WIND RIVER

Wind River Workbench, On-Chip Debugging Edition

The introduction of 32-bit and 64-bit processor technologies has created new challenges that require an innovative approach to JTAG debug and analysis. Wind River offers the industry's leading Eclipse-based development environment for on-chip debugging that takes you from early hardware bring-up to test and manufacturing. The advanced hardware diagnostics and patent-pending multicore debugging capabilities of Workbench help you unlock the power of today's advanced microprocessor technology.

Wind River Workbench, On-Chip Debugging Edition is based on the industry-standard Eclipse framework, providing the industry's leading Eclipse-based JTAG debugging environment that can be easily extended through in-house, third-party, open source, or commercially available Eclipse plug-ins. For example, you can integrate software version control and change management solutions such as CVS and ClearCase. You can also integrate UML-based software visualization and code generation tools, enhanced static analysis solutions, and simulation/virtualization software.

Wind River on-chip debugging solutions enable you to take advantage of the growing Eclipse ecosystem to improve the collaboration between hardware, firmware, and software developers, ensuring a smooth handoff in the debug process.

Table of Contents

Centralized, Project-Oriented Environment

Workbench On-Chip Debugging provides a centralized, standards-based development environment that auto-

Key Features

- This Eclipse-based JTAG debugging solution reduces development costs and extends easily with third-party or open source tool integration.
- The centralized project-oriented environment simplifies end-to-end debugging.
- Advanced editing capabilities reduce the complexity of the edit-compile-debug cycle.
- Connection and configuration management streamlines connections to multiple targets.
- Industry-leading features support debug of an entire system from a single console.
- Flexible flash programming quickly resolves common problems.
- Internally and externally buffered trace enables debugging of complex and intermittent code defects.
- Patent-pending multicore technology debugs even the most complex 32-bit and 64-bit multicore processors.
- It has extensive scripting capabilities.
- It includes a wide range of processor and operating system support.

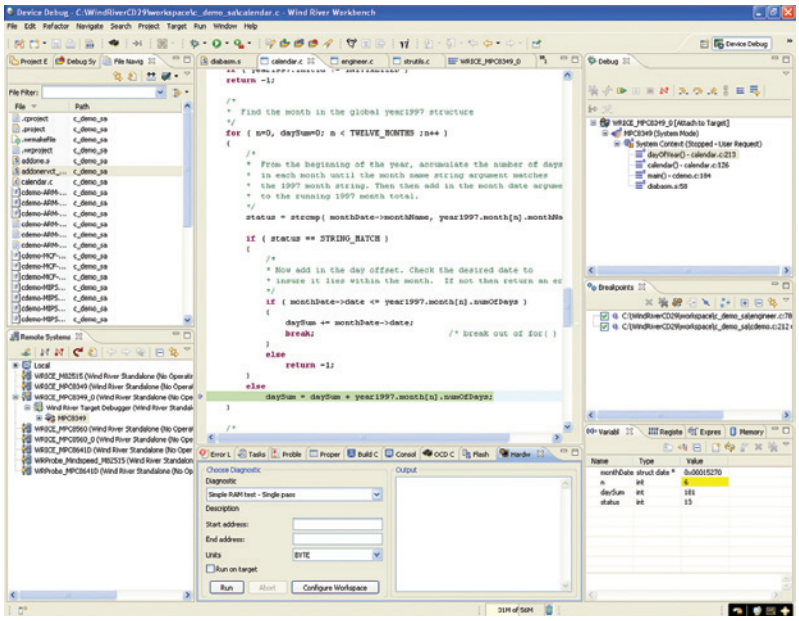


Figure 1: Wind River Workbench, On-Chip Debugging Edition

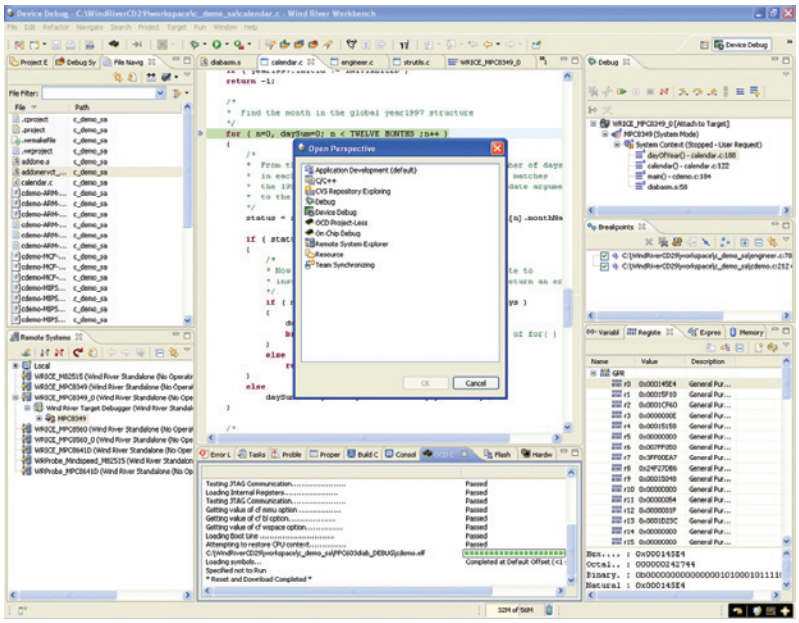


Figure 2: Workbench On-Chip Debugging allows you to specify perspectives relevant to a certain task

