General Description

WinDriver is a driver development toolkit that simplifies the creation of monolithic device drivers. WinDriver includes a graphical development environment, APIs, diagnostic and debug utilities and samples, which enable you to quickly develop a high performance driver, without being a “driver guru”.

Product Features

- **Immediate hardware access**: Access hardware through a graphical user mode application, without having to write a line of code
- **Generation of hardware specific code**: The WinDriver DriverWizard generates your skeletal driver code, customized to your hardware.
- **Easy addition of functionality and logics to the generated skeletal code**: In user mode and within your favorite development environment
- **Performance optimization**: Kernel PlugIn enables transfer of user mode code to the kernel level, thereby achieving optimal performance
- **Graphical tools**: DriverWizard, an intuitive user mode application, simplifies hardware access and driver code generation *
- **Debugging**: Graphical Debug Monitor to monitor kernel level activity
- **Multi operating system support**: Supports Windows 98/Me, Windows NT/2000/XP/Server 2003 and Windows CE, Linux, Solaris and VxWorks
- **Cross operating system compatibility**: The developed driver is source compatible between all supported operating systems without any code modifications
- **Code generation for common development environments**: Visual C++, Visual Basic, Delphi, gcc and others.
- **Hardware independent**: Supports any PCI / CompactPCI / CardBus/ ISA / EISA based hardware
- **64-bit data transfer support**: On x86 computers running 32-bit Windows with 64-bit PCI bus

Product Benefits

**Shorter development cycle; easier learning curve; faster time to market**

- Isolates hardware bugs from the driver development process
- No kernel mode programming
- Achieve kernel mode (Ring 0) performance, by using the Kernel PlugIn feature
- No need to master the operating system internals, kernel development tools like the Microsoft DDK and the Solaris DDI, or the bus protocols
- Write and maintain one driver code base for all supported operating systems
- Includes diagnostics files and samples for jump-starting your development, including specific support for leading chipset vendors: PLX, QuickLogic, Altera, PLDA, Galileo, AMCC
- Free, full featured, 30-day evaluation version: The code created with the evaluation version will be ready for commercial distribution upon purchase of a registered product
- Free, expert technical support, for the duration of the evaluation period

WinDriver is also available for USB based hardware. Refer to www.jungo.com for further details.
Also from Jungo, KernelDriver, for standard device drivers.

* VxWorks and WindowsCE license enables using the Driver Wizard on a Windows 2K/XP/2003 PC to generate the code.
**Technical Specifications**

- Detection of Plug-'n-Play devices installed on a host, including devices residing beyond a PCI-to-PCI bridge
- Provision of data for each detected device: Location; Vendor ID and Device ID; Resources, including interrupts, IO and memory
- Hardware verification and debugging via an intuitive wizard:
  - Read / Write to the PCI Configuration space, memory buffers, IO ranges
  - Listen to interrupts
  - Define and access new registers
- Windows WDM compliant; Supports Plug-and-Play and power management notification handling
- Generation and installation of .inf / .kdf files
- Contiguous and Scatter Gather DMA under Windows, Linux and VxWorks operating systems
- Throughput of up to 100,000 interrupts per second (using Kernel PlugIn)
- Dynamic driver loading
- Supports multiple PCI-bus platforms

Jungo Software Technologies Inc., 3031 Tisch Way, Suite #1007, San Jose CA, 95128 U.S.A
Tel: 1-877-514-0537, Fax 1-877-514-0538 (USA)   Tel: +972-9-8859365, Fax +972-9-8859366 (Int'l)
Web: http://www.jungo.com/   E-mail: marketing@jungo.com
The device driver developed with WinDriver (yourapp.exe/.dll) accesses hardware through the WinDriver kernel module (windrvr.sys/.vxd) using the standard WinDriver API functions.

Performance critical sections (time critical data transfers) are handled in the Kernel PlugIn and are executed in the kernel mode, thereby eliminating overhead.

This architecture enables development and debugging of driver code in the user mode, using the WinDriver functions, and migration of only the performance critical sections of the code to the kernel mode, using the simple Kernel PlugIn mechanism.