



Windows 2000/XP and the IOMAP

Accessing IO-Ports under Windows 2000/XP

Copyright © 2002 SYBERA

Whitepaper No. 21122-01

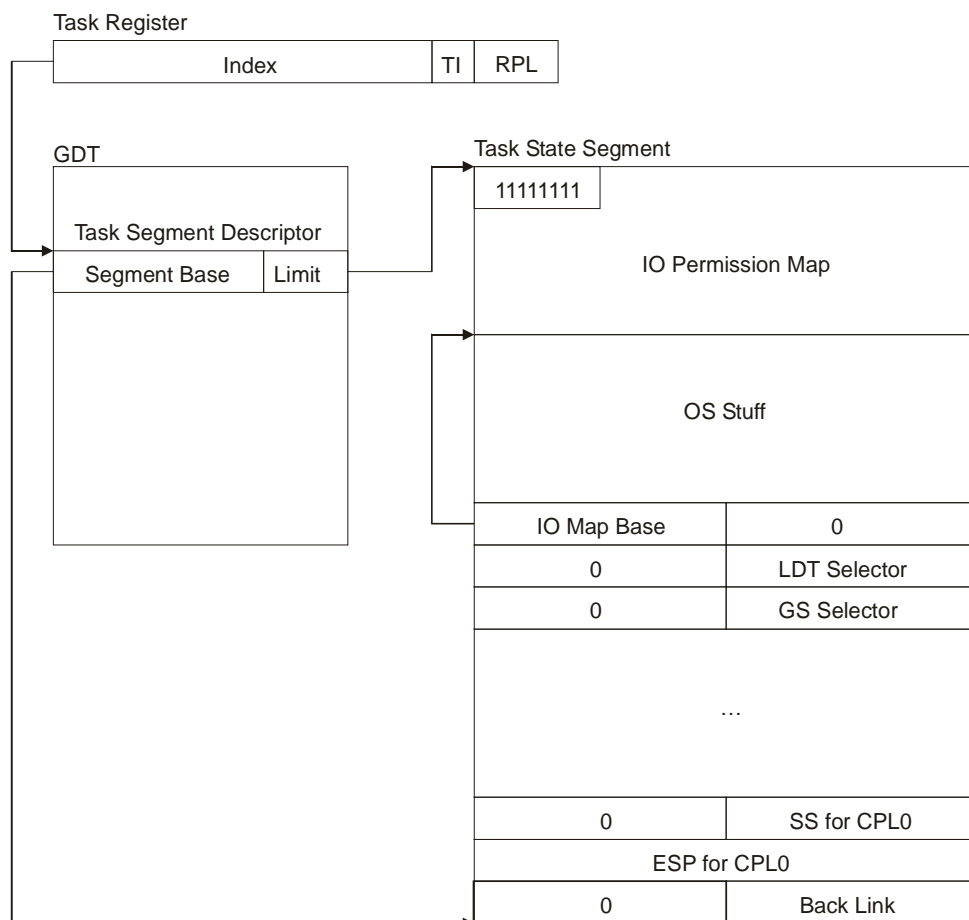
One of the most surprising issues is how Windows 2000/XP denies accessing IO-Port addresses. Considering Windows 95 accessing IO-Ports could be simply done at USER-Level. Under Windows NT accessing IO-Ports could only simply be done at KERNEL-Level, and under Windows 2000/XP IO-Port access depends on the Plug&Play configuration. All this behaviour is founded in the way how the protected mode of the x86 CPU works in conjunction with the operating system.

There are two ways the operating system is able to deny accessing IO Ports:

One mechanism is called IOPL Flag. To access IO ports the IOPL-Flag (IO privilege level) must match the condition of the CPL (current privilege level) of the code page descriptor. While USER-Mode code pages typically are running with CPL = 3, KERNEL-Mode pages are running with CPL = 0. Only if the condition $CPL \leq IOPL$ is TRUE, then access to IO port is possible. This was implemented in Windows 95 (IOPL = 3, IO port access is possible at USER-Mode level) and Windows NT (IOPL = 0, IO port access is possible at KERNEL-Mode level).

