

MULTI Compiler 2013.5 Release Notes for Embedded V850 and RH850



**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.

Copyright © 1983-2013 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 partial listing of Green Hills Software and periodically updated patent marking information, please visit http://www.ghs.com/copyright_patent.html.

PubID: release_notes_v800-496213

Branch: <http://toolsvc/branches/release-branch-2013-5-bto>

Date: October 4, 2013

Contents

1. Changes in Version 2013.5	1
System Requirements	2
Windows	2
Linux	2
Solaris	3
Product Compatibility	3
New Features and Enhancements	4
Static Assertions	4
Specify a Compiler Inside a Project File	4
Ability to Specify the Size of wchar_t	5
Stack Smashing Protection	5
Updated gstack Utility Program for Better C++ Analysis	5
New Attribute and Intrinsic to Remove Specific Functions from Profiling Reports	5
Changes In Behavior	6
GNU Compatibility Updated for Better Compatibility with Version 4.3	6
 2. Changes in Version 2013.1	 7
Product Compatibility	8
New Features and Enhancements	8
Relaxed Size Restrictions	8
protrans Supports Converting Section Dumps to .pro Files	9
Changes In Behavior	9
Some C and C++ Libraries Have New Names Based on Floating Point Support	9
Removed and Deprecated Features	9
 3. Changes in Version 2012.5	 11
Product Compatibility	12

New Features and Enhancements	12
Updated gstack Utility Program	12
New Block Profiling System	12
New ax Modifier	13
New Linker Options to Control Linker p_align Behavior	13
New Stack Check Attribute	13
Changes In Behavior	13
Clean Builds Recommended	13
Error Compiling ind_initmem.c	14
Vector's Clear Method No Longer Deallocates	14
Implicit Inclusion is No Longer Enabled By Default	15
Eclipse Plug-in Option Type Changes	15
ghide No Longer Hides COMMON Symbols	15
Removed and Deprecated Features	15

4. Changes in Version 2012.1 17

Product Compatibility	18
New Features and Enhancements	18
Separate Releases Allow Greater Flexibility When Upgrading	18
Faster Builds	18
Define Macros in Any Project File	19
New Coding Standard Features	19
UTF-8 Support	19
Changes In Behavior	19
New Licenses Required	19
Clean Builds Recommended	20
gbuild Defaults to Parallel Builds	20
Different malloc() Implementation Used By Default in Stand-Alone Projects	20
__MULTI_DIR__ Customization File Macro Changes	20
Building Previous Releases of INTEGRITY or u-velOSity May Produce New Diagnostic Messages	20
Pointers Are Now Unsigned by Default	21
memmove Moved to libind.a	21
Removed and Deprecated Features	21

5. Changes in Version 5.4	23
New Features and Enhancements	24
UTF-8 Support	24
Updated MULTI Eclipse Plug-In	24
Changes in Behavior	24
Removed Features	24
Driver No Longer Accepts .bld Files	24
6. Changes in Version 5.3	25
Product Compatibility	26
New Features and Enhancements	26
New -Omoredebug Optimization Strategy	26
Updated Compiler Front-End	26
Updated Fast Flash Programmer	26
General Improvements	27
Changes in Behavior	27
New Warnings Enabled by Default	27
Changes to Line Continuations in GNU C Mode	27
Changes in std::string	28
Options Enabled by Default	28
Map File Format Changes	28
Loops with an Iteration Variable Shorter than int May Execute Slower than in Previous Releases	29
__alt_init Hook Removed From Stand-Alone Run-time Library	29
Removed Features	29
Deprecated Features and Discontinued Support	30
Deprecated K&R C Support	30
Deprecated -gnu Option	30
7. Changes in Version 5.0.6	31
New Features and Enhancements	32
Improved Code Generation	32
Improved Dependency Analysis	32
DoubleCheck Static Source Code Analyzer	32

Fast Flash Programmer	32
Changes in Behavior	33
Changes to Driver Option Defaults	33
Options Enabled by Standard C++ Dialects	34
Libind.a Splitup	34
Building System Libraries from 4.x Distributions	34
Global Const Variables Placed in Read-Only Data	35
Unbundled Features	35
Unbundled in 5.0.6	35
Deprecated Features and Discontinued Support	36
Deprecated Options	36
The ghexfile Utility is No Longer Available	37
Manual Template Instantiation Mode Options No Longer Supported	37
Programming flash from grun	37
8. Known Issues	39
Installation and Licensing	40
Green Hills Debug Probe CD Compatibility	40
Name of Install Directory Cannot Contain Spaces	40
Host Support	40
UNC Paths Are Not Supported in the MULTI Debugger	40
UNC Paths to Virtual Machines Are Not Supported	40
Target Support	41
Toolchain Issues	41
No Function Prototypes For INTEGRITY 5 strdup() and strlen()	41
gstack Provides Warnings for Green Hills Library Functions	41
gstack Provides Warnings for INTEGRITY	41
Exported Templates With Inline Functions May Produce Incorrect Code	41
ThreadX Build Failure When Customizing Startup and System Libraries	41
Non-flat Directory Structure in Multi-core Builds Causes Warnings in gcores	42

Output Filename in Multi-Core Builds with extension different to .mca Causes Warnings in gcores	42
Target Connection Issues	43
Flash Driver Library libflash.a Limited to Four CFI Drivers	43
Debugging Issues	43
Auto Trace Collection Reduces Performance	43
Target Agent May Cause Failures When Programming Flash	43
References Not Available for Inlined Functions	44
Cannot Debug Executables in Eclipse Projects with a Space in the Name	44
File System Issues	44
Two Dots (..) Not Supported After Symbolic Links in Paths	44
NFS May Cause Poor Performance	44
 9. V850 and RH850 Release Notes	 47
V850 and RH850 Toolchain	48
Changes and Known Issues in 2013.5	48
Changes and Known Issues in 2013.1	50
Changes and Known Issues in 2012.5	54
Changes and Known Issues in 2012.1	57
MULTI 6.1.4 Compatibility	59

Chapter 1

Changes in Version 2013.5

Contents

System Requirements	2
Product Compatibility	3
New Features and Enhancements	4
Changes In Behavior	6

System Requirements

The MULTI IDE may have different system requirements than the Compiler. For information about IDE requirements, see the release notes for the MULTI IDE.

Windows

Running a full 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 Service Pack 2 or greater (IA-32, 32-bit mode only), Windows Vista (32- or 64-bit mode), Windows 7 (32- or 64-bit mode), or Windows 8 (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 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, 10.04, or 12.04 (32-bit or 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 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

Green Hills Compiler 2013.5 is officially supported with INTEGRITY versions 5.x, 10.0.2, and 11.0.4. For information about compatibility with newer versions of INTEGRITY, see the INTEGRITY release notes.

Green Hills Compiler 2013.5 is officially supported with ThreadX 5.5.x and newer.



Warning

Using Compiler 2013.5 with a Green Hills Debug Probe requires Green Hills Debug Probe software version 4.4.4 or newer. Do not install Green Hills Debug Probe software version 4.2.x or earlier over your Compiler 2013.5 installation.



Note

If your MULTI IDE is newer than the compiler, see the IDE release notes for up-to-date product compatibility.

New Features and Enhancements

Static Assertions

This release adds a C11/C++11 style `static_assert` in permissive C and C++ language dialects, that allow you to statically check conditions at compile time with a standard mechanism. For more information, see “Static Assertions” in Chapter 15, “Macros, Pragma Directives, and Ininsics” in *MULTI: Building Applications for Embedded V850 and RH850*.

Specify a Compiler Inside a Project File

You can now specify the location of the Compiler installation directory that should be used to build your project from the Project Manager using the **gbuildDir** option inside a project (**.gpj**) file as a Group 1 option. For example:

```
#!gbuild
gbuildDir=c:\ghs\comp_201354
primaryTarget=ppc_standalone.tgt
[Project]
...
```

This option only affects projects built from the Project Manager; it is ignored by the **gbuild** utility program.

