or Remove OSA Support	11
Command to Modify Target-Specific Mode Variables	11
Added Options to the wait Command for Scripting	11

Added Options to the load, unload, and prepare_target	
Commands	12
Reduced Trace Disk Usage and Trace Session File Sizes	12
Tracing 64-bit INTEGRITY Targets Now Supported	12
Further Parallelization of Trace Processing	12
Faster Reporting of Memory Leaks and Allocations	12
Line Numbers Now Displayed in the Editor	12
Display of Task Priority, Stack Usage, and Stack Size	13
Long Target List Entries Now Easy to Tell Apart	13
Improved Target List Display of OSA Tasks on Certain Multi-Core	
Targets	13
New Register access Type	13
Python Interpreter Upgraded to 2.7.3	14
Subversion 1.7.x and 1.8.x Supported	14
Multiple TCP Ports Supported per License Client	14
Simpler License Manager Configuration	14
Changes in Behavior	14
Any-Task Breakpoints Set By Default in INTEGRITY	14
Watchpoints Implemented Using Any-Task Hardware	
Breakpoints	15
Target List Located on Left By Default	15
New Methods for Manually Dumping and Processing Profiling	
Data	15
Several Profiling Commands Consolidated Under the 'Profile'	
Command	15
Continual Update of Profile Window No Longer Available	16
Data Explorer No Longer Treats Instance Variables as Local	
Variables	16
Data Explorer and 'print /t' Now Display Variable typedef	
Names	17
Run-Mode Partnering Now Supported with -checksum	17
Debugger Attaches to Tasks that Hit Any-Task Hardware	
Breakpoint	17
Register Information Now Shared	17
Non-INTEGRITY TimeMachine No Longer Accessible from Run	
Mode	18
Child Windows Are Displayed in Front of the Debugger on Windows	
OS	18

Improved `!` Command Output	18
Issue with Windows 10 Dongle-Locked Licenses Resolved	19
Changes to Terminology Used in the Documentation	19
Stricter Error Checking on Invalid Pointer Math Expressions	19
Deprecated Features	19
Removed Features	20
 2. Changes in MULTI 6.1.6	 21
New Features and Enhancements	22
Support for Large Virtual Function Tables	22
Support for Freeze-Mode Debugging of INTEGRITY	22
 3. Changes in MULTI 6	 23
New Features and Enhancements	24
Faster Builds	24
Ability to Retrieve More Trace Data from New SuperTrace	
Probes	24
Faster Trace Processing	24
Separate Releases Allow Greater Flexibility When Upgrading	25
64-Bit Support in the Debugger	25
128-Bit Hardware Registers Supported	25
Security and Relationship Views Added to Integrate	25
Local Variable View Updates as You Navigate the Call Stack	25
Ability to Restore Breakpoints	26
Greater Window Availability	26
New System Variables	26
Target List Remembers Node Expansion/Collapse	26
Sortable Columns in the Target List	26
Improved Support for Preprocessor Macros in the Debugger	27
Browsing References Improvement	27
if Command Supports else if	27
Licensing Improvements	27
Faster Scanning of Subversion Checkouts	28
Changes in Behavior	28
IDE and Compiler Installed into Separate Linking Directories	28
Upgrading to MULTI 6	28
License Clients No Longer Broadcast for a License Manager	28

Changes to Computer- and Dongle-Locked Licensing	28
Changes to Trace Retrieval	29
Trace Decoding No Longer Defaults to Reading Unknown Opcodes	29
Changes to TS_Packet May Require TimeMachine API Application Updates	29
Behavior of Primary and Secondary Debugger Windows	29
Large Numbers No Longer Broken Up by Default	30
Default Number of Scroll Back Lines Reduced	30
Storage of PID and TID Debugger Aliases	30
Deprecated Features	31
Removed Features	31

4. Changes in MULTI 5.0.6 33

New Features and Enhancements	34
MULTI Project Manager	34
Flexible Target Connections	34
Improved Target Management from the Debugger	34
Target Traffic Pane	35
TimeMachine Enhancements	35
Improved Kernel Object Awareness	36
Global Profiling	36
Integrated Python Engine	36
Extended Version Control Support	37
Windows Taskbar Organization	37
Improved Help Navigation	37
Changes in Behavior	37
Use of Dongle-Locked Licenses over Remote Desktop	37
Halt Issued Before Setup Script in Pre-5.0.6 Versions of MULTI	37
Unbundled Features	38
Deprecated Features	39
Removed Features	39

5. Known Issues	41
Host Support	42
IDE Components May Fail to Launch on Unsupported Linux Distributions	42
MULTI Installation	42
Name of Install Directory Cannot Contain Spaces	42
Project Manager Issues	42
Project Manager Performance on Very Large Project Files	42
Undefined <code>__OS_DIR__</code> Macro When Opening Some ThreadX Projects	43
Projects with Build Configurations Incompatible with MULTI 6.x and Earlier	43
Editor Issues	43
Limitation of Brace Matching with Preprocessor Directives	43
File Permissions Under Cygwin	43
Version Control Issues	44
Subversion Authentication on Windows	44
Problems Observed with Subversion 1.4.x	44
Missing Version Information for libxml2.so.2	44
Target Connection Issues	45
Serial Ports Supported for Use with rtserv	45
Debugging Issues	45
Source Pane Displays Unexpected File	45
Maximum Number of Attached Tasks	46
Up to 65534 Macros Supported	46
Very Large Files Cause Slow Response Times when Single-/Double-Clicking Variables	46
Limitation of Browsing Traced Memory Accesses	46
Limitation of Browsing Traced Function Calls	47
Cannot Retrieve Trace During a Blocking Run or Step	47
Trace Configuration Files Incompatible with MULTI 4.x	47
Limitations of UNC Paths on Windows	48
Limitations for Debugging Large Enumeration Types	48
Limitation of Mixing PIC or PID Code with ABS Sections	48
Debugging AltiVec-Specific Data Types	48

