

# WIND® POWER IDE



WIND RIVER

WIND RIVER HARDWARE TOOLS

## An integrated development environment for every phase of your project.

Wind River's **WIND®POWER IDE** is a complete integrated development environment for hardware bring-up and embedded applications. By combining state-of-the-art debugging facilities with the simplicity of a modern development environment, Wind River's **WIND POWER IDE** brings C/C++ source-level debugging and embedded application development to a new level.

**WIND POWER IDE** provides a highly visual and automated environment that accelerates the development of even the most complex applications, so creating applications is fast and easy for developers of all experience levels. **WIND POWER IDE** provides a single development environment that's consistent across all supported workstations and PCs. On each of the supported platforms, the features and the usage are identical. There's no need to worry about host-to-host incompatibilities.

**WIND POWER IDE** contains all of the tools needed to complete a major programming project:

- *Project Manager*: Provides intuitive project facility tools to build complex programming projects and shorten development time
- *Text Editor*: Provides a full-featured text editor tightly integrated with the graphical symbol browser

- *Graphical Symbol Browser*: Provides source code browsing capabilities integrated with the **WIND POWER IDE** text editor and debugger
- *Source-Level Debugger*: Provides a high-performance windowed source-level debugger equipped with the latest productivity-enhancing graphical features to shorten board bring-up and application development time
- **DIAB C/C++ COMPILER** suite: Includes the industry-leading C/C++ language **DIAB® COMPILER**, including a Standard Template Library (STL) and a variety of other tools
- *Instruction Set Simulator*: Integrated instruction set simulator for jump-starting application development

### **WIND POWER IDE Project Manager**

The **WIND POWER IDE** Project Manager provides a powerful graphical user interface to simplify organizing, configuring, and building complex programming projects, automating many aspects of managing a project.

The Project Manager component performs automatic dependency analysis and generates makefiles. The powerful graphical user interface enables the inexperienced user to configure a project by selecting from menus of options covering everything from opti-

## Features

- Superior optimization technology generates fast, compact, high-quality code
- Proven reliability and flexibility to meet extreme design constraints
- Compatibility with the latest ANSI C++ specs (ISO/IEC 14882:1998[E] C++ standard) and the ANSI C spec (X3.159-1989)
- Standards conformance (ANSI and EABI) for maximum tool interoperability
- Complete control of code and data memory allocation
- Options to pack or byte-swap structures to match existing data types
- Supports position independent code (PIC) and data (PID)
- Implement ISRs from C code
- File I/O for bare board applications (no OS)
- Proven performance with VxWorks®

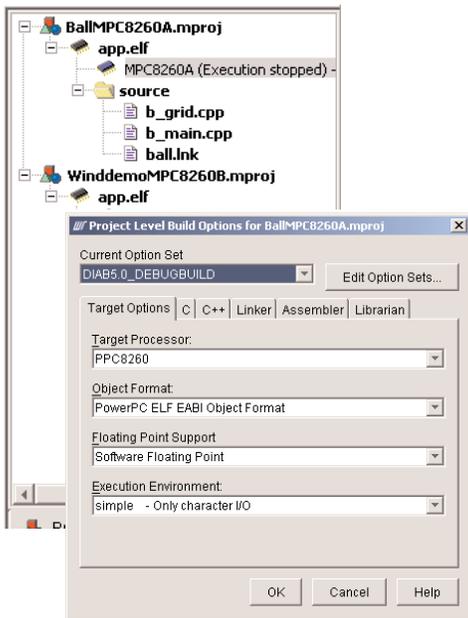
### Supported hosts\*

- Windows® NT4.0 + SP5 and above/2000/XP
- Solaris™ 2.7/2.8/2.9

### Supported targets

- MIPS64\*  
PMC-Sierra RM9000  
MIPS 5Kc  
MIPS 20Kc  
Toshiba TX99
- PowerPC\*  
IBM 7xx  
Motorola 74xx, 82xx, 85xx  
Xilinx Virtex II Pro
- MIPS32\*  
4Kc, 4Km, 4Kp, 4Kec, 4Kep, 4Kem, PNX8550

\* Contact your Wind River Sales Representative for availability



mization level, debugging level, and language-specific features to target type (executable or library) and much more. The new Project Wizard takes the developer step-by-step through a series of questions to create an empty project or a template as a starting place for the application. The templates include a linker command file and makefiles that make it possible to associate debug connections easily. Templates are provided for every supported CPU and programming language supported by the **WIND POWER IDE** compiler.

