



unibrain

The 1394 Innovators

Technical support: support@unibrain.com Sales Information: sales@unibrain.com

www.unibrain.com

June 2004

Unibrain introduces to the IEEE-1394 users and developers community, the latest **ubCore™ version 3.2**, the first and only FireWire (IEEE-1394) driver suite that fully supports the **1394b** standard. **ubCore™** is a set of low-level drivers with a top-down layered structure that provides the physical interface to various 1394 devices and includes the following advantageous features:

- Full 1394b support at 800Mbps.
- Complete 1394a and 1394b SBP2 disk support for Windows 98SE/Me/2000/XP.
- Comprehensive set of tools and applications.
- Seamless integration to all Windows OS.
- User friendly installation wizard.

ubCore™ combined with FireAPI™ (Unibrain's 1394 Development Toolkit), provides to the software developer a powerful set of functions and tools that enables him to integrate 1394 technology into his product. Unibrain has built a number of 1394 applications based on FireAPI™ and included in ubCore™:

- FireViewer™: a 1394 Bus Viewer utility that displays the current topology of the local 1394 bus.
- FireCommander™ a 1394 command interpreter tool that is used to pass commands to the 1394 stack. Benchmarking support is included.
- FireNet™: Software to network two or more computers via 1394a or 1394b.
- Fire-i™: Software to Control and Display any 1394 Digital Camera (IIDC).
- Fire-Share™: Software to allow sharing of any 1394a or 1394b SBP2 disk among multiple users.

Software Compatibility of ubCore™

- Windows NT4/98SE/Me/2000/XP

Hardware Compatibility of ubCore™

- Unibrain's or any third party 1394a or 1394b OHCI PCI/PCMCIA compliant adapter
- Any Desktop/Portable PC with built-in 1394a or 1394b OHCI support.
- Unibrain's FireBoard400™ (PCI Lynx 2) 1394a PCI adapter.

Free Evaluation of ubCore™

The ubCore™ software you are about to install is the complete and fully functioning version for all the Windows environments. Upon initial installation, usage is restricted to an evaluation period of 30 minutes, at the end of which the driver stack will automatically turn off (without affecting the rest of your computer operations in any way). You can repeat the demo as many times as you wish by simply rebooting your PC.

Product Keys

To enable unrestricted use or "authorized" use of ubCore™ software, a Product Key is required for each and every computer that has installed ubCore™. To purchase additional Product Keys please contact your distributor or visit www.unibrain.com.

Important before you proceed

If you plan to use your 1394 ports for DV camcorders only, do not install the ubCore™ 1394b drivers.

If you plan to use your 1394 ports with various Firewire devices (external hard drives, DVD-ROM, DVD Recorders, cameras etc.) and DV camcorders, please refer to [page 24 \(Notes about DV camcorder users\)](#) on how to adjust your computer for each device.

Table of Contents

IMPORTANT BEFORE YOU PROCEED	3
IF YOU PLAN TO USE YOUR 1394 PORTS FOR DV CAMCORDERS ONLY, DO NOT INSTALL THE UBCORE™ 1394B DRIVERS.	3
TABLE OF CONTENTS.....	4
TECHNICAL SUPPORT INFORMATION.....	6
FEATURES OF UBCORE 3.2	7
UBCORE™ VERSUS MICROSOFT 1394 STACK.....	9
SETUP CHECKLIST.....	11
UBCORE™ INSTALLATION.....	12
INTRODUCTION.....	12
UBCORE™ INSTALLATION PROCEDURE.....	13
VERIFICATION OF UBCORE™ INSTALLATION	18
VERIFICATION ON WINDOWS NT4	18
VERIFICATION ON WINDOWS 98SE/ME/2000/XP	20
DIRECTORY STRUCTURE OF UBCORE™ INSTALLATION.....	21
COEXISTENCE OF UBCORE™ AND MICROSOFT 1394 STACK /	24
NOTES ABOUT DV CAMCORDER USERS	24
THIRD PARTY 1394B PCI CONTROLLERS WORKING WITH UBCORE™	28
UNINSTALLING UBCORE™	29
UPGRADING UBCORE™	29
UBCORE™ AUTHORIZATION PROCEDURE	30
1394A AND 1394B SBP2 DISK DRIVER.....	31
INSTALLING SBP2 DISK DRIVER IN WINDOWS 98SE/ME.....	32
1394 SBP2 EXTERNAL DISK CONNECTION	34
SAFE REMOVAL OF 1394 SBP2 DISKS	38
FIREVIEWER™-1394 BUS VIEWER	39
THE NODE TREE	39
COLOR AND PATTERN CODING	41
ADAPTER SELECTION	42
WINDOW APPEARANCE	42
NODE SIZE	42
PORTS.....	42
CAPTION AND MENU BAR	43
ALWAYS ON TOP.....	43
NODE PROPERTIES	43
BUS ANOMALIES.....	45
MENU	46
VIEW MENU.....	46
HELP MENU.....	46
FIRECOMMANDER™- 1394 COMMAND INTERPRETER.....	47

RUNNING THE PROGRAM	47
COMMAND SYNTAX	48
AVAILABLE COMMANDS.....	49
<u>FIRENET™-1394 NETWORKING</u>	<u>56</u>
INSTALLATION ON WINDOWS NT4	56
VERIFY INSTALLATION	58
INSTALLATION ON WINDOWS 98SE/ME/2000/XP.....	59
UN-INSTALLING FIRENET™	62
<u>FIRE-I™ APPLICATION FOR 1394 IIDC CAMERAS</u>	<u>63</u>
FIRE-I™ COMPATIBILITY	63
FIRE-I™ APPLICATION GUIDE.....	64
CONNECTED CAMERAS	64
VIDEO FORMAT	64
VIDEO MODE.....	64
FRAME RATE.....	64
SPEED	65
MEMORY PRESETS.....	65
CAMERA CONTROL.....	65
DISPLAY POPUP MENU	66
FIRE-I™ SHORT-KEYS	69
<u>FIRESHARE™</u>	<u>70</u>
IMPORTANT WARNING!.....	70
INTRODUCTION.....	70
USING FIRESHARE™	71
DISK COLORING	72
SHORT MESSAGING	74

Technical Support Information

All technical support questions should be directed to the following email address:

support@unibrain.com

This address is monitored throughout the day by Unibrain's 1394 support team, who will respond quickly and with the required technical accuracy, to your questions.

Please refrain from sending any queries directly to Unibrain engineering staff, in order to maintain our support database and make sure that your email will be responded to, in case any specific person is unavailable. Please, try to be as specific as possible in your inquiries to this email address, and to restrict questions to technical issues.

For pricing and marketing information, please e-mail at:

sales@unibrain.com

For ordering any Unibrain product please visit:

<http://www.unibrain.com/>

Features of ubCore 3.2

ubCore™ version 3.2 provides to both existing customers and to new users, the next generation of Unibrain's innovative 1394 driver suite with the following advantageous features:

1). Full 1394b support at 800Mbps speed rate. ubCore™ 3.2 is the first and only IEEE-1394 (FireWire) driver platform providing full support of the latest 1394b standard. This support includes amongst others:

- New Self-ID Packets
- Changes to PHY Register Set
- Request Higher speed packets
- Request bigger packets at higher speeds
- Pipelining of packet transmit requests
- Handling faster receipt of BOSS asynchronous packets
- Path Speed being Port dependent and not Node Dependent
- Adapt to longer length cables and larger variations in PHY repeater delays
- Picking the most optimal packet format
- Initiating and resuming from the "Standby" power mode

2). Full 1394b support at S100 (CAT5) and S200 (POF) speed rates. ubCore™ 3.2 is the first and only IEEE-1394 (FireWire) driver platform providing full support of S100 speed rate using CAT5 cable as the interconnection mean between two nodes. Moreover ubCore™ fully supports the S200 speed between two nodes connected via Plastic Optical Fiber (POF).

With sophisticated and recursive built in algorithms, ubCore™ builds a correct speed map of each port of every node in the 1394b bus. This speed map determines the maximum speed that a node can be accessed. ubCore™ can do transactions at any speed, and uses the maximum speed of a specific node that works. This automatic mechanism enables the correct communication of every node in a 1394b bus even if not all devices operate at S100 or S200 speeds.

3). Complete SBP2 support for 1394a and 1394b SBP2 hard disks. SBP-2 ("Serial Bus Protocol", 2nd version) is the best model for computer peripherals. Its use is strongly preferred, largely because it greatly reduces interrupt load on the CPU, and provides a scalable DMA model. The general architecture provides a command and data transport mechanism that exports the command processing and queuing to the peripheral, effectively making each device a powerful scriptable DMA device. This is done by mapping some area of the CPU's memory directly to 1394's memory space, building a linked-list of commands and pointers to data buffers in that area, and then giving the address of that list to the device. The device then reads the commands directly from the CPU and takes the appropriate action, including reading data from or writing data directly to CPU memory.

Note: The SBP2 drivers does not support the Windows NT 4.0 Operating System

Unibrain's 1394 SBP2 driver is fine tuned for speeds of 800Mbps providing the fastest 1394b interface to external S800 SBP2 Hard disks. It is also compatible and fully supports Removable Devices and other SBP2 compliant devices like Apple's iPod.

Moreover Unibrain provides the Fire-Share™ 1394 SBP2 application for multi-share of 1394 SBP2 hard disks. As you probably know, while it is possible for a PC to attach to many external 1394 SBP2 disks, it is impossible to simultaneously attach one 1394 SBP2 disk to many PCs connected via FireWire. In other words the SBP2 drive will be attached and recognized only by one PC, in a FireWire network. With Fire-Share™ you are now able to have mutually exclusive usage of the 1394 SBP2 disk, and once used, you can release it notifying the other PCs so this disk becomes available for other user.

4). IIDC 1394 Digital Cameras Specification 1.30 support

Along with ubCore™ 3.2, you receive the full IIDC 1394 stack to control and display 1394 digital cameras.

5). Native networking capabilities provided by FireNet™

Unibrain's FireNet™ software, one more application implemented over ubCore™, is THE FIRST FireWire™ Networking Solution for ALL Windows platforms. FireNet™ is a system-independent software product that gives you full Ethernet compatibility at speeds of up to 800Mbps. The basic features of FireNet™ are:

- Complete Ethernet emulation.
- Supports various network protocols: TCP/IP, NetBEUI, IPX/SPX, etc.
- High speed – up to 8 times the performance of 100Mbps fast Ethernet.
- No hubs, no routers, complicated cabling, or complex installations.
- Fast and Easy installation.

6). Support on ALL Microsoft operating systems: NT4/98SE/Me/2000/XP: ubCore™ is the only existing 1394b software platform for the Windows environments.

7). Complete hardware support: ubCore™ 3.2 supports Unibrain's or any third party 1394a or 1394b OHCI PCI/PCMCIA compliant adapter or any Desktop/Portable PC with built-in 1394a or 1394b OHCI support. It also supports Unibrain's FireBoard400™ (PCI Lynx 2) 1394a PCI adapter.

ubCore™ versus Microsoft 1394 stack

As mentioned, ubCore™ is a complete driver stack providing support for all 1394a **and 1394b** devices. With your Microsoft operating system you get also 1394a support but no 1394b, The following comparison table presents the basic ubCore™ features and advantages in contrast with the features of the Microsoft 1394 stack.

