

# WinDriver

## QUICK START GUIDE

*A 5-Minute introduction to writing USB device drivers.*



### Who should use WinDriver?

1. Hardware developers – Use the Driver Wizard to quickly test your new USB hardware.
2. Software developers – Use the Driver Wizard to generate the device driver code to drive your USB based hardware. Later, use the WinDriver tools to test and debug your driver.

### Which operating systems does WinDriver USB support?

WinDriver currently supports Windows 98 and Windows 2000. Check the Jungo web site for updates on new operating systems support for the USB bus.

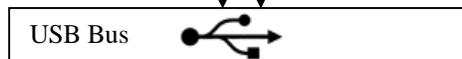
### Where can I get more in-depth information?

1. For more in-depth information about USB, WinDriver USB and a free 30 day evaluation of WinDriver USB, please see the Jungo web site at <http://www.jungo.com/>
2. The complete WinDriver manual can be downloaded at <http://www.jungo.com/manuals.html>

## 6 steps to building your driver:

### 1 Set Up:

- Attach your USB device to the PC.
- Install WinDriver USB.

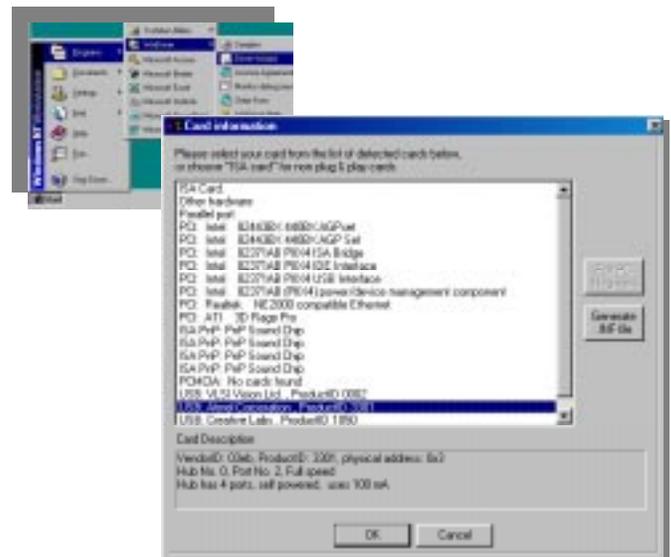


### 2 Select Your Hardware:

- Start the “Driver Wizard” by choosing WinDriver | Driver Wizard from the Start Menu.

The Driver Wizard will show all of the plug and play cards in your machine.

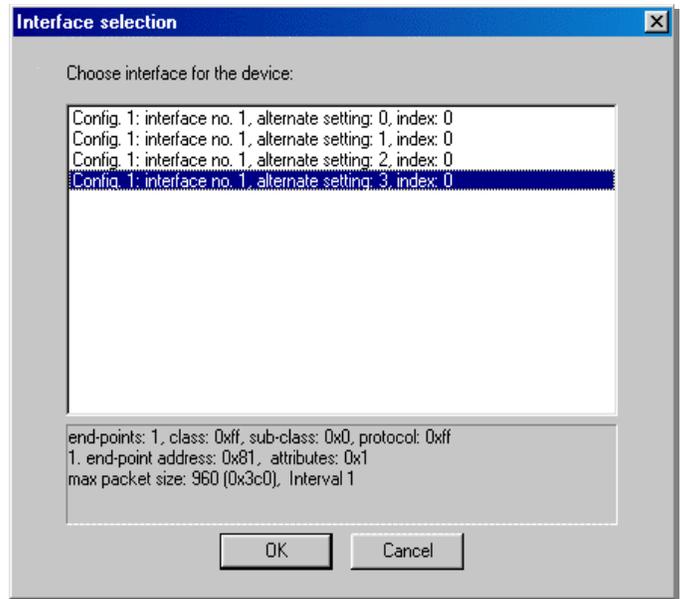
- Choose your USB device from the list.
- In some cases the Wizard will notify you about the need to generate an .INF file in order to work with your hardware. The Wizard automates this process for you. (See the Q&A section at the end of this document).



# 3

## Detect / Define your Hardware:

- Driver Wizard will automatically detect your USB device's interfaces, (including configurations, interfaces and alternate settings).
- Choose the desired interface. (For USB devices with only one interface configured, the wizard automatically selects the detected interface and the 'interface selection' screen will not be displayed).

