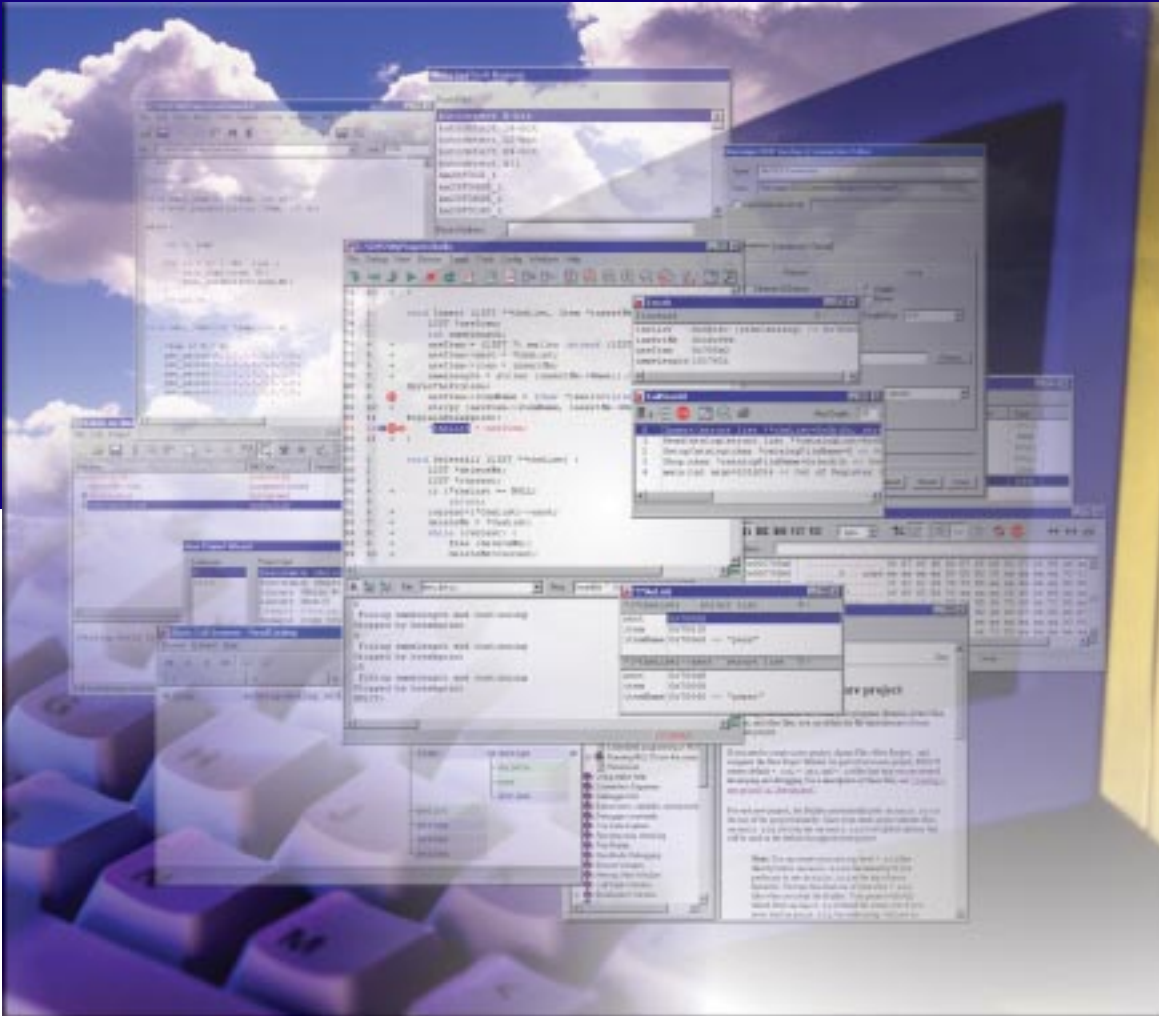


# MULTI<sup>®</sup>

## INTEGRATED DEVELOPMENT ENVIRONMENT



▪ TOTAL SOLUTIONS FOR EMBEDDED DEVELOPMENT ▪

  
**Green Hills**  
• SOFTWARE, INC. •

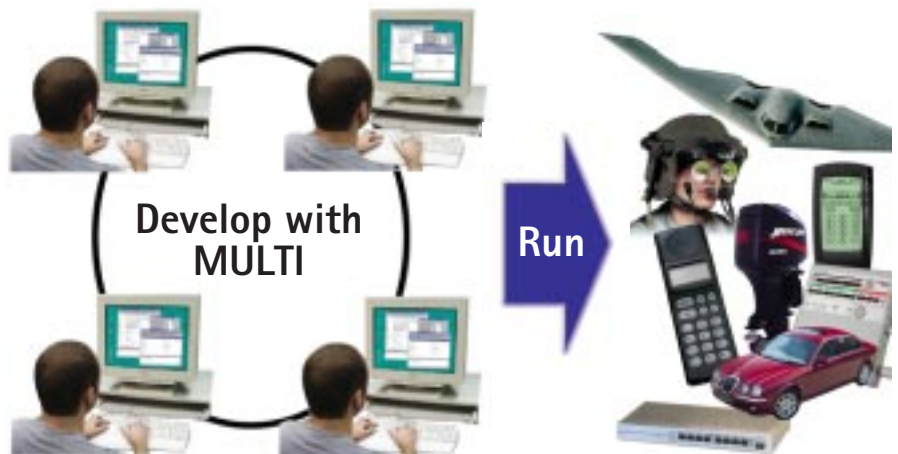
# MULTI® INTEGRATED DEVELOPMENT ENVIRONMENT

**M**ULTI® is a complete integrated development environment for embedded applications using C, C++, Embedded C++, Ada 95, and FORTRAN. MULTI runs on Windows, Linux, or UNIX hosts and supports remote debugging to a variety of target environments. MULTI provides a direct graphical interface with all Green Hills compilers, and supports multi-language development and debugging. MULTI contains all of the tools needed to complete a major programming project:

- **Project Builder** - Provides an intuitive GUI to configure and build complex programming projects.
- **Source-Level Debugger** - Provides a powerful windowing source-level debugger that enables program loading, execution, run control, and monitoring
- **Performance Profiler** - Provides detailed information to enable the developer to identify the code where execution consumes the greatest amount of time as well as how many times a given procedure or block of code was executed.
- **EventAnalyzer™** - Enables programmers to understand the complex real-time interactions of their systems.

***“INTEGRATED DEVELOPMENT ENVIRONMENTS LIKE MULTI ARE PROVING KEY TO THE DEVELOPMENT OF THE SOPHISTICATED PROGRAMS THAT RUN ON HIGH-PERFORMANCE ARM MICROPROCESSORS.”***  
**RICHARD PHELAN - ARM LTD.**

- **Run-time Error Checking** - Provides useful information on a wide variety of run-time errors.
- **Code Coverage Analysis** - Improves product quality by identifying areas of code that have or have not been executed.



*From medical products, to automotive systems, to avionics, and to telecommunications, the MULTI Integrated Development Environment is ideal for any embedded application.*

- **Version Control System** - Provides a seamless integration with the rest of the MULTI components making version control of projects efficient and unobtrusive.
- **Text Editor** - Provides a full-featured, developer-centered editor with an easy-to-understand interface that doesn't force users to memorize arcane key sequences or editing modes.
- **Graphical Browser** - Provides source code browsing capabilities integrated with the MULTI Editor and Debugger.

Taken separately, each component of MULTI is one of the most advanced tools of its class. But MULTI is much more than a useful collection of software tools; it is a truly integrated development environment. MULTI's components are aware of each other and communicate among themselves in new and very useful ways, making the whole much greater than the sum.

