Catalyst
Enterprises, Inc.
Setting new Standards

Using PCIUW from a C++ Code

The PCIUW utility may be executed from within a C program code. The functions below describe how to read, save and write back to the registers. These functions are for Microsoft C++ compilers. The PCIUW.EXE and necessary DLLs must reside in the directory where this program is executed from.
The necessary steps are:

1- Read all devices
2- Search for the desired device in the CFG file
3- Save the information for the device to a file, by specifying the bus number and the device number in the save command
4- Write from the file to reload the hardware

The format of the CFG file is described below to allow locating the desired device(s).

A- Description of function calls to read, save and reload devices

```
 void CTestDlg::OnButtonRun()
     {
        /*run the utility to read all devices */
        int ReturnCode = 0 ;
        ReturnCode=WinExec("pciuw",SW_SHOW);
        /*run the utility to show help dialog */
        // ReturnCode=WinExec("pciuw /?",SW_SHOW);
        /*run the utility to save the device configuration information of the device on BusNo = 1 and
         device No = 0 */
        // ReturnCode=WinExec("pciuw /s 1 0",SW_SHOW);
        /* Bus number=1 and Device = 0 here are examples, to find the specific of what your device
        number is search the CFG file to located your device, using the cfg file description below, after
         you run the utility to read all devices. The device related information may also be located from a
         command line outside of the c program and then used in the code, so long as the system
        configuration does not change. */
        /*run the utility to load configuration information from the last saved file into the hardware */

        // ReturnCode=WinExec("pciuw / w",SW_SHOW);
        if(ReturnCode < 31)
        AfxMessageBox ("run command failed.");
      }
```

B- CFG file description

PCIUW program when executed saves its configuration information in “set@@@.cfg” file. This document describes the format of this file.

Catalyst
Enterprises, Inc.
Setting new Standards

FILE FORMAT

Order	Length	Byte(s)	
1	File type label	20	"PCIUW settig File"
2	Header_data	64 * 452	
3	ActiveDevice	4	Int
4	ActivePointer	4	
5	Counter	4	
6	MaxCounter	4	
7	ActiveSelect	4	
8	ActiveFunction	64 * 4	
9	ActiveBus	64 * 4	
Sum		29480	

DETAILED LAYOUT OF HEADER DATA

Type	Name	Len	Start Address
Int	Dev_func	4	0
Int	BusNo	4	4
Int	MultiFunction	4	8
Char	Dummy	1	12
	Alignment Bits	**3**	**13**
Unsigned int	Device_Id	4	16
Unsigned int	Vendor_Id	4	20
Unsigned int	Command	4	24
Unsigned int	Status	4	28
Char	Revision_Id	1	32
Char	Programming_interface	1	33
Char	Sub_class	1	34
Char	Base_class	1	35
Char	Cache_line	1	36
Char	Latency_timer	1	37
Char	Header_type	1	38
Char	BIST	1	39
Unsigned long	Base_Address1	4	40
Unsigned long	Base_Address2	4	44
Unsigned long	Base_Address3	4	48
Unsigned long	Base_Address4	4	52
Unsigned long	Base_Address5	4	56
Unsigned long	Base_Address6	4	60
Unsigned long	CardBus	4	64
Unsigned int	SubsystemVendorID	4	68
Unsigned int	SubsystemID	4	72
Unsigned long	Expansion_ROM	4	76
Unsigned long	Reserve3	4	80
Unsigned long	Reserve4	4	84
Char	Interrupt_line	1	88
Char	Interrupt_pin	1	89
Char	Min_Gnt	1	90
Char	Max_Lat	1	91

Catalyst
Enterprises, Inc.
Setting new Standards

__int8	PrimaryBusNo	1	92
__int8	SecondaryBusNo	1	93
__int8	SubordinateBusNo	1	94
__int8	SecondaryLatTimer	1	95
__int8	IOBase	1	96
__int8	IOLimit	1	97
__int16	MEMBase	2	98
__int16	MEMLimit	2	100
__int16	PreMEMBase	2	102
__int16	PreMEMLimit	2	104
__int16	IOBaseUp16	2	106
__int16	IOLimitUp16	2	108
__int16	SecStatus	2	110
__int16	BridgeControl	2	112
	Alignment Bits	**2**	**13**
__int32	Breserved	4	116
__int32	BexpRom	4	120
__int32	PreBaseUp32	4	124
__int32	PreLimitUp32	4	128
Char	Extra[192]	192 * 1	132
Char	CstringDriver[128]	128 * 1	324
Sum	Sum of bits	452	

Note:
The function ‘Dev_func” represents a complex structure for the fields for Device Number and Function, out lined below.

```
dev_func = XXXXX  XXX
           ----------   ------
              |           |
              |           → Function No.
              → Device No
```

If device is a multi function device, the ‘MultiFunction’ field will be set to 1 and ‘Function No’ is used to set functions 1 to 7.