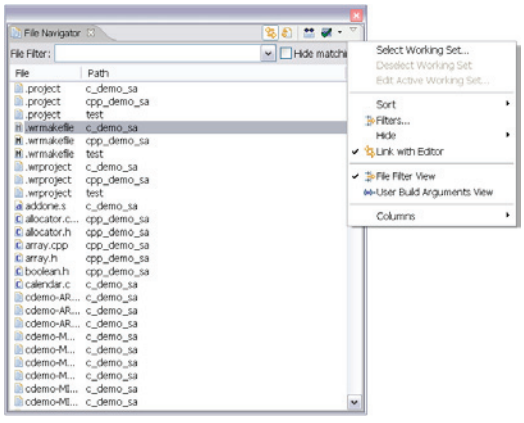


Figure 3: Workbench file navigator allows you to quickly find symbols and functions

mates the debugging and analysis process. Its project-oriented methodology simplifies the handoff between the different phases of development, from initial hardware bring-up to firmware design, OS implementations, and platform and application development.

The Workbench development environment efficiently manages hardware and software development projects. Its flexible, configurable project interface enables you to customize project data based on type of project and development activity. Wizards and utilities walk you through the creation of new projects to get you up and running quickly.

From a single console, you can see information about multiple project files, multiple targets, and the software running on those targets. Because different stages of development require different debug views and capabilities, you can specify a particular developer perspective that provides only the information relevant to a specified task.

The on-chip debugging perspective delivers the starting point for JTAG debugging, including the views required for connection to Wind River emulators, as well as the specific tasks associated with JTAG debugging.

Perspective in Eclipse

Perspective is a specific term in the Eclipse platform that refers to the visible actions and views within a window. It includes an input attribute, which defines the resources in a workspace; and a type attribute, which defines the actions and views in the user interface.

Project Navigation Capabilities

Advanced project navigation capabilities reduce the complexity of managing multiple projects. You can organize and manage all the software components in a device software development project. Moreover, you can create and share projects among other development team members, reducing setup and configuration time and improving collaboration. When combining software debugging with on-chip debugging, you can tighten the project handoff between hardware and software developers.

Integration with CVS enables you to identify when other developers are editing the same code, in order to prevent editing conflicts. The Workbench file navigator feature also allows you to quickly search and find symbols and functions within all files loaded in the project workspace, shortening development time.

Comprehensive Build System

The Workbench On-Chip Debugging Build System provides the tools, options, and parameters for device software build management, enabling you to define everything from global build parameters to fine-grain control of an individual file. By using a single build environment, you can reduce the time associated with managing multiple environments. With build capabilities built directly into the software, you have a centralized solution for the entire edit, compile, and debug process.

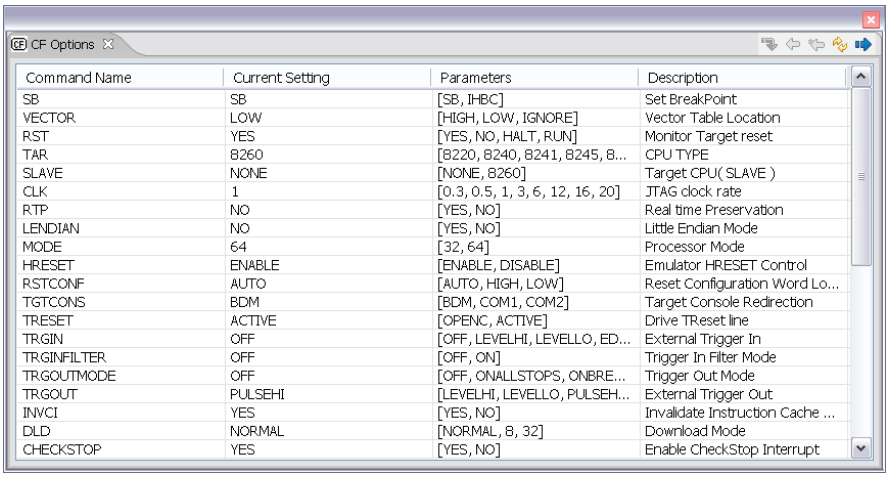
Advanced Editing Capabilities

The editor provides state-of-the art editing, including vi emulation. Performance-enhancing features, such as code completion, parameter hinting, and syntax highlighting of source files, improve edit-compile-debug cycle time and reduce errors.
The editor is tightly integrated with the Project System, Build System, Source Code Analyzer, and Debugger in Workbench, enabling you to move back and forth easily between various source-code debugging tasks.