With MULTI 6 and earlier, this option silently defaults to the Compiler installation directory that the MULTI IDE is linked with in the event that a bad path is supplied.

For more information, see “Group 1 Directives” in Appendix C, “Green Hills Project File Format” in *MULTI: Building Applications for Embedded V850 and RH850*.

Ability to Specify the Size of `wchar_t`

You can now select the size of the `wchar_t` type using the **-wchar_u16** and **-wchar_s32** options to conserve memory when necessary. For more information, see “WChar Size” in Chapter 3, “Builder and Driver Options” in *MULTI: Building Applications for Embedded V850 and RH850*.

Stack Smashing Protection

The new **-stack_protector** option provides protection against stack smashing attacks by using a *stack canary* — a random number on the stack — that is compared to an expected value in function epilogues. For more information, see “Check for Stack Smashing Attacks” in Chapter 3, “Builder and Driver Options” in *MULTI: Building Applications for Embedded V850 and RH850*.

Updated **gstack** Utility Program for Better C++ Analysis

The **gstack** utility program has been updated to provide better support for analyzing function calls involving C++ virtual functions, startup code, and libraries. For more information about **gstack**, see “The **gstack** Utility Program” in Chapter 11, “Utility Programs” in *MULTI: Building Applications for Embedded V850 and RH850*.

New Attribute and Intrinsic to Remove Specific Functions from Profiling Reports

The new `ghs_noprofile` attribute and `__ghs_noprofile_func()` intrinsic allow you to mark functions so that any code paths they appear in are not considered when MULTI builds reports from coverage data gathered with **-coverage=**.

If you are using MULTI 6.1.4, you must enable the **-Xprofileignorefatalblocks** switch to use this feature. With this version of MULTI, the following caveats apply:

- The **Coverage** tab will always display 0 for **% Covered**, but the **Block Detailed** tab will continue to display count information for the block.
- Coverage line coloring in the Debugger is reversed for functions with the `ghs_noprofile` attribute.
- The `ghs_noprofile` attribute is ignored by **TimeMachine** profiling.

Changes In Behavior

GNU Compatibility Updated for Better Compatibility with Version 4.3

The GNU C and C++ dialects have been updated to provide closer compatibility with the 4.3 version of the GNU compiler. To revert to a version of these dialects that is more closely compatible with version 3.3, use **--gnu_version=30300** in conjunction with the **-gcc** or **-g++** language dialect option. Make sure that you have also selected a GNU dialect before specifying this option.

Additionally, dependent name processing is now on by default when using the GNU C++ dialect, even when emulating version 3.3 of the GNU compiler. To disable dependent name processing, use **--no_dep_name**.

Furthermore, the parsing of function template definitions is done lazily in the GNU C++ dialect. Even though the parsing of a template in its generic form (without argument substitution) is typically done to resolve non-dependent names, this parse is only done for templates that are actually instantiated, and the parse itself is deferred until the entire input file has been processed. This may result in the apparent suppression of errors in unused function templates. This may be disabled with the **--no_defer_parse_function_templates** option.

For more information, see:

- “GNU C/C++ Extensions” in Chapter 12, “Green Hills C” in *MULTI: Building Applications for Embedded V850 and RH850*
- “Dependent Name Lookup” in Chapter 13, “Green Hills C++” in *MULTI: Building Applications for Embedded V850 and RH850*
- “Deferral of Function Template Parsing” in Chapter 3, “Builder and Driver Options” in *MULTI: Building Applications for Embedded V850 and RH850*

Chapter 2

Changes in Version 2013.1

Contents

Product Compatibility	8
New Features and Enhancements	8
Changes In Behavior	9
Removed and Deprecated Features	9

Product Compatibility

Green Hills Compiler 2013.1 is officially supported with INTEGRITY versions 5.x, 10.0.2, and 11.0.4. For information about compatibility with newer versions of INTEGRITY, see the INTEGRITY release notes.

Green Hills Compiler 2013.1 is officially supported with ThreadX 5.5.x and newer.



Warning

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



Note

If your MULTI IDE is newer than the compiler, see the IDE release notes for up-to-date product compatibility.

New Features and Enhancements

Relaxed Size Restrictions

The compiler no longer restricts arrays and structures to 256 megabytes in size. It now allows uninitialized arrays and structures declared as up to 2 gigabytes on 32-bit targets, and over 2 gigabytes on 64-bit targets.

The **--large_vtbl_offsets** option eliminates the 64 kilobyte limit on classes with multiple virtual inheritance. For more information, see “Virtual Function Table Offset Size” in Chapter 3, “Builder and Driver Options” in *MULTI: Building Applications for Embedded V850 and RH850*.

There may be some issues with debugging information for large arrays and structures, such as warnings issued from **dblink**, problems viewing these objects in the MULTI Debugger, or problems when these objects are involved in command line procedure calls. In addition, large programs may not load into the simulator if they consume too much RAM on the host.

protrans Supports Converting Section Dumps to .pro Files

If you use one of the **-coverage=** options to collect profiling information from your program and dump that program into a file on your host computer, you can now use **protrans** to convert that file into a **.pro** file using the **-cc**, **-cd**, or **-cf** option. For more information, see “The protrans Utility” in Chapter 11, “Utility Programs” in *MULTI: Building Applications for Embedded V850 and RH850*.

Changes In Behavior

Some C and C++ Libraries Have New Names Based on Floating Point Support

Some C and C++ libraries have been renamed based on the level of floating point support they provide. For more information, see “Libraries” in Chapter 16, “Libraries and Header Files” in *MULTI: Building Applications for Embedded V850 and RH850*.

Removed and Deprecated Features

The following features have been deprecated:

- Call count and call graph profiling (**-p** and **-pg**).
- Run-mode debug over a freeze-mode debug connection with hardware targets running INTEGRITY 5. Run-mode connections are still supported over Ethernet and serial ports.
- Socket emulation over freeze-mode host I/O with hardware targets running INTEGRITY 5. Socket emulation is still supported over run-mode debug, or you can use the GHnet v2 TCP/IP stack on the target.

Chapter 3

Changes in Version 2012.5

Contents

Product Compatibility	12
New Features and Enhancements	12
Changes In Behavior	13
Removed and Deprecated Features	15

In the Green Hills Compiler 2012.5 release, we focused on improving the **gstack** utility program, improving profiling data collection, and increasing C++ performance.

Product Compatibility

Green Hills Compiler 2012.5 is officially supported with INTEGRITY versions 5.x, 10.0.2, and 11.0.4. For information about compatibility with newer versions of INTEGRITY, see the INTEGRITY release notes.

Green Hills Compiler 2012.5 is officially supported with ThreadX 5.5.x and newer.



Warning

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



Note

If your MULTI IDE is newer than the compiler, see the IDE release notes for up-to-date product compatibility.

New Features and Enhancements

Updated gstack Utility Program

The **gstack** stack analysis program has been greatly improved with the ability to analyze recursive functions, function pointers, and interrupts. Its output has also been improved to provide detailed information to guide you in annotating your program for the most accurate results. For more information, see “The gstack Utility Program” in Chapter 11, “Utility Programs” in *MULTI: Building Applications for Embedded V850 and RH850*.

New Block Profiling System

The **-coverage=** option provides a faster method for measuring both performance and code coverage than the legacy **-a** method. If you only want to test for coverage and not execution counts, **-coverage=flag** provides a significantly more efficient

method than **-a**. For more information, see “Enabling Instrumented Coverage or Performance Profiling” in Chapter 2, “Developing for V850 and RH850” in *MULTI: Building Applications for Embedded V850 and RH850*.