Viewing Long Doubles on x86 Systems	48
`update <interval>' Not Supported with INTEGRITY	
TimeMachine	49
Any-Core Hardware Breakpoints Not Applied to TimeMachine ...	49
Limitations of History From RH850 Software Trace	49
Quitting During Command Line Function Call May Leave Program in Undefined State	50
Daylight Saving Time May Result in Erroneous Statement (Windows Only)	50
Compatibility with INTEGRITY	50
MULTI Ignores INTEGRITY 10 Installer Screen	50
Limitations for Loading and Unloading Applications on INTEGRITY 5.x	51
Collecting Memory Leak Information on INTEGRITY 5.x	51
RunTask May Not Have an Effect on Dynamically Downloaded Tasks	51
MULTI RTSERV Auto-Connect and High Priority Tasks	52
Halting or Attaching to Task Cancels a System Call with rtserve ...	52
System Halt with Host I/O May Make Target Unresponsive	52
Run-Mode Breakpoints May Confuse Trace	52
MULTI-Python Integration Issues	53
mpythonrun Unable to Output to stdout/stderr in Cygwin	53
ctypes and Tcl/Tk Not Included in MULTI for Solaris	53
MULTI-Python Integration on Linux May Require Compatibility Library	53
File System Issues	54
Tilde Expansion in Cygwin	54
Two Dots (..) Not Supported After Symbolic Links in Paths	54
NFS May Cause Poor Performance	54
Desktop Environment Issues	55
Numerous Instances of MULTI Can Exhaust the Windows Desktop Heap	55
Locking Assertion Failure May Occur within libxcb	55
Taskbar Organizer and Windows 7 with Aero Peek	55
Taskbar Organizer and nVidia Desktop Manager	56
Windows Do Not Dock Completely	56
nVidia X Server With Compiz Window Manager May Crash	56
Drawing problems with Exceed 2008 (13.0)	57

Nouveau Drivers for nVidia Graphics Cards May Cause Drawing Problems	57
SELinux Restricts Dynamically Loaded Code	58
Font Display Problems with Ubuntu 12.04 and Xvnc	58
MULTI May Lose X Connection After Running for Several Hours	59

Chapter 1

Changes in MULTI 7

Contents

System Requirements	2
Product Compatibility	5
New Features and Enhancements	5
Changes in Behavior	14
Deprecated Features	19
Removed Features	20

System Requirements



Note

Your Green Hills Compiler may have different system requirements. For more information, see the Compiler release notes.

Windows

Running a full MULTI 7 installation on Windows requires:

- A modern, multi-core x86 processor
- One of the following:
 - Windows 10 (64- or 32-bit mode)
 - Windows 8 or 8.1 (64- or 32-bit mode)
 - Windows 7 (64- or 32-bit mode)
 - Windows Vista (64- or 32-bit mode). On Windows Vista, .NET Framework 3.5 and/or 4.5 is required. .NET Framework 4.5 requires Windows Vista Service Pack 2 or greater.



Tip

.NET Framework 3.5 may already be installed on Windows Vista through Windows Update.

- 4 GB of RAM (16 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 1 GB available on the primary drive for temporary files and DLLs. (100 GB of free space and a solid state drive are recommended.)
- A monitor running at a display resolution of 1024x768 or higher. (Two high-resolution monitors are recommended.)
- A DVD-ROM drive or access to the Green Hills FTP site. If neither is available, contact Green Hills Support.
- To install, you must be a member of the Administrators group (or have access to the Windows System directories).

Linux

Running a full MULTI 7 installation on Linux requires:

- A modern, multi-core x86 processor
- One of the following:
 - Ubuntu 16.04.1 LTS (64- or 32-bit mode)
 - Ubuntu 14.04.1 LTS (64- or 32-bit mode)
 - Ubuntu 12.04.5 LTS (64- or 32-bit mode)
 - CentOS 7.x (64- or 32-bit mode)
 - CentOS 6.x (64- or 32-bit mode)

32-bit compatibility libraries are required when running on 64-bit systems.

If you are using Ubuntu 14 or later, run:

```
sudo dpkg --add-architecture i386
sudo apt-get update
sudo apt-get install libc6:i386 libncurses5:i386
sudo apt-get install libstdc++6:i386 libx11-6:i386 lib32z1
sudo apt-get install libxcursor1:i386
```

If you are using Ubuntu 12, run:

```
sudo apt-get install ia32-libs
```

If you are using CentOS 6.0 or earlier (or Red Hat Enterprise Linux 6.0 or earlier), run:

```
sudo yum install glibc.i686 libX11.i686 zlib.i686 libstdc++.i686
```

If you find that the tools are not working, especially because of a licensing issue, you may need to install additional 32-bit compatibility libraries. For example, if you are using LDAP with SSS, you must install **libnss_sss**, which is usually part of the **sssd-client** package.