JTAG Editor

In addition to software editing capabilities, Workbench On-Chip Debugging offers a JTAG editor to create and modify board configuration files. The editor allows you to connect a Wind River JTAG emulator to a board with multiple devices on its scan chain. The editor then provides a graphical view of the layout of devices on the scan chain, expediting the update of board configuration files.

When you only have one or two devices in a scan chain, it is easy to set up the connections and configurations. As devices are added, complexity increases—and so does the chance for error. A graphical view enables you to easily see and configure which devices are in the chain, so you know how to configure the emulator to support the correct number of devices. The JTAG emulator will also need to know how to address the various devices on the scan chain, so it can access specific devices for a debug task while the rest of the devices are set in bypass mode. The editor saves this information and downloads it to the emulator.



Command Name	Current Setting	Parameters	Description
SB	SB	[SB, IHBC]	Set BreakPoint
VECTOR	LOW	[HIGH, LOW, IGNORE]	Vector Table Location
RST	YES	[YES, NO, HALT, RUN]	Monitor Target reset
TAR	8260	[8220, 8240, 8241, 8245, 8...	CPU TYPE
SLAVE	NONE	[NONE, 8260]	Target CPU(SLAVE)
CLK	1	[0.3, 0.5, 1, 3, 6, 12, 16, 20]	JTAG clock rate
RTP	NO	[YES, NO]	Real time Preservation
LENDIAN	NO	[YES, NO]	Little Endian Mode
MODE	64	[32, 64]	Processor Mode
HRESET	ENABLE	[ENABLE, DISABLE]	Emulator HRESET Control
RSTCONF	AUTO	[AUTO, HIGH, LOW]	Reset Configuration Word Lo...
TGTCONS	BDM	[BDM, COM1, COM2]	Target Console Redirection
TRESET	ACTIVE	[OPENC, ACTIVE]	Drive TReset line
TRGIN	OFF	[OFF, LEVELHI, LEVELLO, ED...	External Trigger In
TRGINFILTER	OFF	[OFF, ON]	Trigger In Filter Mode
TRGOUTMODE	OFF	[OFF, ONALLSTOPS, ONBRE...	Trigger Out Mode
TRGOUT	PULSEHI	[LEVELHI, LEVELLO, PULSEH...	External Trigger Out
INVCI	YES	[YES, NO]	Invalidate Instruction Cache ...
DLD	NORMAL	[NORMAL, 8, 32]	Download Mode
CHECKSTOP	YES	[YES, NO]	Enable CheckStop Interrupt

Figure 4: Workbench On-Chip Debugging graphical display allows visualization of all CF options associated with the Wind River emulator for a specified target

Connection and Configuration Management

Workbench On-Chip Debugging centralizes all target configuration and connection management in a single graphical view, eliminating the need to toggle between multiple GUIs or manually configure connections via CLI. These connections can be cores, processors, processes, or simulators. You can even manage multiple connections simultaneously.

Eclipse Launch Capability

The Quick Target Launch Utility in Workbench On-Chip Debugging leverages the powerful Eclipse launch capability, enabling device software developers to easily define cross-target launches. Developers may also select from a list of predefined launches, either taking control of an already running target or starting a target from reset, which results in the user's quick movement from target launch selection to debugging complex hardware and software problems.

Launches may be used to establish a target connection, download an image to the target, load symbol information into the debug engine, or run scripts or any combination of these tasks. In addition, users may create batch launches, which group together individual launches and enable Workbench On-Chip Debugging to establish connections or download software to multiple cores or processors simultaneously.

On-Chip Debugging Command Shell

If you prefer using CLI, the On-Chip Debugging Command Shell provides CLI access to the target via a Wind River emulator. The Command Shell and associated logging capabilities allow you to make target and emulator configuration adjustments. You can also write low-level scripts for target initialization and download sequences to automate commonly occurring tasks.

The Command Shell is used to load target register files into the emulator then execute low-level commands through the emulators. Sessions in the Command Shell can be recorded and replayed as scripts. There are two command log options: input logging and full logging. Input logging only records input commands, while the full logging option records input commands and their associated results. These files can be saved and played back later.

Console for Target Status Reporting

Workbench On-Chip Debugging also includes a console specifically for JTAG connections, providing data on target connection status and the events executing on the target, verifying that a given task executed as instructed.

Graphical Display of All Configuration File Options

To expedite the configuration process, Workbench On-Chip Debugging provides a target register configuration file (CF) for the majority of supported processors. You can customize these configuration files using the Workbench On-Chip Debugging CF Options view.

With the graphical display, you can visualize all the CF options associated with the Wind River emulator for a specified target, including command name, current setting, parameters, and description data.

Expanded Register View Capability

Get your target up and running quickly with the built-in information on the bit-level registers. With one click, you can access a wide range of information on peripheral registration configuration options. This eliminates the time-consuming task of sorting through processor documentation.