New ax Modifier

The **ax** librarian now includes the **u** modifier for compatibility with the GNU archiver.

New Linker Options to Control Linker p_align Behavior

The new **-max_p_align**, **-any_p_align**, and **-no_p_align** options allow you to specify how the linker determines the maximum value of the **p_align** field in ELF program headers. For more information, see “Linker-Specific Options” in Chapter 9, “The elxr Linker” in *MULTI: Building Applications for Embedded V850 and RH850*.

New Stack Check Attribute

The new C/C++ **stackcheck** attribute allows you to enable stack checking on a per-function basis. For more information, see **stackcheck** in *MULTI: Building Applications for Embedded V850 and RH850*.

Changes In Behavior

Clean Builds Recommended

To mitigate any problems that may occur due to small changes in behavior from previous versions, we recommend that you clean and rebuild your programs from scratch when you build them for the first time with the upgraded compiler.

Error Compiling `ind_initmem.c`

When porting a customized **libstartup** from an earlier version to Green Hills Compiler 2012.5, one of the following messages may be given when compiling **`ind_initmem.c`**:

```
#error This file should only be included from another file in the same directory
#error Please include <stddef.h> instead of ghs_null.h
```

To fix this problem, edit **`ind_initmem.c`** to replace:

```
#include <ghs_null.h>
```

with:

```
#include <stddef.h>
```

For more information, see “C and C++ Header Files” in Chapter 16, “Libraries and Header Files” in *MULTI: Building Applications for Embedded V850 and RH850*.

Vector's Clear Method No Longer Deallocates

Prior to Green Hills Compiler 2012.5, vector's `clear()` method deallocated memory, resetting the capacity to zero, even if the `reserve()` method had been called. In Green Hills Compiler 2012.5, `clear()` no longer deallocates memory.

To deallocate a vector's memory, call the `swap()` method instead. This technique is portable across versions of the Green Hills Compiler and other STL implementations:

```
std::vector<T> v;
...
std::vector<T>().swap(v);

std::vector<T> v;
...
std::vector<T>(v).swap(v);
```

This change to `clear()` can be reverted by defining the `__GHS_VECTOR_CLEAR_DEALLOCATES` macro to 1. This macro might be removed in a future version of the compiler.

Implicit Inclusion is No Longer Enabled By Default

Implicit inclusion is no longer enabled by default. If you require this feature, pass the `--implicit_include` option to the driver.

Eclipse Plug-in Option Type Changes

In the Green Hills Eclipse plug-in, the types of a few options have changed. If you have existing Green Hills Eclipse Managed Make projects that you created using an older version of the Green Hills Eclipse plug-in and you encounter options that do not work well (for example, they display erroneous or garbled text values in the C/C++ Build Settings dialog) we recommend creating a new project using the 2012.5 version of the plug-in and re-importing your sources into the new project, rather than reusing the old project with the new plug-in.

ghide No Longer Hides COMMON Symbols

The **ghide** utility no longer hides `COMMON` symbols. For more information about **ghide**, see “The ghide Utility Program” in Chapter 11, “Utility Programs” in *MULTI: Building Applications for Embedded V850 and RH850*.

Removed and Deprecated Features

The following features have been deprecated:

- The **Legacy Coverage Profiling (-a)** driver option.

Chapter 4

Changes in Version 2012.1

Contents

Product Compatibility	18
New Features and Enhancements	18
Changes In Behavior	19
Removed and Deprecated Features	21

In the Green Hills Compiler 2012.1 release, we focused on improving build performance, enhancing existing optimizations, and making architectural changes that allow you to upgrade the MULTI IDE and Compiler separately. This version of the compiler also adds GHS Standard Mode and coding standard profiles, two new features that help you improve code quality.

Product Compatibility

Green Hills Compiler 2012.1 is officially supported with INTEGRITY versions 5.x, 10.0.2, and 11.0.4. For information about compatibility with newer versions of INTEGRITY, see the INTEGRITY release notes.

Green Hills Compiler 2012.1 is officially supported with ThreadX 5.5.x and newer.



Warning

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



Note

If your MULTI IDE is newer than the compiler, see the IDE release notes for up-to-date product compatibility.

New Features and Enhancements

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. This release structure also allows Green Hills Software to release compilers with new CPU support on a faster schedule.

Faster Builds

Build speeds have increased significantly in this release of the MULTI Compiler. Build speed on Windows hosts has increased, and parallel build speed has improved

up to 50%. In addition, **gbuild** now builds in parallel by default, so users that did not explicitly pass **-parallel=** in previous versions of MULTI may see much larger improvements. In one real-world case, a customer reported a 4.5x increase in build speed.

Define Macros in Any Project File

You can now define macros in any project file, not just your Top Project.

New Coding Standard Features

The Green Hills Compiler now ships with GHS Standard Mode, an internally-designed collection of diagnostics that help you avoid common pitfalls while developing programs. We use this standard internally because its rules are pragmatic to apply to existing projects and helpful for improving code quality.

The compiler also provides coding standard profiles, a feature that allows you to create your own coding standards by packaging diagnostic severity levels into a single file to distribute among all developers in your organization. For more information, see Chapter 14, “Coding Standards” in *MULTI: Building Applications for Embedded V850 and RH850*.

UTF-8 Support

The Green Hills Compiler now supports multi-byte characters encoded in UTF-8. For more information, see “Host and Target Character Encoding” in Chapter 3, “Builder and Driver Options” in *MULTI: Building Applications for Embedded V850 and RH850*.

Changes In Behavior

New Licenses Required

The MULTI licensing system was updated in this release. If you are upgrading from an older release, you must obtain new licenses. For more information, please contact Green Hills support.

Clean Builds Recommended

To mitigate any problems that may occur due to small changes in behavior from previous versions, we recommend that you clean and rebuild your programs from scratch when you build them for the first time with the upgraded compiler.

gbuild Defaults to Parallel Builds

gbuild now builds programs in parallel by default, and automatically determines the number of builds to run at once based on the number of CPUs on your host machine. You do not need to specify **parallel=*n*** to **gbuild** unless you want to limit the number of CPUs used for building programs.

Different malloc() Implementation Used By Default in Stand-Alone Projects

The compiler uses a new `malloc()` implementation by default for all Stand-Alone projects. This implementation generally provides better performance, but has a slightly larger library footprint. To use the old implementation instead, pass the **-no_fast_malloc** option.

__MULTI_DIR__ Customization File Macro Changes

If you use the predefined `__MULTI_DIR__` macro in your customization files, replace each instance with `__TOOLS_DIR__`, which points to your MULTI Compiler installation directory. There is no longer a macro defined for the MULTI IDE installation directory.

Building Previous Releases of INTEGRITY or u-velOSity May Produce New Diagnostic Messages

When building u-velOSity 2.2.9 or earlier, INTEGRITY 5, or INTEGRITY 10, the compiler might output new errors or warnings. To avoid compatibility problems, set the **-act_like** option to `5.0` when building an INTEGRITY project, compiling an INTEGRITY BSP, or building u-velOSity system libraries. You can do this in one of two ways:

- Add the following line to **target/integrity.bld** or **target/uvelocity.bld**:

```
:defines=__ghs_driver_act_like=5.0
```

- Set the **-act_like=5.0** option in your Top Project.

If you encounter other warnings, you may need to further adjust the compiler diagnostics options.

Pointers Are Now Unsigned by Default

Pointers are now unsigned by default for all architectures, and shipping libraries are built with **--unsigned_pointer**. If you require signed pointers, pass the **--signed_pointer** option, but keep in mind that operations that rely on memory spanning 0 may not work consistently between user code and the libraries.

memmove Moved to libind.a

The `memmove()` function has been moved from **libansi.a** to **libind.a**. If your build fails because `memmove()` cannot be located, include **libind.a**.