### WIND POWER IDE Text Editor

**WIND POWER IDE** includes a full-featured, user-configurable, windowed text editor with features such as syntax coloring and auto-indenting. Syntax coloring helps quickly identify language keywords and constructs, including comments, strings, constants, and more.

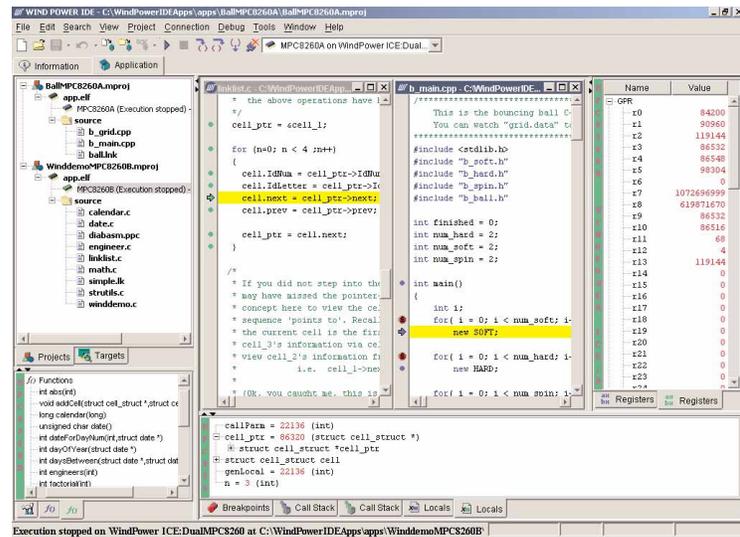
The **WIND POWER IDE** text editor implements all of the standard functions that are expected from an editor, including a powerful search feature that can find values within multiple files.

The **WIND POWER IDE** text editor is fully configurable, so the developer can change the key bindings, font type, font size, color scheme, syntax coloring, and more. The **WIND POWER IDE** text editor also provides a single, consistent editor

interface for all host and target development combinations. It's an integral part of the overall IDE and can be invoked and controlled as an object from other components within **WIND POWER IDE**.

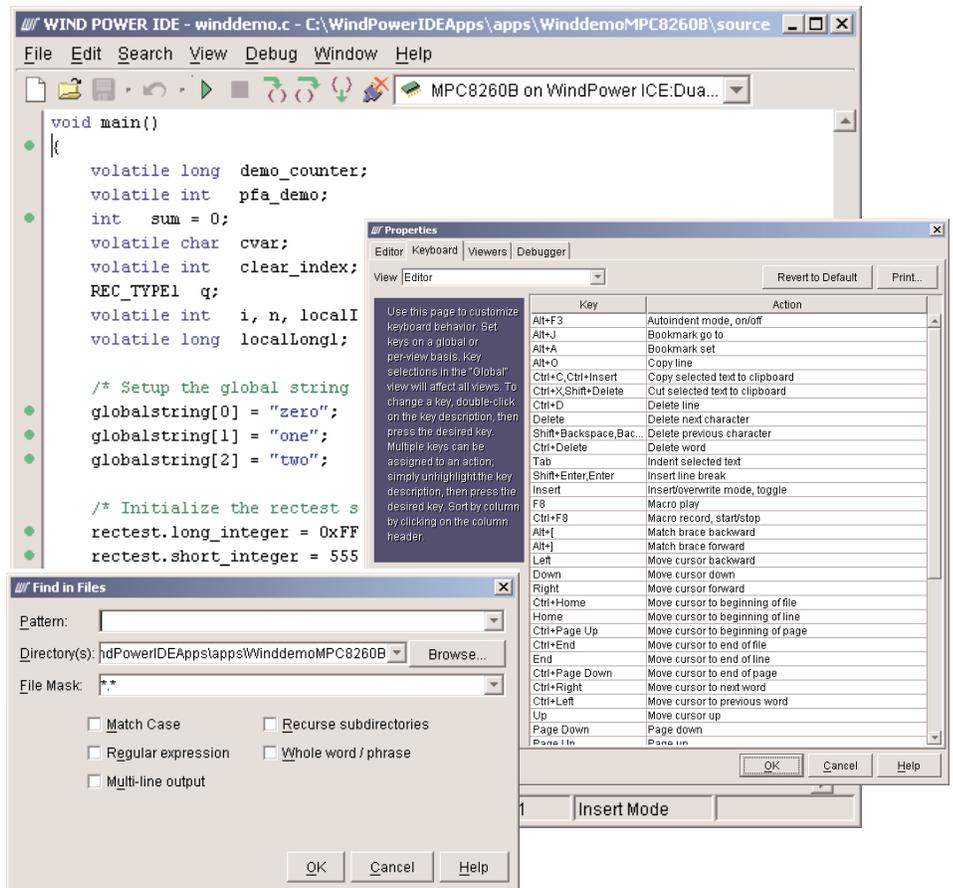
### WIND POWER IDE Graphical Symbol Browser