In addition, you can create a graphical representation of custom peripheral register groups to more effectively manage custom configurations. Specific capabilities include the following:

- Target registration configuration file utility to modify the target initialization file
- Custom register files to manage additional peripherals
- Bit-level details on a specific processor's peripheral register configuration options

Binary Upload and Compare Utility

The Binary Upload and Compare Utility enables developers to quickly extract information from any area of memory on the target system, assuaging worries about overwriting the boot ROM or boot loader shipped with the processor vendor's reference board. This capability is particularly useful when trying to save/store boot ROMs or boot loaders that may be shipped with a target for safekeeping on the host PC.

The Binary Upload and Compare Utility enables developers to graphically select an area of memory on their device, including sectors of flash, and upload the image into a file on their host PC. Once the image is stored on the host PC, it can be used by the Compare Utility to identify and resolve file corruption issues on the target system. The Compare Utility compares images on the target with the content of binary files stored on the host PC. Any differences between these files are recorded, and the information is displayed in the Workbench On-Chip Debugging Editor view.

Analysis and Diagnostics

Advanced Hardware Diagnostics

At the hardware level, Workbench On-Chip Debugging provides comprehensive hardware diagnostic capabilities that eliminate the need for certain standalone hardware diagnostics tools, reducing capital and training costs. By integrating preconfigured test routines and scope loop tests into a single interface, you do not have to spend valued development time writing your own test tools for data and addressing bus diagnostics.

Wind River provides robust and integrated software that quickly isolates and resolves hardware layout and bus issues. You are guided by a set of wizards to configure and run the various diagnostic utilities:

- Address and data bus tests to verify address and data bus performance and quickly identify and resolve issues
- Cyclic redundancy check (CRC) calculations for a defined section of memory
- RAM tests ranging from simple to full: writing a consecutive pattern of three values into a defined area of memory then reading it back for errors, or executing in a single pass or continuous test pattern
- Scope loop tests that put a known pattern on the address and/or data bus for monitoring by an external data scope

Source Code Analyzer

The Source Code Analyzer simplifies the process of documenting the code structure of a file system to support effective integration of large project file systems. This is especially valuable when multiple developers are writing code; code is reused among projects; or newly developed code must be integrated with a legacy code base. You can quickly and completely understand the code written by someone else and integrate it effectively into the existing project. In addition, you can instantaneously see the impact of a proposed change in the existing code, improving overall development productivity and reducing errors.

Statistical Code Profiling

Built-in performance analysis and code coverage profile software identify system bottlenecks to optimize software execution on a target processor. Unlike a typical profiling solution, no configuration or instrumentation is required. It is easy to select a function from the statistical code profiler then see where the function resides in the Source Editor view.

Profiling is as simple as identifying a section of code. The Workbench On-Chip Debugging software does the rest, displaying the results in a chart, table, or histogram that includes the following:

- Function name
- Full path to the function
- Start and end addresses for the function
- Percentage of time within the function for all functions to be executed on the target

Cache Memory Analysis

Tracking cache coherency issues with main memory can be a challenge for a single-core device application. The complexity of multicore implementations makes it almost impossible without sophisticated cache memory analysis. The cache analysis capabilities in Workbench On-Chip Debugging monitor execution on one or more targets and identify differences between data stored in memory versus data stored in cache. You can quickly identify the source code and function for each instruction in cache and easily toggle to its location in the Editor/Source Code view or symbol browser.

Comprehensive JTAG Debugging Engine

Debugging is the most time-consuming and costly phase of any development cycle. Wind River provides a single console for debugging multiple projects across multiple targets. With these debug collaboration capabilities, different developers can easily share information, dramatically improving debug cycle time.

The advanced debug engine supports multiple debug connection types, including JTAG tools, agents, and simulators, providing maximum flexibility in debugging both hardware and software from the early hardware diagnostic stages to application development. You can debug from any host OS to any target OS. Multicore debugging is easier because you can set cross-correlated breakpoints and track information across multiple targets.

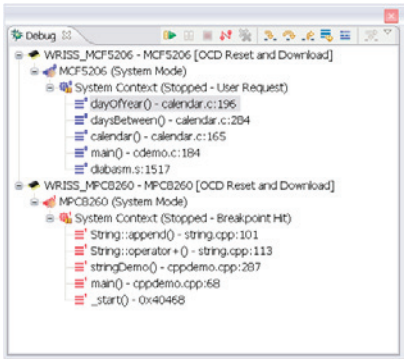


Figure 5: Debug view monitors, controls, and manipulates active tasks

Debug visualization shows at a glance the processors being debugged and the active debug tasks through color-coded and numbered debug sessions. It provides the following information:

- **Standard run-control capabilities:** Start/Resume, Stop/Suspend, Terminate, Step-In, Step-Over, Step-Return, Source Mode, Function Mode, and Assembly Mode debugging
- **Data on the target or process attached to the target:** Stack frame, threads, processes, and tasks
- **System information:** Memory views, Register views, Watch view, local and global variables
- **Multicore Debug view of the stack frame:** Every process or target running under debugger control displayed in the Debug view, color-coded, and numbered