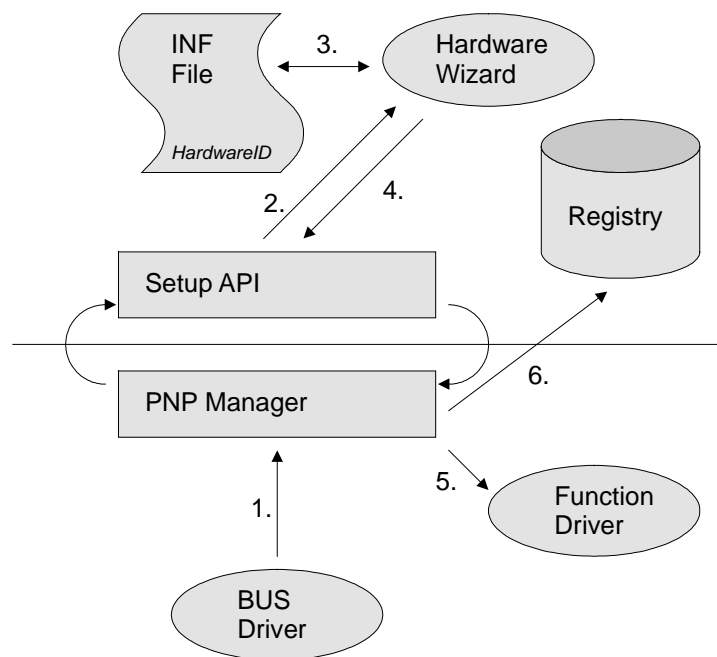
Now the surprising part came with Windows 2000/XP. Even when running at KERNEL-Mode (e.g. Device Drivers) access to IO ports with legacy drivers seems not to be possible any more. A lot of legacy drivers (e.g. NT drivers) do not work any more under Windows 2000/XP. Especially legacy drivers for PCI adapters with own resource management get into troubles. The reason therefor is a second mechanism implemented into the protected mode of the x86 CPU, called „IO Permission Map“ (IOMAP). The IOMAP is a part of the Task State Segment which is managed by the operating system.

e.g. `__asm STR cx`



Now, of course to bypass this restriction of the OS we could change the IO Map Base of the corresponding Task State Segment. But maybe there is an easier way when understanding how Windows 2000/XP maintains the IOMAP.

In contrary to WindowsNT on 2000/XP there is a dynamic resource management due to the Plug&Play mechanism. This management relies on the corresponding INF-File of a PNP-Driver (WDM Driver) which provides a HardwareID (e.g. for PCI the HardwareID consists of VendorID and DeviceID) to identify the PNP device resources to use. Windows 2000/XP provides a BUS-Driver which scans the PNP devices and compares the found configuration with the given HardwareID of the INF-File. If both match then Windows 2000/XP enables the IOMAP for the found device resources.



1. BUS-Driver scans the PNP devices and finds a new device
2. The SetupAPI calls the Hardware-Wizard
3. The Hardware Wizard reads in the corresponding INF-File
4. The HardwareID is given to the System
5. The AddDevice routine of the Function-Driver will be called
6. The device resources will be reported and the corresponding IOMAP enabled