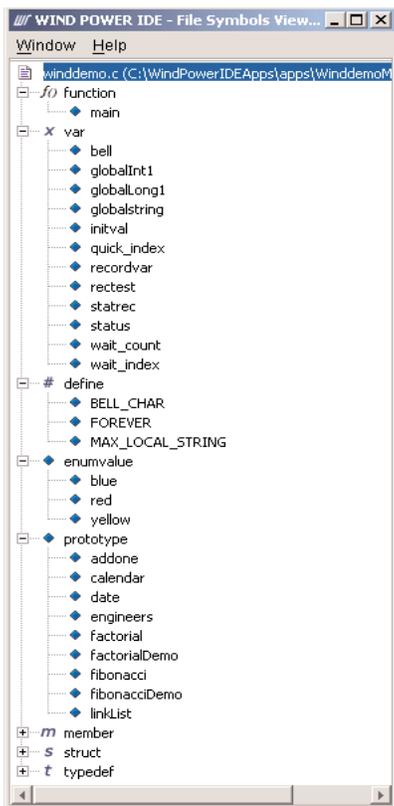
Industry observers estimate that software developers spend nearly half their time searching for basic information buried in application code. As applications grow in complexity, the time required to find, analyze, and modify code grows as a proportion of total engineering effort. The **WIND POWER IDE** graphical symbol browser reduces this largely unproductive time by integrating code browsing and searching into a single tool.



The **WIND POWER IDE** graphical symbol browser provides fast, semantic code navigation that makes it possible to find specific code structures, so finding a symbol or pattern among hundreds of directories and files is fast and easy.

The seamless integration between the **WIND POWER IDE** graphical symbol browser and the text editor means that all changes in the code are immedi-





interface to make debugging fast, flexible, and easy.

### Window-based Environment

The **WIND POWER DEBUGGER** enables developers to operate more efficiently with user-friendly debugging, multiple windows, point-and-click capabilities, and outline format.

**WIND POWER DEBUGGER'S** interface allows users to customize the GUI to fit their needs: to create custom buttons, toolbars, and menus, and to "float" windows that are an integral part of the debugger so that they become independent windows on the workstation.

This provides increased visibility and control over the display of information in the debugger. Windows that have been separated from the debugger can also be "docked" to rejoin the main debugger window controls.

The **WIND POWER DEBUGGER'S** desktop allows users to focus on complex debugging tasks. Each desktop contains just the set of views needed for the task at hand. The application desktop provides a high-level view of the target software, and the hardware desktop provides a low-level view of the target hardware.

### Seamless Integration

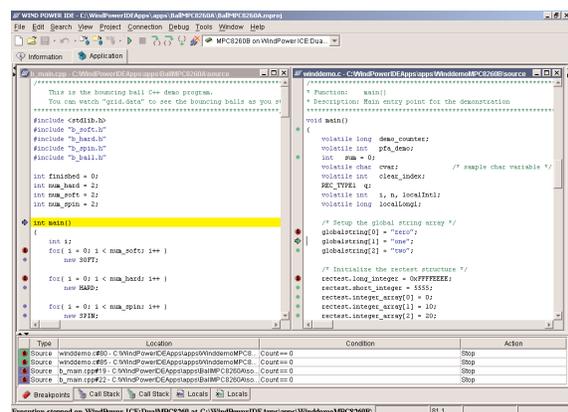
The **WIND POWER DEBUGGER** is fully integrated with **WIND POWER ICE** debugging solutions as well as the Tornado® environment, resulting in optimized run control and faster downloads.

