Wind River Product Technical Overview
Problem

**Inefficient software development and deployment:**

- Developers lack the capability needed to efficiently complete the task at hand or find and correct problems.
- Development tools fail to support “real world” applications because of limited scalability.
- Changes and additions made to complex code bases are difficult and cause hidden problems.
- Closed interfaces for tools limit integration of the development environment.
- Developers are using different tools within and across projects, inhibiting collaboration and effectiveness.
Customer Situation

Solution ➤

**Standardization with Wind River Workbench:**

- Broad range of powerful capabilities from HW bring-up through application development in one environment
- IDE capabilities proven to scale to real-world application
- Code Analysis provides insight to code usage and relationships
- Eclipse open framework for easy integration of 3rd party tools
- Support for the most popular target operating systems and processors
Customer Situation

Result ➤

Impact on development projects:

• Focus on product innovation, not on tool integration

• Debug complex products with coordination

• Skills with development tools are portable among projects

• Reduction in training cost and time
Wind River’s DSO Offering

Wind River Professional Services

Wind River Enterprise Licensing Model

Wind River Partner Ecosystem

WIND RIVER PLATFORM

Integrated Development Environment

Industry-Specific Middleware

VxWorks  Linux

Wind River enables companies to develop and run device software faster, better, at lower cost, and more reliably.

WIND RIVER
Wind River Workbench 2.2

- Highly optimized environment for developing VxWorks 6.0 and Linux device software
- Makes debugging, porting and bring-up of VxWorks 6.0 and Linux devices more efficient
- Easily customizable and extensible with 3rd party plug-ins via the Eclipse framework
Eclipse

Eclipse is an open platform for tool integration that leverages open-source licensing and a community of tool developers.

Eclipse provides
- GUI framework and tool integration capability
- Openness, extensibility, standardization
- Integration via plug-ins for over 350 different software tools (ClearCase, PVCS, and SlickEdit, and others)
# Project Facility and Build System

## Project Facility

- Fits into any existing environment
- Easy configuration of the VxWorks 6.0 Kernel and application projects
- Easy setup of Linux kernel and application projects
- Graphically configurable

## Build System

- Extremely powerful and customizable Makefile generator
- Imports existing Makefiles to quickly navigate to compiler errors in the IDE
- Support for scripts and nightly updates

---

**Wind River**
Source Analysis

Features
- Symbol browsing
- Function call trees
- Dependency graphs
- Class and diagrams
- Include hierarchies
- Code completion
- Conditional code highlighting

Benefits
- Better understand Linux kernel updates and open source software
- Better understand legacy software
- Port to custom hardware more quickly
- Simplify code reviews
Navigate instantly
- Through millions of lines of code
- Thousands of files and directories
- Without getting lost
- Every code symbol at your finger tip

Better comprehend source code
- Excellent dependency visualization
- Rapid employee ramp-up

Make code reuse reality
- Understand VxWorks 6.0 and Linux source code
- Port to custom hardware more quickly
- Simplify code reviews

How do developers spend their time?

Time Savings with Wind River Workbench

Source: Software Quality: Producing Practical, Consistent Software
Mordecai Ben-Menarche & Garry S. Marlis, Thomson Computer Press

Wind River
Wind River Debugger

Only debugger that can concurrently view, control and debug multiple:
• Tasks (VxWorks 6.0)
• Real Time Processes (VxWorks 6.0)
• Processes and threads (Linux)
• Processors and operating systems

Superior Wind River Target Agent (WDB)
• Single instance of the debug agent for both task (user) and system mode debugging
• Only debugger that can switch between Kernel and User mode debug sessions via a single agent

Asynchronous Controls/Messaging
No need to wait for a command to complete

Smart Window Update Policy
• Only update what is needed
• Minimize target impact
Wind River Compilers for VxWorks

Wind River Compiler (WRC) 5.2.2
• Default compiler for building the VxWorks 6.0 kernel, libraries, and BSPs
• C++ COMDAT improvements
• Integrated run-time error checking

GNU 3.3.2
• New version for VxWorks 6.0
• Optional license monitoring and management
• More closely aligned with Open Source Community
• Exceptions are options required to configure VxWorks

