Go-HotSwap™ Quick Start Guide

A 5-Minute introduction to using Go-HotSwap

The following pages will guide you through the process of installing Go-HotSwap and creating your hot swap aware device driver.

Go-HotSwap is a software tools suite that adds the necessary software modules required to enable CompactPCI hot swap, and provides the tools and development environment to develop hot swap aware drivers.

Go-HotSwap supports Windows 2000/X/Server 2003, Linux 2.4/2.6 and Solaris 8.0/9.0. Check the Go-HotSwap web page (`http://www.jungo.com/hotswap.html`) for updates on new operating systems support.

Go-HotSwap includes Jungo's WinDriver driver development tool, designed to dramatically reduce the time spent on device driver development for peripheral hardware connected through the PCI/Compact PCI/PCI Express/ISA and USB buses (`http://www.jungo.com/windriver.html`). WinDriver features a unique cross platform technology: the same hot swap aware driver written with WinDriver, will run on all supported operating systems. Just compile and run!

Four Steps to Developing your Hot Swap Driver:

1. Set Up

(1) Verify that your card is connected to the machine.

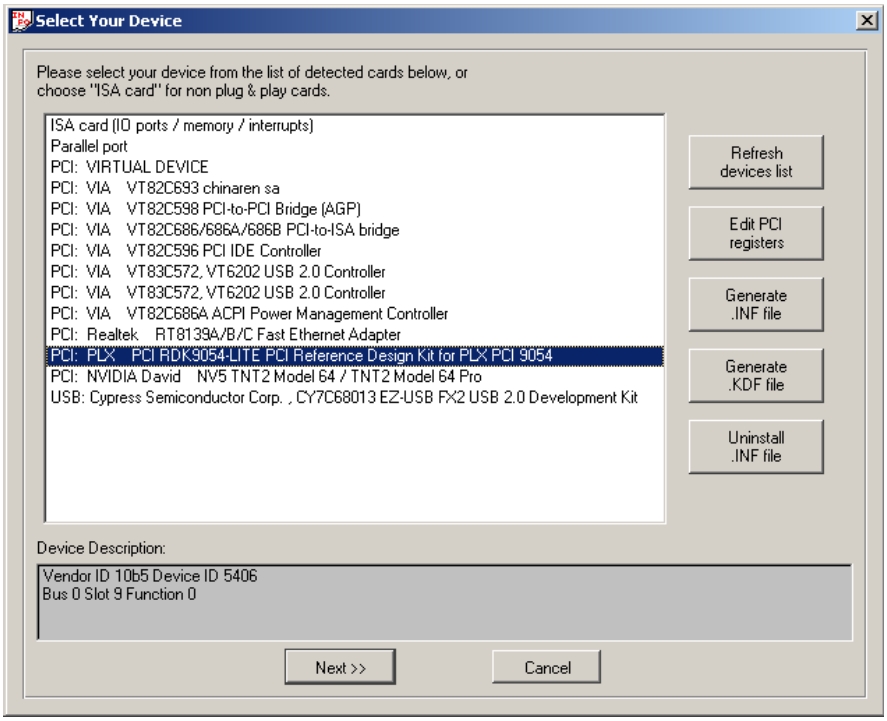
(2) Install Go-HotSwap.

2. Test Your Hardware and Create a Driver

Test and debug your hardware and generate driver code for your specific device with DriverWizard:

(1) Run DriverWizard and select your card:

- Start DriverWizard from the **Start** menu – **Start | Programs | GoHotSwap | DriverWizard** (on Windows) or by running **GoHotSwap/wizard/wdwizard**.
- In the dialogue box that appears, choose **Create New Host Driver Project**.
- Choose your card from the list of Plug-and-Play devices displayed in the wizard:



(2) On Windows: Generate an INF file for your card:

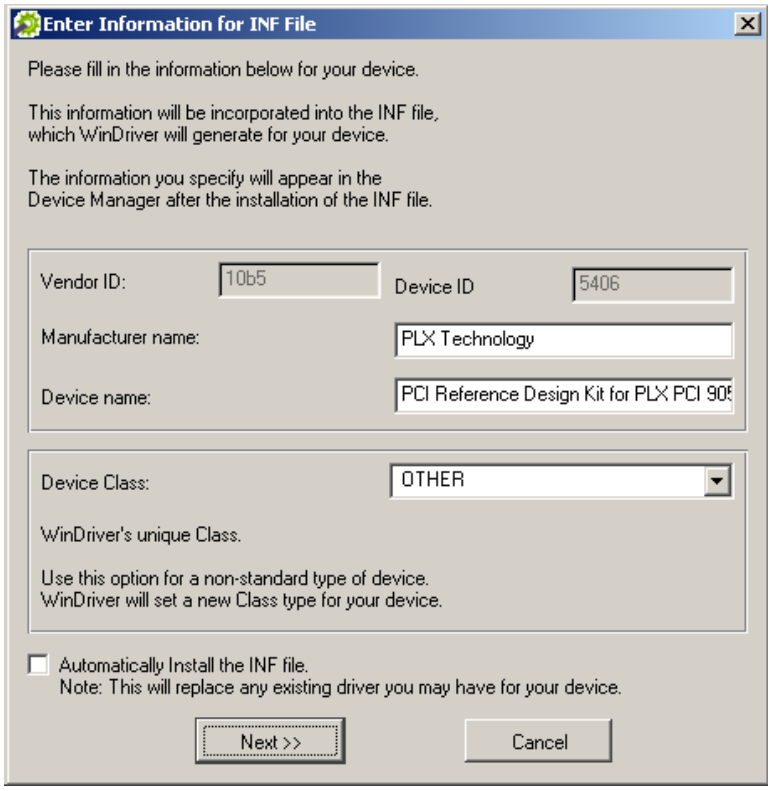
When developing a driver for a Plug-and-Play device (CompactPCI/PCI/PCMCIA/CardBus) on Plug-and-Play Windows operating systems, such as Windows 2000/XP/Server 2003, in order to correctly detect the device's resources and communicate with the device using Go-HotSwap/WinDriver, you need to create an INF file that registers your device to work with the Go-HotSwap/WinDriver kernel engine. DriverWizard automates the INF generation and installation process. To generate an INF file with DriverWizard, follow these steps:

i. Click the **Generate .INF file** button in the wizard's **Select Your Device** dialogue.

ii. DriverWizard will display information detected for your device – vendor ID, device ID, device class, manufacturer name and device name – and allow you to modify the manufacturer and device names and the device class information.

iii. You can choose to automatically install the INF file from the DriverWizard by checking the **Automatically Install the INF file** option in the wizard's INF generation dialogue. If the automatic wizard INF file installation fails, DriverWizard will notify you and provide manual installation instructions. If you select not to automatically install the INF file from the wizard you will also need to install the file manually, either using the **GoHotSwap\util\wdreg.exe** utility, or using Windows' **Add New Hardware Wizard** or **Upgrade Device Driver Wizard**, as explained in the WinDriver documentation.

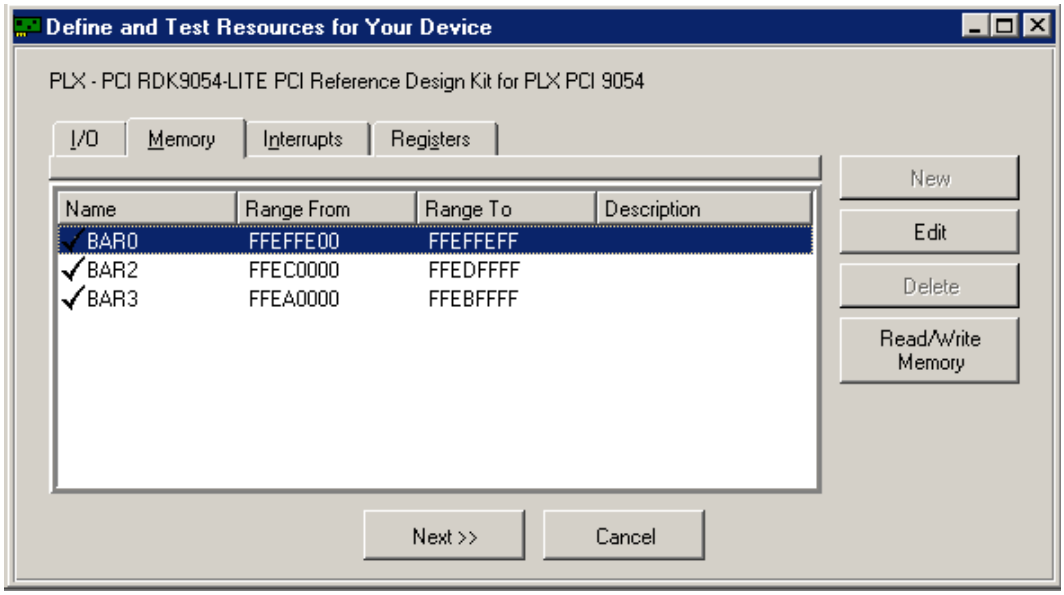
iv. Click **Next** in the INF generation dialogue in order to generate the INF file and install it (if selected).