### Full-Featured Debugging

The **WIND POWER DEBUGGER** provides a rich set of debugging features designed to help the developer quickly find and fix bugs, including:

- **Breakpoints:** Breakpoints are easily set in source code by clicking on a "code-dot" in the window margin that indicates the point where breakpoint insertion is possible. Once the breakpoint has been set, the "code-dot" changes appearance

to indicate the setting. Removal of the breakpoint is just as easy; simply click the "dot" and the breakpoint is automatically



removed. Performing a right-click on the breakpoint enables the behavior of the breakpoint to be changed to make it conditional. It can also be changed to a hardware breakpoint, or attached as an action to the breakpoint that is performed once the breakpoint is hit.

- **Single-stepping:** The **WIND POWER DEBUGGER** supports the following single-stepping mechanisms:
  - Step Into: traces execution of every individual instruction
  - Step Over: does not trace into the called function
  - Step Out: brings execution back to the calling function
- **Tooltips:** Tooltips enable the developer to view crucial information easily. Data Tooltips display a quick, one-time view of a variable, while Icon Tooltips display an item's function when the cursor is placed over it.
- **Simple module and function browsing:** The tight integration between the **WIND POWER IDE** Graphical Symbol Browser and the **WIND POWER DEBUGGER** enables access to an internal table of all modules, global variables, and functions in a given debug context. With a single right-click, it's possible to edit code, run it to a target address, or set breakpoints at entry or exit.

ately reflected in the browser. No recompilation is necessary.

With the **WIND POWER IDE** graphical symbol browser, the mouse can be used to navigate between the different symbols. Just place the mouse cursor on a symbol and right-click to invoke the text editor, which will open the file and highlight the exact location of the selected symbol.

### WIND POWER IDE Source-Level Debugger (WIND® POWER DEBUGGER)

By combining a state-of-the-art IDE with the simplicity of a windowed environment, Wind River's **WIND POWER DEBUGGER** brings C/C++ source-level debugging to a new level. The **WIND POWER DEBUGGER** assembles a wide array of high-powered tools and features into a powerful graphical user interface to help get projects to market ahead of schedule and under budget.

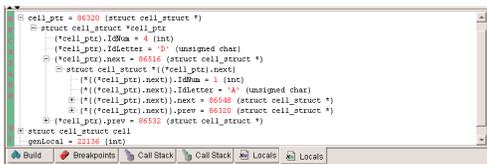
All of the **WIND POWER DEBUGGER** hardware and software features provide simple access and execution. Any debug operation desired is done through an intuitive "point-and-click"

- **Display stack trace:** The “Call Stack” view provides an easy display of all procedures active in the calling chain, and enables the developer to follow the



progress of a program through its hierarchical call structure. The trace information includes the name of each procedure, module name, line number at source-level, physical address in memory, and the name and value of each argument.

- **Local variables display:** The local variables display shows the

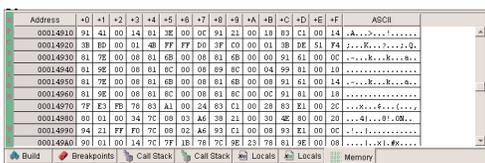


variables local to the current function. As running code moves from function to function, the contents of the local variables view change to display the local variables of the current function being viewed.

- **Displaying data:** WIND POWER DEBUGGER offers three ways to view data:

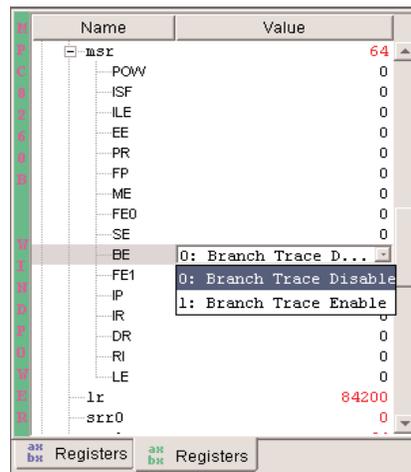
- **Data Tooltip:** displays the values of a variable directly from the source code.
- **Instant Watch:** provides a view of the variable’s data in a popup view that allows pointers to be followed.
- **Watch view:** for monitoring and updating data in a separate window.