New Eclipse compiler options dialogue for both WRC and GNU
• Select target architecture
• Specify compile options, e.g., speed, size, etc.
• Define debug information

WIND RIVER
Wind River System Viewer

- Graphical visualization of all system activity over time
- Multiple data collection and upload methods including ring-buffer and post-mortem
- Data export for advanced analysis
- Operating System Support
  - Linux (Linux Trace Tool data)
  - VxWorks 6.0 (including triggering)
Wind River VxWorks Simulator

Complete simulation of VxWorks 6.0
- Real Time Processes (RTP)
- Message Passing
- Error Management

Ability to create virtual networks
- Create virtual systems
- Connect VxWorks simulation environment to real system components

Hardware layer simulation
Wind River OCD Tools

Debugging enhanced with the addition of On-Chip Debugging support

- Support for Board and Operating System Bring-up
- CPU and Board initialization
- Analyze and debug system crashes
- Program Flash devices in-circuit
- Built-in diagnostics for board testing

Hardware supported

**Wind River ICE**
- High Speed Ethernet connectivity
- Wind River JTAGServer™ support for multiple JTAG devices
- Wind River JTAGAccelerator™

**Wind River Probe**
- USB 2.0 Hardware supported
- USB 2.0 (1.x compatible)
- USB Powered, no additional power supplies required
- Support for ‘suspend’ on laptops
- 100MHz JTAG Clock support
ProfileScope
  • Statistical Profiling

MemScope
  • Memory Analysis with Leak Detection

Stethoscope
  • Graphical Data Monitoring Tool that provides visualization of data

TraceScope
  • Code Execution tracing tool for VxWorks 6.0

CoverageScope
  • Complete code coverage information for testing VxWorks 6.0
Support and Training

Support…

• Worldwide technical support team
• Online support
• Access to updates, patches, major upgrades

Training…

Wind Sprint
• On-site installation
• Hands-on orientation
• Project and support advice

Public Workshops
• Hands-on application of tools and concepts
• Real-time design issues
• How to develop drivers, porting, or applications

Custom On-site
• Knowledge transfer to workplace
• Save time and expense away from office

WIND RIVER
<table>
<thead>
<tr>
<th>Feature</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>End-to-End Integration</td>
<td>Set of tools for the entire development lifecycle, from hardware bring-up to product test and manufacture</td>
</tr>
<tr>
<td>Multiple Contexts</td>
<td>Supports multiple OSes, CPUs, architectures, cores, languages and connection types</td>
</tr>
<tr>
<td>Extensible, Scalable Framework</td>
<td>Based on Eclipse, seamlessly integrates 3rd-party and in-house plug-ins for total customization and scalability</td>
</tr>
</tbody>
</table>
Wind River
General Purpose Platform,
VxWorks Edition
Technical Overview
Wind River’s DSO Offering

Wind River Professional Services
Wind River Enterprise Licensing Model
Wind River Partner Ecosystem

Wind River Platform
- Integrated Development Environment
- Industry-Specific Middleware
- VxWorks
- Linux

Wind River enables companies to develop and run device software faster, better, at lower cost & more reliably

WIND RIVER
General Purpose Platform Components

VxWorks 6.0
- VxWorks 5.5 Compatibility
- VxWorks core values
- State-of-the-art memory protection
- Standards based:
  - POSIX compliance
  - IPv4/IPv6
  - Common communication interface

VxWorks RTOS

Workbench IDE
- Eclipse 3.0 Framework
- Project System
- Build System
- Editor
- Source Code Analyzer
- Debugger
- Wind River Compiler & GCC Compiler
- System Viewer
- VxWorks Simulator
- Host Shell
- Kernel Shell
- On-Chip Debugger
- Support for Linux Targets
- Scope Tools (option)

Hardware Support

Services, Support and Training
# General Purpose Platform Components

## Workbench IDE

<table>
<thead>
<tr>
<th>Shared Memory</th>
<th>DOS File System</th>
<th>Networking</th>
</tr>
</thead>
<tbody>
<tr>
<td>Distributed Messaging</td>
<td>Flash File System</td>
<td>IPv4 IPv6</td>
</tr>
<tr>
<td>Message Channels</td>
<td>Error Management</td>
<td>POSIX Library</td>
</tr>
</tbody>
</table>