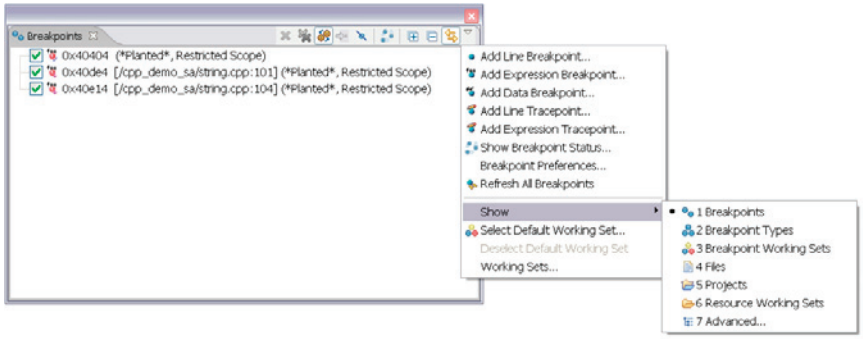


Figure 6: Graphical view of all breakpoints on a target improves management of multiple breakpoints

Common Scripting Framework

Workbench On-Chip Debugging comes with a Host Shell scripting framework to control all debugger activities. It supports standard scripting languages, such as GDB command syntax for low-level debug commands and Tcl or C interpreters for high-level flow control. The Host Shell is supported on Linux, Solaris, and Windows hosts.

Breakpoint Options

Breakpoints are a valuable tool used to stop the target or process running on the system when a specific user-defined event occurs. Workbench On-Chip Debugging supports hardware and software breakpoints configured as either data or expression breakpoints. A graphical view of all breakpoints on a target provides more effective management of multiple breakpoints. You can set breakpoints from the Breakpoint, Source/Edit, Register, and Memory views.

Specific breakpoint features include the following:

- Hardware breakpoints
- Software breakpoints
- Expression, line, or data breakpoints
- Enablement and disablement of any event in the system
- Save to a file and redeploy among multiple projects
- Multicontext awareness, allowing breakpoints to stop the specific core or processor in context with an event
- Ability to stop the entire system when the event the breakpoint is associated with occurs
- Range of icons to manage breakpoint configuration and settings
- Host Shell script execution upon hitting a breakpoint

Flash Programming

Workbench On-Chip Debugging simplifies the configuration of flash memory, enabling you to store images into flash memory on the target board faster than the traditional manual CLI process allows. The flash programming utility supplies common algorithms and diagnostics to pinpoint flash programming issues, enabling you to do the following:

- Configure the flash address and RAM workspace, supporting flexible memory configuration and user-defined allocation of flash programming algorithms into RAM
- Choose files for download; you can select and manage the files to be flashed onto the target from a single console
- Execute erase and program operations by selecting one or more sectors
- Verify the resulting data residing in flash with the file to be flashed
- Program NOR and NAND devices

Support for new or unusual flash devices can be easily added by utilizing the provided sample source code and build instructions.

Trace Support

Workbench On-Chip Debugging provides support for internal and external buffered trace features for select system-on-a-chip (SoC) devices. You can quickly spot challenging intermittent defects that take a long time to analyze with traditional debug methods. The network-based JTAG emulator Wind River ICE 2 can be extended with the optional Wind River Trace 2 module to provide externally buffered trace capabilities on ARM9 processors with ETM v1. Workbench On-Chip Debugging also supports Freescale MPC85xx internal trace capabilities with all Wind River emulators.

You can view the events that occurred in the execution of the trace, such as the start of the trace and end of the trace, along with a description of the event. You can then select an event from the trace event field to locate the specific event occurrence in the trace buffer. Specific trace fields include the following:

- **Event occurrences (unlabeled):** Type of trace event
- **Address:** Address or line number of trace event
- **Absolute time:** Elapsed time since the beginning of the trace
- **Delta time:** Change in absolute time since the last trace entry
- **Instruction (unlabeled):** Executed instructions that can be configured to display code at the function, source, or disassembly stage

The flexible trace configuration options enable you to configure and set up the display of trace data, clear the trace buffer, reset the Trace view, save the trace to file, set trace rules, and configure trace filtering.

Multicore Technology

Multicore refers either to a single chip containing multiple logical devices capable of executing code, or multiple processors on a single target board. In addition, in system-level design, you may have multiple processors across multiple boards. The emergence of these multicore processors is creating new debug challenges, requiring innovative technology that leverages the open environment of the Eclipse framework.

Multicore Debugging Challenges

Multicore offers more functionality and higher performance, but it also poses new debug challenges for hardware and software developers:

- How to effectively visualize and manage the edit-compile-and debug cycle across multiple cores
- How to optimize the JTAG interface with many SoCs, leveraging a single JTAG interface to save on costs
- How to manage real-time performance requirements for multicore debugging
- How to support multiple processors from different vendors
- How to support the debug of multiple operating systems across different cores

The Workbench On-Chip Debugging solution offers a single project-oriented framework for debugging the most complex multicore scenarios. It allows you to view all your multicore projects from a single interface, whether the cores are on a single board or on multiple boards in an integrated system.