- 4 GB of RAM (16 GB is recommended.)
- 2 GB of free disk space per installation. (100 GB of free space and a solid state drive are recommended.)

- A monitor running at a display resolution of 1024x768 or higher. (Two high-resolution monitors are recommended.)
- A graphical display running the X Window System
- A mounted DVD-ROM drive or access to the Green Hills FTP site. If neither is available, contact Green Hills Support.
- An Ethernet connection
- Write permissions to the installation directory

Solaris

Running a MULTI 7 installation on Solaris requires:

- A sun4u SPARC-based workstation
- One of the following:
 - Solaris 10 (64- or 32-bit mode)
 - Solaris 9 (64- or 32-bit mode)
- 2 GB of physical memory (4 GB is recommended.)
- 2 GB of free disk space per installation
- A graphical display running the X Window System
- A mounted DVD-ROM drive or access to the Green Hills FTP site. If neither is available, contact Green Hills Support.
- An Ethernet connection
- Write permissions to the installation directory



Note

History, INTEGRITY TimeMachine, and the debug snapshot capability are not supported on Solaris.

Product Compatibility

The MULTI 7 IDE is officially supported with:

- Green Hills Compiler versions 2012.1 through 2017.1
- INTEGRITY IoT versions 2016.0 and 2017.0
- INTEGRITY versions 5.x, 10.0.2, 11.0.2, 11.0.4, 11.2.4, and 11.4.4
- u-velOSity versions 2.2.9 and 2.6.4

For information about compatibility between the MULTI 7 IDE and newer versions of these products, see the release notes for the newer products.



Warning

Using MULTI 7 with a Green Hills Debug Probe requires Green Hills Debug Probe software version 4.4.4 or later. Do not use Green Hills Debug Probe software version 4.2.x or earlier with your MULTI 7 installation.

New Features and Enhancements

History

MULTI 7 includes a new debugging and performance analysis tool called History, which provides a visual, time-based overview of your system. You can use History to:

- Fix bugs
- Improve performance
- Gain a better understanding of your system

For more information, see Chapter 23, “Viewing System Data with History” in *MULTI: Debugging*.

INTEGRITY TimeMachine

MULTI 7 introduces the INTEGRITY TimeMachine Debugger, which, like the TimeMachine Debugger, allows you to debug backwards in your code. The new INTEGRITY TimeMachine Debugger:

- Provides full OS integration, with awareness of kernel events and shared memory
- Works with run-mode connections to give you a smoother and faster debugging experience
- Instantaneously transitions into INTEGRITY TimeMachine mode when you step or run back
- Does not depend on hardware trace data, which means that data values are always available, data is never lost, and you are not limited to using CPUs and boards with trace capabilities

For more information, see Chapter 19, “Debugging with INTEGRITY TimeMachine” in *MULTI: Debugging*.

Debug Snapshot

MULTI 7 gives you the ability to save a debug snapshot to a file on your host machine and view the saved snapshot in the Debugger without a target connection. This is especially useful if you want to share a debugging session with another engineer, or if you need to repurpose your hardware before you are finished debugging a problem.

For more information, see Chapter 24, “Saving a Debug Snapshot” in *MULTI: Debugging*.

Synchronous Run Control

With a run-mode connection to INTEGRITY IoT, halting one task for debugging purposes causes other tasks in the system to stop in a non-invasive pause. Stepping a halted task or resuming the system causes paused tasks to become schedulable again. This means that interactions between tasks are minimally impacted by debugging. For more information, see “Stopping and Starting Tasks Synchronously”

in Chapter 8, “Executing and Controlling Your Program from the Debugger” in *MULTI: Debugging*.

With a supported freeze-mode connection to a multi-core hardware target, all cores are run or halted at the same time. Halting all cores simultaneously means that you can debug one core without worrying that operations running on another core will affect shared memory. For more information, see “Synchronous Debugging on a Multi-Core Target” in Chapter 25, “Freeze-Mode Debugging and OS-Awareness” in *MULTI: Debugging*.

Profile Window

The **Profile** window and many profiling features have been redesigned to make it easier than ever to:

- Find performance problems
- Find coverage gaps and track test coverage of specific code
- Share profile data with other users
- Collect and merge profile data from several tests or profile recordings

For more information, see Chapter 17, “Recording and Viewing Profiling Data” in *MULTI: Debugging*.

Project Build Configurations

Projects can now define multiple build configurations. Each of these configurations sets different Builder and driver options so that you can build one project in different ways. For example, a debug configuration might set options that configure a project for maximum debugging ability, while a release configuration would optimize for speed and size. Project build configurations are supported with Compiler 2015.1 and newer. For more information, see “Displaying Project Build Configurations” in Chapter 2, “Managing and Building Projects with the Project Manager” in *MULTI: Managing Projects and Configuring the IDE*.

INTEGRITY Monolith Debugging

Because many INTEGRITY systems in final deployment incorporate a monolith application, MULTI 7 contains special support for run-mode debugging of INTEGRITY monolith applications. When you open an INTEGRITY monolith in the Debugger, the run-mode virtual AddressSpaces and Tasks defined statically in that monolith are shown in the Debugger even before the kernel has booted, and a subset of run-mode debugging features are available. For more information, see “Working with INTEGRITY Monolith Applications” in Chapter 22, “Run-Mode Debugging” in *MULTI: Debugging*.