- **Memory view:** Memory view is the fastest way to display and modify the contents of target memory. Features include automatic alignment, find in memory, the ability to compare



two memory regions or memory and a file, uploading memory contents to a file, filling memory with a known value, freezing the memory view to prevent target access, invoking multiple memory views, cutting, pasting, and more. The memory can be formatted in a variety of ways, including hexadecimal, decimal, octal, ASCII, binary, big/little endian, and others.

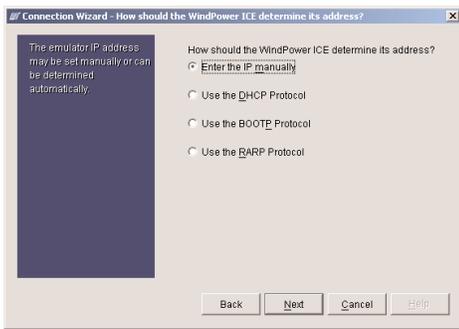
- **Register view:** The register view provides extensive information on CPU core and peripheral registers, as well as up to 128 user-defined custom registers. All of the registers displayed can also include bit level details on register contents. Bit level details are formatting details that break down and describe the contents of bit-mapped registers, making it easy to interpret the register contents. Bit field values are displayed as English-language equivalents of bit field patterns. Picking a value from a pull-down list or manually entering the register value can result in bit field value changes within the registers.



- **Flash programming:** Program on-board Flash devices from within the same graphical user interface used to debug the application. No boot code is required to run on the target system in order to use the programming features of the WIND POWER DEBUGGER.

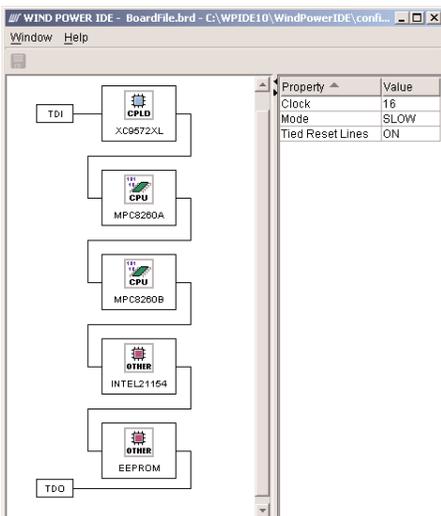
- **Object file format:** The WIND POWER DEBUGGER supports ELF/DWARF 1 and 2 object file output formats.
- **Board bring-up:** The WindPower Debugger helps developers deal with the complexity of bringing up a board by providing complete control over all board settings, including initial register values and memory configuration. After initial target register values are defined, the debugger restores these values each time the user connects to their board. Then an assembler source file can be created from these settings as an addition to the project. The WIND POWER DEBUGGER also includes a comprehensive set of hardware diagnostics and robust flash programming to support an extensive list of flash devices.
- **Multi-core/CPU debugging:** The WIND POWER DEBUGGER combined with WIND POWER ICE enables debugging of multi-core System on Chip (SoC) and multiple-CPU targets. Every core has its own independent register view, memory view, stack view, disassembly view, source view, and more. It’s not necessary to run multiple instances of the debugger (as other products require). And it’s possible to debug a mixture of different types of cores. Features like “stop all cores” and “run all cores,” as well as single-stepping some cores while the other cores are running provide the power to synchronize and control debugging sessions to match the target behavior. The WIND POWER DEBUGGER is designed to support multiple debug sessions running on one or more hosts simultaneously.
- **Mixed language debugging:** The WIND POWER DEBUGGER supports mixed language debugging in C, C++, and Assembly Language. When moving between source modules written in different languages, the WIND POWER DEBUGGER

automatically analyzes the language of the file in view and adjusts the expression evaluation



and data display accordingly.

- **Target connection wizard:** Connecting to hardware just got easier with the new **WIND POWER DEBUGGER** Connection Wizard. The Connection Wizard simplifies and automates the task of defining new connection definitions based on hardware and communication parameters.
- **Board file editor:** The board file editor is used to create a board description file that contains the information pertaining to the devices located on the scan chain for the target. Information created includes the type of device (such as microprocessor, CPLD, or FPGA), the target type, the instruction register length, and operations that are valid for each device. Use the board file editor to create a board description file that exactly describes the custom scan chain layout.



The board file editor also enables **WIND POWER ICE** to communicate to the target.