Take, for example, the simple case of the edit-compile-debug cycle. When an error is discovered with MULTI's Debugger, an Editor can be brought up

on the offending line with a single button click. As soon as modifications are made to the file, the Editor automatically checks it out of the Version Control System. Another button click saves the changes and checks the file back in, prompting for a comment. A final click builds the program and reinitializes the Debugger.

## COMMON DESKTOP ENVIRONMENT

MULTI provides the user with a single development environment that is constant across all workstations and PCs. MULTI works the same on every platform: the features, the commands, and the usage are all identical. The user no longer has to worry about host-to-host incompatibilities in languages and *makefiles*.

With MULTI it really is possible to have a large team of programmers, each using a workstation from a different vendor, working together to develop a single program. The Program Builder and source control system operate across the network of mixed machines to tie your programmers into a single coordinated development team.

## COMPILER SUPPORT

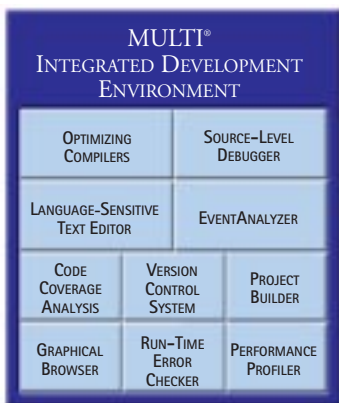
MULTI can be used with Green Hills Software's optimizing C, C++, Embedded C++, Ada 95, and FORTRAN compilers or with other compilers that adhere to EABI standards. Our compilers produce EABI compliant ELF object modules.

## RTOS-AWARE

MULTI supports a wide variety of real-time operating systems (RTOS). Because MULTI is fully RTOS aware, designers can debug and tune their applications at a task level. Within the MULTI Debugger, designers working with RTOSes from Green Hills Software's, a third party, or a home grown RTOS, can start and stop tasks, set task-specific breakpoints, and monitor OS resources like buffers, queues, and streams. They can also debug new tasks as they are spawned and watch processes as they communicate through pipes, semaphores, etc.

***"MORE EEMBC SEMICONDUCTOR MEMBERS HAVE CHOSE THE GREEN HILLS COMPILER TO GENERATE THEIR BEST EEMBC BENCHMARK SCORES THAN ANY OTHER COMPILER IN THE INDUSTRY."***

***MARKUS LEVY - EEMBC***



## MULTI-PROCESS SUPPORT

Debugging multi-process programs provides its own unique challenges, and MULTI provides a number of features to assist programmers in this area. The most important of these is that MULTI provides a separate debug window for each process. Users can independently single step, set breakpoints and examine variables in each process, and watch the processes as they communicate with each other using pipes, message queues and shared memory. MULTI can attach to a process that is already running and halt it. MULTI can also detach from a process and allow it to continue on its own.

## MULTIPROCESSOR SUPPORT

MULTI is uniquely suited for developing programs on systems consisting of multiple processors because it makes no distinction as to whether the processes being debugged are on the same or on different processors. Since each debug window independently asks its debug server for the model of the processor to which it is attached, MULTI is able to debug systems with multiple mixed processors.

*Because it treats each process being debugged independently, MULTI is uniquely suited for developing programs on systems with multiple, mixed processors.*

## SUPPORTED PROCESSORS

Green Hills Software provides C, C++, Embedded C++, Ada 95, and FORTRAN optimizing compilers, MULTI Integrated Development Environment, and the Tool Chain for many processor architectures:

### CISC:

- CPU32
- 680x0/683xx
- x86/Pentium

### CISC/RISC:

- ColdFire
- FR20
- i960
- PowerPC
- V850

### RISC:

- Alpha
- ARM/Thumb, StrongArm, XScale
- Lexra
- M•CORE
- MIPS, MIPS16, MIPS32, MIPS64
- RAD6000
- RH32
- SH
- SPARC/SPARCite

### DSP:

- AltiVec
- FirePath
- SH-DSP
- ST100
- StarCore
- TriCore
- ZSP

## SUPPORTED HOST SYSTEMS

Green Hills Software supports development on a variety of popular PC and UNIX workstation hosts:

- Linux
- PA-RISC/HP-UX
- SPARC/Solaris
- Windows 9x/ME/NT/2000/XP

# SOURCE-LEVEL DEBUGGER

## MULTI SOURCE-LEVEL DEBUGGER

The MULTI Source-level Debugger is a powerful windowing debugger that enables program loading, execution, run control, and monitoring. MULTI provides debugging in C, C++, Embedded C++, Ada 95, FORTRAN, and assembly. MULTI's basic functions are intuitive and easy to use; programmers spend less time learning the tools and more time generating code.

MULTI offers advanced features for viewing data during execution, easy direct links to the MULTI Text Editor, customizable menus and command buttons, as well as an open interface to editors and version control software of choice.

### MAIN DEBUGGER WINDOW

The main debugger window consists of three panes. The upper pane is the tool bar and below the source pane. The bottom command pane is used to enter typed commands and display the results of commands.

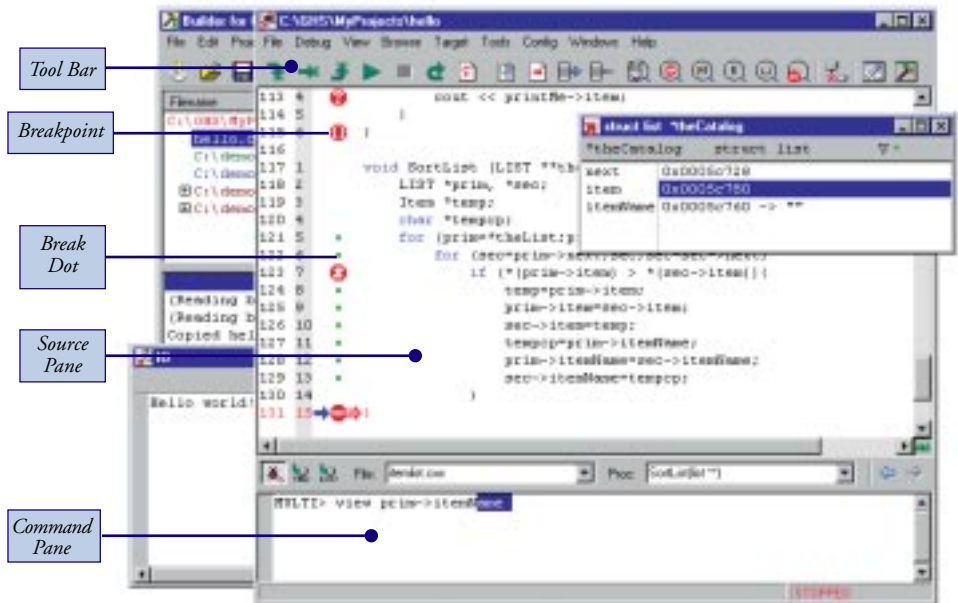
### FULL-FEATURED DEBUGGING

MULTI provides an exhaustive set of debugging features, designed to make it easy to find and fix bugs:

- **Setting Breakpoints** - A single-click on the green "break-dot" next to each executable line of source or assembly code sets a breakpoint. A stop sign will replace the dot to show that a breakpoint is set. Another click on the stop sign clears the breakpoint.
- **Conditional and Command Breakpoints** - Type language syntax onto the command pane or use the dialog box GUI to specify conditions and action clauses for breakpoints.

- **Single-Stepping** - Stepping "into" (*Step* button) traces execution of every individual instruction even when functions are called. Stepping "over" (*Next* button) does not trace into the called function. Stepping "out of" (*Return* button) brings execution back to the calling function.
- **Examining Variables** - Clicking on the name of a variable in the source pane causes the value of the variable to be displayed in the command pane. Pointers are recognized as a special case and the value of the object pointed to is also displayed. Clicking on the name of a structure causes the whole structure to be displayed, with every structure or class element labeled.
- **Browse To Source Code For Functions** - Clicking on a function name automatically brings up the editor on the source code for that procedure.