Removed and Deprecated Features

The following features are no longer available:

- Windows 2000 support
- Elan licenses
- The **gbldconvert** utility that converts **.bld** files to **.gpj** files. If you are migrating from MULTI 4 or earlier and need to convert your **.bld** files, contact Green Hills Support for more information.

The following features have been deprecated:

- **gproxy**
- **grun**
- P&E Lightning card support in **peserv**
- The `truncate()` function in **libsys.a**.
- **--cxx_include_directory** (use the **-I** driver option instead).

Chapter 5

Changes in Version 5.4

Contents

New Features and Enhancements	24
Changes in Behavior	24
Removed Features	24

New Features and Enhancements

UTF-8 Support

The compiler now supports UTF-8 encoding for extended characters, in addition to EUC-JP and Shift-JS. For more information, see “Extended Character Support” in the *MULTI: Building Applications* book.

Updated MULTI Eclipse Plug-In

The MULTI Eclipse plug-in now supports Eclipse 3.7.x and CDT 8.0.x. For more information, see the documentation about the MULTI Eclipse plug-in in the *MULTI: Building Applications* book.

Changes in Behavior

Removed Features

Driver No Longer Accepts .bld Files

The driver and Project Manager no longer build projects in the legacy **.bld** format, and the **gbldconvert** utility that converts files from **.bld** to the newer **.gpj** format has been deprecated. Support for converting **.bld** files will be removed in the next release.

Chapter 6

Changes in Version 5.3

Contents

Product Compatibility	26
New Features and Enhancements	26
Changes in Behavior	27
Removed Features	29
Deprecated Features and Discontinued Support	30

Product Compatibility

The MULTI 5.3 compiler has been tested for compatibility and is officially supported with:

For information about the compatibility of the MULTI 5.3 compiler with older versions of these products, contact your Green Hills sales representative. For information about compatibility with newer versions of these products, see the release notes for the appropriate product.

New Features and Enhancements

New -Omoredebug Optimization Strategy

The new **-Omoredebug** optimization strategy has been added in this release and is enabled by default for new projects. This strategy enables optimizations that do not affect debug ability, and provides enhancements that extend the liveness of variables and provide additional breakdots in the MULTI Debugger. Note that in some cases, **-Omoredebug** may cause your program's stack size to increase.

Updated Compiler Front-End

MULTI Compiler 5.3 is based on a newer version of the EDG C++ front-end. The new front-end includes many bug fixes, improvements in standard compliance, and improvements in GNU compatibility. The default severity levels of some diagnostics may have changed.

Updated Fast Flash Programmer

The fast flash programmer includes several enhancements:

- faster SREC handling
- delta flashing capabilities (only re-flash changed sectors)
- improved logging
- support for new flash parts

For more information about which flash parts are supported, contact your Green Hills sales representative.

General Improvements

The following features have seen major improvements:

- Wholeprogram optimizations
- Interprocedural optimizations

Changes in Behavior

New Warnings Enabled by Default

The `--prototype_warnings` option is now set by default. This option may generate warnings that indicate bugs in your code.

Changes to Line Continuations in GNU C Mode

In versions of the MULTI compiler earlier than 5.2, a backslash (\) followed by a space was never parsed as a line continuation. As of 5.2, in the GNU C or C++ dialect, the compiler parses backslash as a line continuation character whether or not it is followed by a space.

If you compile your code using the GNU C or C++ dialect and it has a line that contains a backslash followed by a whitespace, it might create an unintentional line continuation. The following example shows one of these cases (the space is indicated by an underscore, because it does not display in print):

```
#include <stdio.h>
int main(void) {
    int i = 42; // use space after backslash, as in C:\_
    i = 0; // ignored in GNU
    if (i != 0) {
        printf("i = %d, should be 0\n", i);
    }
}
```

When the compiler encounters one of these cases, it issues diagnostic 1400. If you need to hide this diagnostic, pass the option **--diag_suppress=1400** to the driver.

Changes in `std::string`

Due to changes in `std::string`, the C++ libraries included with this release are incompatible with the C++ libraries shipped with previous versions of MULTI.

Options Enabled by Default

The following options are now enabled by default in MULTI 5:

- **--alternative_tokens**
- **--no_implicit_extern_c_type_conversion**
- **--no_guiding_decls**
- **--readonly_typeinfo**
- **--readonly_virtual_tables**

The following options are enabled by default unless you pass **--guiding_decls**:

- **--no_old_specializations**
- **--no_implicit_typename**

Map File Format Changes

The format of map files generated by the **elxr** linker has changed in version 5.3. The map file now lists the size of each symbol in addition to its address (in the format *address+size*).

To force the linker to produce map files in the old format, pass the following option to the driver:

```
-lnk=-old_mapfile
```

This option is not supported in conjunction with the **-Mu** or **-MI** linker options.

Loops with an Iteration Variable Shorter than int May Execute Slower than in Previous Releases

If you have enabled speed optimizations and your program has a loop that uses an iteration variable of a type smaller than `int`, that loop may execute slower than it did with previous versions of MULTI. To work around this issue, use an iteration variable of type `int` or larger, or `int_fastN_t` or `uint_fastN_t`, defined in `stdint.h`.

`__alt_init` Hook Removed From Stand-Alone Run-time Library

The call to `__alt_init` from the stand-alone run-time routine `__ghs_ind_crt1` has been removed as of the 5.2 release. We recommend renaming `__alt_init` to `__ghs_board_devices_init`, which is called at approximately the same point in the startup sequence. A second approach is to add the following C code to your project:

```
#if defined(__GHS_VERSION_NUMBER) && (__GHS_VERSION_NUMBER >= 520)
void __alt_init(void);
void __ghs_board_devices_init(void) {
    __alt_init();
}
#endif
```

If neither of these approaches work for you, you can customize **libsys.a** and modify `__ghs_ind_crt1` in **ind_crt1.c** to call `__alt_init` prior to calling `main()`. For more information, see “Creating a Project with a Customizable Run-Time Environment” in “Libraries and Header Files” in the *MULTI: Building Applications* book.

Removed Features

The following features are no longer included in MULTI, but may still be available for purchase separately. For more information, contact your Green Hills sales representative.

- Elan licensing tools
- HP-UX host support
- The following driver options:

- **--array_new_and_delete**
- **--common_implicit_initialization/--no_common_implicit_initialization**
- **--distinct_template_signatures**
- **--early_tiebreaker/--late_tiebreaker**
- **--enum_overloading/--no_enum_overloading**
- **--explicit**
- **--extern_inline/--no_extern_inline**
- **--friend_injection/--no_friend_injection**
- **--include_once/--include_never**
- **--initpoinsters**
- **--memory_description_file**
- **--new_for_init/-old_for_init**
- **-Owholeprogram_libs**
- **--relative_xof_path/--no_relative_xof_path**
- **--short_lifetime_temps/--long_lifetime_temps**
- **--std_qualifier_deduction**
- **--typename**
- **-use_vp_as_dbg_source_root/-no_use_vp_as_dbg_source_root**

Deprecated Features and Discontinued Support

Deprecated K&R C Support

Support for the K&R C dialect (**-k+r**) is deprecated and will be removed in the next MULTI release. Certain old-style features, such as K&R-style parameter declarations, will continue to be supported outside of the K&R C dialect.

Deprecated -gnu Option

The **-gnu** driver option is deprecated. To get the same behavior as **-gnu**, pass both the **-gcc** and **--g++** options instead.

Chapter 7

Changes in Version 5.0.6

Contents

New Features and Enhancements	32
Changes in Behavior	33
Unbundled Features	35
Deprecated Features and Discontinued Support	36

New Features and Enhancements

Improved Code Generation

MULTI 5 comes complete with Green Hills Software's world class compilers, including new interprocedural optimization and link-time optimization. MULTI 5 boasts compilers which generate some of the industry's smallest and fastest code. Additionally, you can use MULTI 5 to enforce programming guidelines such as the MISRA 2004 coding standards.