## VxWorks RTOS

## Hardware Support
- Broad architecture support
- Processor Abstraction Layer
- Comprehensive BSPs and device drivers for reference designs and COTs boards

## Professional Services, Support & Training
- Quick start training
- On-site training
- Worldwide technical support
- Online support
- Access to updates, patches and major upgrades

---

**WIND RIVER**
General Purpose Platform Components

**Memory Management**
- VxFusion

**Message Channels**
- Connection oriented, bi-directional messaging for task-task communication on a single node
- Designed for future multi-processor use

**Networking**
- Commercial-quality IPv4/IPv6 dual stack
  - TCP, UDP, PPP, 802.3 and 802.11 drivers
- Configurable for IPv4 only
- Pre-integrated v6 components:
  - SNMP, Telnet, SNTP, ARP/ND, HTTP Server, Rsh/Rlogin, IGMPv2/MLD, DHCP v4/v6, IPsec, 802.1x, FTP/TFTP, DNS
- Zero Copy buffer libraries
- BootP – For system initialization
- Auto-IP configuration for IPv4
- Standard IPv4 and IPv6 APIs
- RIPv1/v2 and RIPvng
- Transition Mechanisms
- Separation of IPv4 and IPv6 Logical Networks
- IPv6 Downed Interface Processing
- True blocking Sockets calls
- Multithread safe

**Workbench IDE**
- Shared Memory
- DOS File System
- Networking

**VxWorks RTOS**
- Distributed Messaging
- Flash File System
- IPv4 IP

**Hardware Support**
- Message Channels
- Error Management
- POSIX Library

**Services, Support and Training**
General Purpose Platform Components

- **Workbench IDE**
  - Shared Memory
  - DOS File System
  - Networking
  - Distributed Messaging
  - Flash File System
  - IPv4
  - IPv6

- **VxWorks RTOS**
  - Message Channels
  - Error Management

- **Hardware Support**

- **MS-DOS File System**
  - Cache write-through option
  - CheckDisk utilizes "clean bit"
  - Unicode file name support
  - Optional transactional file system layer provides lightweight journaling

- **POSIX Library**
  - Increased POSIX compliance

- **Error Management**
  - Error detection and reporting
    - ISR/Task overrun and under-run
    - Code corruption
    - Null-pointer usage
    - Heap block overrun
    - RTP error
    - Heap leakage
  - API for application errors
  - Extensible

WIND RIVER
## Hardware & Host Support

<table>
<thead>
<tr>
<th>Architectures</th>
<th>Intel Architecture, MIPS and PowerPC</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Architecture Families</strong></td>
<td></td>
</tr>
<tr>
<td>• Pentium 2,3,4</td>
<td></td>
</tr>
<tr>
<td>• MIPS 5Kx, tx49xx, bcm125x</td>
<td></td>
</tr>
<tr>
<td>• Freescale PowerPC 60x, PowerPC 7xx, PowerPC 74xx, PowerPC 82xx, PowerPC 52xx, PowerPC 85xx</td>
<td></td>
</tr>
<tr>
<td>• IBM PowerPC 405, PowerPC 44x</td>
<td></td>
</tr>
<tr>
<td><strong>Hosts</strong></td>
<td></td>
</tr>
<tr>
<td>• Windows 2000 professional</td>
<td></td>
</tr>
<tr>
<td>• Windows XP</td>
<td></td>
</tr>
<tr>
<td>• Solaris 2.8, 2.9</td>
<td></td>
</tr>
<tr>
<td>• Red Hat Enterprise Workstation 3.0</td>
<td></td>
</tr>
</tbody>
</table>

WIND RIVER
VxWorks 6.0 New Features

- Kernel execution environment compatible with 5.5
- Real-time Process environment for user-mode code
- MMU-based memory protection
- Error detection and reporting facility
- Better POSIX compliance, esp. in RTPs
- dosFS improvements
- Transaction-based Reliable File System
- ROMFS Filesystem
- New and enhanced IPC facilities
- Shared library support
- Object ownership and resource reclamation
- Kernel (target) shell enhancements
- New processor/device support
- Improved OS configuration and build facilities
VxWorks 6.0 – Real-time Processes