Simplified Multi-Core Configuration

The MULTI 7 Debugger can use information you provide in a multi-core configuration file (*.ghsmc) to automatically download one or more executables to your multi-core target and load symbol files in the Debugger. This simplifies target setup considerably. For more information, see “Using a Multi-Core Configuration File to Load Executables onto a Multi-Core Target” in Chapter 25, “Freeze-Mode Debugging and OS-Awareness” in *MULTI: Debugging*.

Automatic Setup of Certain Multi-Core Targets

If you are running an INTEGRITY application on an SMP multi-core target with the new synchronous debugging feature enabled, you can debug the application just like you would on a single-core target: the Debugger automatically runs the target setup script (if any), downloads the application on the first core of your multi-core target, and makes the application's symbol information visible on all cores. For more information, see “Synchronous Debugging of an INTEGRITY Application on an SMP Target” in Chapter 25, “Freeze-Mode Debugging and OS-Awareness” in *MULTI: Debugging*.

Choose Which Target List Entries are Displayed

You can now choose which entries are displayed in the target list. Click the star alongside each entry of interest, and then click the **Only show starred items** button (★), located to the right of the **Status** column. This button toggles the exclusive display of starred entries, their ancestors, the currently selected target list entry, and

stopped entries. This feature is especially useful when your system has a large number of target list entries, but you only want to view a small number of them. For more information, see “The Star Column” in Chapter 2, “The Main Debugger Window” in *MULTI: Debugging*.

Choose Which Data Explorer Entries are Displayed

You can now choose which entries are displayed in the Data Explorer. Click the star alongside each item of interest, and then click the **Only show starred items** button (★) to toggle the exclusive display of starred items and their ancestors and descendants. This feature is especially useful if you are debugging an application that involves data structures with many fields, or complex functions with many local variables. For more information, see “Filtering Data Explorer Variables” in Chapter 11, “Viewing and Modifying Variables with the Data Explorer” in *MULTI: Debugging*.

Run to the Next/Previous Modification of a Variable

In MULTI 7, you can run to the next or previous modification of a variable from the Debugger source pane, the Data Explorer, or the command pane. For more information, see the descriptions of **Run to Next Modification** and **Run Back to Previous Modification** in “The Tools Menu” in Chapter 11, “Viewing and Modifying Variables with the Data Explorer” in *MULTI: Debugging*. Or see the same descriptions in “The Variable Shortcut Menu” in Appendix A, “Debugger GUI Reference” in *MULTI: Debugging*.

Set Watchpoints from the Data Explorer

The new **WP** column in the Data Explorer allows you to set, toggle, and delete watchpoints—hardware breakpoints set on a variable's address that stop your program when the address is modified. For more information, see “Setting Watchpoints on Data Explorer Variables” in Chapter 11, “Viewing and Modifying Variables with the Data Explorer” in *MULTI: Debugging*.

Quickly Navigate Complex Data Structures in the Data Explorer

Context-sensitive arrows in the **Variable** column of the Data Explorer now allow you to quickly navigate complex data structures. Right-facing arrows reroot on child nodes, and left-facing arrows reroot on parent nodes. For more information, see “Navigating Complex Data Structures” in Chapter 11, “Viewing and Modifying Variables with the Data Explorer” in *MULTI: Debugging*.

Data Explorer Column Settings Remembered Within and Across Sessions

The Data Explorer now remembers column visibility and ordering within and across sessions.

Commands to Time Target Operations

You can use the new **starttime** and **endtime** commands to quickly get an estimate of how long a target operation takes. For more information, see the descriptions of **starttime** and **endtime** in Chapter 2, “General Debugger Command Reference” in *MULTI: Debugging Command Reference*.

Command to Control Breakpoint Installation After Target Setup

You can use the new **install_bp_on_request** command in setup scripts to control precisely when breakpoints are installed. This is useful if, during the initialization of your target, you need to execute code on the target while it is not safe to install breakpoints. For more information, see “install_bp_on_request” in Chapter 3, “Breakpoint Command Reference” in *MULTI: Debugging Command Reference*.

Command to List and Optionally Attach to OSA Tasks

Like the **osaexplorer** command, the new **osatasklist** command provides information about OSA tasks; however, it typically accesses less target memory than **osaexplorer**. Additionally, it allows you to debug tasks in freeze mode. For more information about this command, see “osatasklist” in Chapter 15, “Scripting Command Reference” in *MULTI: Debugging Command Reference*.

Command to Clean up OSA Information or Remove OSA Support

The new **osacleanup** command cleans up OSA information or removes OSA support. This lessens or eliminates the effect of the OSA package on system performance, thereby speeding up operations like single-stepping. For more information, see “osacleanup” in Chapter 15, “Scripting Command Reference” in *MULTI: Debugging Command Reference*.

Command to Modify Target-Specific Mode Variables

The new **asmmode** command changes target-specific mode variables, which may affect the behavior of the disassembler and/or other target-specific behavior in the Debugger. The primary use for this is to set a specific instruction set mode for disassembly (such as Thumb when using ARM). Normally MULTI determines this from debug information, so it should only be necessary when looking at instructions that have no debug information. For more information, see “asmmode” in Chapter 8, “Display and Print Command Reference” in *MULTI: Debugging Command Reference*.