Features	Microsoft 1394 stack	ubCore™ 3.2 /FireAPI™ 3.2
1394a Software support	NO Windows NT4	Windows NT4, 98SE, Me, 2000, XP, 2003
1394b Software support	NO (speed is dropped to S100 on Windows XP SP2)	Windows NT4, 98SE, Me, 2000, XP, 2003
1394a Hardware support	OHCI only	Lynx2 and OHCI compatible boards
1394b Hardware support	NO	FireBoard800™ and any other 1394b compliant device based on the Texas Instruments TSB82AA2 1394b to PCI controller.
IIDC 1394 digital cameras	Partial Support on 98SE/ME/2000/XP	Fire-i™ application and IIDC drivers for Windows NT4, 98SE, Me, 2000, XP. (FireAPI™ includes FULL source code)
1394 SBP2 Hard Disk support	(speed is dropped to S100 on Windows XP SP2) No sharing application	NATIVE S800 speed..!! FireShare™ application
1394 Native Networking	Windows ME/2000/2003 (speed is dropped to S100 on Windows XP SP2)	FireNet™ drivers for Windows NT4, 98SE, Me, 2000, XP, 2003 NATIVE S800 speed..!!
1394 Topology Tool	NO	FireViewer™ application, a 1394 Bus Viewer utility that displays the current topology of the local 1394 bus. (FireAPI™ SDK includes full source code)
1394 Control Tool	NO	FireCommander™ application, a 1394 command interpreter tool that is used to pass commands to the 1394 stack. (FireAPI™ SDK includes full source code)
Serial Bus Manager	NO	SBM™ driver, providing the set of functions to control bus management and isochronous resource management. (FireAPI™ SDK includes full source code)

Virtual 1394 Device support	Primitive support on Windows Me, XP	Windows NT4, 98SE, Me, 2000, XP, 2003
User mode API for applications development	NO	YES (FireAPI™ SDK)
Loop back functionality.	NO	YES

Table 1: ubCore™ versus MS 1394 stack

Setup Checklist

Before proceeding with the product installation, please verify that:

Check No.	Description	Check? (√)
1	Your host PC operating system is Windows NT4/98SE/Me/2000/XP.	
2	For all operating systems (but especially for Windows 98SE and Windows ME), please be sure to have latest updates installed (through http://www.windowsupdate.microsoft.com/)	
3	Your PC has a FireWire (i-Link) IEEE-1394 interface based on: <ul style="list-style-type: none"> • Unibrain's or any third party 1394a or 1394b OHCI PCI/PCMCIA compliant adapter or • any built-in 1394a or 1394b OHCI support or • Unibrain's FireBoard400™ (PCI Lynx 2) 1394a PCI adapter. 	
4	You have disabled the Plug and Play Operating System flag on your BIOS setup, in case that you are going to install ubCore™ into Windows NT 4.0 system. Please also verify that your system has installed Service Pack 6!	
5	IMPORTANT! Always UNPLUG the connecting FireWire cables between all the FireWire computers and devices before you begin installing or uninstalling ubCore™ software.	
6	IMPORTANT! ALWAYS COMPLETELY UNINSTALL ANY PREVIOUS VERSION OF ubCore™ BEFORE INSTALLING THE NEW ONE. Please refer to section "Upgrading ubCore™"	

Table 2: Setup Checklist

ubCore™ Installation

Introduction

For all operating systems there are common basic tasks during installation that you should follow. Those tasks are in essence the next:

Task No.	Description	Check? (√)
1	Verify the Setup Check list described in the previous Table 2	
2	Follow the installation instructions described into the next paragraphs.	
3	IMPORTANT! IN CASE YOU ARE ABOUT TO INSTALL ubCore™ TO A SYSTEM ALREADY HAVING ubCore™, THEN THE CORRECT PROCEDURE IS TO COMPLETELY UN-INSTALL THE OLD VERSION, REBOOT YOUR SYSTEM AND THEN FOLLOW THE STEPS OF THE INSTALLATION PROCEDURE IN ORDER TO IN STALL THE NEW VERSION.	
4	Once finished your installation (you need to restart your system) you have to check the correct installation of ubCore™. There are many procedures to do, but the most simple and straightforward is the usage of the FireViewer™ application. Please refer to the section “Verification of ubCore™ installation”	
5	After verifying the correct installation of ubCore™ it is time to fully authorize it. Please recall from previous section that the ubCore™ software you already have installed is a complete and fully functioning version that its usage is restricted to an evaluation period of 30 minutes, at the end of which the driver stack will automatically turn off (without affecting the rest of your computer operations in any way). You can repeat the demo as many times as you wish by simply rebooting your PC. To enable unrestricted use or “authorized” use of ubCore™ software, a Product Key is required for each and every computer that has installed ubCore™. Product keys may be purchased online from Unibrain or from any distributor. Please refer to the section “Authorization procedure” for more details.	

Table 3: Basic Installation Tasks

ubCore™ installation procedure

This part explains how to install ubCore™ 3.2 in your system under the Windows XP operating system. Windows NT4/98Se/Me/2000 have a similar installation. Please notice that the ubCore™ installer automatically identifies the target Operating System and installs the required files accordingly.

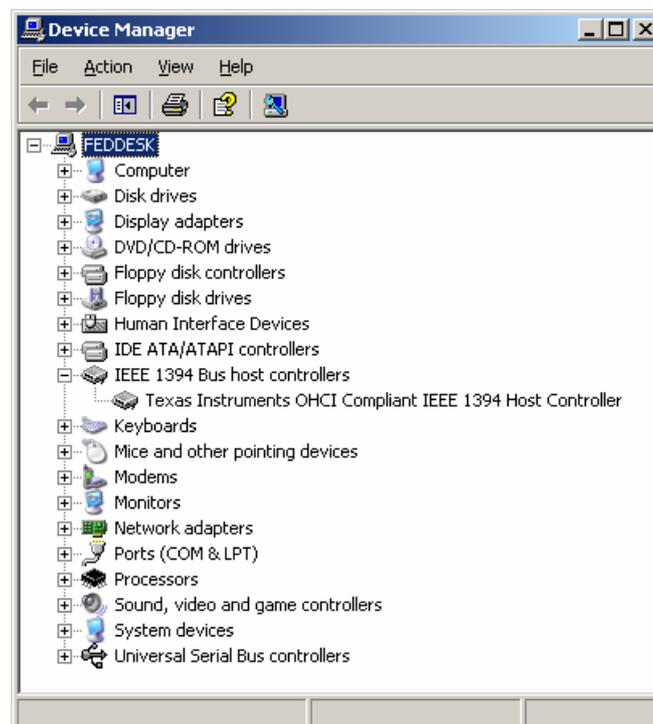
Important note for Windows 98 Second Edition users:

Before you install ubCore™ drivers, you must make sure your system is updated with the latest patches/system updates from Microsoft. Please do the following:

1. Visit <http://windowsupdate.microsoft.com> and install all main security & hotfix updates along with the latest Internet Explorer version (6.1 SP1).
2. - Download and install the 1394 patch from Microsoft for Win98SE:
http://www.microsoft.com/windows98/downloads/contents/WURecommended/S_WUFeatured/1394/Default.asp

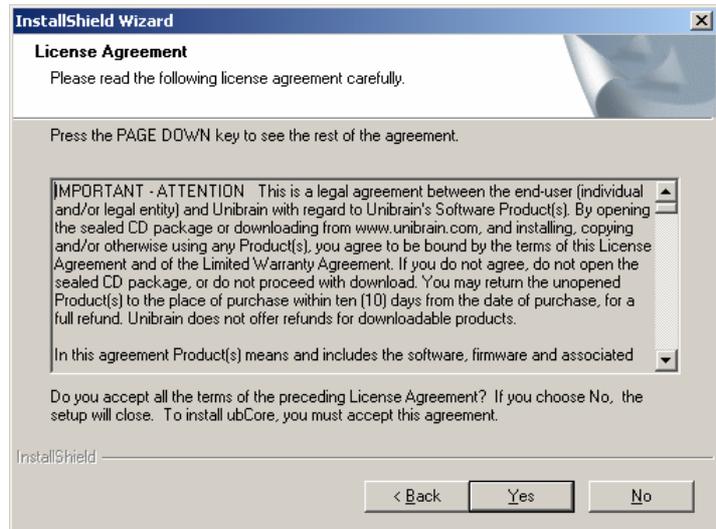
After you restart your computer on both cases, you will be able to start the ubCore™ setup procedure.

The first step is to verify that the Microsoft 1394 stack is installed in your system. Provided that you have the required 1394 hardware in your PC, the Windows XP system will recognize it as an IEEE 1394 Bus host Controller and will install it automatically as can be seen from the Device Manager (see next figure).

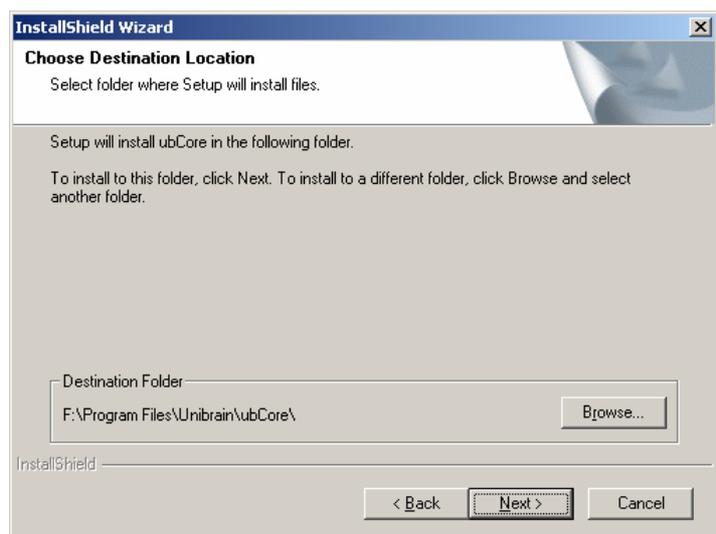


Locate the file “ubCore.exe” executable file and launch it by double-clicking on it. The Welcome screen will be displayed. To continue with the installation, click on “Next”.

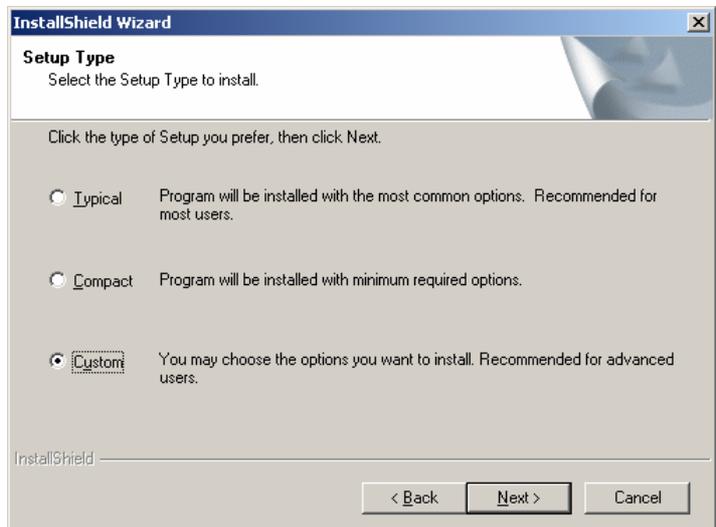
The ubCore™ License Agreement is being displayed next. Read this Agreement thoroughly. If you agree with the terms stated in this License Agreement, click on “Yes”.



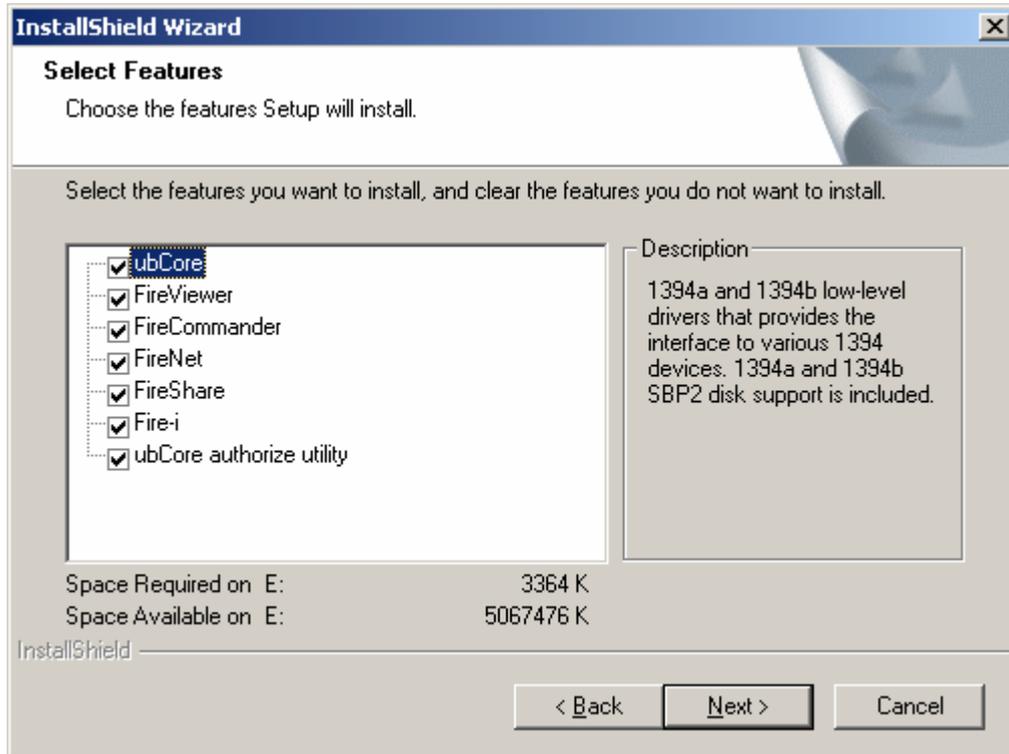
Enter your Customer Information, click Next and Choose the Destination Location of the ubCore™ files.



Choose the type of installation. The default selection is “Typical”, but in case you want to modify the configuration, choose the option “Custom”.