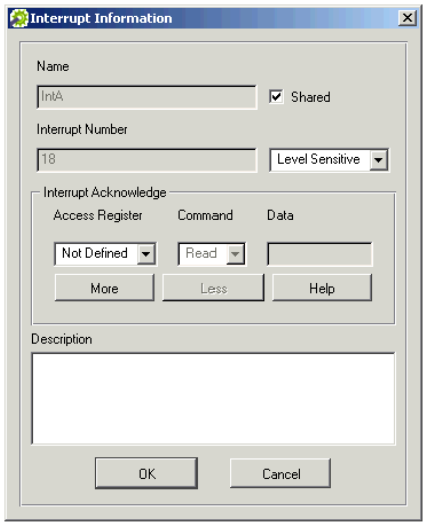


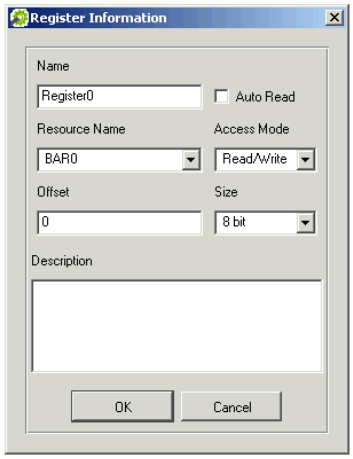
v. When the INF file installation completes, select and open your device from the list described in step 2.1 above.

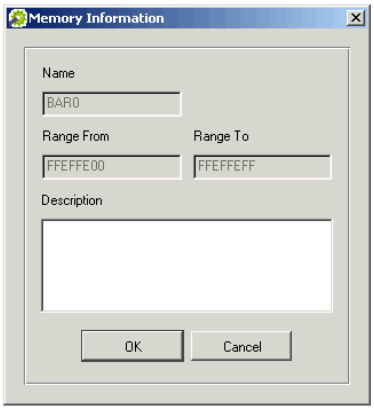
(3) Detect / define your hardware's resources:

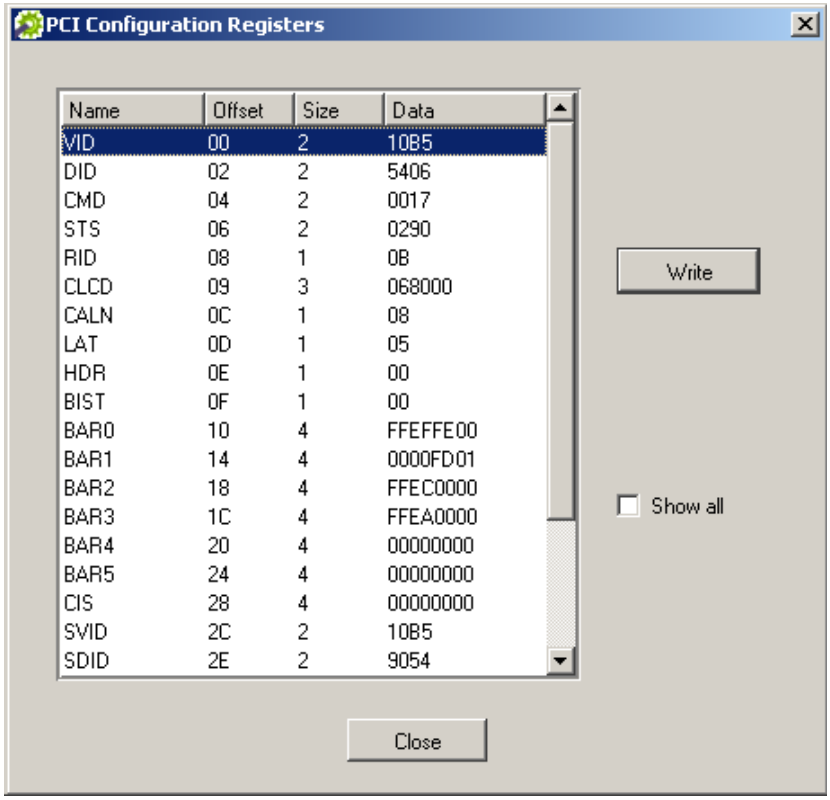
i. DriverWizard will automatically detect your Plug-and-Play hardware's resources (I/O ranges, Memory ranges, PCI configuration registers and Interrupts). You can define additional information yourself, such as defining registers for your device as well as assigning read/write commands for these registers to the interrupt. (For non-Plug-and-Play hardware (ISA) – define your hardware's resources manually).