Added Options to the wait Command for Scripting

Several new options are available for use with the **wait** command, which blocks command processing until a specified action has occurred. New options are:

- **-timeoutcmd {commands}** — Executes *commands* if a timeout occurs before the specified action.
- **-pause** — Blocks command processing until the system is paused.
- **-debugsnapshot** — Blocks command processing until the debug snapshot save is complete.
- **-load** — Blocks command processing until all dynamic downloads for the current connection are complete.
- **-unload** — Blocks command processing until the last module to be unloaded is fully unloaded from the target.

For more information, see “wait” in Chapter 2, “General Debugger Command Reference” in *MULTI: Debugging Command Reference*.

Added Options to the load, unload, and prepare_target Commands

New options to the **load**, **unload**, and **prepare_target** commands allow you to block command processing until a load/unload operation is complete or until a specified timeout has occurred. For more information, see the documentation for **load**, **unload**, and **prepare_target** in “General Target Connection Commands” in Chapter 17, “Target Connection Command Reference” in *MULTI: Debugging Command Reference*.

Reduced Trace Disk Usage and Trace Session File Sizes

MULTI 7 dramatically reduces trace disk usage and trace session file sizes.

Tracing 64-bit INTEGRITY Targets Now Supported

MULTI 7 adds support for tracing 64-bit INTEGRITY targets.

Further Parallelization of Trace Processing

Trace processing has been further parallelized in MULTI 7. This results in a 20-30% speedup in trace processing for certain targets when you are decoding on a host PC with at least 4 cores.

Faster Reporting of Memory Leaks and Allocations

If you are using MULTI 7 and Compiler 2014.1 or later, reporting of many memory leaks or allocations can be over 30 times faster than in previous releases.

Line Numbers Now Displayed in the Editor

By default, line numbers are now displayed in the Editor to make it easy to shift from looking at code in the Debugger to code in the Editor. For information about the configuration option that controls the appearance of line numbers—**Show line number**—see “The More Editor Options Dialog Box” in Chapter 8, “Configuration Options” in *MULTI: Managing Projects and Configuring the IDE*.

Display of Task Priority, Stack Usage, and Stack Size

If you are using INTEGRITY version 10 or later, the target list displays task priority, stack usage, or stack size when you right-click any column header in the target list and select **Priority**, **Stack used**, or **Stack size**, respectively. If you are using INTEGRITY IoT, **Priority** and **Stack size** are available.

Long Target List Entries Now Easy to Tell Apart

Compressed target list entries that do not fit in the **Target** column are now abbreviated so that you can easily tell entries apart and find what you are looking for. This is especially helpful when the target list is located on the left side of the Debugger window, where there is typically less room for it.

Improved Target List Display of OSA Tasks on Certain Multi-Core Targets

If you are running an INTEGRITY application on an SMP multi-core target with OS-aware debugging and the new synchronous debugging feature enabled, the Debugger's target list displays all OSA tasks in a single, system-wide task hierarchy. This differs from the display that is available when synchronous debugging is not enabled. In that case, a task hierarchy is nested under each core, so that if your multi-core target contains four cores, for example, each task appears in the target list four times. Not only is it easier to find and track tasks in the single, system-wide task hierarchy, the task statuses provided are always accurate as a consequence of the fact that all cores run and halt at the same time.

New Register access Type

A new register `access` type, `command`, allows you to define a register whose value is the output of a specified MULTI Debugger command. This is particularly useful in conjunction with the `mprintf` command. For more information, see the description of `access` in “The register Section” in Appendix F, “Register Definition and Configuration Reference” in *MULTI: Debugging*.

Python Interpreter Upgraded to 2.7.3

MULTI 7 contains an upgraded Python interpreter based on the 2.7.3 release of Python.

Subversion 1.7.x and 1.8.x Supported

MULTI 7 adds support for Subversion 1.7.x and 1.8.x.

Multiple TCP Ports Supported per License Client

The use of multiple TCP ports per license client is now supported. For more information, see “Specifying a Floating or Named-User License Pool” in Chapter 3, “Configuring End User Machines” in *MULTI: Licensing*.

Simpler License Manager Configuration

You can now use the **License Manager Settings** dialog box to configure a License Manager to serve licenses from multiple license files.

Changes in Behavior

Any-Task Breakpoints Set By Default in INTEGRITY

If you are using INTEGRITY or another operating system that supports any-task and task-specific breakpoints, any-task breakpoints are usually preferred, and they are now the default. For information about modifying this behavior, see the configuration option **clickToSetAnyTaskBp** in “Other Debugger Configuration Options” in Chapter 8, “Configuration Options” in *MULTI: Managing Projects and Configuring the IDE*. See also “Breakdots, Breakpoint Markers, and Tracepoint Markers” in Chapter 8, “Executing and Controlling Your Program from the Debugger” in *MULTI: Debugging*.

Watchpoints Implemented Using Any-Task Hardware Breakpoints

Watchpoints are implemented using any-task hardware breakpoints on targets that support them. Because an any-task hardware breakpoint can be hit by any task in the address space in which it is set, you can capture writes to memory locations of interest regardless of which task the write comes from. Furthermore, you can capture writes even if the writing task did not exist when you set the watchpoint.