### DIAB C/C++ COMPILER suite

Wind River's **DIAB COMPILER** suite combines industry-leading optimization technology with the flexibility and control needed to fully exploit today's complex CPUs. The **DIAB COMPILER**'s design is based on a modular architecture that results in proven reliability and flexibility for embedded applications, as well as interoperability with other development tools.

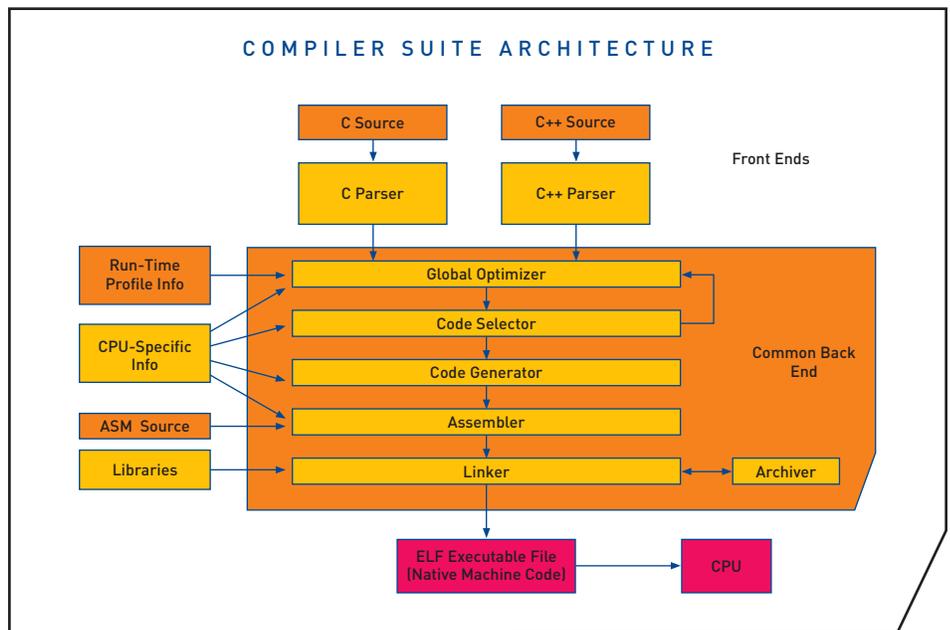
### Modular Architecture

The **DIAB COMPILER** provides language-specific front ends for C and C++ that parse the original source code into a

silicon with very short lead times. In short, the **DIAB COMPILER**'s modular architecture enables the developer to immediately gain maximum performance from a compiler/silicon investment.

### Superior optimization technology

Wind River's **DIAB COMPILER** suites produce exceptionally fast, compact, high-quality object code. A very large number of highly refined, global, local, CPU-specific, and application-specific (profile-driven) optimization techniques enable the programmer to fine-tune the compiler's output to match the application's requirements. Users can select various optimizations to balance execution speed with code size while intelligent defaults can generate optimal code "out of the box."



common token-based representation of the source. Optimizations are applied to this intermediate language representation. Also, the fully optimized code is converted into the appropriate machine code through a robust, table-driven back-end module. Wind River's close relationship with silicon partners, combined with the **DIAB COMPILER**'s modular design, make it possible for the **DIAB** tool chain to provide highly optimized compilers for new

### DIAB COMPILER tool chain components

In addition to the compiler, the **DIAB COMPILER** tool chain is a complete development suite that includes the following programs/utilities:

- **Driver:** An intelligent wrapper program used to invoke the compiler, assembler, linker, and so on.
- **Assembler:** A full-featured macro assembler that is invoked automatically by the driver program

or as a complete standalone assembler for generating object modules. Key features include:

- Conditional macro assembler with over thirty directives
- Unlimited number of symbols
- Debug information for source level debugging of assembly programs
- **Linker:** Offers precise control over the allocation, placement, and alignment of code and data in memory. The linker:
  - Links object modules into absolute or re-locatable modules
  - Reads/writes/mixes ELF and COFF object files
  - Generates fully EABI compliant ELF/DWARF output for tool interoperability
- **Libraries:** The following extensive libraries are included in the tool chain:
  - Complete C++ library (STL)
  - Complete, re-entrant C libraries compliant with ANSI/ISO, POSIX, and SVID standards
  - Full complement of math libraries, including IEEE-754 Appendix functions
  - Efficient floating-point libraries for fast execution of calculations
  - Source code supplied
- **Archiver/Librarian:** A standalone archiver provides all functions needed to create and maintain libraries
- **Profiler:** Profiling options contained in the compiler instrument application code, which when executed save profile information that can be viewed by the profiler utility.