Value of Eclipse in Multicore Debugging

A common debug framework with a strong ecosystem is the foundation of a multicore debug strategy. While some take a proprietary approach in integrated development environments, Wind River leverages the Eclipse framework to reduce the complexity of the edit-compile-debug process.

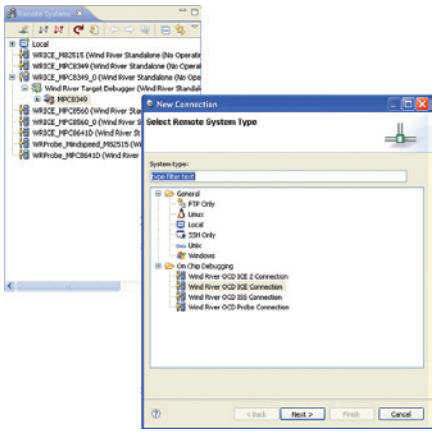


Figure 7: Target connection manager simplifies attachment to multiple cores and supports multiple connection methods

Target Connection Manager

An easy-to-use target connection manager simplifies the attachment to multiple cores and supports multiple connection methods, including the JTAG emulator, Wind River's Transparent Mode Driver (TMD), software agent, or simulator. The Wind River solution connects to up to 128 cores in a system and debugs up to eight of those cores simultaneously through a Wind River emulator JTAG connection, solving the problem of managing multiple cores through a single JTAG interface. In addition, the Wind River Connect extension module for the Wind River emulators connects up to four different scan chains simultaneously to provide the scalability required in system-level multicore development.

The Target Connection Manager is based on the Eclipse Remote System Explorer (RSE) framework, providing you with remote access to target file systems through a number of network-based protocols.

Advanced Debugging Features

When you're debugging multiple cores, managing and tracking hardware and software breakpoints becomes even more critical. Workbench On-Chip Debugging synchronizes the run-control of a system to start and stop the entire system. You can set cross-correlated breakpoints and track sessions with multiple targets. The debug sessions are color-coded and numbered, so you can quickly spot an issue on an impacted core.

With advanced multicore diagnostics, you can inject errors, such as taking down a specific core, and analyze the impacts on other cores in the system to quickly spot software and hardware dependencies across multiple cores.

Cache memory management tracks cache coherency issues with main memory across multiple cores, including the ability to monitor instruction execution and identify the difference between data stored in memory versus cache. By tying the Cache view back to the editor, you can identify the source code and function for each instruction in the Cache view, improving the troubleshooting of cache coherency problems across multiple cores.

High-Performance JTAG Server

Traditional JTAG servers have limitations in multicore debugging environments, such as low performance and a limited range of semiconductor processor debug support. Wind River brings you all the advantages of a JTAG server: the ability to leverage the debug capabilities of the processor and manage multiple cores with a single JTAG connection, without any of the problems associated with older single-core JTAG server technology.

Wind River's patent-pending technology optimizes the performance of the JTAG

server, providing the speed and reliability required for real-time application debugging. The scope of processor support ensures deep debug and analysis capabilities on the leading SoCs.

OS and Processor Support

Workbench On-Chip Debugging supports many operating systems, including Wind River's VxWorks, Wind River Linux (including Wind River Real-Time Core), Linux 2.4 and 2.6 kernels, and Express Logic's ThreadX. Wind River Professional Services ports are available for OSE and Nucleus.

Linux and VxWorks support provides visibility and debugging at the OS level, as well as applications executing on the target, allowing debugging of complex application-kernel interactions.

Linux user-mode debugging support is also provided for debugging application code on Linux devices. Workbench On-Chip Debugging gives developers the ability to debug the Linux kernel, user applications, and shared libraries, without requiring kernel instrumentation. The software also enables developers to see both system and user application contexts when debugging in Linux and eases debugging of system-application interaction issues.

The Third-Party Operating System (TOS) API allows Wind River Professional Services to add support for an in-house proprietary operating system or a third-party commercial operating system. Kernel object data structure is captured using XML, enabling debug support for all defined kernel objects and their task context associated views. The API comes with documentation, an XML file template, and a VxWorks 6 reference file.

Wind River Hardware

Wind River Probe

Wind River Probe is the most comprehensive tool on the market for board bring-up, flash programming, and production/test. It uses embedded on-chip debugging services in a microprocessor and Wind River JTAG accelerator technology to deliver high-performance debugging over a USB

Figure 8: Wind River Probe

Figure 9: Wind River ICE 2

connection, offering fast download speeds (in excess of 1.5Mbps on some processors), high debug throughput, and unprecedented debug efficiency. Probe can also support the internal trace buffer provided on Freescale's MPC85xx processor family.