- **Display Stack Trace** - Shows a window displaying a stack trace. This allows the programmer to follow the progress of a program through its hierarchical call structure. The Stack Trace displays the name of each procedure, and the name and value of each argument and of local variables.
- **Incremental Debug** - Divides application debug symbol information generated by the compiler into separate files relevant to the code that is currently being debugged. This increases responsiveness when debugging very large projects. Incremental Debug also provides: access from an open API to the advanced debug and variable lifetime information, debugging support of optimized code, and seamless debugging of applications across different host platforms.
- **Local Variables Display** - Shows the variables local to the current function. When moving from function to function, the contents of the window will change to display the local variables of the current function being viewed.



The MULTI Debugger uses windows extensively to greatly simplify complex debugging tasks, most of which have been reduced to a single mouse click. While you can enter a full range of typed commands into the command pane, in practice your hands will rarely need to touch the keyboard.

- **Data Explorer Windows** - Double clicking on a variable name creates a window that displays the variable, its type, and current value. The variable may be of any type (integer, structure, array, or class). The window display is updated to show value changes every time the program stops. If the display window contains a pointer, double clicking on it opens a new data explorer window and displays the variable pointed to, making it easy to follow a linked list or other complex data structure.

Data explorer windows can be “frozen” to save a view of values at a given point in order to compare them with values at subsequent points in the program’s execution.

Taking full advantage of MULTI’s new advanced Incremental Debug format and improved processing of C++ symbol information, MULTI has implemented several advanced C++ debugging capabilities, including:

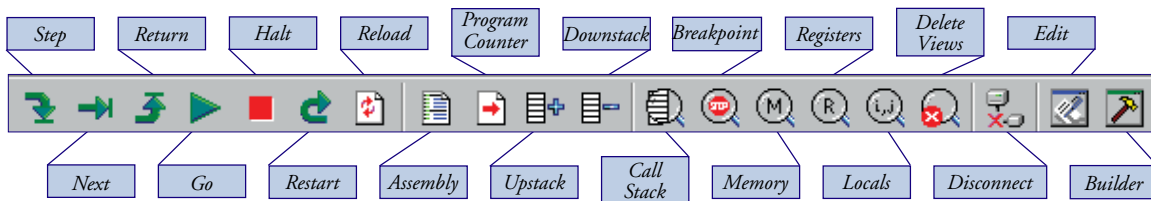
- Full namespace debugging support
- C++ function navigation with ambiguity resolution
- Advanced class browsing
- Debugging of C++ exception throws and long jumps

The same powerful display capabilities available for viewing variables in a C program have been extended to fully support the display of C++ objects.

## COMMAND LANGUAGE COMMAND-LINE PROCEDURE CALLS

From the MULTI command pane, you can execute any debugging command also accessible via a button or menu item in the debugger. You can save these commands to a script that can be run at the command prompt or automatically when your program is debugged. In this way you can automate your most common debugging tasks.

Additionally, you can evaluate expression and call functions within your program. The function is called and runs on your target hardware or simulator. You can call any function linked into your program, including C++ class-member functions.



The MULTI Source-level Debugger Tool Bar

## EXTENSIVE C++ SUPPORT

MULTI is a true C++ development environment with C++ support seamlessly integrated at every level. MULTI’s Source-level Debugger evaluates C++ expressions and understands C++ namespace resolution, overloaded function resolution, and copy constructors and coercion operators. Casts and coercions are performed implicitly, and ambiguities regarding overloaded operators and functions are automatically resolved. MULTI is fully aware of the complex features of the C++ language. This makes it unnecessary to mentally translate C++ constructs into the mechanisms used by the compiler. MULTI can also call C++ member functions from the command pane.

Clicking on the name of a C++ object displays its members in the command pane; double clicking on the name of the C++ object displays its members in a special display window where they can be modified or monitored as the user steps through the program.

C++ programmers can access the following important information through MULTI’s graphical user interface:

- C++ inheritance graphs
- C++ classes, casts and coercions
- Static fields in classes
- Anonymous unions in classes
- Implicit *this* and *casts*
- Overload ambiguity resolution
- Calling overloaded operators and functions
- References
- True object types

## TEMPLATE FUNCTIONS

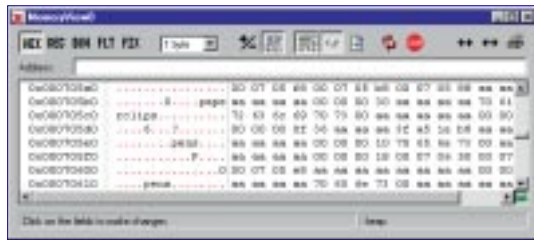
MULTI supports the debugging of template functions. With templates, C++ automatically generates multiple instances of a function, one for each combination of argument types actually used in the program. Every time the function is referenced, the C++ compiler determines which specific function to use, based on the argument types.

**“WE SELECTED MULTI AS THE DEVELOPMENT ENVIRONMENT FOR OUR HIGHLY INTEGRATED NET+OS PRODUCT BECAUSE IT OFFERS OUR CUSTOMERS EXCELLENT VISIBILITY INTO THEIR APPLICATION CODE WITH ALL THE FEATURES TO CATCH EVEN THE TOUGHEST BUGS. MULTI SIMPLY GETS OUR CUSTOMERS TO MARKET FASTER.”**

**DICK ANDERSON - NETSILICON, INC.**

## MEMORY VIEWER

MULTI enables the user to view the contents of memory in a flexible and powerful manner. MULTI's *memview*, which can be invoked through a user-defined button or command line statement, provides a full address/contents table of memory that can be formatted in a variety of ways including hexadecimal, decimal, ASCII, binary, real, and many more. Memory regions can be specified symbolically or in



*Memory Viewer with pull down*

terms of absolute or relative addresses. Multiple *memview* windows can be opened to view several memory regions simultaneously.

## MIXED LANGUAGE DEBUGGING

MULTI supports mixed language debugging in C, C++, Embedded C++, Ada 95, FORTRAN, and Assembly. When moving among modules of different languages, MULTI recognizes the language of the currently viewed file and adjusts its expression evaluation and data display accordingly. Also, command line input uses the syntax of the language being debugged. Ada and

FORTRAN programmers will appreciate a debugger that understands these languages and doesn't require them to struggle with a debugger intended for C programs. For example, arrays may be displayed in either row major order or column major order, according to the conventions of the language.

## MULTIPROCESSOR DEBUGGING

MULTI was designed to support multiprocessor, distributed systems.

Depending on the capabilities of the target connection, one instance of MULTI can debug application code running on multiple processors. A growing industry trend is to place multiple processors, homogeneous or heterogeneous on the same die, utilizing a single JTAG connection for debugging all of the processors.

## CROSS REFERENCE DEBUGGING

Allows users to find the definition of, declaration of, assignment of, or reference to any symbol in their program. This kind of debugging offers the power of being able to see what role any variable or procedure plays in the user's program.

## FIELD DEBUGGING

MULTI supports non-intrusive debugging of live systems. Developers can gather information from a running system without the risks involved in halting the system or any process. Data gathering information occurs and accrues on the target, without requiring an active connection to the host development system. The data can then be safely uploaded to the host development system for later analysis.



*Register View Window, highlighting changes*

## REGISTER VISIBILITY

The MULTI Debugger provides a fully customizable, hierarchical register view window. This window enables the programmer to visualize all register resources (hardware and memory mapped) that are of interest. This is useful when dealing with custom peripherals and processors that have many peripherals on-chip (such as the PowerPC 8260).

## UML DESIGN TOOLS SUPPORT

MULTI is fully integrated with the Rational Rose® RealTime UML development environment. Green Hills' optimization technology compiles the model generated C or C++ source code. MULTI's target connectivity downloads the compiled model to the target system. MULTI cooperates with Rose RealTime, to simultaneously allow UML level debugging of the model and source level debugging of the deployed components.