Define and Test Resources for Your Device
PLX - PCI RDK9054-LITE PCI Reference Design Kit for PLX PCI 9054
I/O
Memory
Interrupts
Registers
Name
Range From
Range To
Description
BAR0
FFEFFE00
FFEFFEFF
BAR2
FFEC0000
FFEDFFFF
BAR3
FFEA0000
FFEBFFFF
New
Edit
Delete
Read/Write Memory
Next >>
Cancel


Interrupt Information
Name
IntA
Shared
Interrupt Number
18
Level Sensitive
Interrupt Acknowledge
Access Register
Command
Data
Not Defined
Read
More
Less
Help
Description
OK
Cancel


Register Information
Name
Register0
Auto Read
Resource Name
Access Mode
BAR0
Read/Write
Offset
Size
0
8 bit
Description
OK
Cancel

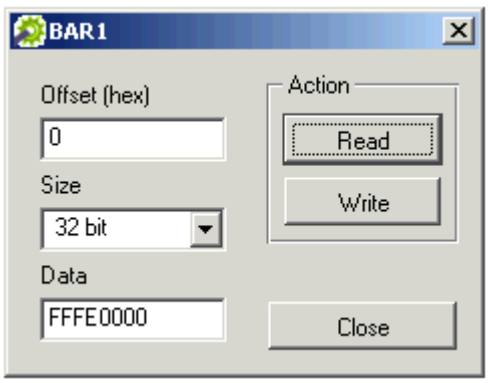

Memory Information
Name
BAR0
Range From
Range To
FFEFFE00
FFEFFEFF
Description
OK
Cancel

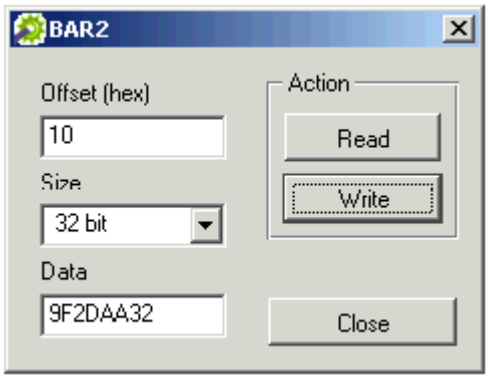


(4) **Test your hardware:**

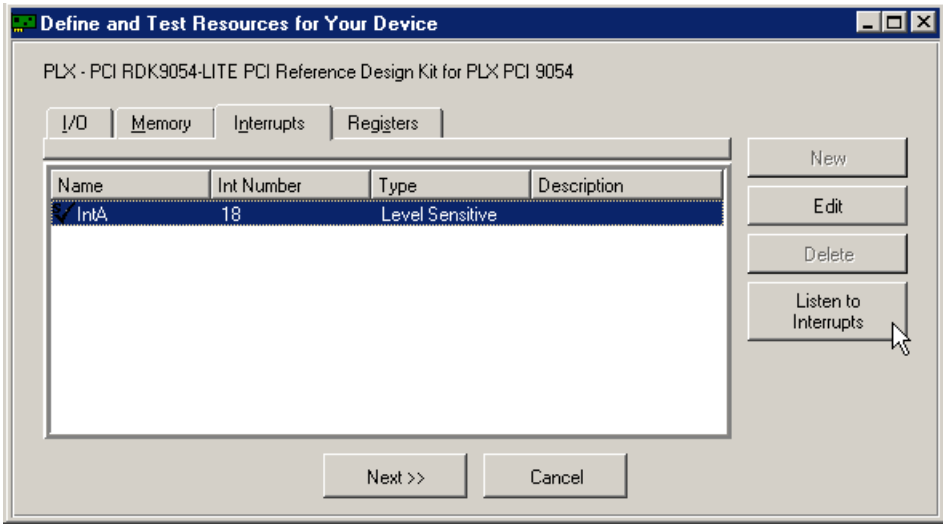
Use DriverWizard to diagnose your hardware and verify that it is working properly:

- Read and write to the I/O ports, memory space and your defined registers.