Wind River ICE 2

This advanced network-based JTAG emulator has been designed for today's complex processor environments, including 32-bit and 64-bit single and multicore implementations. Simple enough to support the most basic debugging needs but robust enough to debug multiple JTAG devices simultaneously in a single scan chain of up to 128 individual EJTAG/JTAG devices, Wind River ICE 2 excels in complex system debugging and multisite development.

The same emulator used in the development process can be deployed in test and manufacturing to ensure end-to-end integration. Through its JTAG server capability, Wind River ICE 2 supports multiple JTAG/EJTAG devices on a single scan chain as well as multiple debugger connections to target devices. Wind River ICE 2 also offers the ability to support capture of internal trace buffer information to provide added run-time visibility.

Wind River Trace 2

The Wind River Trace 2 external trace module extends the capability of Wind River ICE 2 to include real-time trace capability for supported processors and provide better visibility into hardware/software interaction on the target platform. This enables developers to identify and resolve the most difficult program-flow problems such as when software is randomly crashing the target or when the root cause is not easily found using standard system-level debugging methods via register and memory access. Benefits include the following:

- 1GB trace buffer for storage of instructions and timestamp information
- Ability to capture real-time trace at clock speeds up to 200MHz
- Fast hardware-based post-processing for efficient navigation of trace buffer
- Integration with Workbench On-Chip Debugging for program-flow monitoring and user-specified trace configuration and event filtering

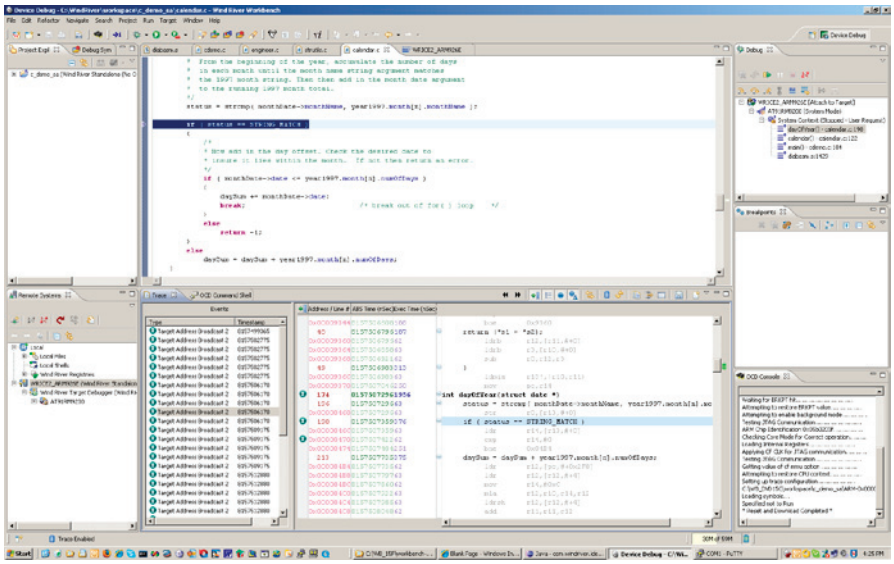


Figure 10: Wind River Trace 2 capture window

Technical Specifications

Host OS Support*

- Red Hat Enterprise Linux Workstation 4 (Update 3), 32-bit x86
- Red Hat Enterprise Linux 5, 32-bit x86 and 64-bit x86-64
- Ubuntu 8.04 Hardy Heron
- Solaris 9, 32-bit SPARC
- Solaris 10, 32-bit SPARC
- openSUSE 10.3, 32-bit x86
- SUSE Linux Enterprise Edition 11, 32-bit, x86
- Fedora Core 7, 32-bit x86
- Windows XP Professional with Service Pack 2, 32-bit, x86
- Windows Vista, 32-bit x86

**Wind River Probe is supported on a subset of Linux hosts. Contact your Wind River sales representative for further information.*

Target OS Support

- VxWorks 6.3, 6.4, 6.5, and 6.6
- VxWorks 5.5
- VxWorks 653
- Wind River Linux platforms (including Wind River Real-Time Core for Linux)
- Linux 2.4 and 2.6 kernels (Workbench On-Chip Debugging only)
- Native Linux development on Red Hat Enterprise Linux 4
- ThreadX 4.0 and 5.0 (Workbench On-Chip Debugging only)

Customizable target OS awareness capability for Workbench On-Chip Debugging enables support for other target operating systems to be added.

VxWorks 6.x

- List of kernel tasks and object/stack summary for each
- List of RTPs and object summary for each
- List of RTP tasks and object/stack summary for each
- List of semaphores and object summary for each
- List of message queues and object summary for each
- List of I/O devices and object summary for each
- List of I/O drivers and object summary for each
- List of I/O file descriptors and object summary for each
- List of ISRs and object summary for each
- List of memory partitions and object summary for each
- List of triggers and object summary for each
- List of watchdogs and object summary for each
- List of sockets and object summary for each
- List of shared libraries and object summary for each

VxWorks 5.5

- Task/process
- Wind River Linux
- List of processes
- List of threads
- User-mode application contexts showing application variables, stack trace, and labels

ThreadX

- Thread list
- Mutex list
- Semaphores
- Block pools
- Byte pools
- Event flags (type of message queue, not as robust)
- Timers (similar to watchdog timers)

Processor Support

Wind River on-chip debugging solutions support a wide range of processor architectures including ARM, ColdFire, MIPS, and PowerPC. Wind River is continually adding new processor support capabilities. For the latest details on supported processors, contact your Wind River sales representative.