This profile data can also be automatically fed back to the compiler for additional code optimization based on execution paths

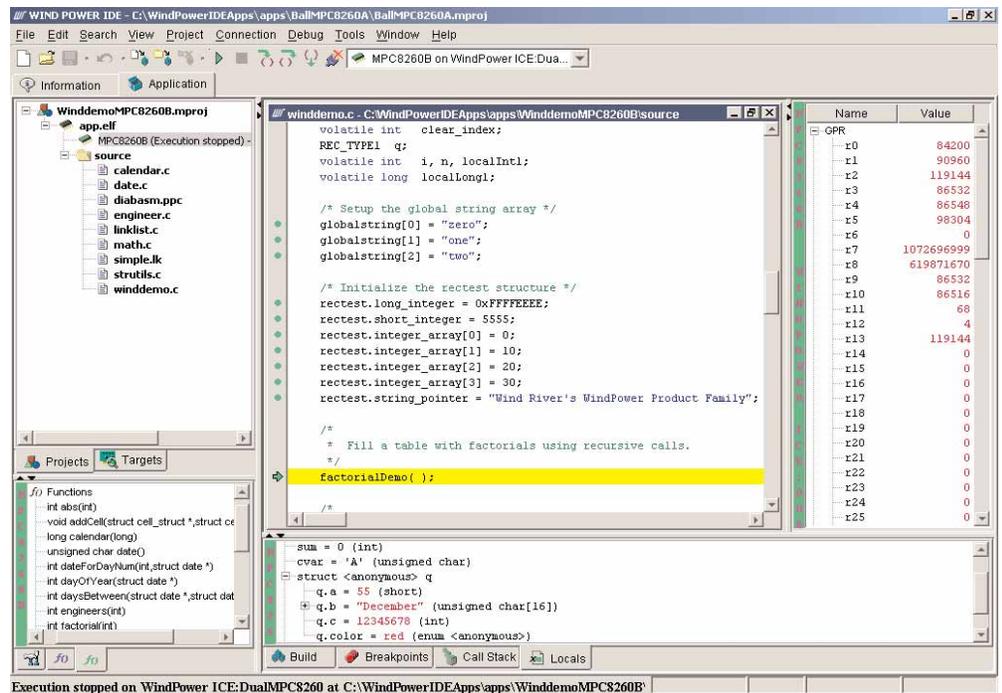
- **Object file converter:** In addition to providing conversion routines for generating S-Record format, this utility program also contains options for:
  - Viewing symbol tables
  - Code size details
  - C++ symbol name demangler
  - Munch routine
  - More

### WIND® POWER INSTRUCTION SET SIMULATOR

The **WIND POWER INSTRUCTION SET SIMULATOR** provides a quick and easy way to begin developing code without the requirement for access to hardware. The ability to

develop software without requiring hardware provides a number of significant benefits to software engineers, including the ability to run code before custom hardware is ready, running/testing code when hardware resources are limited, and learning how to use the development environment without first having to get hardware running.

The **WIND POWER INSTRUCTION SET SIMULATOR** provides full instruction simulation and fully supports standard C library I/O. It is fully integrated with the **WIND POWER IDE**, and also provides a full command-line interface. The **WIND POWER INSTRUCTION SET SIMULATOR** is available for the Motorola PowerPC family.



**WIND RIVER**

**Wind River Worldwide Headquarters**  
 500 Wind River Way  
 Alameda, CA 94501 USA  
 Toll free 1-800-545-WIND  
 Phone 1-510-748-4100  
 Fax 1-510-749-2010  
 Inquiries@windriver.com  
 Nasdaq: WIND

For additional contact information, please see our Web site at [www.windriver.com](http://www.windriver.com).

Wind River, the Wind River logo, Tornado, and VxWorks are registered trademarks of Wind River Systems, Inc. Any third-party trademarks referenced are the property of their respective owners.

For further information regarding Wind River trademarks, please see: [www.windriver.com/corporate/html/trademark.html](http://www.windriver.com/corporate/html/trademark.html)

©2003 Wind River Systems MCL-DS-WPI-0301