Improved Dependency Analysis

Building programs with complicated dependencies (such as an INTEGRITY kernel) has historically required that the user identify libraries that might have changed since the last build in order to avoid link-time errors. MULTI's Improved Dependency Analysis solves this problem by automatically building out of date libraries if the current program is dependent on those libraries. This capability extends to prebuilt libraries or other libraries linked in with an option such as **-l** or **-kernel**.

DoubleCheck Static Source Code Analyzer

The DoubleCheck static source code analyzer reviews code, searching for problems which may lead to bugs in the final product. DoubleCheck is fully integrated with the MULTI IDE. It executes automatically when you build your application. Any problems discovered by DoubleCheck are reported to you alongside syntax errors reported by the compiler. This creates an opportunity for you to immediately fix bugs, saving many hours of debugging time later in your development process.

Fast Flash Programmer

The MULTI IDE's Fast Flash Programmer solves several problems which occur in firmware development. It significantly cuts down on the time and effort required to repair corrupted boot code, fix bugs in firmware and program production boards. The Fast Flash Programmer uses little or no RAM and runs with greater speed than its predecessor. Additionally, the Fast Flash programmer provides effective feedback

when hardware causes a programming failure. Operation of the Fast Flash Programmer can be scripted using the MULTI command interface for greater flexibility and power.

Changes in Behavior

Changes to Driver Option Defaults

Builder Option	MULTI 4.2.4 Default	MULTI 5.0.6 Default
Alternative C++ Tokens	Off (--no_alternative_tokens)	On (--alternative_tokens)
Consistent Code Without Debug Information	Option does not exist	Off (-noconsistentcode)
Constant Propagation	Off (-Onoconstprop)	Depends on optimization strategy
DoubleCheck Level	Option does not exist	None (-double_check.level=none)
DoubleCheck Output that Stops Builds	Option does not exist	Off (-double_check.stop_build=off)
Guiding Declarations of Template Functions	On (--guiding_decls)	Off (--no_guiding_decls)
Link-Once Template Instantiation	Option does not exist	Off (--no_link_once_templates)
-Olimit= Without Debug Information	Option does not exist	Off (-noglimits)
Simplify C Print Functions	Option does not exist	On (-Oprintfuncs)
Stack Limit Checking	Option does not exist	Off (-no_stack_check)
Section Overlap Checking	Warnings (-overlap_warn)	Errors on Non-Zero Overlap (-nooverlap)
Support for Implicit Extern C Type Conversion	On (-implicit_extern_c_type_conversion)	Off (-no_implicit_extern_c_type_conversion)
Support for Old-Style Specializations	On (--old_specializations)	Off (--no_old_specializations)
Support for Implicit Typenames	On (--implicit_typename)	Off (--no_implicit_typename)
Treat Doubles as Singles	Option does not exist	Off (-nofloatsingle)

Options Enabled by Standard C++ Dialects

When you select either of the Green Hills Standard C++ dialects (`--std` or `--STD`), MULTI enables various options in order to conform more closely to the C++ standard as defined in International Standard ISO/IEC 14882:1998. MULTI 5 enables a slightly different set of options from MULTI 4:

Options enabled by `--std` and `--STD`:

- `--alternative_tokens`
- `--no_implicit_extern_c_type_conversion`
- `--no_old_specializations`
- `--no_implicit_typename`
- `--no_guiding_decls`

Additional options enabled by `--STD`:

- `--dep_name`

Libind.a Splitup

Some of the contents of the `libind.a` library from previous releases have been split out into a new library. The `libmath.a` library is new in MULTI 5, and contains math routines.

Building System Libraries from 4.x Distributions

The MULTI 4 System Libraries and headers make use of **stdio.h** field names that are not preceded by an underscore. In MULTI 5, the names are all defined with a leading underscore.

If you are using MULTI 5 to build a project with system libraries (`libsys.a`) from a MULTI 4 distribution, you must pass `-D__GHS_OLD_STDIO_FIELD_NAMES` so that MULTI 5's **stdio.h** defines the old names as well.

Global Const Variables Placed in Read-Only Data

In prior versions of MULTI, global `const` variables without an initializer, such as:

```
const int i;
```

were output as common variables (see “Allocation of Uninitialized Global Variables” in Chapter 3, “Builder and Driver Options” in the *MULTI: Building Applications* book). This could cause inconsistencies when using special data areas or when referencing the variable with position-independent code (PIC) or position-independent data (PID).

In MULTI 5, these variables are output as if they have been initialized to zero. The example above would be output in the same way as:

```
const int i = 0;
```

As a result, these variables are typically placed in read-only data.

Unbundled Features

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

Unbundled in 5.0.6

The following features have been unbundled from MULTI:

- FORTRAN compiler
- CodeBalance
- Distributed Build System
- Remote serial connections
- TraceEdge and In-Memory TimeMachine
- Pre-MULTI-4 debug server protocol compatibility
- EST ICE integration
- PMON ROM monitor integration
- IDT ROM monitor integration

- Green Hills ROM monitor integration
- HP-UX host support

Deprecated Features and Discontinued Support

Deprecated Options

The following options have been deprecated in this release:

Builder Option	Driver Options
ANSI C and Standard C++ Extensions	--discretionary_errors, --discretionary_warnings, -pedantic-errors, -pedantic
Auto-Instantiation of Templates	--auto_instantiation, -template=auto, --no_auto_instantiation, -template=noauto
.c Files are C++	-nodotciscxx, -dotciscxx
Ignore #include Directives	--no_include_never, --include_never, --include_once, --no_include_once
Link Map Method	-default_lnk, -no_default_lnk, -default_i, -no_default_i
Minimum Structure Alignment	--struct_min_alignment
New-Style 'for-init' Code	--no_for_init_diff_warning, --for_init_diff_warning
PCH Messages	--pch_messages, --no_pch_messages, --no_pch_silent, --pch_silent
Precompiled Header File	--no_pch, --use_pch, --pch, --create_pch
Signedness of Enum Type	--unsigned_enum_fields, --signed_enum_fields
Unroll Loops Up to 8 Times	-Ounroll8, -Onounroll8
Use ThreadX Demo Library	-no_use_demo_library, -use_demo_library
Use Before 'Set'	--no_use_before_set_warnings, --use_before_set_warnings
Use Debug Library List (.dli) Files	-no_use_dli_files, -use_dli_files

The ghexfile Utility is No Longer Available

The **ghexfile** utility is no longer available. If you used **ghexfile** in a previous version of MULTI, reconfigure your projects to use **gsrec** instead. **gsrec** includes functionality that was formerly included in the **ghexfile** utility. For more information, see “The gsrec Utility Program” in “Utility Programs” in the *MULTI: Building Applications* book.

Manual Template Instantiation Mode Options No Longer Supported

The following **Manual Template Instantiation Mode** options are no longer supported:

- **-tall**
- **-tlocal**
- **-tused**
- **-tnone**

Programming flash from grun

Programming flash memory from grun is no longer supported in MULTI 5. Similar functionality can be obtained by scripting the flash burn command in the debugger.

Chapter 8

Known Issues

Contents

Installation and Licensing	40
Host Support	40
Target Support	41
Toolchain Issues	41
Target Connection Issues	43
Debugging Issues	43
File System Issues	44

Installation and Licensing

Green Hills Debug Probe CD Compatibility

Some previous revisions of the Green Hills Debug Probes software (shipped with the Green Hills Probe, SuperTrace Probe, and Slingshot) are not compatible with MULTI 5. If your Green Hills Debug Probe CD is not marked “Compatible with MULTI 5,” do not install it with MULTI 5. Contact Green Hills customer support for a replacement CD.