If you chose the “Custom installation” option, you will be shown the window below. Choose the components that you wish to install and then press “Next” to proceed.



√ ubCore: 1394a and 1394b low-level drivers that provides the interface to various 1394 devices. 1394a and 1394b SBP2 disk support is included.

√ FireViewer: A 1394 Bus Viewer utility that displays the current topology of the local 1394 bus.

√ FireCommander: a 1394 command interpreter tool that is used to pass commands to the 1394 stack. Benchmarking support is included.

√ FireNet: 1394 networking software.

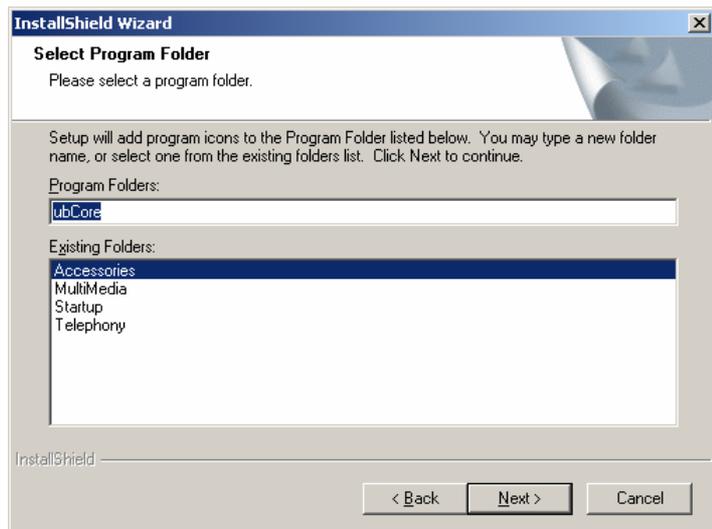
√ FireShare: Software to allow sharing of any 1394a or 1394b SPB2 disk among multiple users.

√ Fire-i: Software to Control and Display any 1394 IIDC Firewire digital camera.

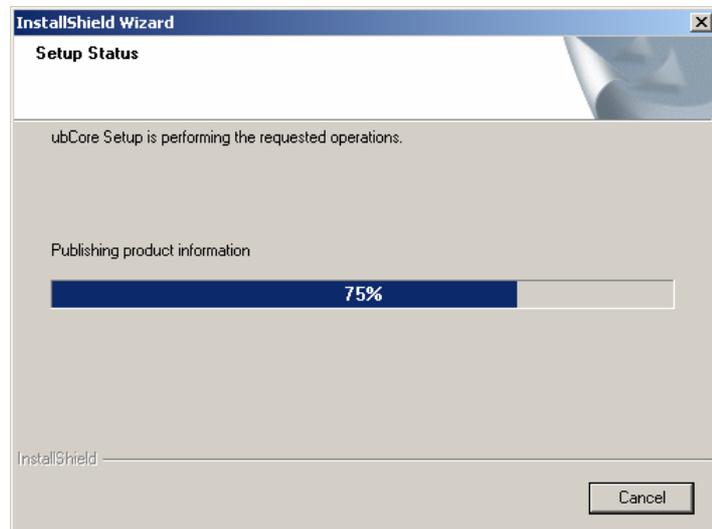
√ ubCore Authorize utility: Utility to authorize ubCore software for unlimited use. A Product Key "PK" number is needed.

If all options are set correctly, click on “Next” to complete the installation process.

You can let Install create a new Program Folder or add the Unibrain ubCore™ icons to any existing group. In the latter case, select the Group you want to add ubCore™. Press Next to proceed.



The installer will start copying the required files in the respective directories.

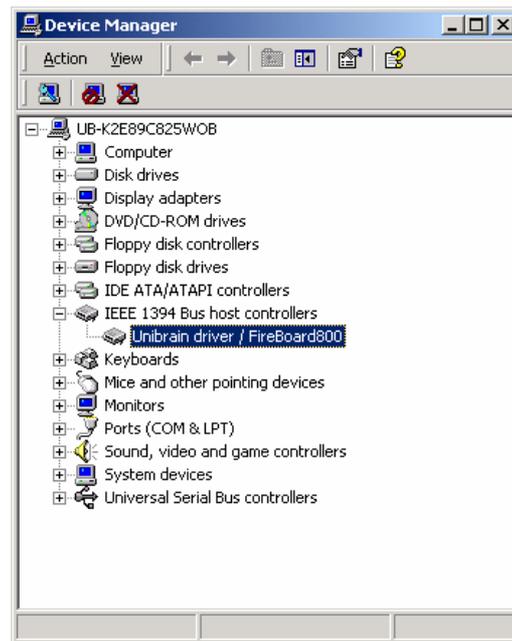


Once the next window, press “Continue Anyway”:



Once the installation procedure is completed, press Finish to end the installation.

The ubCore™ drivers and the accompanying software have been installed and, if you check the device manager, you can observe the presence of Unibrain's driver for the control of the "IEEE 1394 Bus host controllers":



To verify that the ubCore™ suite of drivers are all installed successfully run the FireViewer™ utility from the "Start→Programs→ubCore". For more details please refer to the section below, "Verification of ubCore™ installation".

Verification of ubCore™ Installation

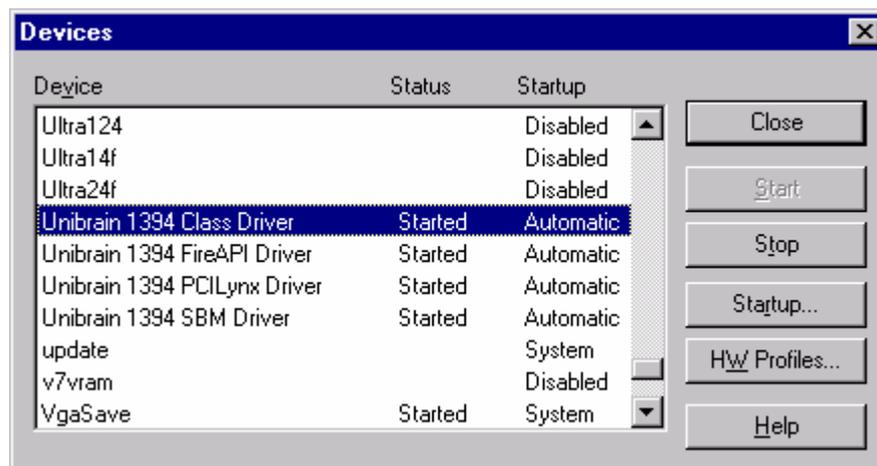
Verification on Windows NT4

The most simple procedure to verify the correct installation of ubCore™ is to run the FireViewer™ utility from the “Start→Programs→ubCore”. Provided that you have one 1394 PCI controller, not connected with any other FireWire device, you should see the following window:



For more details on the FireViewer™ application, please refer to the respective section.

There is also an alternative way to verify that the ubCore™ suite has been correctly installed at your Windows NT 4 system. After reboot open the Control Panel/Devices and go to the entry "UB"



In case you have installed a FireBoard400 or a FireCompact400 four Unibrain Drivers should have started automatically:

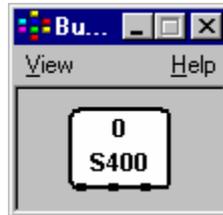
- **Unibrain 1394 PCILynx Driver – UBLynx.sys**
PCILynx-2 miniport for FireBoard™400 or FireCompact400
- **Unibrain 1394 Class Driver – UB1394.sys**
Kernel Mode (ring 0) 1394 API
- **Unibrain 1394 FireAPI Driver – UBUMAPI.sys**
User Mode (ring 3) 1394 API
- **Unibrain 1394 SBM Driver – UBSBM.sys**
Serial Bus Manager client driver

In case you have installed a Unibrain's FireBoard800™ or a FireBoard400-Blue™ or a FireCard™ or any other third party OHCI compliant 1394 board, then you should have:

- **Unibrain 1394 OHCI Driver – UBOHCI.sys**
OHCI miniport for OHCI compliant 1394 PCI controllers
- **Unibrain 1394 Class Driver – UB1394.sys**
Kernel Mode (ring 0) 1394 API
- **Unibrain 1394 FireAPI Driver – UBUMAPI.sys**
User Mode (ring 3) 1394 API
- **Unibrain 1394 SBM Driver – UBSBM.sys**
Serial Bus Manager client driver

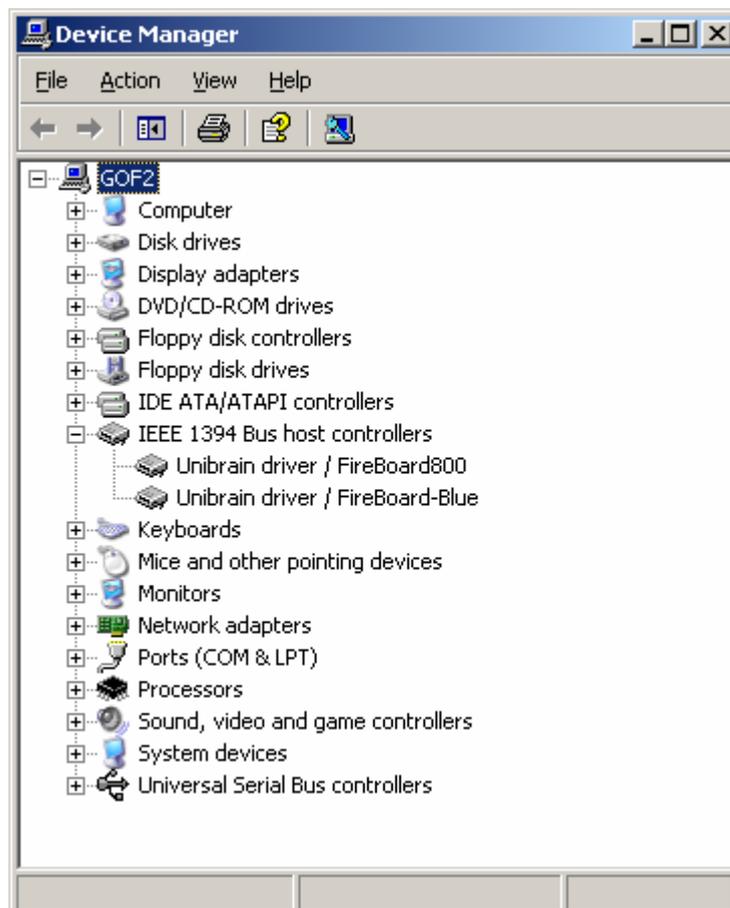
Verification on Windows 98SE/Me/2000/XP

The most simple procedure to verify the correct installation of ubCore™ is to run the FireViewer™ utility from the “Start → Programs → ubCore”. Provided that you have one 1394 PCI controller, not connected with any other FireWire device, you should see the following windows:



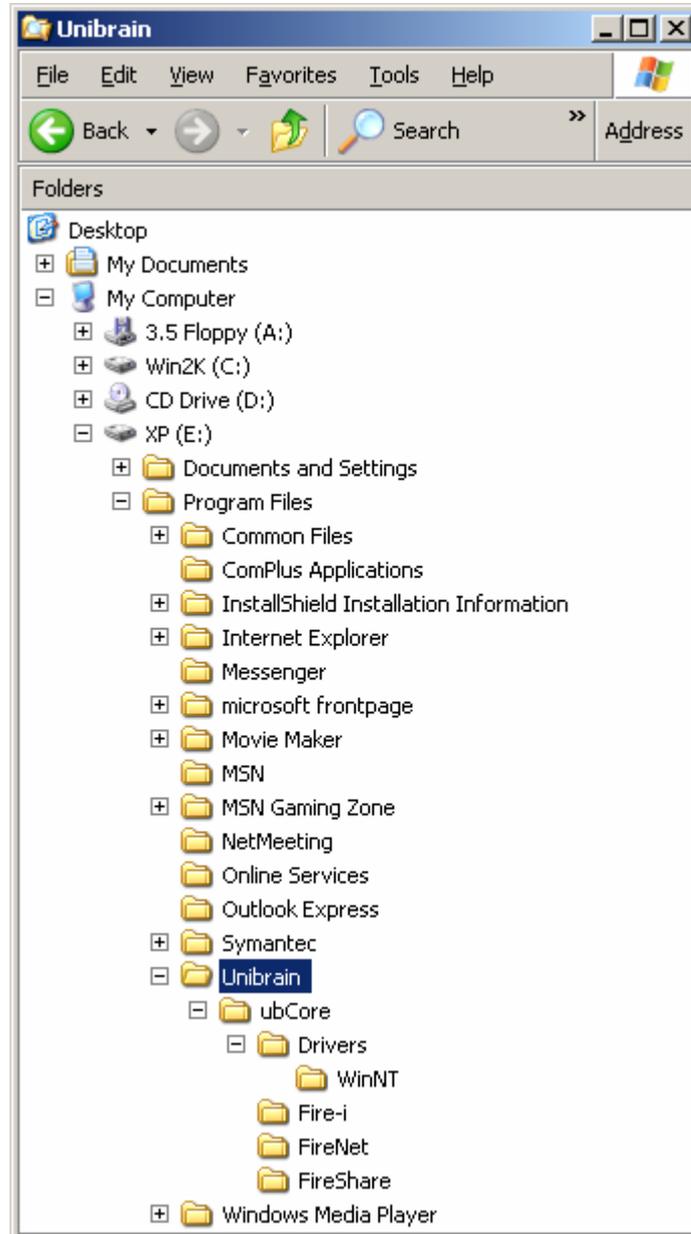
For more details on the FireViewer™ application, please refer to the respective section.