**"...[THE] MULTI IDE ENABLED US TO PROVIDE A TURNKEY DEVELOPMENT AND TARGET ENVIRONMENT THAT MAXIMIZES PROGRAMMER PRODUCTIVITY AND ENHANCES RELIABILITY AT AN AFFORDABLE PRICE."**

**ANDREW KEYSER - BFGOODRICH**

## TARGET ENVIRONMENTS

MULTI supports Host-Target debugging in a wide variety of combinations. From Windows or UNIX hosts, MULTI supports remote target debugging through On-chip Debug, Commercial RTOSes, Instruction Set Simulators, In-Circuit Emulators, and ROM Monitors. In each case, MULTI communicates using a standard debugging API to a software “debug server”, which uses a target specific interface to connect MULTI to the target environment.

## ON-CHIP DEBUG SOLUTIONS

MULTI can cross-debug to on-chip debug (OCD) targets using the high-performance Green Hills Probe™, Slingshot™, Agilent Emulation Probe, Macraigor Raven, Hitachi E10 and E10A, EST visionICE, or an inexpensive parallel interface (“Wiggler”). OCD targets include most popular 32-bit and 64-bit microprocessors.

## REAL-TIME OPERATING SYSTEMS

MULTI is fully integrated with a wide variety of commercial RTOS targets through the use of MULTI’s multitasking debug servers:

- Green Hills Software’s INTEGRITY®
- ThreadX®
- OSE™
- VxWorks®, Tornado®, TornSIM, and VxSim
- User-developed home grown

## INSTRUCTION SET SIMULATORS

MULTI can cross-debug to Instruction Set Simulators for most target processors.

- ARM, Ltd. ARMulator
- MIPS Technologies, Inc. MIPSsim
- Green Hills Instruction Set Simulators

## IN-CIRCUIT EMULATORS

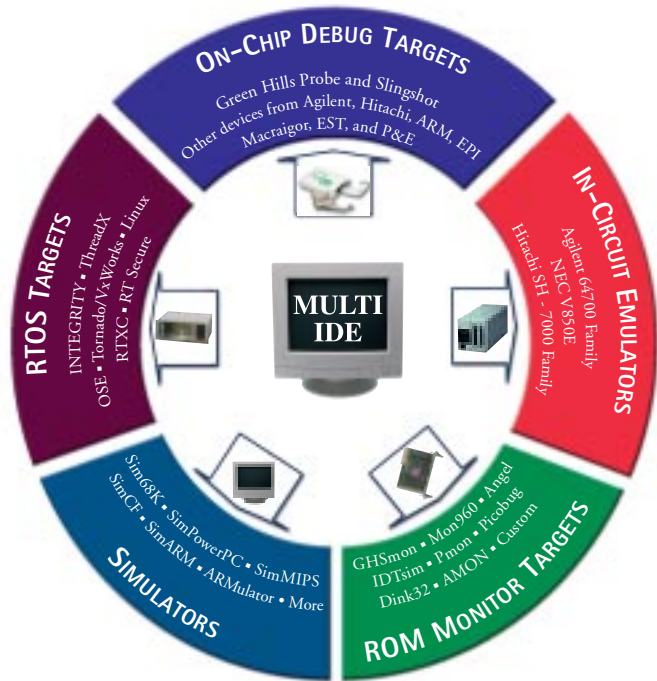
MULTI can cross-debug to In-Circuit Emulators:

- Agilent 64700 ICE for 680x0/683xx and i386DX/EX
- NEC V850 ICE for V850

## ROM MONITORS

Using RS232 and in some cases Ethernet or parallel ports, MULTI can cross-debug to the following ROM Monitors’ targets:

- **MON960** - Intel i960
- **IDTSim** - IDT MIPS
- **Angel** - ARM
- **LSI Pmon** - LSI MIPS
- **Picobug** - M•CORE
- **AMON** - Hitachi SH PowerPC
- **GHSMon** - a wide variety of processors



*MULTI can be used to develop programs on many different kinds of targets, including simulators, ROM monitors, in-circuit emulators, and real-time operating systems. Every target system is controlled by a debug server, which generally resides on a host workstation. To control a specific target, MULTI establishes a connection with the appropriate debug server.*

## TARGET CONNECTION EDITOR

Connecting to hardware just got easier with MULTI’s Target Connection Editor. Simply select the debug device or method and tell MULTI how you want to connect. An easy to use interface guides users through the process of connecting to hardware for the first time. Once a connection method has been created, save it in MULTI’s Target Connection Organizer for future use. When a project is created using the New Project Wizard, potential connection methods are created for the user.



*The Target Connection Editor lets the user create, save and reuse connection commands that make it easy to access target hardware from MULTI.*



## MULTI VERSION CONTROL SYSTEM (MVC)

The MULTI Version Control System (MVC) is seamlessly integrated into the MULTI IDE making version control of projects efficient and unobtrusive. The most common version control operations are automatic. For example, the MULTI Text Editor will “checkout” a file being edited when the first change is made. If the edit is aborted, the checkout will be aborted as well. When the file is closed, MULTI prompts the user for comments describing the edit. MULTI then automatically checks the file back in. MVC provides version control of application source files, and enables change history and graphical display of version differences. MVC provides:

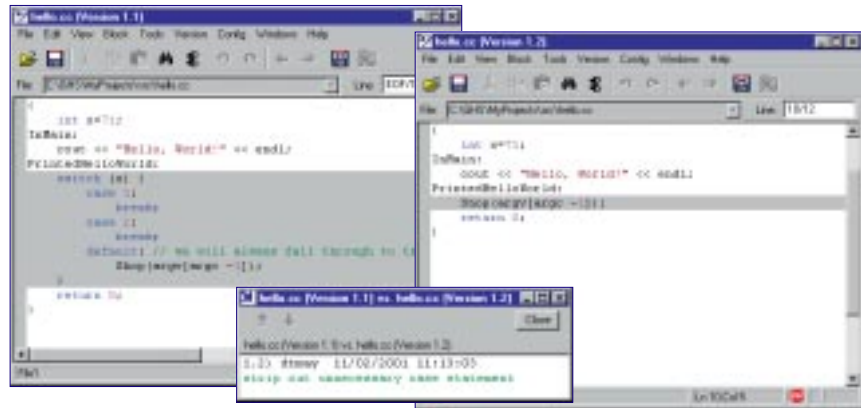
- Automatic checkout and checkin from Editor
- Prompt for comments
- Display of change history
- Show last edit for region of code
- Merge branches

## MANAGING RELEASES

When shipping an intermediate version of a program, it is customary to copy and save all of the files associated with that distribution. With MULTI, this is no

## MULTI CODE COVERAGE ANALYSIS

MULTI improves product quality by identifying areas of code that have or have not been executed. Code Coverage results can be generated for a single execution of the program or accumulated over a set of runs to give aggregate coverage information. Areas of code that have not been executed are highlighted in the debugger to stand out from the rest of the source code. High level language (C, C++, Ada 95, Embedded C++, or FORTRAN) source code or Assembly Language source code can be displayed at the click of a button, showing Code Coverage down to the machine instruction.



The ShowLastEdit command lets you determine how selected lines or parts of lines were last edited. A single click, opens two new edit windows: one with the version just prior to the edit and another with the edited version. In both windows, the affected text highlighted. The version control entry is also displayed so that the date of the modification, name of the user who made the changes, and the user's comments are all documented.