Name of Install Directory Cannot Contain Spaces

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

Host Support

UNC Paths Are Not Supported in the MULTI Debugger

On Windows hosts with MULTI 6.1.4 or older, creating new projects or debugging projects over a UNC path is not supported. In general, we recommend mapping UNC paths to network drives before trying to access them in the MULTI IDE.

UNC Paths to Virtual Machines Are Not Supported

When remotely accessing a MULTI or Green Hills Compiler installation on a Windows virtual machine using a UNC path, you may get errors that the application could not be found or that it could not obtain a license. To work around this issue, map the path to a network drive and use the drive letter instead of the UNC path.

Target Support

Toolchain Issues

No Function Prototypes For INTEGRITY 5 `strdup()` and `strnlen()`

There are no function prototypes available for `strdup()` and `strnlen()` on INTEGRITY 5.

gstack Provides Warnings for Green Hills Library Functions

Some functions in the Green Hills libraries are not annotated for **gstack**. If you link your program with these libraries, you may receive warnings about these Green Hills library functions when using **gstack**.

gstack Provides Warnings for INTEGRITY

INTEGRITY has not been modified to take advantage of new **gstack** features. If you use **gstack** with an INTEGRITY program, you may receive warnings about INTEGRITY functions.

Exported Templates With Inline Functions May Produce Incorrect Code

If your code contains the following:

- A C++ `extern inline` function that contains a class definition that contains a static function definition that contains a static variable
- C++ export templates

the compiler may produce incorrect code.

ThreadX Build Failure When Customizing Startup and System Libraries

If using ThreadX and your project includes customizable **Startup** and **System** library source code, you may get build errors like “[elxr] (error) symbol `__ghsLock`

multiply defined in: ...libsys.a(ind_lock.o) ...tx.a(tx_ghs.o)”. This combination is not supported.

Non-flat Directory Structure in Multi-core Builds Causes Warnings in gcores

If a full path is provided in the output filename of a Core or a SharedMemory, the directory component of the output filename will be truncated and a warning will be issued. Those truncated items will be generated in the project's root directory and will not be cleaned properly by the command **gbuild -clean**.

For example, given the following **.gpj** corresponding to a SharedMemory subproject file:

```
#!gbuild
[SharedMemory]
    -o shared/shared_imp.so
shared.c
```

The output of that build will be placed in **shared_imp.so** instead of in **shared/shared_imp.so** and the following warning will be issued:

```
gcores: warning: The path shared/shared_imp.so has
been flattened to shared_imp.so.
```

Output Filename in Multi-Core Builds with extension different to .mca Causes Warnings in gcores

If the output filename (specified with the option **-o**) of a MultiCoreArchive has no extension, or the extension is different to **.mca**, the extension will automatically be set to **.mca**. That changed output will not be cleaned properly by the command **gbuild -clean**.

For example, if the option **-o demo.mcb** is provided in a MultiCoreArchive project file, the output filename will be changed to **demo.mca** and the following warning will be issued:

```
gcores: warning: The output filename demo.mcb has
been changed to demo.mca
```

Target Connection Issues

Flash Driver Library **libflash.a** Limited to Four CFI Drivers

The flash driver library distributed with the MULTI IDE offers support for Common Flash Interface (CFI)-compatible flash devices. Drivers for these devices are configured at run time. Previous versions of **libflash.a** supported only targets with a single CFI flash device. MULTI 5 now includes support for up to four CFI flash devices. Calls to `FLASH_loadDriver()` on a fifth CFI flash device will return an error code. Multiple calls to `FLASH_loadDriver()` on a single flash base address will return the same CFI driver, even if the hardware is changed or remapped in memory. These limitations do not affect non-CFI drivers. You can force a non-CFI driver by setting the `disableCFI` flag in `FLASH_loadDriver_2()`.

Debugging Issues

Auto Trace Collection Reduces Performance

Automatically-enabled trace collection reduces the performance of simulated targets by a significant amount. This is especially noticeable with INTEGRITY simulators on Solaris.

To work around this issue, disable automatic trace collection. For instructions, see the documentation about enabling and disabling trace collection in the *MULTI: Debugging* book.

Target Agent May Cause Failures When Programming Flash

The MULTI **Fast Flash Programmer** dialog contains an input field for the RAM base address. During a program operation, the flash programmer downloads target agents to this address. If the memory used by the target agent conflicts with reserved memory sections, programming may fail. Error messages printed by the flash programmer may indicate that either a hardware exception was hit or the target agent stalled. Check the link map of your executable or the memory map of the board to confirm that the RAM base address entered in the flash dialog is located after the end of any reserved sections.

References Not Available for Inlined Functions

Issuing the **browseref** or **xref** command or right-clicking a symbol (a function, type, member, variable, etc.) and selecting **Browse References** allows you to view all references of a symbol. However, references are not stored, and consequently not displayed, for inlined functions.

Workaround: Disable inlining by compiling your program with **--no_inlining**. Note that functions defined in header files may be implicitly inlined by the compiler if this option is not used.

Cannot Debug Executables in Eclipse Projects with a Space in the Name

You cannot debug executables in Eclipse projects with a space in the name. Do not put a space in the name of an Eclipse project.

File System Issues

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.

Chapter 9

V850 and RH850 Release Notes

Contents

V850 and RH850 Toolchain	48
MULTI 6.1.4 Compatibility	59

V850 and RH850 Toolchain

Changes and Known Issues in 2013.5

gsrec now supports multi-core archives

gsrec now supports operating on multi-core archives created by the **gcores** utility to produce HEX and S-Record files. Please see “GCores Generate Additional Output” in Chapter 3, “Builder and Driver Options” in *MULTI: Building Applications for Embedded V850 and RH850* for more information.

AUTOSAR and OSEK Operating System Awareness

Provides operating systems awareness (OSA) for ORTI-enabled operating systems such as many AUTOSAR and OSEK operating systems. See its documentation in “AUTOSAR and OSEK Operating System Awareness” in Chapter 11, “Utility Programs” in *MULTI: Building Applications for Embedded V850 and RH850* for more information.

Interrupt routines and nofloat and -no_callt

Interrupt routines written in C using `#pragma ghs nofloat interrupt or -no_callt` build option should be built with the **-inline_prologue** build option. Otherwise library routines may be used to save and restore registers, and these routines will save and restore system registers relating to the FPU and to the call table unconditionally.

Software Trace Logging Support

Support for the software trace functionality of the RH850 family of processors has been added in this release. Please refer to “Software Trace Logging Support” in Chapter 2, “Developing for V850 and RH850” in *MULTI: Building Applications for Embedded V850 and RH850* for more details.

Preventing the use of double-precision floating point

Two new options have been added to give users greater control over restricting the usage of double-precision floating point in their programs. Please refer to “Warn for types double or long double” in Chapter 3, “Builder and Driver Options” in *MULTI: Building Applications for Embedded V850 and RH850* and “Do not allow types double or long double” in Chapter 3, “Builder and Driver Options” in *MULTI: Building Applications for Embedded V850 and RH850* for more details.

gcores -target option deprecated

For users of the **gcores** utility, the **-target** option has been deprecated. Using it will result in an error or warning. Users should instead use the new **-driver** option. For more information about this new option, please refer to “GCores Driver” in Chapter 3, “Builder and Driver Options” in *MULTI: Building Applications for Embedded V850 and RH850*.

RH850G3x loss of target control with some trace options

For RH850 hardware with 850eserv2, if the "Real-Time Trace Mode" option is disabled and either the "When Trace Buffer Fills Since Last Halt" option is set to "Stop Trace" or a Delay Trace trigger is set and "When Delay Trigger is Hit" is set to "Stop Trace at Delay Trigger", control of the target may be lost. This is due to a limitation of the current hardware. If the problem comes up, disconnect from 850eserv2, power cycle the target, and reconnect to 850eserv2.