There is also an alternative way to verify that the ubCore™ suite has been correctly installed at your Windows 98SE/Me/2000/XP system: At the Device Manager you should be able to see the driver installed under the "IEEE 1394 Bus host controllers" section. For example, in the next figure you can see the "Unibrain driver/FireBoard-Blue" and the "Unibrain driver/FireBoard800" entries, providing you with the information that the two 1394 PCI adaptors in your system (one FireBoard-Blue and one FireBoard800) have been correctly recognized by ubCore™.



Directory structure of ubCore™ installation

After the complete installation of ubCore™ 3.2 the directory structure should be as depicted in the next figure, provided that you use the default destination directory: "E:\Program Files\Unibrain".



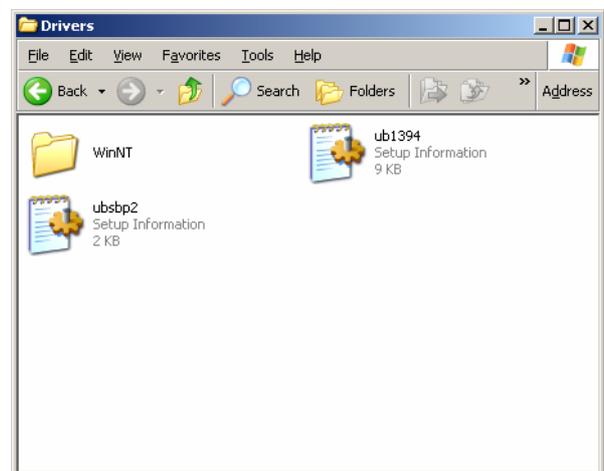
So the ubCore™ subdirectory is structured as follows:

\ubCore\: here are provided the following 3 applications:

- **FireCommander:** a 1394 command interpreter tool that is used to pass commands to the 1394 stack. Benchmarking support is included.
- **FireViewer:** A 1394 Bus Viewer utility that displays the current topology of the local 1394 bus.
- **ubCore Authorize:** Utility to authorize ubCore™ software for unlimited use.

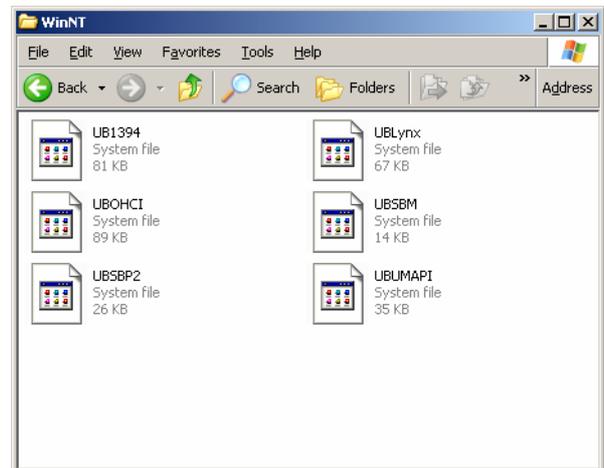


\ubCore\drivers: here are provided the ub1394.inf and ubsbp2.inf system information files.

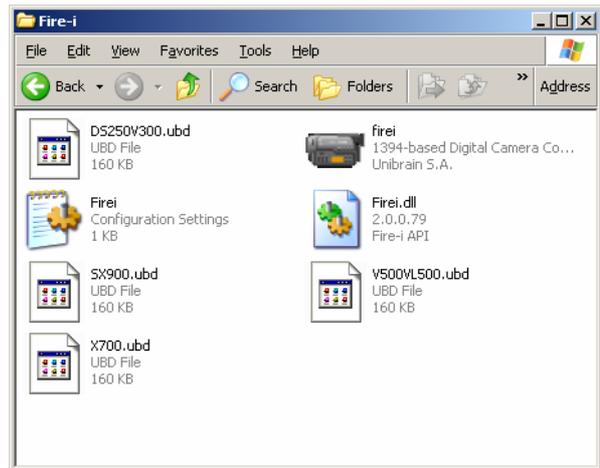


\ubCore\drivers\WinNT: (for NT4/2000/XP OS)
or
\ubCore\drivers\Win98: (for 98SE/Me OS)

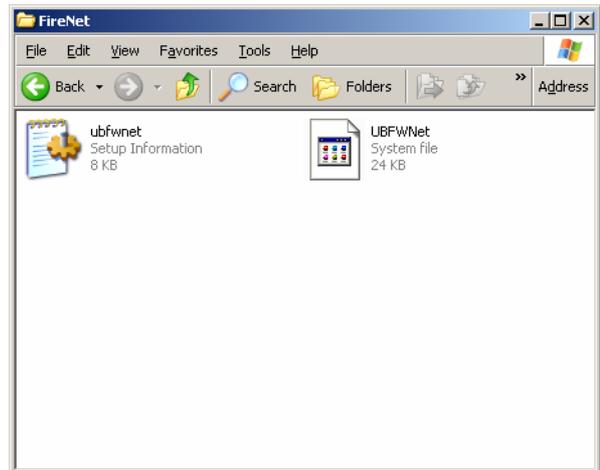
- **UB1394.sys: Unibrain 1394 Class Driver – Kernel Mode 1394 API**
- **UBLynx.sys: Unibrain 1394 PCILynx Driver**
- **PCILynx-2** miniport for FireBoard400™ or FireCompact400™
- **UBOHCI.sys: Unibrain 1394 OHCI Driver**
- **OHCI** miniport for OHCI compliant 1394 PCI controllers
- **UBSBM.sys: Unibrain 1394 SBM Driver – Serial Bus Manager client driver**
- **UBUMAPI.sys: Unibrain 1394 FireAPI Driver – User Mode 1394 API**



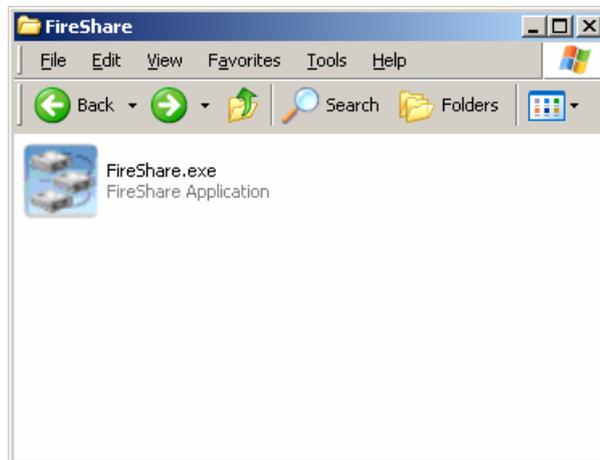
\ubCore\Fire-i: here is provided the Fire-i™ application software to Control and Display any 1394 IIDC camera, along with the required Firei.dll and *.ubd description files for all the SONY 1394 IIDC cameras.



\ubCore\FireNet: here is provided the FireNet™ (1394 Networking software) driver and this is the path you should retrieve during the manual installation of FireNet™. (Please refer to the FireNet™ Installation section for more details)



\ubCore\FireShare: here is provided the FireShare™, Software to allow sharing of any 1394a or 1394b SPB2 disk among multiple users. Moreover the authorization program is located in this folder. Please refer to the FireShare™ chapter for further details.



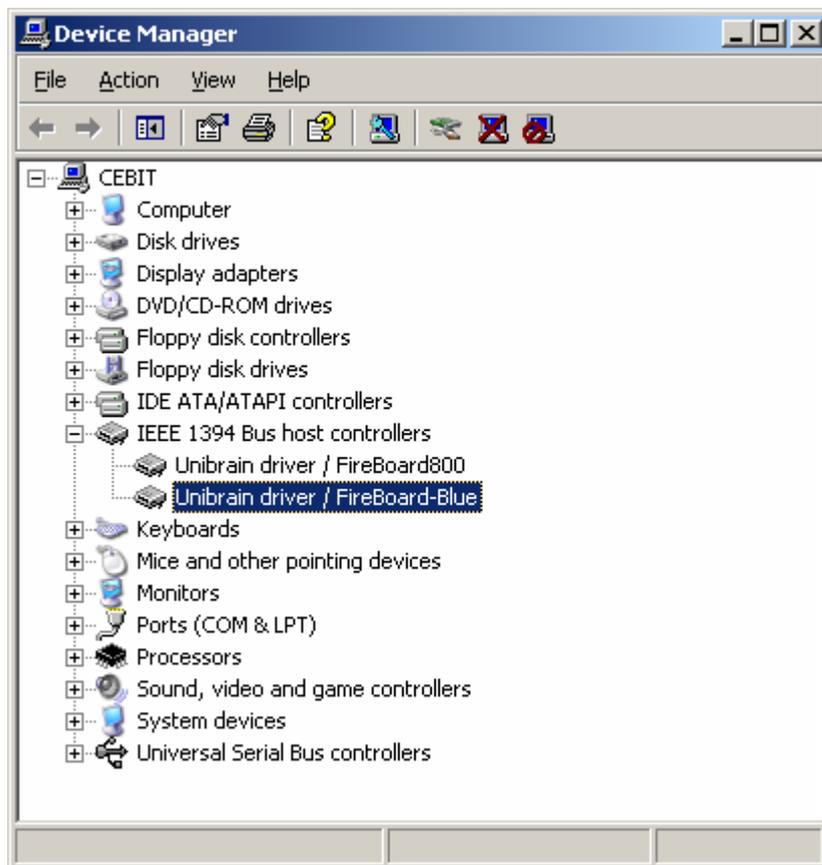
Coexistence of ubCore™ and Microsoft 1394 stack / Notes about DV camcorder users

ubCore™ is a set of drivers & utilities highly optimized for Firewire 800 devices. It also supports most of the 1394a (Firewire-400) devices but however, some devices may not be supported, like DV camcorders.

For this case, because ubCore™ drivers are Plug and Play, they can be very easily interchanged with the Microsoft 1394 drivers. So, for non-supported devices, the user can switch temporarily to the Microsoft drivers and revert back to the ubCore™ drivers at any moment.

The procedure below describes how to switch from the Unibrain ubCore™ drivers to the Microsoft Windows 1394 drivers. The opposite procedure is exactly the same.