Professional Services

Wind River Professional Services helps companies to reduce risk and improve competitiveness. Our team delivers device software expertise within structured engagements that directly address key development challenges and contribute to the success of our clients. Our track record of timely delivery and in-depth understanding of market and technology dynamics makes Wind River a valuable implementation partner for clients worldwide. Based on our commercial-grade project methodology, service offerings include device design, BSP and driver optimization, software system and middleware integration, and legacy application and infrastructure migration.

Workbench Services

Wind River Professional Services knows how to jump-start your development efforts. Even if you opt for a non–Wind River platform, Linux distribution, host operating system, or target architecture, we can help. No matter which development environment you use, Wind River Professional Services can extend Workbench to adapt to your needs with the following offerings:

- Extend Workbench processor support
- Extend Workbench target OS support
- Validate Workbench on Linux host environment
- Validate Eclipse plug-ins
- Integrate agents

Installation and Orientation

Proper installation and orientation of Wind River Workbench, On-Chip Debugging Edition means you won't waste time solving easily avoidable problems before you can begin your next development project. Wind River offers an Installation and Orientation Service to ensure that your project starts on time and without hassle by delivering the following:

- **Onsite installation:** Guided install on your hardware and host platform, along with a sample build process, demonstrations, and examples of customizations
- **Hands-on orientation:** Architecture, development file system, adding open source packages, porting drivers, addressing design issues
- **Advice:** Introduction to Wind River support channels and processes, additional services, project review, and consultation

The Wind River Installation and Orientation Service will expedite your path to productivity, allow you to rest assured that we have eliminated a common source of user error, and help you realize all of the platform's potential.

Education Services

Wind River offers in-depth and comprehensive product training for Wind River Workbench, On-Chip Debugging Edition 3.1. After completing these training courses, customers can successfully design, develop, build, test, and debug device software applications in a target-host environment. Classes are offered in a variety of formats including a wide variety of regularly scheduled public courses and customer-specific onsite courses.

Wind River Personalized Learning Program is a comprehensive, systematic approach to deliver the best possible education to all developers. Unique needs of each team member are assessed, and development plans, training materials, and learning events are recommended to address work-related skill gaps. This approach results in a rapid increase in productivity.

Support Services

Wind River provides full technical support. Our global support organization is staffed with engineers who have extensive experience with Wind River products and device software development. At major support centers worldwide, our local experts can help diagnose problems, provide guidance, or answer "How do I...?" questions. Support is also available 24 hours a day at our Online Support (OLS) website (www.windriver.com/support) or by email at support@windriver.com.

Visit OLS for fast access to product manuals, downloadable software, and other problem-solving resources. Additional features, including patches and technical tips for common problems, are available for all customers upon subscription. OLS visitors can also access a community of developers to discuss their issues and experiences.

If you cannot find the information you need through OLS, contact our global support team for access to the industry's most knowledgeable and experienced support staff.

North America, South America, Asia/Pacific

support@windriver.com
Toll-free tel.: 800-872-4977 (800-USA-4WRS)
Tel.: 510-748-4100
Fax: 510-749-2164
Hours: 6:00 a.m.–5:00 p.m. (Pacific time)

Japan

support-jp@windriver.com
Tel.: +81 3 5778 6001
Fax: +81 3 5778 6003
Hours: 9:00 a.m.–5:30 p.m. (local time)

Europe, Middle East, Africa

support-ec@windriver.com
Toll-free tel.: +800 4977 4977
France tel.: +33 1 64 86 66 66
France fax: +33 1 64 86 66 10
Germany tel.: +49 899 624 45 444
Germany fax: +49 899 624 45 999
Italy tel.: +39 011 2448 411
Italy fax: +39 011 2448 499
Middle East region tel.: +972 9741 9561
Middle East region fax: +972 9746 0867
Nordic tel.: +46 9 594 611 20
Nordic fax: +46 8 594 611 49
UK tel.: +44 1793 831 393
UK fax: +44 1793 831 808
Hours: 9:00 a.m.–6:00 p.m. (local time)

How to Purchase Wind River Solutions

Visit www.windriver.com/company/contact-us/index.html to find your local Wind River sales contact. To have a sales representative contact you, call 800-545-9463 or write to inquiries@windriver.com.

WIND RIVER

Wind River is the global leader in Device Software Optimization (DSO). We enable companies to develop, run, and manage device software faster, better, at lower cost, and more reliably. www.windriver.com