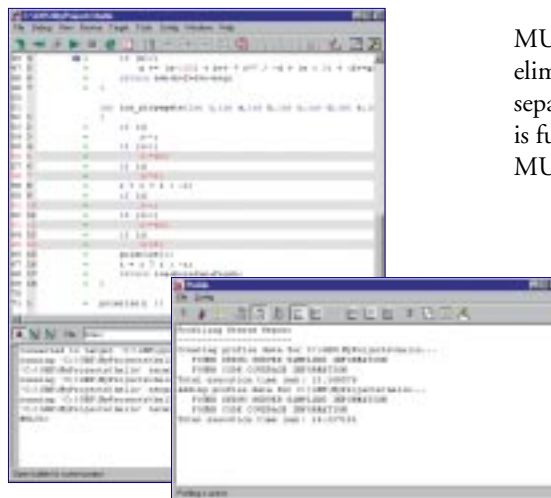
longer necessary. Instead, simply give that version of the program a name (e.g. Release 2.5). MVC interacts with the MULTI Project Builder to use that name to retrieve all of the program components as they existed at the time of the release.

## 3RD PARTY VERSION CONTROL

Users wishing to use 3<sup>rd</sup> party Version Control programs can do so from MULTI through a flexible configuration dialog. For example, one popular Version Control program is Rational's ClearCase®. MULTI can be configured

to invoke ClearCase, rather than MVC, for most version control operations. Thus, MULTI's open interface enables user choice of version control software, while retaining the comprehensive MULTI IDE framework and other development tool components.

***“OUR ENGINEERING STAFF PURCHASED GREEN HILLS’ TOOLS FOR THEIR RICH FEATURE SET, REASONABLE PRICE, AND RESPONSIVE TECHNICAL SUPPORT.”***  
***GEORGE JAQUETTE - DIGITAL HARMONY***



Generate detailed reports from the Code Coverage window

MULTI's integrated Code Coverage eliminates the need for specialized separate products additional cost, and is fully integrated with the rest of MULTI's easy-to-use features.

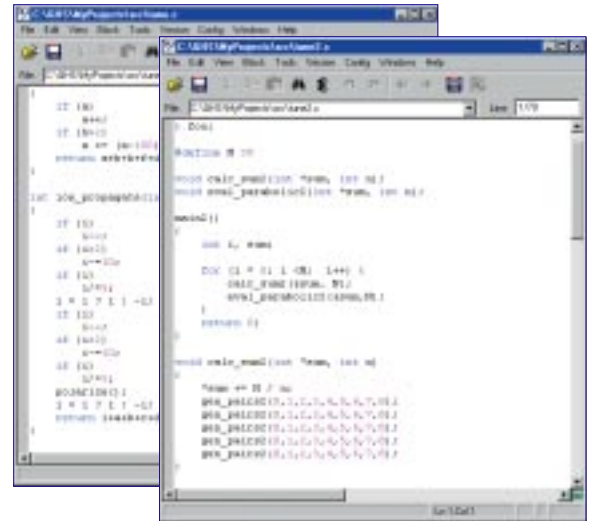
# TEXT EDITOR AND EVENTANALYZER

## MULTI TEXT EDITOR

MULTI includes a full-featured, user-configurable, windowing text editor that includes editor syntax coloring and auto-indenting. Syntax Coloring helps programmers quickly identify language keywords and constructs, including comments, strings, and constants. Auto-indenting automates formatting of source code as the programmer types it in. These features combine to enhance code readability and simplify code writing.

The MULTI Editor implements all of the standard functions that is expected in a programming editor. It is fully configurable: the user can change the mouse behavior, the button bindings,

and the menu bar. MULTI's Text Editor enables a single, consistent edit interface for all host and target development combinations. The Editor is an integral part of the overall MULTI IDE as it can be invoked and controlled as an object by other components within MULTI. MULTI also supports other editors such as *SNIFF+*, *vi*, *emacs*, and *brief*.



The MULTI Text Editor supports an unlimited number of edit windows, each of which can be open on the same or different files. If two windows reference the same file, any changes made in one are automatically reflected in the other. In each window, users can edit an unlimited number of files.

## MULTI EVENTANALYZER™

The EventAnalyzer™ is a powerful utility that enables programmers to understand the complex real-time interactions of their system. Some problems simply cannot be detected with standard debugging techniques. The EventAnalyzer allows the programmer to view and track application events in a full featured graphical viewer. The EventAnalyzer is configurable to view a variety of application and RTOS events and can be ported to many 3rd party or in-house RTOSes. Among the events that can be viewed, in a stretchable time scale, are the following:

- Task context switch
- Semaphore give/take
- Interrupts and exceptions
- Message send/receive
- User-defined events

This data is transferred to the host where it is displayed graphically in the EventAnalyzer GUI. Enabling and disabling event logging, selecting the

events to track, and determining the size of target event buffers can all be fully controlled by the user, both interactively and programmatically.

Data can be logged either in Post-mortem Mode or Live Mode. In Post-mortem Mode, data is gathered in a circular buffer on the target with a minimal effect on the rest of the system; users can upload the event log upon request, such as after a failure. In Live Mode, data is continually sent to the host via TCP/IP, allowing a virtually unlimited history of data to be collected and analyzed as the system is running.

The EventAnalyzer GUI is an effective tool for analyzing and understanding the data, providing convenient navigation capabilities such as zooming, searching, and the ability to hide or display various events and change the look and feel of task status lines and event icons.



The EventAnalyzer displays multicolored icons and status lines to aid the user in understanding the complex, real-time interactions of the system.

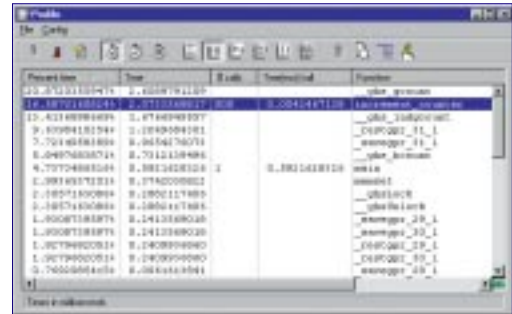
Certain RTOSes can provide the ability to log user-defined events. This can be very powerful since the user can locate important points in their code and log events to help understand the system better. Also, extra data can be logged, such as the values of particular variables. This kind of data tracking in real-time can be invaluable. When the data is later analyzed, the user can search for these user events specifically. User events can either be instrumental into the code by the programmer or set dynamically in the field (called tracepoints) via the MULTI Debugger.

## MULTI PERFORMANCE PROFILER

MULTI provides detailed and summary profiling information to enable the developer to identify the code where execution consumes the greatest amount of time as well as how many times a given procedure or block of code is executing. This helps users select the best optimizations (such as loop unrolling or inlining) for a particular section of code. MULTI's Profiler can gather, arrange, and display profile information in a variety of formats:

- **Call Graph** - Shows calls to and from each function, with percentages of time for each. The Call Graph is shown in tabular or graphical form.

- **Block Coverage** - Shows basic blocks of code and how many times each basic block was executed in a particular run of the program. Unexecuted blocks are highlighted.
- **Standard Calls Report** - Listing shows percentage of overall time spent in each function.
- **Source Lines Report** - Listing shows percentages for each source line, whether in original high-level language form or assembly language form.



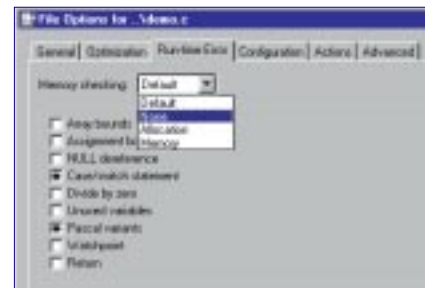
The Profile Control Panel controls the collection of profile data and the generation of reports. The large variety of profile reports give the user several ways of zeroing in on problem areas.

## MULTI RUN-TIME ERROR CHECKING

MULTI's Run-Time Error Checking capabilities can provide useful information on a wide variety of run-time errors. MULTI detects the errors when they first occur, rather than later when they manifest themselves through secondary effects.