Now we found a way to enable the IOMAP to get legacy drivers at work. The thing we need to do is simply give the system a so called „Dummy WDM Driver“ with a corresponding INF-File to satisfy the system needs. Heres a sample for such an INF-File:

```
[Version]
Signature="$WINDOWS NT$"
Class=Unknown
ClassGUID={4D36E97E-E325-11CE-BFC1-08002BE10318}
Provider=%Provider%
DriverVer=03/26/2001,1.0.0.0

[Manufacturer]
%Mfg% = Sybera

;*****
;This section needs to be changed with the corresponding
;VendorID and DeviceID
;*****
[Sybera]
%Product%=ShaWdm.Install,PCI\VEN_10B5&DEV_9050

[SourceDisksNames]
1=,,,

[SourceDisksFiles]
shawdm.sys=1 ;Change name of WDM Dummy Driver

[DestinationDirs]
ShaWdm.Files.NT=12

[ShaWdm.Install.NT]
CopyFiles=ShaWdm.Files.NT

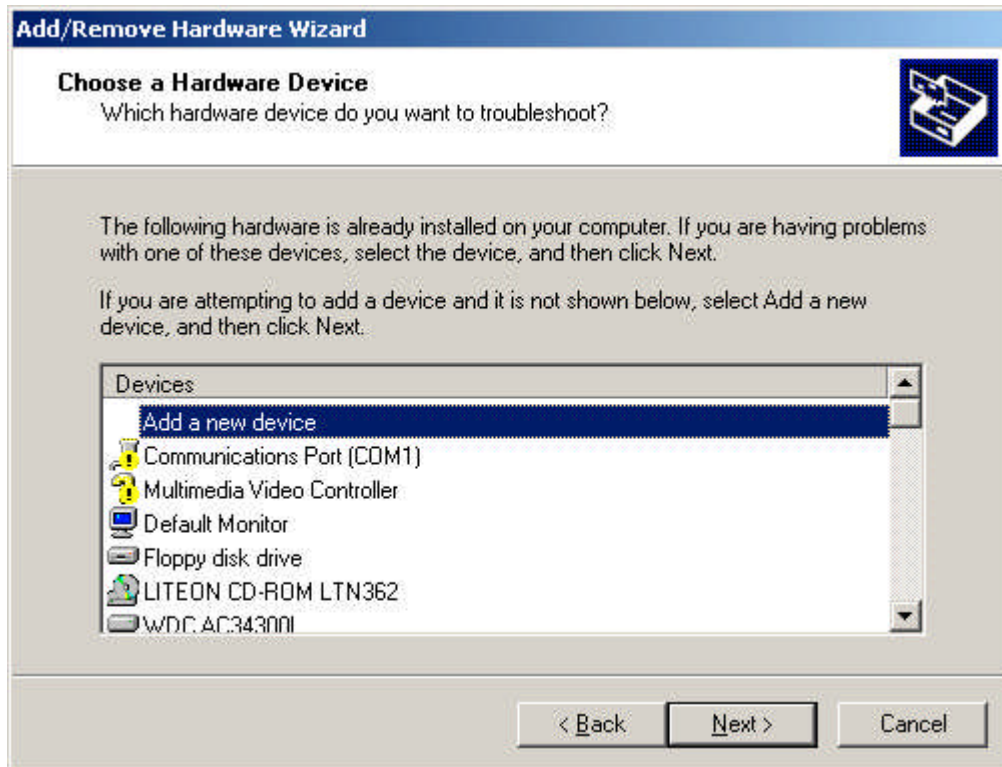
[ShaWdm.Install.NT.Services]
AddService = ShaWdm,2,ShaWdm.Service ;Change name of WDM Dummy Driver

[ShaWdm.Files.NT]
shawdm.sys

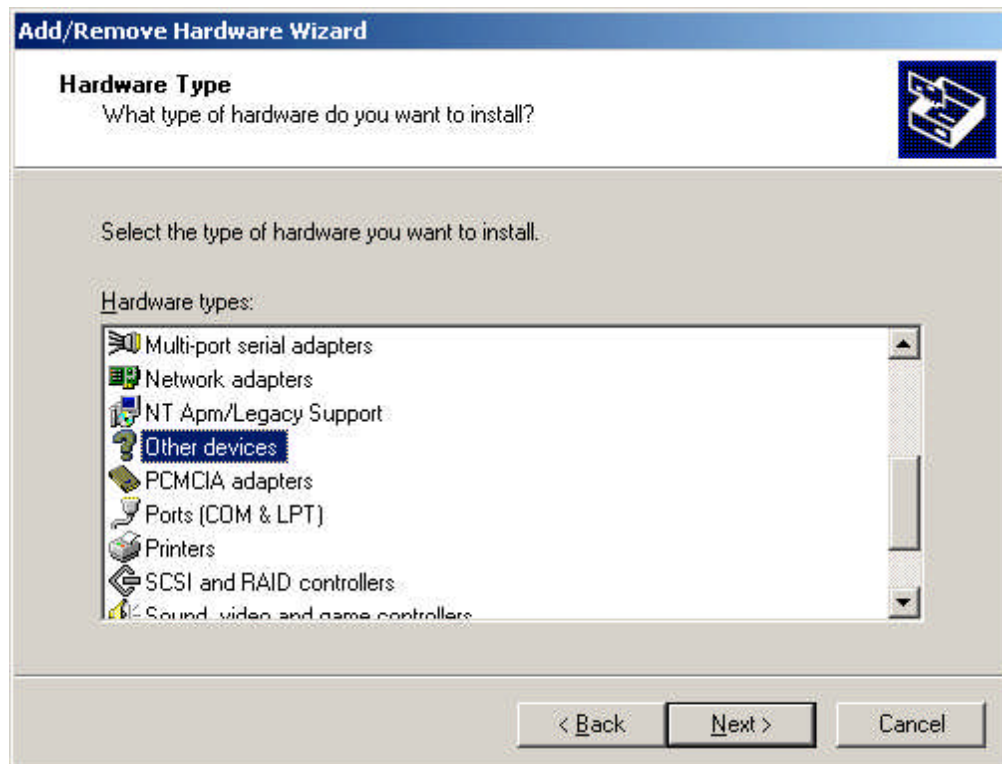
[ShaWdm.Service]
DisplayName = "SHA Win2000 Support Driver"
ServiceType = 1
StartType = 3
ErrorControl = 1
ServiceBinary = %12%\shawdm.sys ;Change name of WDM Dummy Driver

[Strings]
Mfg="Sybera"
Provider="Sybera"
Product="SHA Win2000 Support Driver"
```