• Real-time Processes (RTPs) are containers for user-mode applications.
• Each RTP has own copies of code, data, stacks, heap and resources.
• RTPs are not scheduled – tasks within RTPs are.
• RTPs are launched from a fully-linked relocatable executable loaded from a file system (a la UNIX).
VxWorks 6.0 – Kernel Enhancements

- Object Management (private and public scope)
- ISR Objects
- Optimized mutex semaphore for processes
- Task preemption prevention in processes (taskRtpLock/Unlock)
- Priority-inheritance enhancement
- Configurable kernel work queue size
VxWorks 6.0 Memory Management

- Non-executable stack pages (certain CPUs.)
- Stack overrun and underrun detection.
- NULL pointer dereference detection (certain CPUs.)
- Text segment write protection
- Heap and partition manager instrumentation (run-time checking)
- Kernel heap allocator improvements (best-fit vs. first-fit)
- User-space heap and partition allocators
VxWorks 6.0 – ED&R

- New error detection and reporting facility
- Persistent error logs
- Configurable behaviors on a task, process, and system level
- Exception handlers instrumented to log information
VxWorks 6.0 – File System

- Improved dosFs file system
- Safest-order writes of metadata and user data to minimize chances of corruption
- Optional cache write-through
- Support for O_SYNC flag on open() operations
- FIOSYNC ioctl fullyflushes block device caches
- Support for Unicode file names
- Enhanced CHKDSK (FAT recovery, performance, etc.)
- TRFS lightweight journaling filesystem
- ROMFS read-only filesystem

WIND RIVER
• C interpreter enhancements for process information
• Handles long long, short, float, signed, unsigned types
• Path completion (if filesystem supports it)
• New UNIX shell like interpreter
• Allows custom interpreters
• Multiple shell support
• Secure access (user id/password protection)
• Fault management support (ED&R)
• VI or EMACS style command line editing
• C++ symbol handling enhancements

WIND RIVER
Migrating from Tornado 2.2.x to GPP 3.0
Migrating from 2.2.x to 3.0

- VxWorks 6.0 is source code compatible with previous version, but:
  - Some deprecated features have been dropped
  - New tool versions
  - Performance (code size/speed) changes
Migrating from 2.2.x to 3.0

• **Obsoleted Features**
  - RT11 Filesystem (unsupported in 5.5)
  - VxVMI (made redundant)
  - BOOTP (still supported for boot loader)
  - BSD 4.4 Ethernet drivers
    - Also, other BSD driver support routines
    - etherhooks
    - if_sm replaced with smEnd
• **GNU 3.3.2 (was 2.96+)**
  • -O3 still unsupported. Code that previously worked with this option may now break.

• **Wind River Compiler (formerly Diab) 5.2.2**

• **For C code:**
  • Kernel applications use native VxWorks C library
  • RTP applications use Dinkum C library

• **For C++ code:**
  • Both use Dinkum C++ / Embedded C++ libraries
  • SGI STL replaced by Dinkum STL
Migrating from 2.2.x to 3.0 - Size

• **VxWorks 6.0 build is bigger than 5.5**
  • Around 15-20%
  • Additional features within existing APIs
  • Network stack is bigger

• **Shouldn’t be an issue for most applications**
  • Review configuration; remove unneeded components
  • Future releases will have some improvements
Migrating from 2.2.x to 3.0 - Performance

• Goal: “Maintain existing 5.5 characteristics”
  • Overarching 6.0 goals: backward compatibility and memory protection

• Memory allocation made deterministic
  • Achieves deterministic allocation & less fragmented heap
    • Free blocks in AVL tree vs. linked list (best-fit algorithm)
    • Slower (20%) than 5.5 in low fragmentation state

• OS primitives have performance degradation compared to 5.5
  • OS primitives generally slower (a few are faster)
  • Creation/deletion routines generally much slower (see malloc)
  • User-mode system call adds overhead
  • This refers to OS primitives NOT networking, C library calls, etc!!!

• Priority for enhancement in future releases
  • Areas for improvement already identified
  • Investigating multiple component implementations: size vs speed
Thanks!