MULTI can insert special checks into the program to watch for and report a broad variety of run-time errors. By pinpointing the source line at which the error first occurs, error checking saves many hours of time hunting down very intricate problems. Error checking saves a lot of time finding simpler bugs as well, and that time adds up quickly over the course of a programming project. MULTI can check for several errors including:

- **Allocation** - Reading or writing unallocated memory.
- **Free** - Freeing unallocated memory.
- **Memory Leak** - Detecting dynamically allocated memory blocks not referenced by any program pointer.
- **Array Bounds** - Accessing an element beyond the declared bounds of an array.
- **Assignment Bounds** - Storing a value into a variable or bit field too small to hold it.
- **NULL Dereference** - Using a null pointer to access memory.
- **Switch/Case Statement** - Having a parameter to a switch/case statement which does not match any of the case labels.
- **Divide By Zero** - Dividing by zero.
- **Watchpoint** - Accessing a specified memory address.
- **Unused Variables** - Having a local variable which is never used, or which is read before being initialized.
- **Return** - Exiting from a non-void function without returning a value.



Each type of error check can be separately enabled. Most error checks can be enabled or disabled on a module by module basis, so that the check overhead is not incurred on time critical code.

## MEMORY LEAK DETECTION

When all pointers to a block of memory allocated with *malloc()* but not released with *free()* are discarded, then that memory can no longer be accessed or freed and is called a memory leak. MULTI's *findleaks* command identifies all pointers in the program and matches them against the addresses of all allocated memory blocks. Any memory block not pointed to by any pointer is reported as a leak. The report identifies the source line originally responsible for allocating the offending block; the *malloc()* on that line is missing a corresponding *free()*. Clicking on a line of the *findleaks* report will cause the associated source code to be displayed in the debugger source pane.

**“GREEN HILLS SOFTWARE DELIVERS A COMPREHENSIVE INTEGRATED DEVELOPMENT ENVIRONMENT WITH THEIR USER-FRIENDLY MULTI TOOLS—HELPING TO SPEED TIME TO MARKET FOR OUR CUSTOMERS.”**

**PHIL BROWNFIELD - MOTOROLA SEMICONDUCTOR PRODUCTS SECTOR**

# RUN-MODE DEBUGGING

## RUN-MODE DEBUGGING

MULTI provides advanced run-mode debugging support for embedded applications. Run-mode debugging simplifies the design of sophisticated multithreaded applications by enabling designers to debug those applications on a thread-by-thread basis while the application continues to run.

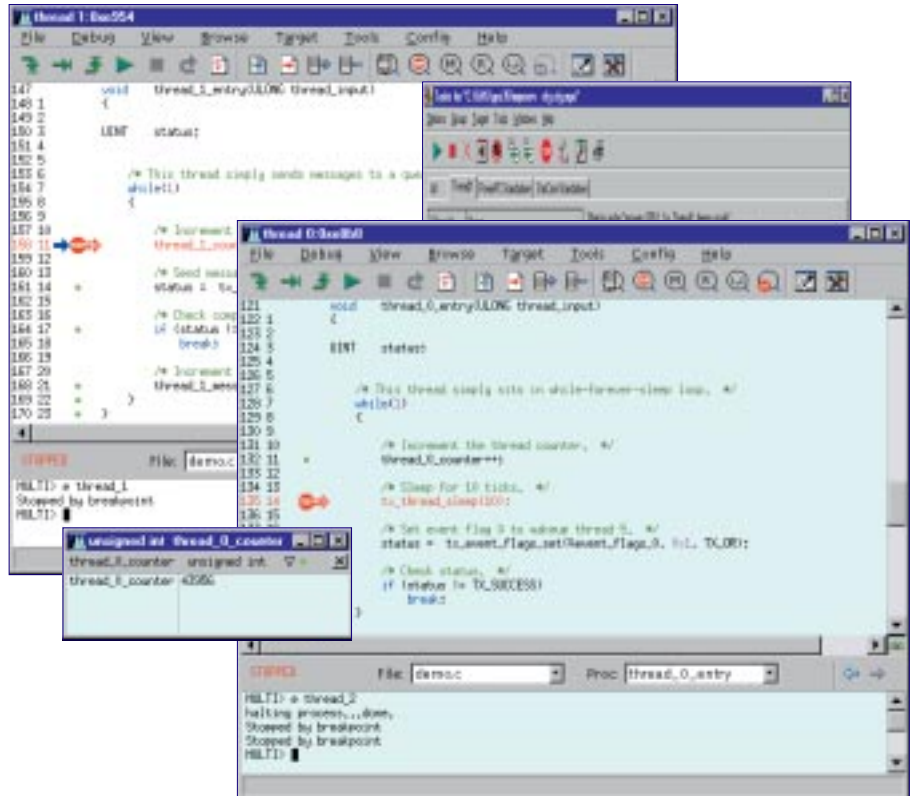
Traditional debugging approaches like BDM (Background Mode Debugging) and JTAG require that programmers halt their system and stop all threads before they can examine any particular data or thread. Run-mode debugging provides a better gauge of real-world system behavior by enabling programmers to halt and examine specific threads in their system while the remaining threads continue to run. Each time a thread hits a breakpoint, MULTI halts that thread and opens a source code debug window specific to that thread. Programmers can then single-step or browse through the halted thread and read memory and registers, all without disrupting program execution. This capability allows the user to debug a single thread even when it uses code that is common to other threads.

### CAPABILITIES

Run-Mode Debugging provides visibility into the multiple threads (tasks/processes) of the application running under a particular RTOS. Many commercial RTOSes are supported, and proprietary RTOSes and kernels can be adapted to use INDRT for Run-Mode Debugging as well. Run-Mode provides greater control over the application by letting the system continue to run while some threads are stopped for examination.

***“RUN-MODE DEBUGGING ADDS A NEW DIMENSION TO MULTI’S ABILITY TO GIVE PROGRAMMERS A BIRDS-EYE VIEW OF WHAT HAPPENS IN DYNAMIC REAL-TIME EMBEDDED SYSTEMS AT A THREAD LEVEL.”***

***WILLIAM LAMIE - EXPRESS LOGIC***



*MULTI's Run-Mode Debugging capability enables the developer to highlight one or more threads for debugging while allowing the rest of the application, and the kernel, to continue to service interrupts and run.*

Run-Mode also provides a Task List showing all loaded threads (tasks) in the system. Users can select individual threads for debugging, and a color-coded debug window is created, corresponding to the selected thread. The user can have several such windows open on his desktop simultaneously, and easily switch among them for the focus of the debug session, all the while allowing the rest of the application to continue running.

### HOW IT WORKS

Run-Mode Debugging is implemented for each RTOS in a manner that provides Ethernet and/or serial connection from the host to the target. This connection supports target I/O, file access, as well as debug control and data. One of these approaches is used to implement Run-Mode Debugging in a particular RTOS environment:

- The RTOS may contain a debug API, enabling MULTI to communicate with it directly through a special “server” interface program that is provided with MULTI. This server may be developed by Green Hills for use with a particular RTOS, or developed by the RTOS provider to enable users of that RTOS to use MULTI for Run-Mode debugging.
- Addition of the INDRT module into the RTOS. INDRT is a Green Hills “debug agent” that resides within the kernel or application space of the RTOS and communicates with MULTI via the RTserv debug server, provided with MULTI. Ethernet and/or serial communications are provided, and all the Run-Mode capabilities are available via either type of interface.

## FREEZE-MODE DEBUGGING

MULTI supports “Freeze-Mode” debugging for most target environments. Freeze-Mode is a subset of the more advanced Run-Mode, and enables the developer to examine the contents of memory and registers, and to download, set break-points, step, run, and halt the application program. This is particularly useful in situations where no RTOS is in use, or for debugging interrupt routines and the kernel in commercial or proprietary RTOS environments. Typically, Freeze-Mode debugging is used with OCD, BDM, and JTAG-equipped processors.