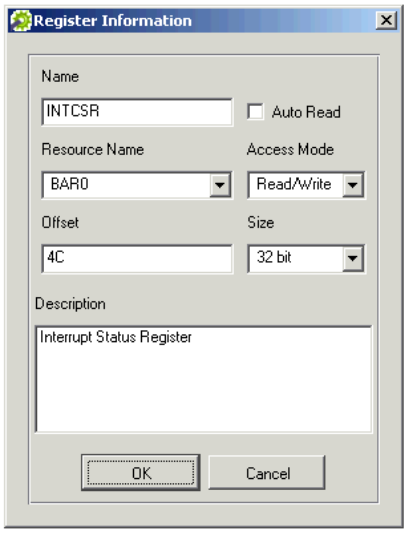




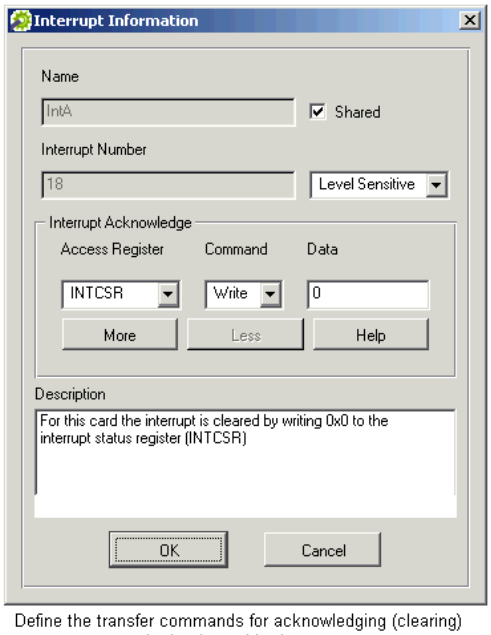
- "Listen" to your hardware's interrupts.



NOTE: For level sensitive interrupts, such as PCI/CompactPCI interrupts, you must use DriverWizard to define the interrupt status register and assign the read/write command(s) for acknowledging (clearing) the interrupt, before attempting to listen to the interrupts with the wizard, otherwise the OS may hang! The specific interrupt-acknowledgment information is hardware-specific.



Define the interrupt status register

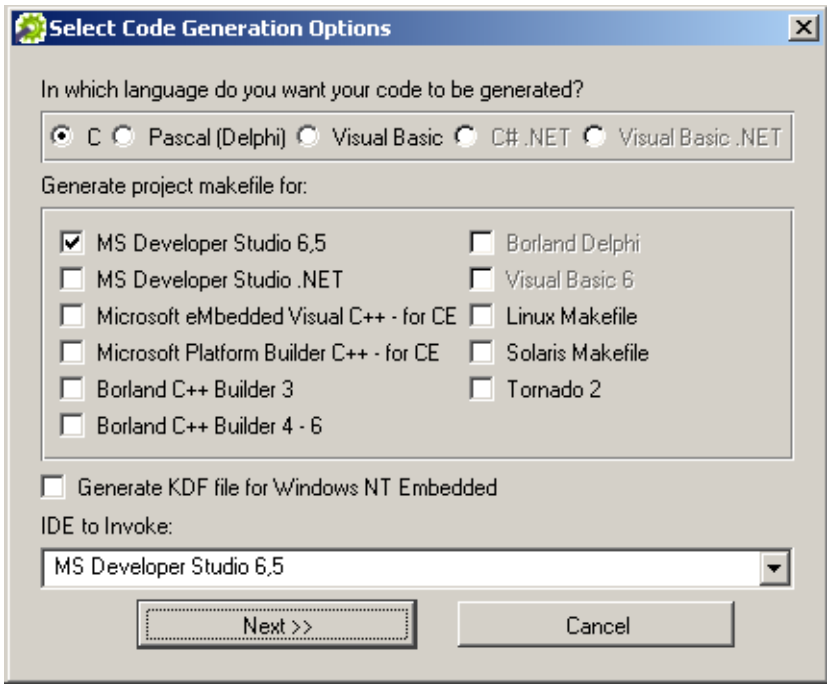


Define the transfer commands for acknowledging (clearing) the level sensitive interrupt