Name change from RH850/F1A to RH850/F1L

The New Project Wizard now refers to the RH850/F1L board, not the RH850/F1A. The corresponding command line option is `-bsp=rh850_f1l`.

Changes and Known Issues in 2013.1

RH850G3x support

Support for the RH850G3K, RH850G3M and RH850G3H devices is now included in this release. The previous RH850 setting available in 2012.5 (**-cpu=rh850**) is now treated as an RH850G3K (same as **-cpu=rh850g3k**). When connecting to the simulator or other debug server targeting the RH850G3H or RH850G3K, the debugger may still report them as the RH850G3M. This does not affect support for these devices, and will change in a future release to reflect the correct CPU. When using the **ccrh850** driver, if no CPU is specified on the command line, then the default setting will be RH850G3K. Please note that whenever the CPU is specified as RH850G3K, software floating-point will be enabled by default (**-fsoft**). To enable hardware floating-point instead, please specify **-fhard** or **-fsingle** as appropriate for your floating-point unit. Please refer to “Floating-Point Mode” in Chapter 3, “Builder and Driver Options” in *MULTI: Building Applications for Embedded V850 and RH850* for more details on these FPU settings.

RH850G3x trace support

The **850eserv2** debug server included in this release supports the collection of trace from trace-enabled hardware using the Renesas E1 emulator. For multi-core configurations, the debug server currently has the limitation that any trace configuration setting made will affect all cores. Per-core trace setting support is planned for a future release. Multi-core trace support is also included in this release, however, it is provided in preliminary form only, and should be considered alpha release quality.

RH850G3x loss of target control with some trace options

For RH850 hardware with 850eserv2, if the "Real-Time Trace Mode" option is disabled and either the "When Trace Buffer Fills Since Last Halt" option is set to "Stop Trace" or a Delay Trace trigger is set and "When Delay Trigger is Hit" is set to "Stop Trace at Delay Trigger", control of the target may be lost, and attempting to halt would give errors such as "user system err (cannot break)". This is due to a limitation of the current version of the Renesas EXEC library shipped with this release. If the problem comes up, disconnect from 850eserv2, power cycle the target,

and reconnect to 850eserv2. This will be resolved in a future release of the Renesas EXEC library. Users experiencing this problem should contact support@ghs.com to see if an updated Renesas EXEC library patch is available.

PUSHSP/POPSP support

The options **-push_pop** and **-no_push_pop** have been added to control the compiler's use of the `PUSHSP` and `POPSP` instructions. See “Use pushsp and popsp instructions” in Chapter 3, “Builder and Driver Options” in *MULTI: Building Applications for Embedded V850 and RH850* for more information.

PREPARE/DISPOSE support

The option **-prepare_dispose** is now on by default whenever the option **-no_callt** is used. This represents a change from previous default behavior. See “Epilogues and Prologues via prepare and dispose” in Chapter 3, “Builder and Driver Options” in *MULTI: Building Applications for Embedded V850 and RH850* for more information.

SIMD support in interrupt routines

The compiler in this release does not generate code to save or restore the SIMD coprocessor state for interrupt routines. Interrupt routines should either not modify the SIMD coprocessor state, or should use assembly code to manually save its state beforehand.

FPU version 3 support

The version 3 FPU supported on most RH850 devices is now supported and enabled by default when compiling for RH850 devices. Users who are using an early-generation RH850 device may need to enable the legacy version 2 FPU support. Consult your device manual for more information. For information about how to enable the legacy FPU support, please see “Use FPU 3.0 instructions (RH850 and later)” in Chapter 3, “Builder and Driver Options” in *MULTI: Building Applications for Embedded V850 and RH850* for more information.

Coprocessor initialization

The compiler library startup code for RH850 processors will now initialize the PSW.CU0 and PSW.CU1 register fields to enable these coprocessors on implementations which support them.

Legacy simulator for V850 and RH850 (sim850)

The legacy **sim850** simulator has been discontinued. It has been replaced with the new multi-core simulator for V850 and RH850, **simrh850**. A **sim850** executable is still included in this release for compatibility with project connections from previous releases. However, it simply invokes **simrh850** with the appropriate options. Users should transition existing projects from previous releases to use the new simulator. The new simulator, in addition to full multi-core and shared memory support, is also significantly faster at simulation than the legacy simulator. For more information about using the new simulator, please refer to Chapter 5, “Multi-core Simulator for V850 and RH850 (simrh850) Connections” in *MULTI: Configuring Connections for V850 and RH850 Targets*.

simrh850 coprocessor simulation

The **simrh850** simulator now simulates the coprocessor fields of the PSW register available on RH850 processors, specifically the PSW.CU0 and PSW.CU1 fields. By default, the simulator will rely on information provided by the compiler in the ELF file to set determine whether or not it will simulate the appropriate coprocessor as enabled or disabled. If the coprocessors are disabled and an associated coprocessor instruction is encountered, the simulator will simulate a coprocessor exception and print an error. Users can forcefully disable the coprocessors in the simulator using the **-fpu=none** and **-no_rh850_simd** options. Please refer to Chapter 5, “Multi-core Simulator for V850 and RH850 (simrh850) Connections” in *MULTI: Configuring Connections for V850 and RH850 Targets* for more details.

simrh850 hyper-threading simulation

The **simrh850** simulator now reflects recent changes to the RH850 specification and resets the HTSCCTL.RND register field to zero. In previous releases it was

reset to 0x3F. Users of previous releases should take care to ensure they initialize this register appropriately.

Extended GNU asm support in V850 and RH850

The option `--gnu_asm` is supported and enabled by default. See Chapter 19, “GNU Extended Assembly” in *MULTI: Building Applications for Embedded V850 and RH850* for more information.

Renesas RH850 E1x connections

The New Project Wizard does not support the multi-core E1x board. To connect to the board, invoke 850eserv2 with options similar to the following:

```
connect 850eserv2 -rh850 -ellpd4=11000 \  
-df=c:/ghs/comp_201315/rh850/dr7f701z006ebg.dvf \  
-dclock=4000,0,swoff -id ffffffffffffffffffffffffffffffffff \  
-fastflashload
```

Renesas RH850 Data Trace

Trace packets for loads and stores can sometimes be dropped or issued out-of-order by the hardware, resulting in data trace packets that cannot be correlated with the appropriate instructions in the trace list.

Stop Execution on Trace Buffer Full for Renesas RH850 E1x

When the Buffer Full trace option is set to Stop Execution on the multi-core E1x board, the behavior is unreliable. Sometimes both cores halt with a full trace buffer message at the same time, but other times only a single core will display the full trace buffer message. Retrieving trace in the latter case can lead to problems with decoding the data.

Trace Delay Triggers Across System Calls

When a trace delay trigger is set to disable trace for a certain condition, such as execution of a specific address, trace may be re-enabled upon execution of a system

call (for example, during printf). This may lead to the trace list containing extra instructions or overflowing.

Errors when re-enabling trace on multi-core E1x board

On the multi-core E1x board, when trace is manually disabled and re-enabled repeatedly, it is possible to get an error from 850eserv2 like the following:

```
Target: BRS condition write error 0x404: param err (event num overflow)
```

Changes and Known Issues in 2012.5

RH850 Support

Full support for the RH850 processor family is included in this release. This includes new CPU options (**-cpu=rh850**), new command line drivers (**ccrh850** and **cxrh850**) and a new simulator (**simrh850**). The new simulator supports multi-core simulation of the V850 and RH850 processors, as well as simulation of hyper-threading on RH850 processors. See Part I. Using the MULTI Compiler in *MULTI: Building Applications for Embedded V850 and RH850* and Chapter 5, “Multi-core Simulator for V850 and RH850 (simrh850) Connections” in *MULTI: Configuring Connections for V850 and RH850 Targets* for more details.