Target List Located on Left By Default

The target list is now located on the left side of the Debugger window by default. To move it to the top of the Debugger, click the **Move target list to top** button () located to the right of the **Status** column.

New Methods for Manually Dumping and Processing Profiling Data

The methods for manually dumping and processing profiling data that were available in MULTI 6 are not supported in MULTI 7. This includes use of the **profdump** and **profilemode process** commands. For information about current profiling techniques, see Chapter 17, “Recording and Viewing Profiling Data” in *MULTI: Debugging*.

Several Profiling Commands Consolidated Under the 'Profile' Command

The **profdump**, **profilemode**, and **profilereport** commands are no longer supported, and many of them are no longer necessary. The following table provides alternative commands where available.

Old Command	New Command
profdump	profile stop
profilemode clear	profile clear
profilemode close	profile close
profilemode coverage percent	profile display coverage percent
profilemode import	profile load
profilemode process	profile collect
profilemode start stop	profile start stop

Old Command	New Command
<code>profilereport print</code>	<code>profile print</code>
<code>profilereport save</code>	<code>profile csvexport</code>

For more information, see Chapter 12, “Profiling Command Reference” in *MULTI: Debugging Command Reference* and Chapter 17, “Recording and Viewing Profiling Data” in *MULTI: Debugging*.

Continual Update of Profile Window No Longer Available

The **Profile** window no longer continually updates if you are profiling all tasks in the system or using trace. Profiling data can be recorded, or trace data can be converted into a profile recording, by using the on-demand mechanisms described in:

- “Recording Sampled and Instrumented Profiling Data” in Chapter 17, “Recording and Viewing Profiling Data” in *MULTI: Debugging*, and
- “Converting Trace Data Into a Recording” in Chapter 17, “Recording and Viewing Profiling Data” in *MULTI: Debugging*, respectively.

If you are using trace and have enabled the option **Retrieve trace when buffer fills**, you must first process trace profiling once at the beginning of your session in order to start the processing of trace profiling. Once you have run the target as desired, you can process trace profiling again to obtain the final results. For information about processing trace profiling, see “Converting Trace Data Into a Recording” in Chapter 17, “Recording and Viewing Profiling Data” in *MULTI: Debugging*.

Data Explorer No Longer Treats Instance Variables as Local Variables

In previous versions of MULTI, issuing the command `view $locals$` or `localsview` on a C++ instance method caused the Data Explorer to display all instance variables as if they were local variables. In MULTI 7, the Data Explorer only displays actual local variables and the `this` pointer. To view the members of `this`, expand it just as you would any other structure.

Data Explorer and `print /t' Now Display Variable typedef Names

In previous versions of MULTI, displaying a variable's type via the Data Explorer's **Type** column or printing its type via the **print /t** command showed the final type name, not the typedef name. When used with Green Hills Compiler 2014 or later, MULTI 7 shows the typedef name instead of the final type name. For example, given the following C code, the Debugger displays `myNumber` (not `int`) as the type name of `intVal` and `myObj.count`:

```
typedef int myNumber;

myNumber intVal;
struct myStruct {
    char *name;
    myNumber count;
} myObj;
```

Run-Mode Partnering Now Supported with -checksum

If you are using INTEGRITY IoT or INTEGRITY version 10 or later with Green Hills Probe version 4.4 or later, making an automatic run-mode partner connection no longer sets a breakpoint in the target kernel. This allows automatic run-mode partnering even when you have built your kernel with the **-checksum** option.

This release note supersedes the INTEGRITY note that states, “The Run-mode Partner cannot be enabled in MULTI when invoking CRC or SHA-1.”

Debugger Attaches to Tasks that Hit Any-Task Hardware Breakpoint

The Debugger now automatically attaches to an unattached task if the task hits an any-task hardware breakpoint.

Register Information Now Shared

Prior to MULTI 7, each Debugger process had its own private copy of register information. As a result, changing the register configuration for a particular process would affect only that process.

Because more and more registers—especially memory-mapped registers—are being defined in modern chips, register information is now shared among some Debugger processes (register values are not shared). This reduces memory usage and speeds up the initialization of new processes in the Debugger. Register information is shared among:

- All non-TimeMachine processes in the same AddressSpace

And among:

- All processes created for homogeneous cores under a multi-core connection

Changes to register information, such as adding new registers and changing registers' bitfield structures, are reflected in all the Debugger processes that share the same register information.

Non-INTEGRITY TimeMachine No Longer Accessible from Run Mode

When used from run-mode tasks, the **TimeMachine Debugger** button () , **TimeMachine** menu items, and run control buttons now apply exclusively to the INTEGRITY TimeMachine Debugger. It is no longer possible to enter non-INTEGRITY TimeMachine mode from a run-mode connection.

Child Windows Are Displayed in Front of the Debugger on Windows OS

If you are using MULTI 7 on Windows, child windows are always displayed in front of the Debugger by default. For information about modifying this behavior, see the configuration option **Debugger child windows** in “Debugger Configuration Options” in Chapter 8, “Configuration Options” in *MULTI: Managing Projects and Configuring the IDE*.

Improved `l` Command Output

The Debugger now prints better-formatted output for the **l** command. If you run any scripts that parse **l** command output, we recommend that you update them.