Lets assume that your system has two 1394 PCI devices; one FireBoard800™ (Unibrain's 1394b, 64 bit PCI adaptor) and one FireBoard-Blue™ (Unibrain's 1394a, 32 bit PCI adaptor) and that you have successfully installed ubCore™ in your system. So if you see the Device Manager you should observe a system configuration like the one below:

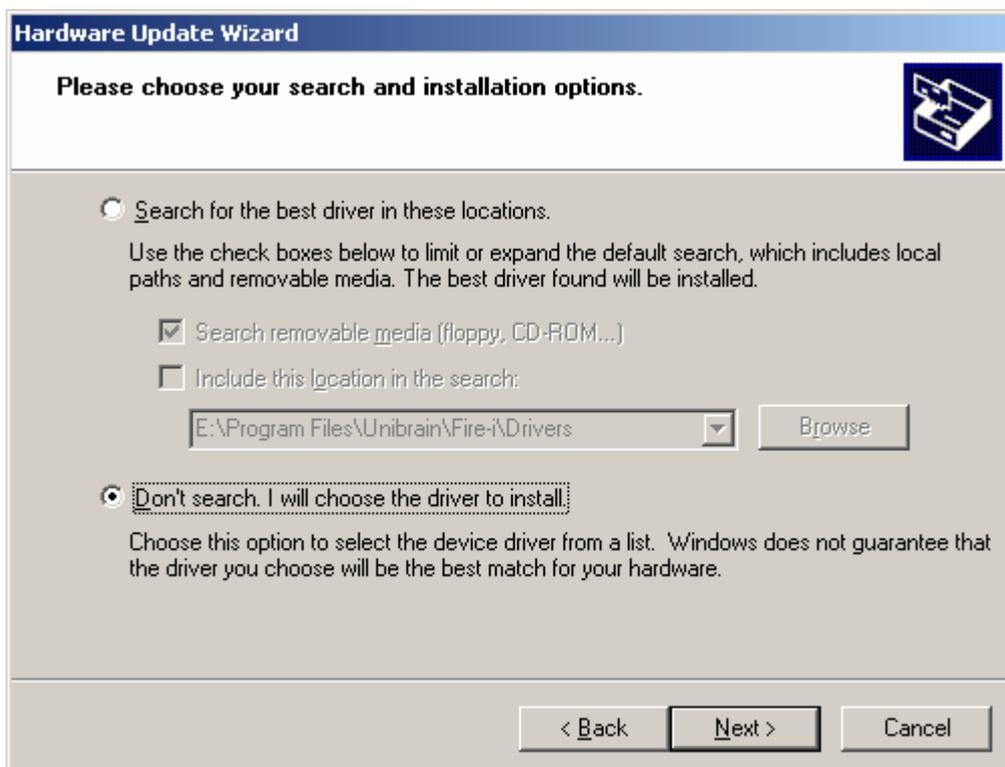


In order to change the driver for FireBoard-Blue™ from Unibrain's ubCore™ to Microsoft 1394 stack drivers, select the "Unibrain driver/ FireBoard-Blue™", right click your mouse button and from the properties select "Update Driver...". The Hardware Update Wizard will appear, as seen in the next picture:

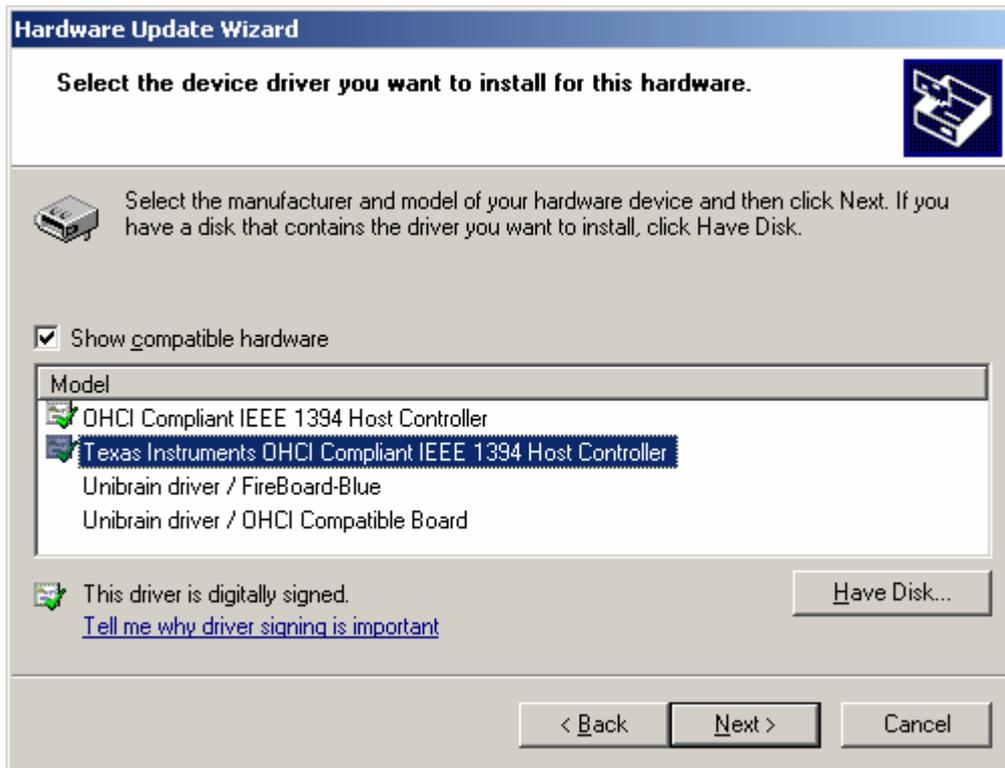


Select the “Install from a list of specific location (Advanced)” radio button and click Next

In the next window, select the “Don’t search, I will choose the driver to install” option:



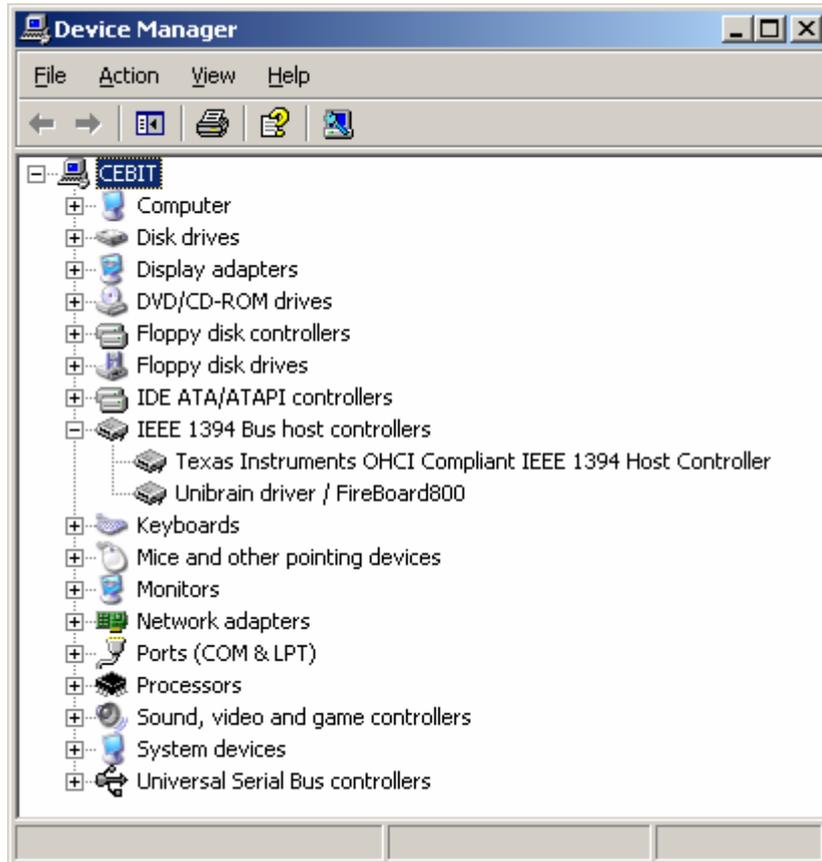
Select the appropriate driver (in our case is “Texas Instruments OHCI Compliant IEEE 1394 Host Controller”) and click Next



This is the completion stage of your hardware update, so please click Finish in the next dialogue.



Following the above procedure you have replaced the ubCore™ 1394 driver with the Microsoft 1394 stack and the Device Manager should appear like the below picture.



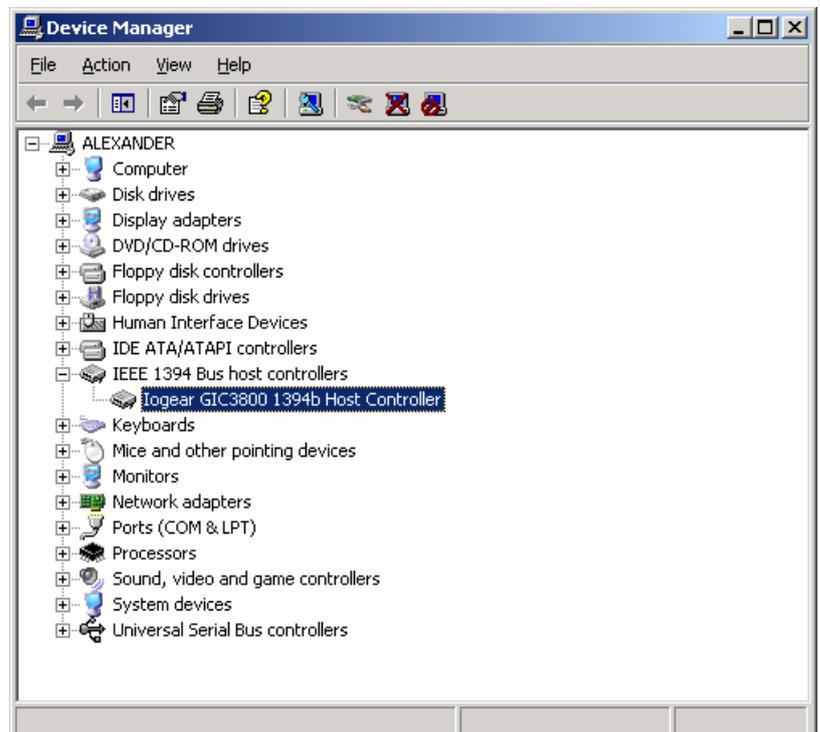
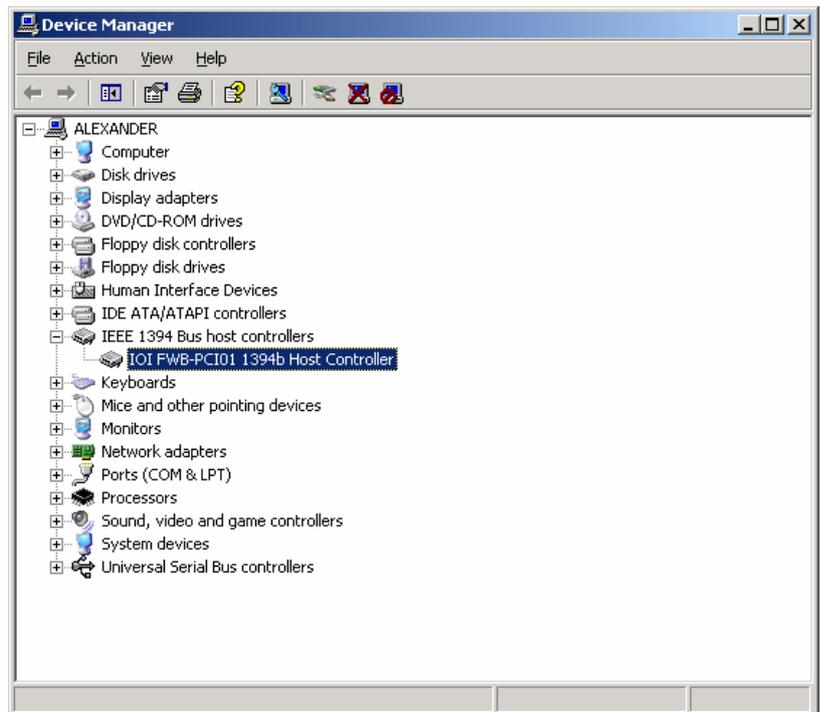
Please notice that for FireBoard800™ we kept the same driver based on Unibrain's ubCore™ because **it is the only existing driver providing full 1394b capabilities!**

Please also recall that with a similar to the above procedure you can switch from Microsoft 1394 stack, back to ubCore™.

Third party 1394b PCI controllers working with ubCore™

As is already mentioned, ubCore™ 3.2 supports Unibrain's or any third party 1394a or 1394b OHCI PCI/PCMCIA compliant adapter or any Desktop/Portable PC with built-in 1394a or 1394b OHCI support.

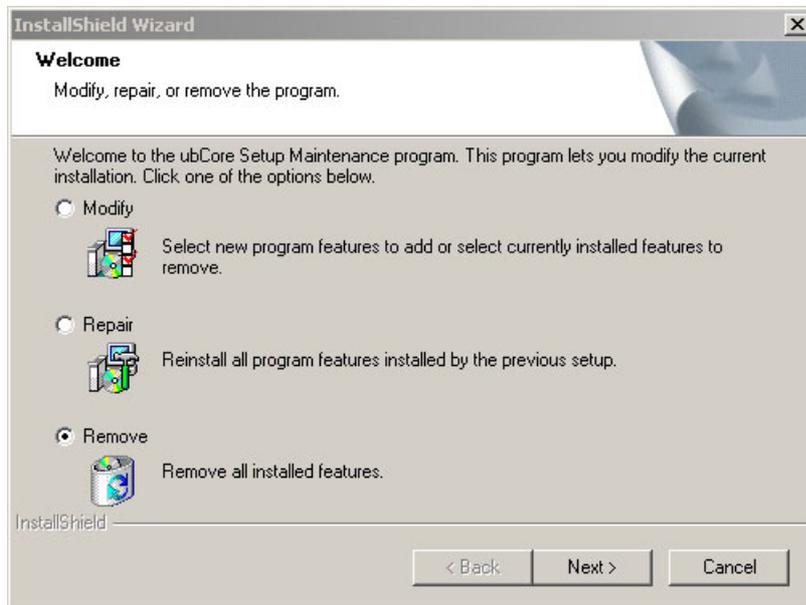
The two adjacent figures display how the ubCore™ 1394 stack recognizes and represents in the Device Manager two 1394b PCI controllers: the FWB-PCI01 board from IOI Technology Corporation and the GIC3800 from IOGEAR.