Multi-core Support

Multi-core building and debugging is supported in this release. This includes the new utility program **gcores**, the new driver options: **-exportall** and **-exported_absolutes_only**, the new project files types: MultiCoreArchive, Core and SharedMemory. See “The gcores Utility Program” in Chapter 11, “Utility Programs” in *MULTI: Building Applications for Embedded V850 and RH850* for more information about building for multiple cores. Multi-core debugging is supported in the new MULTI provided in this release, including as well a new configuration file type **.ghsmc**. Please see “Multi-Core Debugging” in Chapter 27, “Freeze-Mode Debugging and OS-Awareness” in *MULTI: Debugging* for more detailed information.

V850 and RH850 Target Connections

Renesas E1 Emulator support in 850eserv2

E1 Emulator support in 850eserv2 is included in this release. See Chapter 3, “Renesas V850/V850E ICE (850eserv2) Connections” in *MULTI: Configuring Connections for V850 and RH850 Targets* for more details.

Renesas V850/V850E ICE connections (850eserv2)

In the Connection Editor, selecting the box labeled “Write 0's into BSS section” has no effect.

When initially connecting to 850eserv2, if you abort the connection attempt, the target hardware may need to be power-cycled before you can connect to it again.

Limitation in Retrieve Trace when Buffer Fills Option

The trace option “Retrieve trace when buffer fills” has a limitation for V850 targets that trace might not be stopped immediately when the buffer fills. This means that some trace from the beginning of the buffer may be missing.

Some unsupported trace trigger conditions shown for IECUBE2

For trace on IECUBE2, the **Set Triggers** window shows some trigger conditions that are not supported by the hardware. Selecting a condition that involves data reads or writes such as **Data Written** will result in an error message like “Trace event write error 0xc60: param err (invalid event condition)”.

Problem When Using Section Sub-Switch Off to Enable Trace

For trace with the Midas ICE, e.g., for the KIT-V850E2/MN4, using the Section Sub-Switch Off trace options to enable PC or data trace does not work properly. Some instructions/data will not appear while Section Sub-Switch is Off.

SuperTrace Probe with Renesas IE Cube Deprecated

Support for using a SuperTrace Probe to collect trace from a Renesas IE Cube is deprecated in this release and may be removed from a future release.

V850 and RH850 Debugging

Spawn Window Script

A new Python script allows you to spawn a Debugger window for a target listed in the current Debugger window. If the new Debugger window is spawned for the currently selected target, the script attempts to select a different target in the target list.

You can specify the script with or without a target name. If specified without a target name, it is applied to the currently selected target.

You can either use the script as a script or you can import and run it. To run it as a script from the Debugger command pane, enter:

```
py -f compiler_installation/python/ghs_examples/spawn/spawn.py
```

or

```
py -f compiler_installation/python/ghs_examples/spawn/spawn.py \  
    target_name
```

To import and run it from a script, from the **Py** pane, the **Py Window**, or the **MULTI-Python GUI**, use:

```
sys.path.append("compiler_installation/python/ghs_examples/spawn");  
import spawn;  
spawn.spawn();
```

or

```
spawn.spawn("target_name");
```

Changes and Known Issues in 2012.1

Improvements to V850E CALLT usage

In previous releases of the tools the compiler would, when compiled with `-callt` (the default), optionally use the CALLT instruction for certain prologue and epilogue space-saving optimizations. The use of this instruction, and the method with which it is implemented, has changed. This creates some important noteworthy differences in this release that customers making use of this feature should be aware of. Please see “Generate a backwards-compatible (MULTI 5.1.x and earlier) call table in the linker” in Chapter 3, “Builder and Driver Options” in *MULTI: Building Applications for Embedded V850 and RH850* and “Function Call Pragma Directives” in Chapter 2, “Developing for V850 and RH850” in *MULTI: Building Applications for Embedded V850 and RH850* for more details.

V850 Global Registers (-globalreg)

The V850 global registers (those declared with the “register” keyword) are always r20-r24; previous release notes indicated additional registers may be used, but that is no longer the case.

V850E1F Signed Zero Treatment

A new driver option has been added to the tools as of MULTI 5 which provides some control over the treatment of floating point zero values. This option is `-precise_signed_zero_compare` and its corresponding antithesis `-no_precise_signed_zero_compare`. The behavior of the compiler for the V850E1F architecture in older versions of the compiler does not correspond to the default value of this new option. Users wishing to revert to previous behavior should use `-no_precise_signed_zero_compare`. Users should consult their manual before enabling this option so that they are fully aware of the consequences of its use.

V850 TDA Support

`-notda` is now the default compilation mode with V850. The linker will give an error when a user attempts to link a module compiled with `-notda` together with one that uses the TDA.

Users migrating from older releases who want to use TDA should compile with `-single_tda`

EP Register Usage

`-notda` is now the default compilation mode with V850. See the note above on V850 TDA Support.

Additionally, `-allocate_ep` is also now the default compilation mode with V850. In this mode, EP is allocated to as a temporary register.

Users migrating from older releases who want to disable TDA and limit the use of the EP register should compile with `-no_allocate_ep`.

Force Save/Restore of R4/R5 for Interrupt Service Routines

It is now possible to force the save and subsequent restore of the small data area (SDA) and read-only small data area (ROSDA) base registers (r4 and r5) during the prologue and epilogue of interrupt service routines. A new command line driver option `-v850_isr_save_r4r5` enables this behavior. This behavior is off by default.

Earlier releases of the V850 tools would save and restore these registers under certain circumstances. As that was a violation of the ABI, and resulted in inefficient code generation, that behavior was removed. However, customers who relied on this behavior can use this new option to retain this previous behavior. Most customers will not need to use this option.

V850E and Later C/C++ Interrupt Routines Compiled with `-no_callt`

In previous releases, a library compatibility issue existed when compiling a C/C++ interrupt routine with the `-no_callt` driver option, the `-Osize` size optimization, and the default `-no_inline_prologue` driver option enabled. This issue was addressed in MULTI 5, but requires that C/C++ interrupt routines built with these options and targeting V850E and later cores be recompiled to ensure compatibility with libraries going forward.

For reference, C/C++ interrupt routines are those routines declared with the `__interrupt` keyword or the `#pragma ghs interrupt pragma`.

V850E -no_v850e_mul_errata Option

When targeting the V850E processor (**-cpu=v850e**), the compiler does not by default emit instructions of the following forms:

```
mul reg1, reg2, reg3
    mulu reg1, reg2, reg3
    mul imm1, reg2, reg3
    mulu imm1, reg2, reg3
```

where *reg3* is *r0*, or *reg2* is the same as *reg3*. This is to work around an erratum with the V850ES implementation.

The assembler and linker flag any occurrences of the above instructions, but do not attempt to change them.

To disable both the checks in the assembler and linker, and the compiler's avoidance of these instructions, specify the **-no_v850e_mul_errata** driver option.

MULTI 6.1.4 Compatibility

This Compiler release should be used with the version of MULTI provided on this CD or the SH CD. If you have MULTI 6.1.4 installed from a previous version of V800 or SH Compiler 2013.1, 2012.5 or 2012.1 CD, or Blackfin, PPC, ARM, x86, or MIPS Compiler 2013.1, 2012.5 or 2012.1 CD, or MULTI 6.1, you can install a patch to MULTI to make it compatible with this release, using the following steps:

- Change into the **multi-v6.1.4-patch** directory on the CD.
- Run **install.bat** (on Windows) or **install.sh** (on Linux or Solaris).
- When prompted for an Installation Directory, enter your existing MULTI installation directory.

The patch includes trace enhancements specific to V850, support for the RH850 SIMD unit, and support for multi-core building.