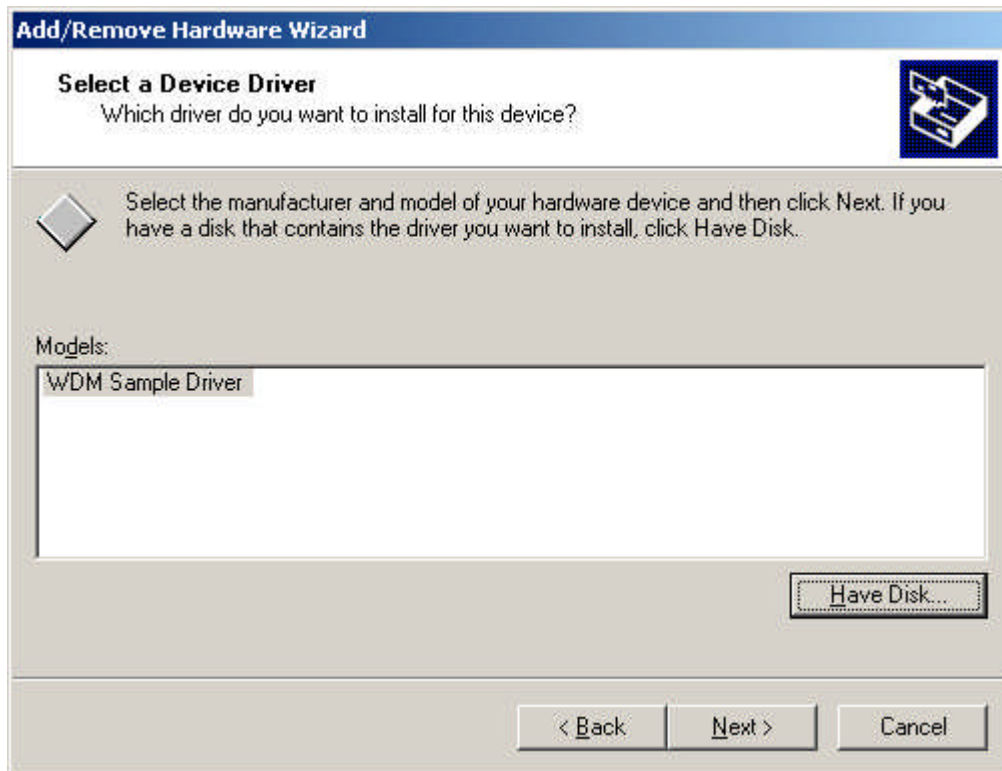
For installation of a „WDM Dummy Driver“ open the Hardware Wizard inside the Control Panel [Control Panel -> Add/Remove Hardware]



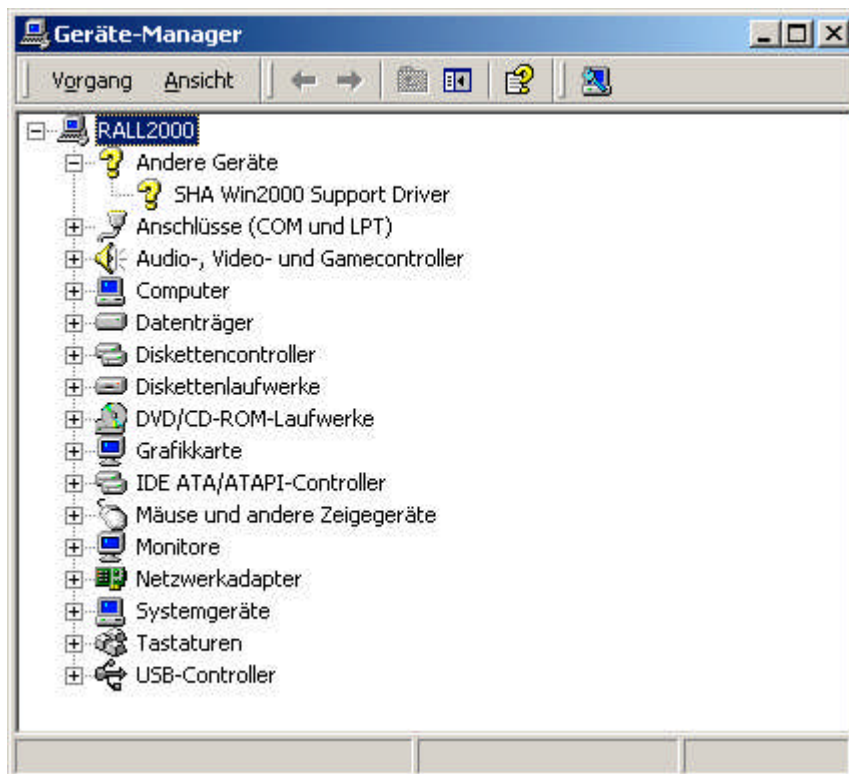
Each WDM Driver corresponds to a device class. For our "WDM Dummy Driver" we just use the class UNKNOWN (Other Devices)



After setting the installation path the WDM Dummy Driver will be displayed:



After installation the driver is entered to the device manager window
[Control Panel -> Administrator Tools -> Computer Management -> Device Manager]



To write a WDM Dummy Driver a lot of stuff can be found at www.sybera.com. Together with such a driver and a corresponding INF-File the world seems turning again.

SYBERA, Nov.22.2002