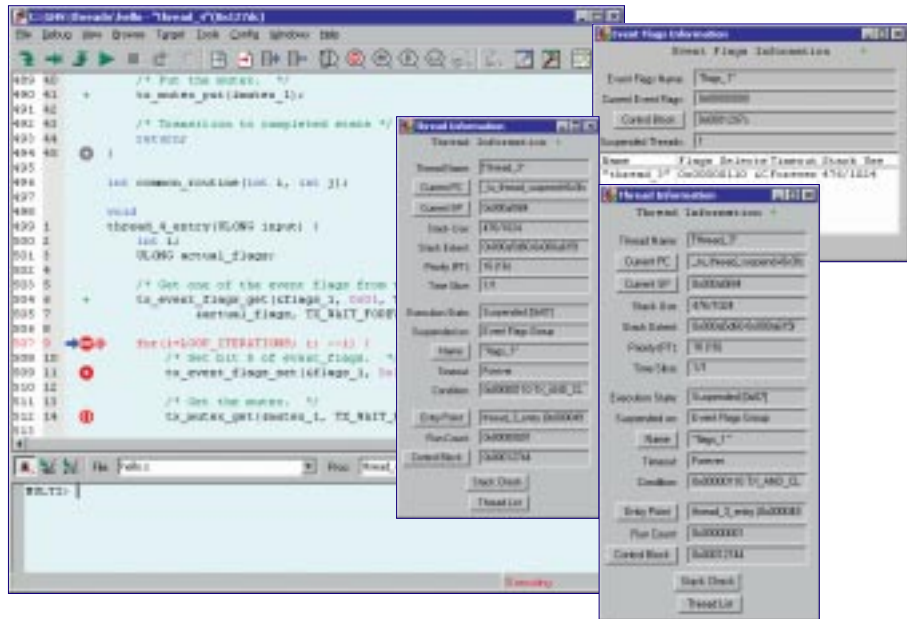
Freeze Mode debugging operations halt the processor for brief intervals, and MULTI takes care of this transparently. Often, the stop-read-restart sequence can be performed “instantaneously” without noticeable interruption to the program flow in the development environment. Freeze Mode debugging supports Kernel Awareness, and Task Awareness, as does Run-Mode, for applications running with an RTOS.

## KERNEL AWARENESS

MULTI provides comprehensive task- and kernel-aware debugging. MULTI can display multiple tasks (“threads”) graphically, including thread rosters, system objects, stack usage, semaphore usage and status, and a variety of other thread- and system-specific information. This capability enables embedded developers to identify thread-specific bugs by setting either system- or thread-specific breakpoints.

**“WE SELECTED MULTI FOR ITS SUPERIOR COMPILER TECHNOLOGY AND COMPLETE TOOL SUPPORT FOR MOTOROLA’S MPC860 PROCESSOR.”**

**EDUARDO LEZAMA - Kb/TEL**



Tasks and other kernel objects can be examined through MULTI’s information windows

The MULTI GUI includes thread-specific windows not available with other development environments, which greatly simplify the analysis and debugging of sophisticated programs. Kernel objects, for example, can be displayed easily with a simple double click of the mouse. View windows, similarly, can be stacked for simultaneous display of multiple objects and variables.

MULTI’s kernel awareness allows developers to continually observe the status of all threads and other kernel components in a system. Each view is fully customized for the kernel and includes a detailed help screen explaining the meaning of each field and how it was derived from the underlying data structures to help developers gain familiarity with the internal workings of the kernel.

Thread information windows, for example, show a thread’s current execution

state: Ready, Executing, Suspended, Completed, or Terminated. For suspended threads, MULTI provides additional helpful information. In that case, the thread information window shows exactly why the thread is suspended and what specific kernel component is responsible. In keeping with MULTI’s ease of use, a single button click brings up a view of the component on which a thread is suspended.

Similarly, in all component information windows, a single button click will show even more information. For example, clicking on a thread’s current PC button causes the debugger source window to move to the corresponding source statement and clicking on a message queue’s read pointer brings up a memory view of that location. Windows continue to display live information that is updated each time the target system stops for any reason. Windows can also be frozen to increase debugging performance on certain targets or to compare the earlier state of a kernel component with its current state.

# HARDWARE DEBUG DEVICES

Green Hills offers two hardware debug devices that enable the MULTI/AdaMULTI Debugger to load, control, debug, and test a target system without the need for prior board initialization, an RTOS, or even a ROM monitor. Developers can choose from the Green Hills Probe™ or Slingshot™.

## THE GREEN HILLS PROBE

Green Hills Software's most advanced hardware debug device, the Green Hills Probe, enables the MULTI/AdaMULTI Debugger to load, control, debug, and test a target system without the need for prior board initialization. Through a JTAG or BDM port, the Probe can debug and control the core state (such as CPU internal registers) as well as the system state (external RAM and flash memory).

In addition to viewing the state, the Probe provides run control and conditional and complex breakpoints both in volatile and nonvolatile memory.

The Probe is ideal for all stages of your product's development: board bring up, where nothing is running on your target, full blown application debugging, and manufacturing test and programming.

The Green Hills Probe's high-speed download enables rapid debug/fix/reload cycles for high productivity. The Green Hills Probe supports both heterogeneous and homogeneous multi-core debugging with a single host connection and a single target JTAG interface.

The Probe can communicate with targets from MULTI host systems running Windows 2000/9x/NT/ME/XP, x86 Linux, Solaris, and HP-UX via Ethernet, USB (Windows 98/ME/2000 only), or RS-232 serial interfaces.

## SLINGSHOT

Green Hills Software's Slingshot is a high-performance, low cost microprocessor debug probe with a single USB host interface. Through a JTAG, EJTAG, or BDM test port, Slingshot can debug and control the core state (such as CPU internal registers) as well as the system state (external ram and flash memory).

Like the Green Hills Probe, the Slingshot provides run control and conditional and complex breakpoints for both volatile and nonvolatile memory in addition to viewing the state.

The Slingshot and Probe hardware debug devices are software compatible, providing flexibility and allowing easy transition with Green Hills Software's microprocessor probe product family.

*"THE GREEN HILLS PROBE IS EASY TO USE AND OUTPERFORMS OUR PREVIOUS PROBE BY A WIDE MARGIN, ENABLING US TO SPEED UP OUR DEVELOPMENT PROCESS SIGNIFICANTLY."*

*GADI SHAVIT -  
ZORAN MICROELECTRONICS*

## GREEN HILLS PROBE FEATURES

- High speed downloads of up to 500 KBps
- Easy configuration and setup
- USB, Ethernet, or RS-232 host interfaces
- Supports a variety of targets
- Source-level debugging
- Local intelligence
- Multicore debugging
- OCD/JTAG target control
- Flash programming
- Built-in hardware diagnostics

## SLINGSHOT FEATURES

- High-speed downloads of up to 150 KBps
- Easy to use
- Variety of targets
- USB host interface
- Full MULTI integration
- Small size
- Low cost



*The Green Hills Probe and Slingshot debug devices mark a dramatic advance in debug technology by offloading many debugger functions onto faster, dedicated hardware.*

### THIRD PARTY PRODUCT SUPPORT

In addition to tight integration with leading 3rd Party RTOSes, MULTI interfaces with numerous complementary third party products.

### USING A 3RD PARTY EDITOR

The MULTI Source-level Debugger is aware of and interacts with other components of the MULTI environment. Clicking on the *Edit* button launches an editor on the current file, with the edit cursor positioned at the current line. Double clicking on the name of a procedure anywhere in the source pane brings up an editor window on the source code for that procedure definition. After making an edit, a single button click will recompile the program and reinitialize the debugger with the new executable.