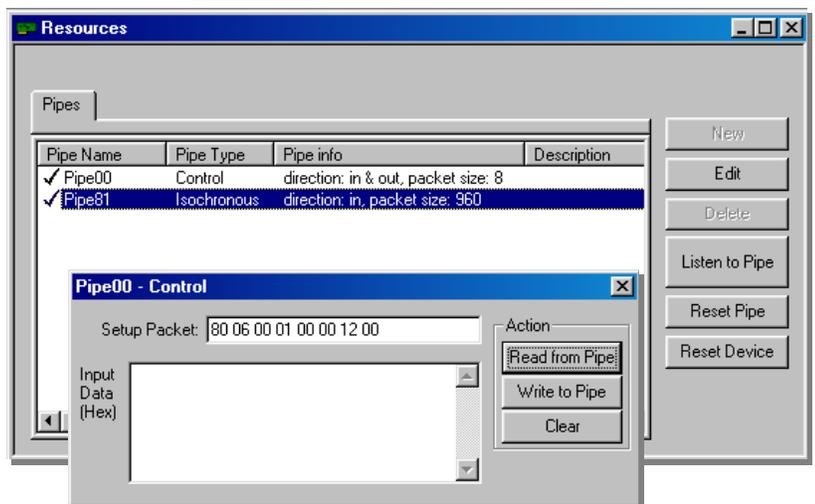


# 4

## Test your hardware:

Before writing your device driver, it is important to make sure your hardware really works. Use the Wizard to diagnose your hardware:

- Read and write to the control pipe.
- Read, write and "Listen" to your device's data pipes.

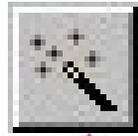


# 5

## Generate the Driver Code:

Use the Driver Wizard to generate your device driver in C/C++ or Delphi. The following code is generated:

- API for accessing your hardware from the application level (and from the kernel).
- A sample application that uses the above API to access your hardware.
- Project make-files for all of the supported operating systems and environments.



```

DWORD CAMERA_ReadPipe81(CAMERA_HANDLE hCAMERA, PVOID
                        pBuffer, DWORD dwSize)
{
    WD_USB_TRANSFER transfer;

    BZERO(transfer); transfer.dwPipe = 0x81;
    transfer.dwBytes = dwSize; transfer.fRead = TRUE;
    transfer.pBuffer = pBuffer; transfer.hDevice = hCAMERA->hDevice;
    WD_UsbTransfer(hCAMERA->hWD, &transfer);

    if (transfer.fOK)
        return transfer.dwBytesTransferred;
    return -1;
}

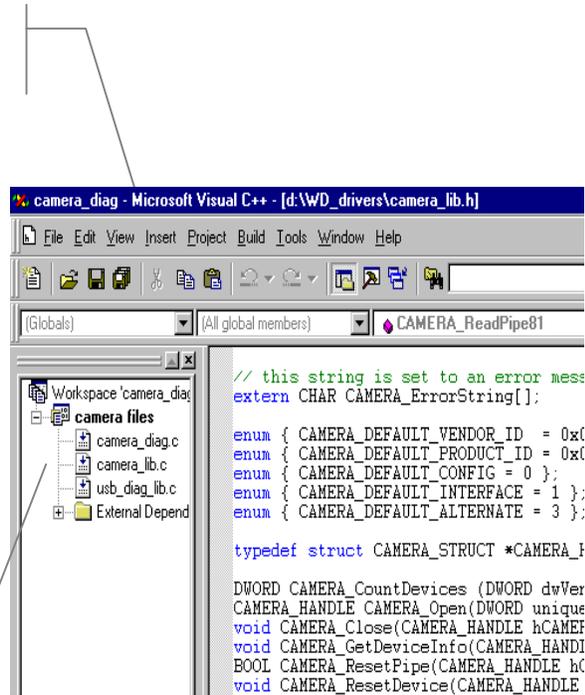
```

# 6

## Compile and Run:

- Compile the sample diagnostics application, and run it! This sample is a robust skeleton for your final driver.
- Modify the generated sample application to suit your application needs.