Uninstalling ubCore™

The procedure to uninstall ubCore™ is straightforward. Just select from “Start→Program Files→ubCore→Uninstall ubCore”

The following Dialogue box will appear. Select the Remove option and press “Next”.



Upgrading ubCore™

The correct procedure to upgrade your ubCore™ is to completely uninstall the old version, REBOOT your system and then follow the steps of the installation procedure in order to install the new version.

ubCore™ Authorization procedure

To enable an unrestricted use of the ubCore™ software, a unique "Name" and "Product Key" combination per PC station is required.

After obtaining a Product Key for each computer, run the "ubCore Authorize utility" application, which can be found in

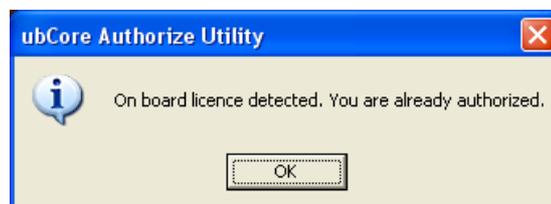
Start Menu → Program Files → ubCore → ubCore Authorize.

Enter your **Name** and the **Product Key** in the appropriate fields and press OK. The application will display a message indicating if the operation was successful or not.

In order to enable unlimited usage of the ubCore™ stack and all the accompanying applications (FireViewer™, FireCommander™, Fire-i™, FireShare™), you have to provide to the ubCore™ Authorize Utility a product key starting with the key word **CORE:**



If you have installed in your system a Unibrain board with the "ubCore 3.x Compliant" label, then you are authorized for unlimited usage of ubCore™ and you do not need to enter Authorization Product Key. If you do so, the next message will appear.



1394a and 1394b SBP2 disk driver

Unibrain's 1394 SBP2 driver is fine tuned for speeds of 800Mbps providing the fastest 1394b interface to external S800 SBP2 Hard disks. It is also compatible and fully supports removable Devices and other SBP2 compliant devices like Apple's iPod.

The current paragraph presents the functionality of the 1394a and 1394b SBP2 disk driver installed with ubCore™.

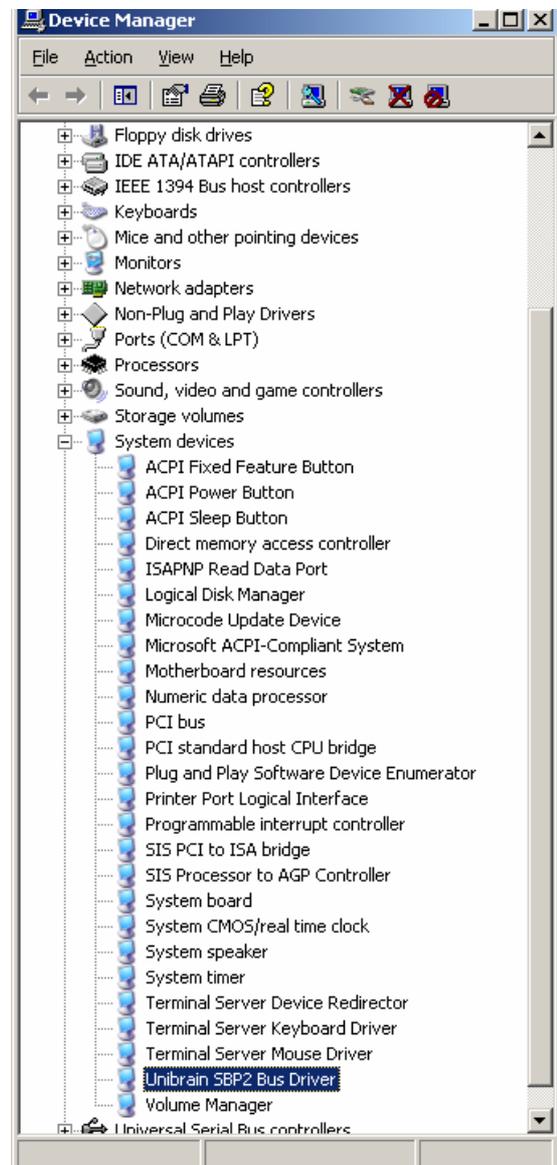
Please notice that this SBP2 driver is by default installed and ready to operate after the ubCore™ installation in Windows 2000 and Windows XP.

IN CASE THAT YOU HAVE Windows 98SE or Windows Me, YOU HAVE TO MANUALLY INSTALL THE SBP2 DISK AS DESCRIBED IN THE NEXT SECTION ENTITLED "Installing SBP2 disk driver in Windows 98SE/Me"

After the correct installation of ubCore™, you can observe the SBP2 driver from the Device Manager under the System devices, enabling the option "Show Hidden Devices".

Please see in this figure the "Unibrain SBP2 Bus Driver" under the System devices.

It is important to emphasize that this driver is installed and displayed even if you have not plugged the external hard disk!



Note: The SBP2 driver does not support the Windows NT 4.0 Operating System.

Installing SBP2 disk driver in Windows 98SE/Me

The following paragraphs describe the steps in order to install the ubCore™ SBP2 disk driver in Windows 98SE/Me. Please notice that for Windows 2K/XP the installation is performed automatically, once installed ubCore™.

- Start the Add/Remove Hardware Wizard from Start→Control Panel



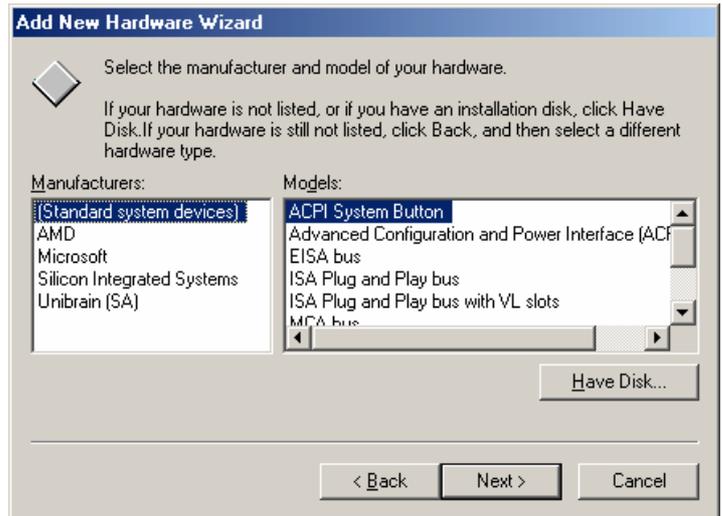
- In the adjacent dialogue box, select “No, I want to select the hardware from a list” click Next



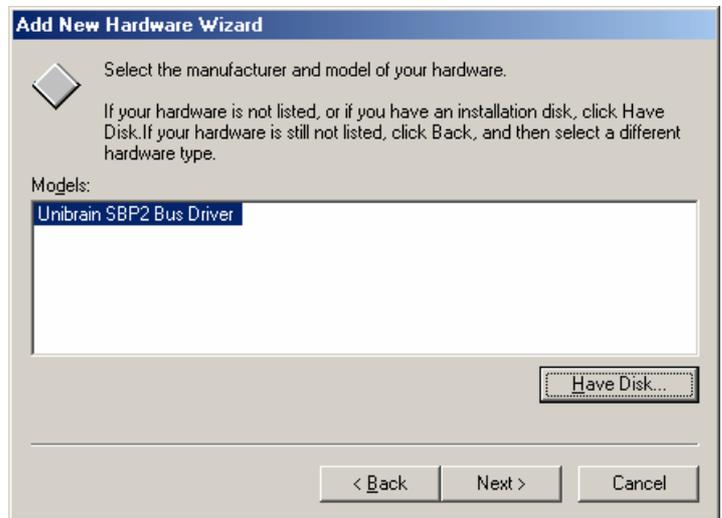
- Select from the Hardware types list the “System devices” and click Next



- In the next dialogue click “Have Disk” and locate the directory in which ubCore™ and the SBP2 driver has been placed

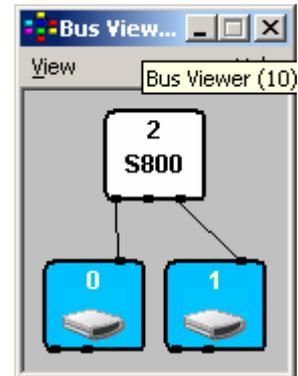


- Once locating the driver, the next dialogue will appear. Click Next and the installation will be completed.

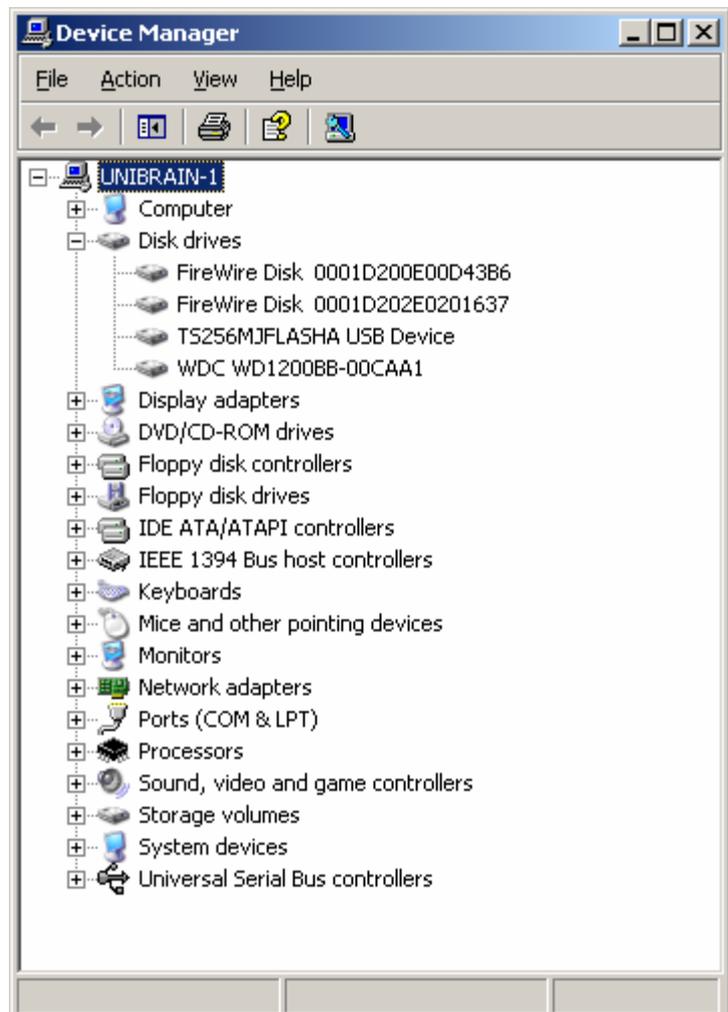


1394 SBP2 external disk connection

Lets assume that you have a FireBoard800™ in your system and you connect two (2) powered-on external 1394b Hard Disks. You can use the FireViewer™ utility to observe (see adjacent picture) the topology with FireBoard800™ in node 2 and the two external hard disks in nodes 0 and 1 respectively; notice that the default color for a 1394b node is blue

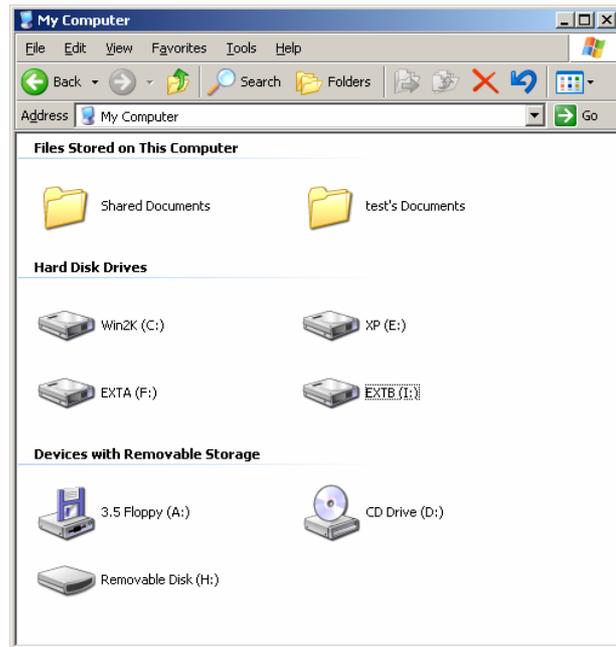


A few seconds after the connection of the two external hard disks, you can see from the Device Manager under the “Disk drives” section, that both disks have been recognized by the operating system and are ready to use.



