Continue

Green hills compiler user manual pdf

Green hills compiler.

The MULTI IDE has been a go-to choice for thousands of developers over three decades, providing an unparalleled integrated development environment to craft, refine, and optimize code for embedded processors. This powerful tool lets developers swiftly locate and rectify troublesome bugs, pinpoint performance bottlenecks, and prevent future issues, ultimately lowering the cost of delivering software on time and minimizing hardware costs by using more efficient code that requires less memory and smaller CPUs. The MULTI IDE supports all popular processor architectures and can run on both Linux and Windows machines locally or in the cloud. It's capable of debugging code from multiple compilers, connecting to targets over various links, and supporting multiple operating systems running on silicon, simulators, or emulators. In complex real-time systems, using the MULTI IDE can save weeks or even months of time spent hunting for sporadic bugs or finding performance bottlenecks, potentially saving a product launch. One of the key features of the MULTI Debugger is its ability to quickly pinpoint problems caused by memory corruption, unexpected dependencies, interrupts, multicore complexity, inter-task corruptions, opaqueness from missing source code or virtualization, missed real-time requirements, and asynchronous hardware events. The debugger includes three powerful capabilities: History viewer, TimeMachine, and Debug Snapshot. The History viewer is a tool that shows how the program got to its current state and what the system was doing at any given time. This provides a clear view into complex heterogenous multicore systems, allowing developers to zoom in deeply on processor behavior at the micro-second level or step back to see system behavior spanning minutes and days. TimeMachine is another powerful tool that enables developers to run, step, and analyze code backward to any problem area shown in History. This feature provides unprecedented visibility into a system by capturing actual program execution data and enabling developers to find difficult bugs in seconds, see hidden bottlenecks and dependencies, and analyze execution times. By using the MULTI IDE and its powerful capabilities like TimeMachine, developers can prevent new problems and improve overall code quality. A tool named DoubleCheck helps developers by spotting programming errors before they run their program, saving time and money. As part of Green Hills optimizing compilers, there's no need to set it up, making it easy to use with minimal hassle. Simply setting an option allows DoubleCheck to automatically review the source code each time it's compiled. This process is faster than traditional static analysis tools, meaning every developer can have it turned on all the time. DoubleCheck is also more reliable than traditional code reviews and can find bugs that might never show up during regular system testing. It complements another tool called Run-time Error Checking by identifying problems that can't be spotted through static analysis alone. This includes issues like memory corruption, which are especially difficult to find and can remain silent for a long time. Run-time error checking alerts you to the cause of the problem when it happens, making it easier to identify and fix bugs early on. DoubleCheck lowers development costs by identifying problems early in the development stage, saving time and resources. Clean code is less likely to contain errors and is easier to test, understand, and modify. All these factors contribute to fewer bugs and greater reliability. Green Hills optimizing compilers also enforce clean coding conventions defined by industry standards like MISRA 2012 and 2004, which include over a hundred rules for safe programming. You can even customize these rules to meet specific requirements. Another powerful tool is the MULTI Debugger, which allows you to examine, monitor, and change source code running on complex multicore target processors and simulators. When used with TimeMachine, it can even run backward in time. The Debugger is seamlessly integrated with other tools within MULTI and can be invoked by clicking inside various MULTI tools. The MULTI Debugger can debug applications, kernels, and device driver code on various operating systems or bare board code. It's also capable of debugging virtualized operating system kernels and their applications. With the ability to traverse the call stack, view cache, browse objects, and view memory and registers, it's a powerful tool for developers. In addition, the OSA Explorer allows you to deeply see into the status of kernel objects, tasks, and resources. It can even debug task interactions at the source level and find specific data points using powerful execution and data breakpoints. The MULTI Debugger can also debug code running in virtual address spaces, making it a versatile tool for developers. When debugging multitasking applications on an OS like INTEGRITY or Linux, MULTI can interact with multiple tasks in run-mode, freeze-mode, or both modes simultaneously. In run-mode, the operating system kernel continues to run as you halt and examine individual tasks. In freeze-mode, the entire target system stops when you examine tasks. MULTI for Linux offers advanced debugging capabilities for embedded Linux software development, enabling engineers to improve their productivity and deliver more reliable products faster. Unlike traditional tools like GDB or Eclipse, which can be cumbersome to set up and use, MULTI provides a seamless debugging experience by halting all cores simultaneously when a debugging condition is encountered. This ensures that shared memory remains unaffected, allowing developers to focus on one core without worrying about interference from other cores. The MULTI IDE simplifies project management, eliminating the need for complicated build infrastructure and streamlining the compilation process. Its intuitive Builder tool automatically analyzes dependencies and compiles files in parallel, taking advantage of modern multicore systems. In addition to its debugging capabilities, MULTI offers a range of tools to help developers optimize their code for performance. The IDE includes industry-leading embedded C/C++ compilers, as well as advanced performance analysis tools. For applications requiring the highest possible performance, Green Hills Software experts can provide customized solutions tailored to specific devices. Green Hills C/C++ optimizing compilers stand out for their exceptional performance. In EEMBC benchmarks, they consistently outperform competitors, generating the fastest and smallest code for embedded processors. Hundreds of advanced optimizations enable significant speed boosts and size reductions. Vendors often publish EEMBC-certified scores generated with Green Hills Compilers, demonstrating their processor's capabilities in real-world applications. To maximize program performance, it's crucial to use top-notch analysis tools like History and Profiler. These tools help identify and eliminate performance bottlenecks, resulting in faster, more efficient applications. To optimize performance, consider the time dimension, which is often overlooked by profiling tools. The innovative History visualization tool provides a high-level view of program flow, making it easy to spot patterns and where time is spent. The powerful Profiler offers additional insights into program behavior, providing data on processor time consumption, coverage gaps, and test coverage. Green Hills Software's custom services can help you reach the highest possible performance by optimizing your design and creating a customized compiler optimization tailored to your goals. By reducing memory and processor costs with our MULTI IDE, you can lower production costs and boost your bottom line. Our superior C/C++ compiler optimizations are tuned to get the most out of your chosen processor. Improving code performance also gives you more CPU headroom to add features, potentially enabling you to choose a lower-cost processor that uses less power and cooling. Additionally, our performance analysis tools provide a comprehensive view of program behavior, helping you focus optimization efforts and maximize real-time requirements while reducing costs and increasing functional safety confidence. The C/C++ IDE and toolchain have achieved top-tier safety certifications from TÜV NORD and exida, meeting rigorous standards such as ISO 26262:2018 (Automotive), IEC 61508:2010 (Industrial), EN 50128:2011, and EN 50657 (Railway). This certification ensures the highest achievable levels of Safety Integrity Level (SIL) 4 and Automotive Safety Integrity Level (ASIL) D. The toolchain is also certified for C/C++ run-time libraries. The benefits of using this IDE include reduced costs and time-to-certification, lower maintenance after product release, and higher quality code targeting the highest ASIL/SIL levels. Additionally, Green Hills provides expert training and consulting services to maximize productivity. These services are customizable in content and duration, offering: * Expert training environment setup * Application design mentorship * In-depth training on INTEGRITY programming and advanced History-guided debugging Courses are taught by experienced professionals with hands-on knowledge of Green Hills tools. Classes can be held at your facility or scheduled around the world. For teams with smaller budgets, Green Hills offers open-enrollment courses for new hires who have joined a team that has already completed a training class. The managed implementation program also provides ongoing mentorship and regular access to Green Hills experts.