Issue with Windows 10 Dongle-Locked Licenses Resolved

In MULTI 7.1.6, a dongle will incorrectly report that it is not attached when used with Windows 10 updated to version 1803 (April 2018) or later. This issue is resolved with MULTI 7.1.6d. To determine if MULTI 7.1.6d is installed, you can run the **gversion** utility on **lm.exe**.

Changes to Terminology Used in the Documentation

- The operations formerly referred to in the documentation as “synchronous operations” are now called “group-based operations.”
- “Procedure(s)” and “proc(s)” are now documented as “function(s)” and “func(s).” This reflects changes made in the GUI.

Stricter Error Checking on Invalid Pointer Math Expressions

The Debugger now prints errors if you attempt to evaluate expressions that perform pointer math on addresses the Debugger cannot calculate (for example, the addresses of variables having automatic storage duration, evaluated in a context where the variable is out of scope). Such expressions previously evaluated to predictable (but usually incorrect) values.

Deprecated Features

The following features are deprecated in MULTI 7:

- **ANSICMODE** system variable
- Support for relocatable modules
- Task Manager in freeze mode
- Group-based operations, formerly referred to as “synchronous operations” (see “Changes to Terminology Used in the Documentation” on page 19)
- **prepareAllCores** configuration option
- **-allcores** option to **prepare_target** command
- **-onecore** option to **prepare_target** command
- **p** option to **l** command

- **procs** and **procedures** options to **browse** command
- **multibar** command
- **setsync** command
- **showsync** command
- **Are initially focused** setting of **Debugger child windows** configuration option (**viewsAreChildrenMode keepFocus**)
- **Save debugger window position** configuration option
- **Show version control information on file chooser dialog box** configuration option
- **paddedHex** configuration option
- **disAsmStyle** configuration option
- **ShowAddress**, **ShowFType**, **ShowAllFields**, **ExpandValue**, **ExpandComplexMemberValue**, **OpenPointer**, and **ShowChanges** settings of **viewDef** configuration option
- **SourceSafe** integration

Removed Features

The following features are not available in MULTI 7:

- **profdump** command
- **profilemode** command
- **profilereport** command

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

Chapter 2

Changes in MULTI 6.1.6

Contents

New Features and Enhancements	22
-------------------------------------	----

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 3

Changes in MULTI 6

Contents

New Features and Enhancements	24
Changes in Behavior	28
Deprecated Features	31
Removed Features	31

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 20, “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 and ETMv3 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. If you are using INTEGRITY IoT, see the *Integrate User's Guide* for information. If you are using an earlier version of INTEGRITY, see Appendix G, “Integrate Views” in the *MULTI: Debugging* book.

Local Variable View Updates as You Navigate the Call Stack

The Data Explorer local variable view now updates 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 Function 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 and later, 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 Debugger 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 (GHSIm)	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**
 - **WGUtills** 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 4

Changes in MULTI 5.0.6

Contents

New Features and Enhancements	34
Changes in Behavior	37
Unbundled Features	38
Deprecated Features	39
Removed Features	39

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 5

Known Issues

Contents

Host Support	42
MULTI Installation	42
Project Manager Issues	42
Editor Issues	43
Version Control Issues	44
Target Connection Issues	45
Debugging Issues	45
Compatibility with INTEGRITY	50
MULTI-Python Integration Issues	53
File System Issues	54
Desktop Environment Issues	55

Host Support

IDE Components May Fail to Launch on Unsupported Linux Distributions

Components of the MULTI IDE may fail to launch on Linux systems using a 32-bit **libc.so** that is built assuming a 16-byte stack alignment. This problem has been observed with Manjaro Linux and may affect other distributions. If you experience it, we recommend that you switch to a supported version of Linux. For more information, see “System Requirements” on page 2.

MULTI Installation

For known issues regarding INTEGRITY installation, see “Compatibility with INTEGRITY” on page 50.

Name of Install Directory Cannot Contain Spaces

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

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.

Projects with Build Configurations Incompatible with MULTI 6.x and Earlier

Projects that use the build configurations feature new in MULTI 7, including projects generated by the **Project Wizard**, cannot be opened by earlier versions of MULTI.

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 8/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.

Very Large Files Cause Slow Response Times when Single-/Double-Clicking Variables

Source files containing many lines of source code (for example, a million) may cause slow response times when you single- or double-click a variable in the Debugger source pane. This is true even when the source code is included in C/C++ source files via `#include` statements.

Workaround: Split the large file into multiple smaller files.

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. 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.

Trace Configuration Files Incompatible with MULTI 4.x

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 8/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.

`update <interval>' Not Supported with INTEGRITY TimeMachine

If you are using INTEGRITY TimeMachine, the **update** command does not allow an interval to be specified. Additionally, if you use the **update** command to specify an update interval and then enter INTEGRITY TimeMachine mode, interval updates are disabled.

Any-Core Hardware Breakpoints Not Applied to TimeMachine

TimeMachine does not support any-core hardware breakpoints. If you set an any-core hardware breakpoint on the live target and then launch TimeMachine, the breakpoint is not applied to TimeMachine. Note that any-core hardware breakpoints are the default when they are supported.

Limitations of History From RH850 Software Trace

RH850 only

On RH850 processors, History can be used to view function and variable information logged via the software trace features of the architecture. This feature has the following limitations:

- History is not available until trace data is retrieved from the target. MULTI will display an error saying that History is not supported until that point.
- Not supported with multi-core connections.
- Not supported in the instruction set simulator.
- Only function call information and events logged with EAGLE macros are available. Function call information is only supported when function entry and exit events are instrumented by the compiler and PC trace is disabled in the trace options.
- Timing information is only valid if timestamps are enabled in the trace options.
- Only information since the most recent trace discontinuity is shown.