In case that both disk are FORMATTED in a valid Windows file system format (e.g. NTFS) then you are able to explore and fully control the hard disks as valid volumes. In our example depicted in the adjacent picture, the two external hard disks have the volumes:

- EXTA mapped in volume F: and
- EXTB mapped in volume I:

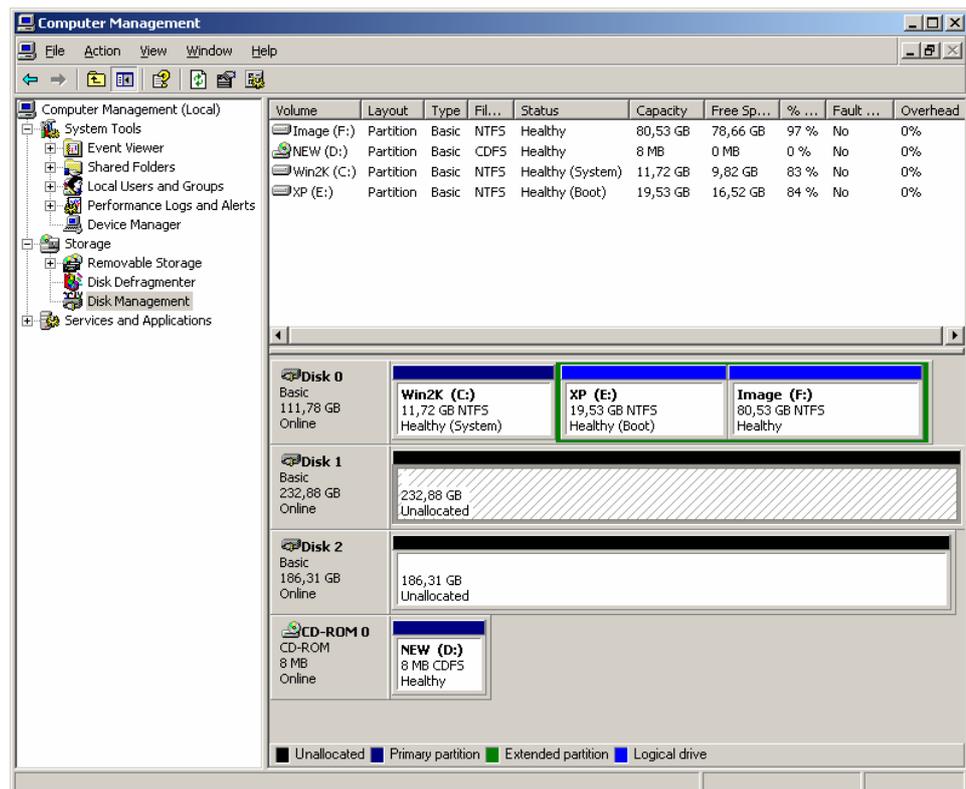


In case that your external hard disks are not FORMATTED, please follow the next set of instructions that apply for both Windows 2000 and Windows XP (our sample screen captures are taken from a Windows XP system):

- 1). Go to Start→Administrative Tools→Computer Management and double click the Disk Management option. If the SBP2 drive is plugged for the very first time, the “Initialize and Convert Disk Wizard” will appear. Click Next to continue.

The number of disks that will appear depends on how many you have connected to your system. Check mark (✓) the disks to initialize and click Next. In the next dialogue click Finish to complete the disk initialization part

2). In the Computer Management screens that will appear, you can see a list of the available disk drives. In our case there is Disk 0 with 3 different partitions, Disk 1, Disk 2 the external Unallocated 1394 SBP2 hard disk just plugged and the CD-ROM 0

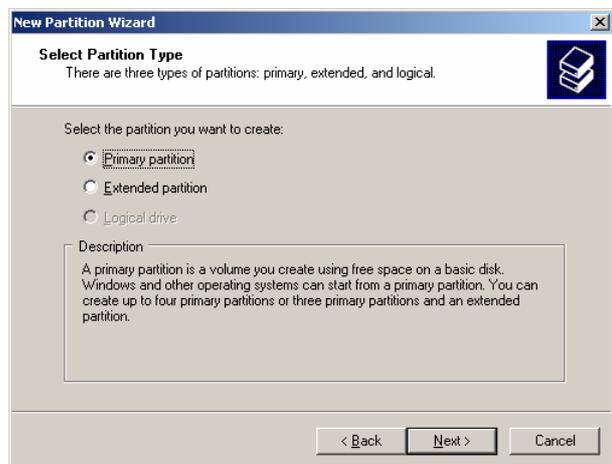


Right click on the Disk 1 that is unallocated and select the New Partition so that the New Partition Wizard will appear.



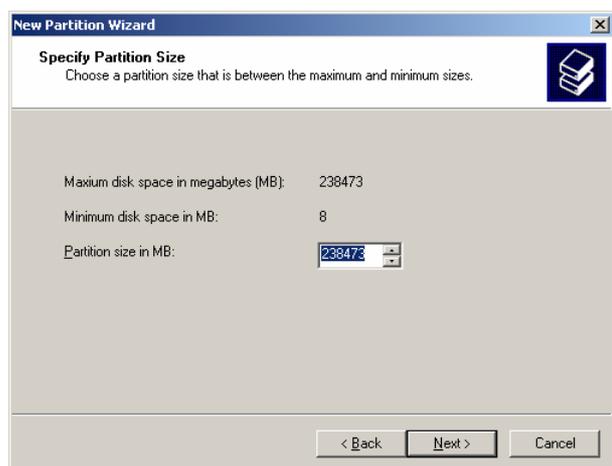
Select the partition type and click Next.

The default type is Primary partition and is the recommended type that you select. Use extended partition if there is a need to create more than four (4) partitions on a single drive



At the "Specify Partition Size" window type in Mbytes the number of the partition size and click **Next**.

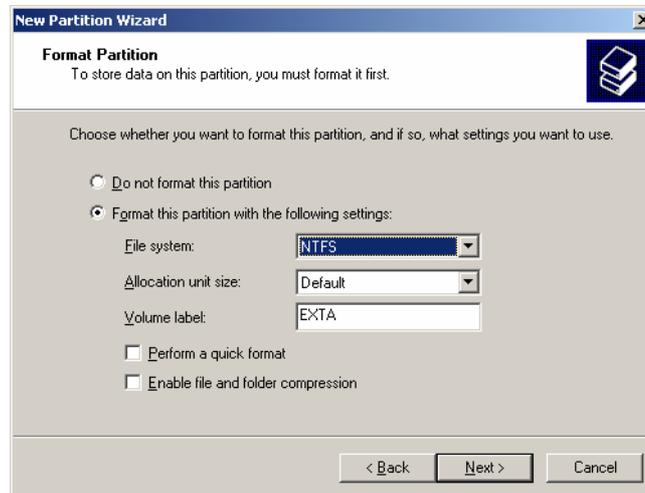
By default, Windows XP will select the total disk space available for your drive (single partition). If you choose to create multiple partitions, change the "Partition size in MB" and click **Next**.



At the next "Assign Drive Letter or Path" dialogue window, assign the drive letter and click **Next**.

By default, Windows XP will select the next available drive letter. Although, you can change the drive letter, it is recommended to leave the default setting.

In the next dialogue select the file system for the specific hard disk. Under “Format this partition with the following settings”, select FAT, FAT32 or NTFS as the File System. For “Allocation unit size” leave the default setting and give your drive a “Volume label”. Once completed, click **Next**.



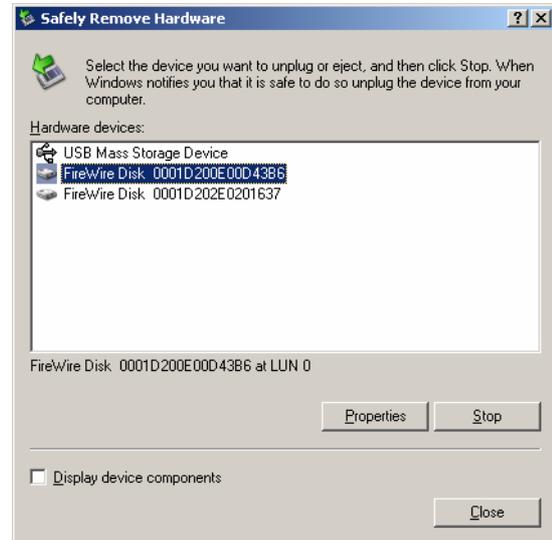
At the last window “Completing the Create Partition Wizard”, click Finish and you can start working with your external hard disk. Please repeat the above procedure for each of the external 1394 SBP2 drives you attach to your system for the first time.

Safe Removal of 1394 SBP2 disks

In order to safely remove you external 1394 SBP2 hard disk(s) please follow the same procedures as for any Plug n Play external device. More specifically:

Double click on the taskbar icon to “Unplug or Eject hardware”. The adjacent dialogue box will appear displaying the PnP external devices. In the current example you can see an USB Mass Storage Device and the two (2) external 1394b hard disks recognized by Unibrain’s SBP2 driver.

In order to remove a hard disk, click the device in the list and then click Stop. The operating system will automatically unplug the device and inform you for this safe removal. If you are using the specific hard disk (e.g. you have open a Word document located in this disk) the operating system will notify you that is unable to remove the device. In this case, close all the applications affecting the hard disk and try again.



FireViewer™-1394 Bus Viewer

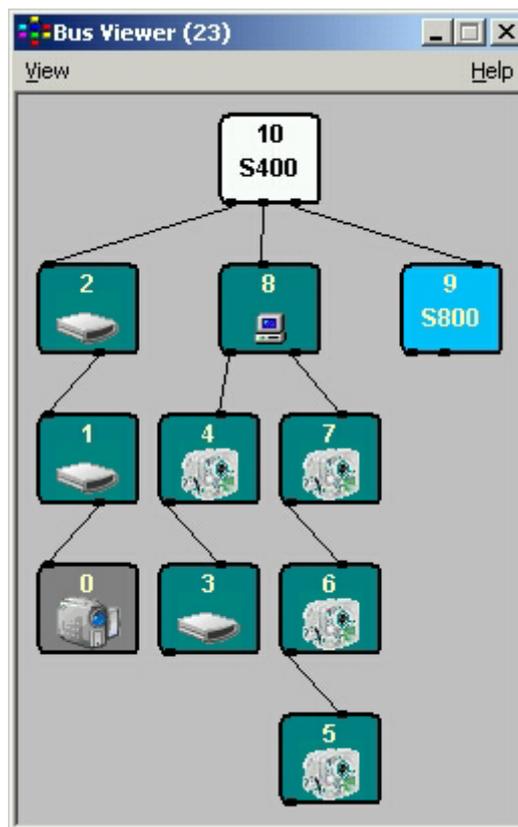
Unibrain's 1394 Bus Viewer utility (FireViewer™) shows the current topology of the local 1394 bus, allowing the user to continuously monitor the bus for changes in the topology and also to retrieve various information for each node.

The section below describes the usage of the FireViewer™ and the information provided by it.

The Node Tree

The bus is displayed as a hierarchical tree in the main application window, with the root node at the top, and the children of each node appearing on the 'row' below the 'row' where the parent is drawn. Inside each node, the program displays the node's physical ID and the node's speed or an icon that represents the type of the node.

In the picture below, a sample bus topology is illustrated. The root node is node 10 equipped with 1394 controller providing speed up to 400Mbps (S400). Node 10 is connected to nodes 2, 8 and 9, which are its children. Nodes 4 and 7 are the children of node 8, etc. Node 9 is a 1394b node with speed up to S800.



If a node is a PC with FireNet™, an IIDC 1394 camera, a Camcorder with 1394 interface, or external SBP2 1394 hard disk, descriptive icons are shown instead of the speed code. The icons are:

Icon	Meaning
	The node is a PC running FireNet™. Moving the cursor over the node will display a ToolTip window containing the computer's UNC name.
	The node is an IIDC 1394 digital camera
	The node is a Camcorder with 1394 interface
	The node is an external SBP2 1394 hard disk

If a computer with FireNet™ contains two or more adapters connected on the same bus, FireNet™ will only be running on one of them, which will be drawn with the FireNet icon. The rest will contain only a speed code.