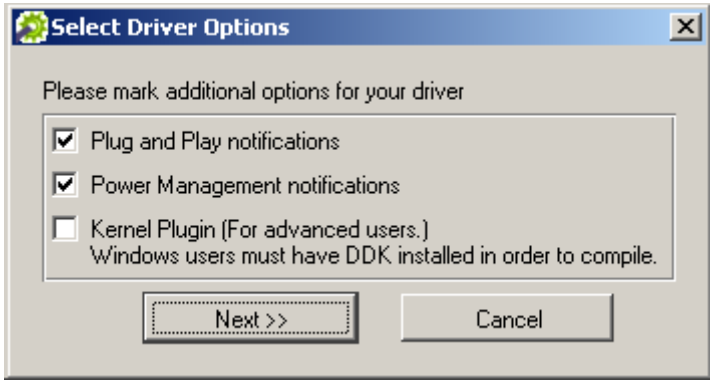
(5) Generate the Driver Code:

i. Use DriverWizard to generate your skeletal device driver. Click on the **Next** button or select the **Generate Code** option from the **Build** menu.

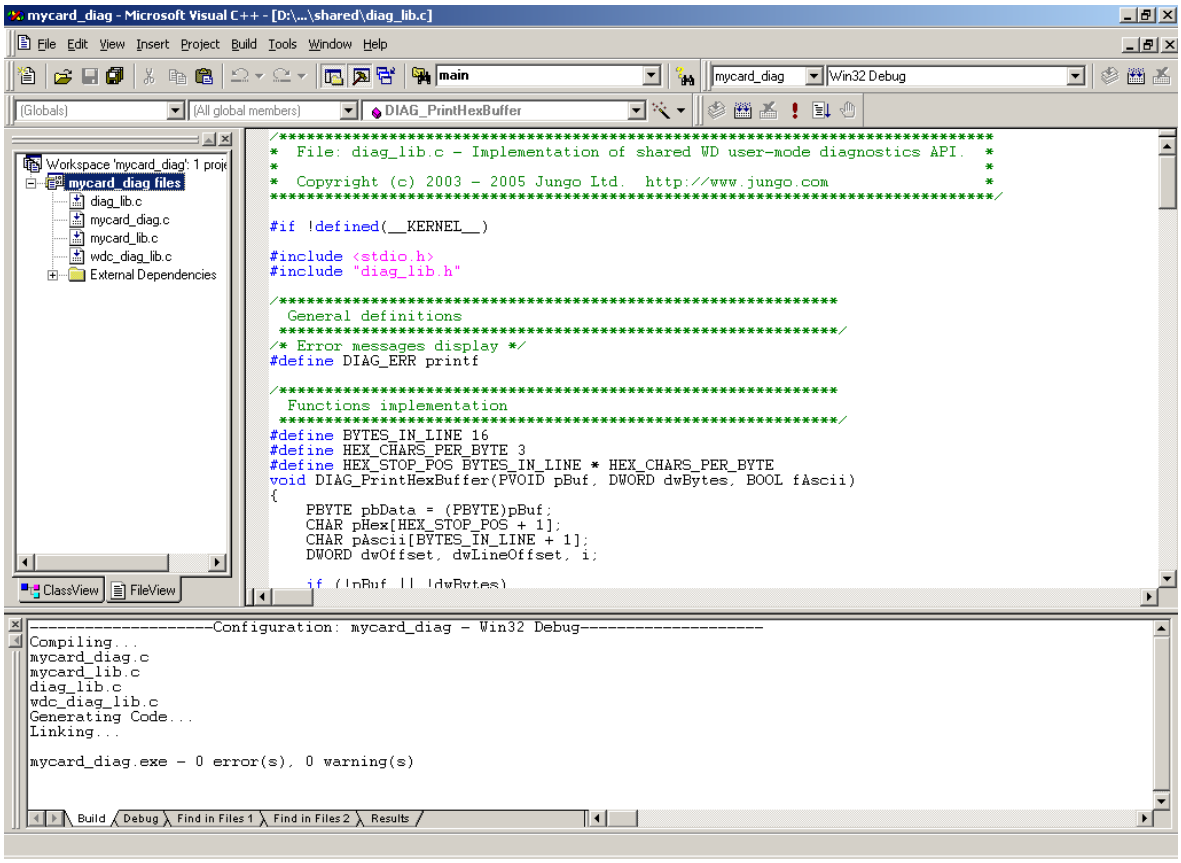
ii. Choose the code language and indicate which development environments you would like to have project/make files for:



iii. Indicate whether you wish to handle Plug-and-Play and power management events from within your driver code and whether you wish to generate Kernel PlugIn code. (Note: In order to build a Kernel PlugIn driver on Windows you must have an appropriate Microsoft DDK installed):



iv. Click **Next**. DriverWizard will launch the desired development environment you chose in step 2(5)ii above.



3. Compile and Run Your Driver

- 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 selected build environments.
  - An INF file for your device (for Plug-and-Play hardware on Windows).
- Use the project/make file that DriverWizard generated with your selected compiler [2(5)iv].
- Compile the application, and run it! The generated code is a robust skeletal code for your final driver.

- Modify the sample application to suit your specific development needs, or start from one of the many samples provided with Go-HotSwap. To add hot swap capabilities to your driver, refer to step 4 below.

4. Add Hot-Swap Capabilities to your Driver

To add hot swap capabilities to your driver/application, follow these steps:

(1) Activate the Hot Swap Engine by calling the function `WD_WatchPciStart()` from your application. This allows the application to receive Hot Swap services.

(2) Register your application to receive notifications about hot swap events, according to a user defined set of criteria.

(3) When a registered how swap even occurs, retrieve information about the event and handle the event as needed.

For detailed information on how to perform these tasks and for a full API reference, refer to the Go-HotSwap and WinDriver PCI user's manuals (**hsman** and **wdman**, respectively). The manuals (in different formats) are found under your **GoHotSwap/docs/** directory and can also be viewed on-line from the Go-HotSwap web page.

You can also refer to the **GoHotswap/hotswap/hs_detect/hs_detect.c** or **GoHotswap/hotswap/hs_activate/hs_activate.c** samples for a demonstration of how to use the APIs.

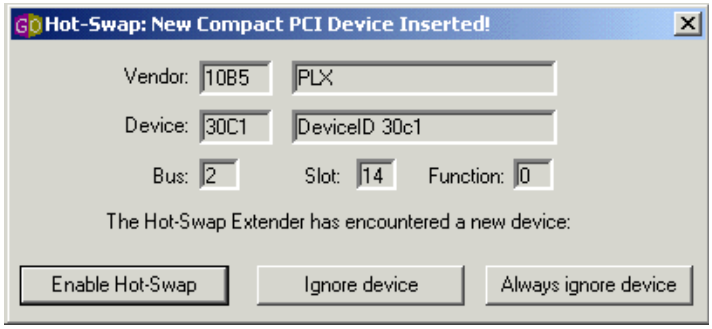
Go-HotSwap Configuration Manager

Go-HotSwap enables you not only to develop hot-swappable drivers but also to configure the Configuration Manager in order to execute tasks upon insertion or removal of devices (start/stop services or run executable and batch files). This utility enables you, amongst other things, to implement hot swap with legacy PCI device drivers, by starting/stopping device drivers.

To activate the GUI version of the Go-HotSwap Configuration Manager (**hs_extender**) on Windows, double-click the Go-HotSwap icon in the Windows task bar or select: **Start | Programs | GoHotSwap | HotSwap Configuration Manager**.

To activate the console-mode version of the Configuration Manager, run the **hs_activate** application from the **GoHotSwap/hotswap/hs_activate/** directory (source code included).

You can now define what actions to perform on each insertion/removal of a device.



When a device is hot swapped, the Configuration Manager will perform the actions pre-defined by the user and saved in its database. If a new type of card is inserted for the first time, the Configuration Manager will prompt upon insertion and wait for the user to define new actions to be carried out upon future insertion/removal.

Legacy Drivers

In order to use Go-HotSwap with legacy drivers, you must first install the vendor-supplied legacy driver INF file. Then activate the Go-HotSwap Configuration Manager, which starts the Hot Swap Engine. Once the legacy INF file is installed and the Hot Swap Engine is activated, the Windows Plug-and-Play Manager will notify the legacy driver of Plug-and-Play events.