### THE MULTI OPEN INTERFACE

MULTI provides an open interface that allows users to easily use their favorite components from within MULTI. MULTI adheres to open standards that enable users to avoid proprietary, closed interfaces:

- ELF/DWARF standard compiler output interface for C, C++, and Embedded C++ debugging
- PowerPC EABI standard, StarCore ABI
- ANSI/ISO/IEC Ada 95
- BDM, JTAG, EJTAG, OnCE on-chip debug interfaces
- ANSI C, C++, Embedded C++, FORTRAN
- Tornado from Wind River Systems
- Gateway from Agilent Technologies

Users can invoke their favorite editors, *make* systems, and configuration management systems all from within MULTI. Integration with other products is accomplished through MULTI's

### THIRD PARTY PRODUCT SUPPORT

**MICROPROCESSORS AND CORES**

Advanced Micro Devices ▪ ARM ▪ Atmel ▪ Cirrus Logic ▪ Fujitsu ▪ Hitachi  
IBM ▪ Infineon Technologies ▪ Intel ▪ Intrinisty ▪ IDT ▪ Lexra ▪ LSI Logic  
Lockheed Martin Federal Systems ▪ MIPS Technologies  
Mitsubishi Electronics America ▪ Motorola SPS ▪ NEC ▪ Philips Semiconductors  
PLX Technology ▪ QED ▪ Q-Logic ▪ SHARP Microelectronics  
STMicroelectronics ▪ Toshiba ▪ TRW

**REAL-TIME OPERATING SYSTEMS**

ThreadX® ▪ OSE™ ▪ Nucleus®  
VxWorks® ▪ RT Secure

**HARDWARE DEBUG SOLUTIONS**

Agilent Technologies ▪ Applied Microsystems ▪ EPI ▪ EST  
Hitachi ▪ Macraigor Systems ▪ NEC ▪ P&E ▪ Tektronix ▪ Yokogawa

**CPU BOARDS**

Embedded Planet ▪ Galileo Technology ▪ Motorola Computer Group  
NETsilicon ▪ Many Eval/Development Boards for Supported Microprocessors

**NETWORKING AND COMMUNICATIONS PROTOCOLS**

Allegro Software ▪ Elmic Systems ▪ Interpeak  
NETsilicon Softworks Group ▪ Objective Interface Systems ▪ Vertel

**COSIMULATION AND COVERIFICATION**

Cadence ▪ CoWare ▪ Mentor Graphics

**APPLICATION MODELING**

I-Logix ▪ Rational RoseRT™ ▪ The Mathworks

**GRAPHICS/USER INTERFACE SOFTWARE**

3D Labs ▪ Seaweed Software ▪ Swell Software

**CODE QUALITY AND MANAGEMENT**

Rational ClearCase ▪ Vector Software

powerful ability to issue shell commands to other applications. These shell commands can be issued from user-customized pull-down menu options or custom action buttons. This flexibility provides an open door to companion applications and shell scripts.

Using MULTI avoids the trap of getting locked in with proprietary tools from one ICE, RTOS or chip manufacturer. With MULTI, users can switch to another ICE, RTOS, or target processor without having to retrain personnel, redo *makefiles*, or modify source code to

account for language variations. These software development costs, at an industry average of \$750 per day, per programmer, far exceeds the purchase price of the MULTI Development Environment.

### ON-LINE HELP

MULTI includes a context sensitive help system, providing assistance in all windows. This help system is a full featured, searchable, indexed, Windows compliant integration that can be used across all supported platforms, including UNIX.

# PRODUCT SUMMARY

## REAL-TIME OPERATING SYSTEMS INTEGRITY • ThreadX®

## HIGH PERFORMANCE OPTIMIZING COMPILERS

C • C++ • Embedded C++ • ETC++  
Ada 95 • FORTRAN

## MULTI AND ADA MULTI INTEGRATED DEVELOPMENT ENVIRONMENT

Source Debugger • Run-Time Error  
Checking • Program Builder • Version  
Control System • Resource Analyzer  
Performance Profiler • Text Editor  
Code Coverage Analysis  
Source and Class Browser

## EMBEDDED COMMUNICATIONS PROTOCOLS

TCP/IP • FTP • PPP • POP3  
SNMP • SMTP • more

## PROCESSOR FAMILIES SUPPORTED

Power PC • ARM/Thumb  
StrongARM/XScale  
MIPS32/MIPS64/MIPS16  
Intrinsity • CPU32 • StarCore  
680x0/683xx • M\*CORE • ColdFire  
Lexra • ST100  
SH • SH-DSP • V850  
i960XScale • x86/Pentium  
SPARC/SPARCLite  
Tricore • Alpha • ZSP  
FujitsuRISC FRxx

## CROSS DEVELOPMENT TOOL CHAIN

Assemblers • Linkers • Librarians  
Run-time Libraries

## SUPPORTED TARGET SYSTEMS

OCD/BDM/JTAG • RTOS  
Simulators • In-Circuit Emulators  
Processor Probes • ROM Monitors  
CPU Boards

## SUPPORTED HARDWARE DEBUG DEVICES

Green Hills Probe  
Green Hills Slingshot  
Agilent Probe • ARM MultiICE  
EPI • Jeeni/Majic  
EST VisionProbe/ICE  
Macraigor Wiggler/Raven

## SUPPORTED PARTNER RTOS

ChorusOS • Embedded Linux  
LynxOS • OSE • RT Secure • RTX  
Tornado/VxWorks

## HOST PLATFORMS FOR EMBEDDED DEVELOPMENT

Windows 9xx/ME/NT/2000/XP  
Linux • SPARC/Solaris  
PA-RISC/HP-UX

# CONTACT INFORMATION

## NORTH AMERICAN SALES OFFICES

California - Los Angeles  
Phone: 909.592.9061

California - San Clemente  
Phone: 949.369.3950

Northern California  
Phone: 831.430.0525

Washington - Seattle  
Phone: 425.837.1699

Colorado - Denver  
Phone: 303.740.8462

Texas - Dallas  
Phone: 972.712.1407

Illinois - Chicago  
Phone: 847.515.2418

North Carolina - Mars Hill  
Phone: 828.689.8508

Massachusetts - Lexington  
Phone: 781.862.2002

## CORPORATE HEADQUARTERS

30 West Sola Street  
Santa Barbara, California 93101  
Phone: 805.965.6044  
Email: [sales@ghs.com](mailto:sales@ghs.com)  
[www.ghs.com](http://www.ghs.com)

## NORTH AMERICAN SAFETY-CRITICAL PRODUCTS SALES OFFICES

California - Laguna Hills  
Phone: 949.460.6442

Florida - Palm Harbor  
Phone: 727.781.4909

## INTERNATIONAL OFFICES

France  
Phone: +33.1.46.96.07.00  
[sales-fr@ghs.com](mailto:sales-fr@ghs.com)

Germany  
Phone: +49.721.98.62.580  
[sales-ger@ghs.com](mailto:sales-ger@ghs.com)

Netherlands  
Phone: +31.33.4613363  
[sales-nl@ghs.com](mailto:sales-nl@ghs.com)

Sweden  
Phone: +46.46.211.33.70  
[sales-se@ghs.com](mailto:sales-se@ghs.com)

United Kingdom  
Phone: +44.1844.267950  
[sales-uk@ghs.com](mailto:sales-uk@ghs.com)

Japan (ADaC)  
Phone: +81.3.3576.5351  
[sales@adac.co.jp](mailto:sales@adac.co.jp)



*Green Hills Software, the Green Hills logo, MULTI and INTEGRITY are registered trademarks and the Green Hills Probe, Slingshot, AdaMULTI, EventAnalyzer, ResourceAnalyzer, ISIM, QuickStart, Integrate, G-Cover, GMART, GSTART and Safety Critical Products are trademarks of Green Hills Software Inc. ThreadX is a registered trademark of Express Logic, Inc. All other trademarks (registered or otherwise) are the property of their respective companies.*

© 2003 Green Hills Software, Inc. v0203