Your Driver Project Name



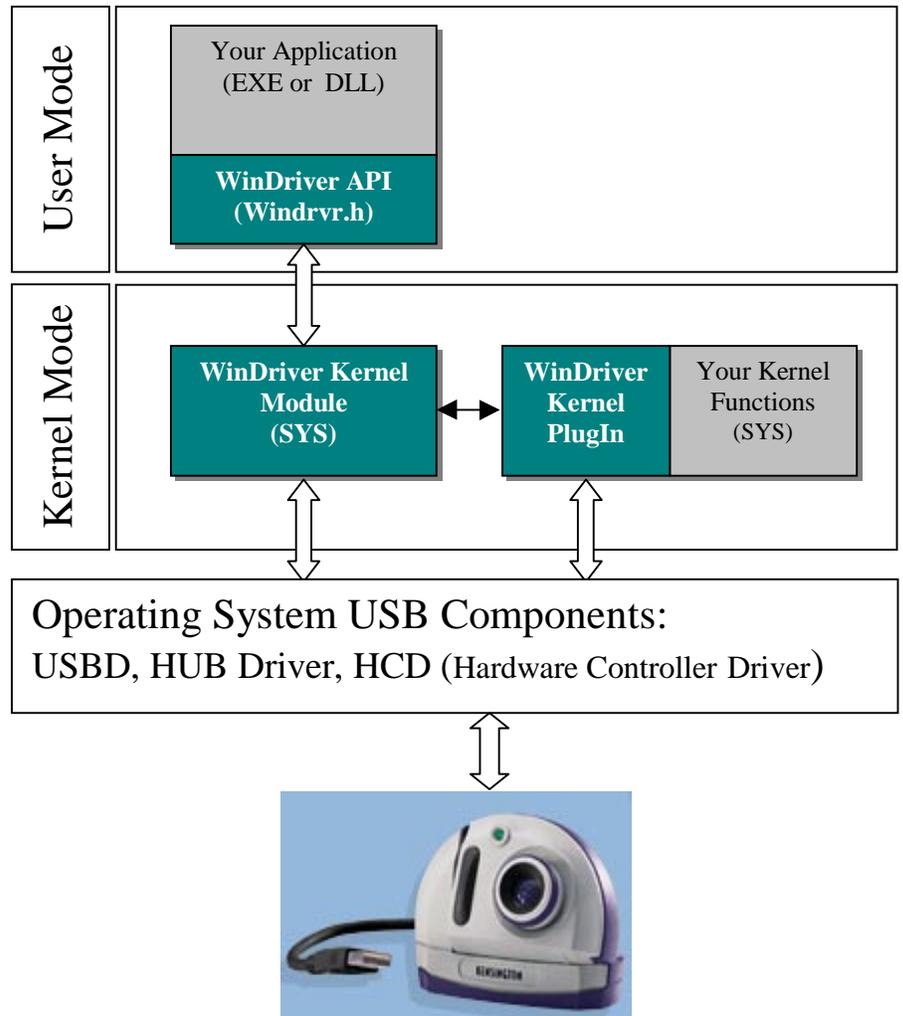
Your hardware access API, and a sample application that uses this API

Intuitive API. Call these functions directly from within your application! Start with XXX\_Open(), use the Read/Write functions, etc., and finish with XXX\_Close().

## Q: How does WinDriver work?

**A:** With WinDriver, your device driver is developed in the User Mode (as part of your application or as a separate DLL). This dramatically shortens development time by enabling you to use your standard Win32 tools (MSDEV, Borland, Delphi, etc.) to develop and debug your driver.

The device driver produced with WinDriver (YourApp.EXE) accesses your hardware through the WinDriver kernel module (windrvr.SYS and wdusb.sys) using the standard WinDriver functions.



## Q: How can I achieve optimal performance with WinDriver?

**A:** After your driver is complete, you may easily transfer the performance critical parts of your driver code to WinDriver's "Kernel PlugIn", which runs those sections at Kernel level, thereby achieving optimal performance. For example – write your data transfer functions in the user mode - debug it with your application debugger, and later move the code into the Kernel PlugIn. This will enable your data transfer functions to be executed in the kernel level, thereby allowing them to operate at maximal performance.

This architecture enables you to develop and debug all of your driver code in the user mode, using the WinDriver functions, and to migrate only the performance critical sections of the code to the User Mode, via the simple Kernel PlugIn mechanism - without any changes in the code.

## Q: What is an .INF file?

**A:** Device information (INF) files are text files, that provide information used by the "Plug and Play" mechanism in Windows 95/98/2000 to install software that supports a given hardware device. INF files are required for "Plug and Play" hardware, such as USB and PCI based devices. The INF file includes all necessary information about the device(s) and the files to be installed. When hardware manufactures introduce new products, they must create INF files to explicitly define the resources and files required for each class of device.

In some cases .INF files supplied with the operating system will suit your device. In other cases, you will need to create an .INF file for your device. The Driver wizard can generate an INF specific for your Card/device. The INF is used to tell the OS that the selected device is now handled by your driver (WinDriver).

## Q: Why should I create an INF file?

**A:**

1. To stop the 'new hardware wizard' of the Windows operating system from popping up after boot.
2. In some cases, the OS doesn't assign physical addresses to USB devices without an INF file. In these cases you will not be able to diagnose your USB device with the DriverWizard until creating the INF file.
3. To load the new driver created for the card/device. Creating an INF file is required whenever developing a new driver for the hardware.
4. To replace the existing driver with a new one.

## Q: How Do I Create and Install the .INF file?

**A:** Use the DriverWizard to generate your specific device's .INF file. Click the 'Generate .INF file' button on the 'Card information' screen (see picture in Step 2 above). The INF file includes your device VID/PID and loads WDUSB.SYS as your device driver.

- When no driver exists for your device, use the operating system 'Add new hardware' wizard to add and register the .INF file created with WinDriver. In the relevant screen enter the path of the new .INF file created with WinDriver.
- When replacing an existing driver, follow the instruction displayed by the Wizard after generating your .INF file.

## Contacting Jungo

Phone: (USA) 1-877-514-0537 (WorldWide) +972-9-8870878  
Fax: (USA) 1-877-514-0538 (WorldWide) +972-9-8870877  
Email: [info@jungo.com](mailto:info@jungo.com) Web: <http://www.jungo.com>