- The **history -invalidate** command is not supported.
- The **eagleconfig** command is not supported.

Quitting During Command Line Function Call May Leave Program in Undefined State

Quitting the Debugger while a command line function call is in progress may leave the program you were debugging in an undefined state.

Workaround: Remove all breakpoints and allow any command line function calls in progress to complete before quitting the Debugger.

Daylight Saving Time May Result in Erroneous Statement (Windows Only)

If the MULTI Debugger is open on an executable when the time changes from standard time to daylight saving time, the Debugger may state that your executable was rebuilt and ask you to reload. This can occur even if the executable has not changed.

This issue affects Windows hosts only.

Compatibility with INTEGRITY

MULTI Ignores INTEGRITY 10 Installer Screen

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, 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 freeze-mode connection.

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 IoT or INTEGRITY version 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.

Run-Mode Breakpoints May Confuse Trace

Run-mode breakpoints are implemented by replacing the targeted instruction with a special breakpoint instruction, which, when executed, causes an exception that triggers breakpoint handling. This may cause minor issues for the trace tools, which are not aware that the targeted instruction was replaced. For example, in the Trace List, the targeted instruction may be shown executing twice, and the corresponding disassembly and/or **Opcode** data may be incorrect.

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.

ctypes and Tcl/Tk Not Included in MULTI for Solaris

The Python 2.7.3 installation that is shipped with MULTI for Solaris does not include the `ctypes` package, which is required by the TimeMachine API, or the Tcl/Tk interpreter and GUI package.

Workaround: Download Python 2.7.3 (which includes the `ctypes` module and Tcl/Tk interpreter) from the Python Web site [<http://www.python.org>], compile and install Python on your 32-bit Solaris machine, install the Tcl/Tk GUI package if you want to use Tcl/Tk, and follow the instructions in “Using a Customized Python Installation” in Chapter 2, “Introduction to the MULTI-Python Integration” in *MULTI: Scripting*.

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.

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 **uvelocity.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.

Windows Do Not Dock Completely

If you are using Windows 10, windows may not snap to the edge of other windows or to the edge of the screen, leaving a gap. This is due to inconsistencies in the Windows API with regard to border thickness. For information about disabling window docking, see “Configuring Taskbar Organization” in Chapter 7, “Configuring and Customizing MULTI” in the *MULTI: Managing Projects and Configuring the IDE* book.

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**.

Nouveau Drivers for nVidia Graphics Cards May Cause Drawing Problems

If you are using the Nouveau open-source driver for nVidia graphics cards, IDE windows may not always redraw correctly. This is due to an apparent problem in the Nouveau driver, which is used by default in certain Linux distributions, including Ubuntu 14.04 LTS.

To determine whether the Nouveau driver is in use, run the following command in a terminal:

```
grep -E -i "connected|card detect|primary dev|Setting driver" /var/log/Xorg.0.log
```

If the output contains `NOUVEAU`, the Nouveau driver is in use for the local display. If the output contains `NVIDIA`, the nVidia binary driver, which does not exhibit the problem, is in use.

Workaround: If you are using Ubuntu 14.04 LTS, follow these steps to switch to using the binary driver:

1. Launch the Dash by clicking the Ubuntu icon at the top of the Launcher.
2. Type `Additional Drivers`, and select the **Additional Drivers** entry under the **Applications** group.
3. When the **Additional Drivers** tab loads, select one of the **Using NVIDIA binary driver - version xxx.xx (proprietary)** radio buttons. If you have the option to select a **(proprietary, tested)** driver, that may be preferable.
4. Click **Apply Changes**.
5. You may be prompted to enter your login credentials for installing software on the machine.

6. After the update is complete, click **Close**.
7. Reboot the machine for the changes to take effect.

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  
...
```

Font Display Problems with Ubuntu 12.04 and Xvnc

If you are using MULTI with the Xvnc server on Ubuntu 12.04, you may notice the following problems:

- Fonts in menu bars and dialog boxes use the same fixed-size system font used for source code display.

This occurs because MULTI uses fixed-width fonts, which are not installed by default with Ubuntu 12.04.

Solution:

```
sudo apt-get install xfonts-75dpi
```

- Fonts in the Help Viewer are difficult to read.

This occurs because the Type 1 fonts that are found first in the Xvnc font path remap the Helvetica font to Nimbus Sans.

Workaround:

1. Copy **/usr/bin/vncserver** to a directory you have write access to. Then open it in an editor.
2. Find the line containing `/usr/share/fonts/X11/75dpi/,.` Move it to the line before `/usr/share/fonts/X11/Type1/,.`
3. Save the file.
4. Restart your Xvnc server. When you run **vncserver**, use your modified version.

MULTI May Lose X Connection After Running for Several Hours

If, when running the MULTI IDE, you run a script that drives the Debugger for several hours, the IDE may exit with the following message:

```
multi: fatal: X: connection to server lost
```

This problem is caused by a known bug within libxcb and xlib in all recent Linux distributions. A fix from the library maintainers may be available in the future. Ubuntu 10.04 LTS is not affected by this issue.