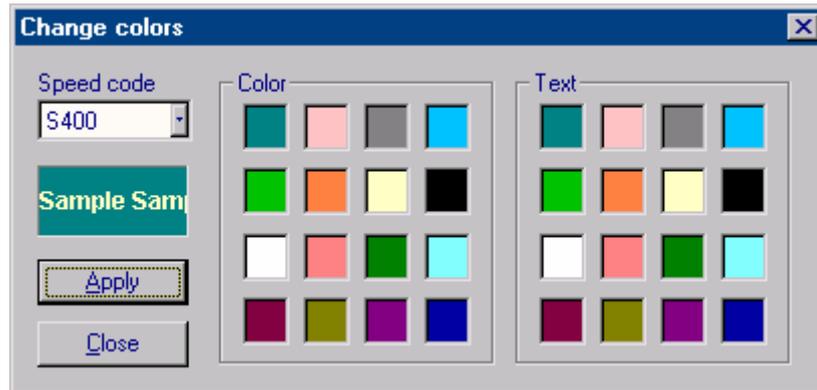
Keep in mind that the NodeID assignment of 1394 is dynamic, and can change after each bus reset. Don't be surprised if the tree changes shape while you work!

Color and Pattern coding

Depending on their speed, nodes are drawn in different colors. All colors will be displayed solid if your graphics adapter supports 256 colors or more. If your graphics mode is set to 16 colors, then the colors will be displayed dithered.

The node on which the program is running (the local node) is always drawn in white with black letters. This way you can easily identify the node ID of the PC you are working on.

The colors for all the other nodes are customizable through the Color dialog box:



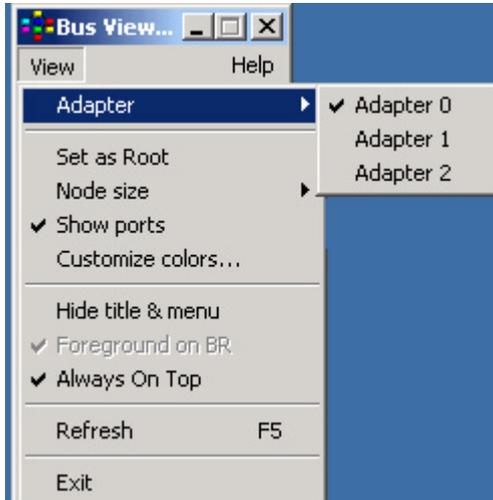
In order to change the color and text color for a speed code:

- 1) Select the speed code from the combo box titled "Speed code"
- 2) Left click on the desired color from the swatch on the left. The "Sample" area changes color according to your selection.
- 3) Left click on the desired text color from the swatch on the right. The "Sample" area changes text color according to your selection.
- 4) When satisfied with the result, press the "Apply" button to save the change for the specific speed code. The main window is redrawn immediately to reflect the changes, which will be retained after you dismiss the dialog box with the "Close button".

Active 1394 nodes (i.e. whose configuration ROM can be read) are shown in solid color. Inactive nodes are drawn with a diagonal stripe. Active nodes are located every time a bus reset occurs, with random order, after a delay of approximately ½ sec. You can update the topology any time by selecting the **Refresh** command from the menu, or simply by pressing the F5 key, while the FireViewer™ window is active.

Adapter selection

The topology shown each time is the one for the current adapter. For multi-adapter systems, you can select the current adapter as in the following picture. Up to 10 adapters are supported but only the present ones are shown in the menu.



Window appearance

Node size

The main program window can draw the bus topology with two different node sizes. There are two choices for node size:

- **Small Nodes:** Only the node ID is shown inside the node. This makes the FireViewer™ window consume less space, making it easier to keep visible all the time.
- **Big Nodes:** The node contains its node ID and speed code or icon.

To switch between the two sizes select “**View | Node size | Small**” or “**View | Node size | Big**” from the menu.

Ports

Check the “**View | Show Port**” menu item to enable the display of port information. When this is enabled, then FireViewer™ shows the ports of each node, and the actual connections between the nodes.

This can be of significant help when you have more than 3-4 PCs on the bus. There comes a point, when you can't remember which nodes connect to each other, and you have to trace the cables around in order to find out where each cable leads to.

Ports are numbered left-to-right. The leftmost port is port 0, the middle one is port 1, and the rightmost port two. In devices with 1 or 2 ports, only one or two ports are displayed (unless the device reports differently as some one-port cameras do).

The port that is connected to the parent node is drawn at the top edge of the node's rectangle.

There is no line leading to an unconnected port. The FireViewer™ shows the first three ports of each node, whether they are present or not.

Select again the “**View | Show Port**” menu item to disable the display of port information.

Caption and menu bar

The window can be shown with or without a caption and menu bar. Double-click the left mouse button inside the window to switch between the two views, or select “**View | Hide title and menu**” and/or “**View | Show title and menu**” from the menu.

When the window does not have a caption and menu bar, you can move it by clicking with the left mouse button into the window and dragging it to the desired position. The “View” menu is accessed via the right mouse button.

When the caption is visible, it shows the number of bus resets that have happened during the machine’s uptime.

Always on top

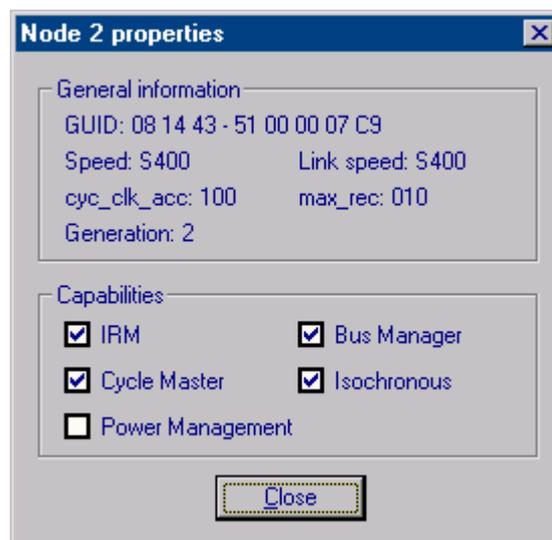
Check the “**View | Always on top**” menu item to make the window topmost. If this menu item is unchecked, you can check the “**View | Foreground on BR**” menu item to have the window come to the foreground whenever a bus reset takes place.

Node Properties

Right clicking on any active node will show a context menu that includes the **Properties** command.

Selecting this command will display a small dialog box with information about that node. The information varies according to the node type.

The next figure shows the contents of this dialog box which are common to all node types:



The first line displays the node’s Globally Unique Identifier (GUID). The rest information is retrieved from the node’s *bus_info_block*. Refer to the 1394 standard’s documentation for a detailed explanation of these items.

For FireNet nodes, the Properties dialog box looks like:



FireNet properties include the UNC name of the computer, its operating system, the FireNet version it is running, the emulated MAC address it uses and its MTU size.

For digital cameras, the Properties dialog box is as follows:



Extra information includes the camera vendor and model, as well as the specification number.

Bus anomalies

When an anomaly occurs on the bus, and a valid topology cannot be derived from the self-id packets, then the tree cannot be drawn accurately.

In this case the last known topology is drawn dimmed. The application tries to communicate with any active nodes on the bus, and draws in solid color the ones that are still active.

This situation might occur from time to time, especially when a device gets unplugged from the bus. It is usually corrected with a bus reset. The SBM driver checks for this condition and initiates a bus reset after a couple of seconds to correct the situation, so usually it won't be necessary to do this yourself. The communication between nodes is never halted due to a topology error, so you should not worry if this situation occurs. In the rare case when the problem persists, then there is some problem with some node on the bus. The only thing you might do in this case is disconnect a few devices from the bus until you restore a valid topology.

In extremely rare cases, the problem might persist. Then you might have to reboot your PC, or shut it down, turn it off, remove all 1394 cables from it and then power it up again. This is a problem that has only occurred during our development, in cases where a serious software failure or driver crash has left the adapter in a state from which it cannot operate even after a warm reboot, but requires a complete power off. This also means disconnecting the adapter from the 1394 bus, because the bus itself supplies power to the adapter even when the PC is turned off.

Menu

The menu bar of the window contains a “View” and “Help” menu. You can also access the “View” menu by right-clicking the mouse button inside the window.

View menu

- **Properties:** This item is available *only* when the “View” menu is opened by right-clicking the mouse on an active node. It opens a dialog box showing information about the node that is accessible via the configuration ROM.
- **Adapter:** Select from the popup menu that opens the adapter that will be the current.
- **Node size:** Select “Small” or “Big” to have the nodes drawn in small or big size, respectively.
- **Show ports:** Check or uncheck this item to enable or disable port drawing respectively.
- **Customize colors:** Opens the Color dialog box.
- **Hide title and menu:** Has the same effect as double-clicking the left mouse button in the window when the window has a caption and menu bar **-or-**
- **Show title and menu:** Has the same effect as double-clicking the left mouse button in the window when the window does not have a caption and menu bar.
- **Foreground on BR:** Active only when the window is not topmost. Check this menu item to have the window come to the foreground whenever a bus reset occurs.
- **Always on top:** Check this menu item to make the window topmost.
- **Refresh:** Select this menu item or press the F5 key to retrieve again the bus topology and redraw the window.
- **Exit:** Select this menu item to terminate the program.

Help menu

- **About:** Shows copyright information, including the current version number of FireViewer™.

FireCommander™ - 1394 Command Interpreter

The 1394 command interpreter is a tool that can be used to pass commands to the 1394 stack, for example sending transaction requests to various nodes and seeing the results, sending PHY packets, initiating bus resets etc. The 1394 command interpreter also has the ability to execute script files containing the commands that it recognizes. This capability can be used for the execution of elementary testing procedures.

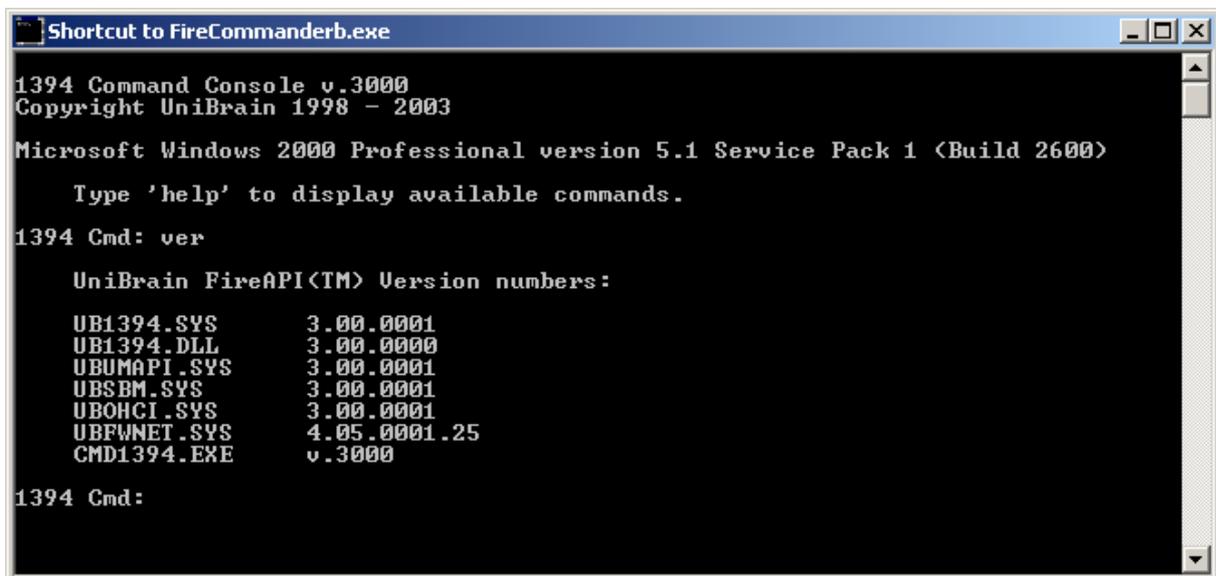
An important feature of FireCommander™ is the built in utility **bench** that performs instant benchmarking and performance measurement of the connection between two (2) 1394 PC nodes. Please refer to this utility description for more details.

The **help** command from within the interpreter provides the list of available commands and a summary of command syntax.

Running the program

Run the FireCommander™ from the “Start → Programs → ubCore”. It runs as a console process, by opening a text window similar to the system’s command prompt. In this window you can type the various supported commands, and then press Enter so that the interpreter processes them.

In the next screen capture you see the initial console information and the Unibrain’s 1394 driver stack version executing the **ver** command



```
Shortcut to FireCommanderb.exe
1394 Command Console v.3000
Copyright UniBrain 1998 - 2003
Microsoft Windows 2000 Professional version 5.1 Service Pack 1 (Build 2600)
Type 'help' to display available commands.
1394 Cmd: ver
UniBrain FireAPI(TM) Version numbers:
UB1394.SYS      3.00.0001
UB1394.DLL      3.00.0000
UBUMAPI.SYS     3.00.0001
UBSBM.SYS       3.00.0001
UBOHCI.SYS      3.00.0001
UBFWNET.SYS     4.05.0001.25
CMD1394.EXE     v.3000
1394 Cmd:
```

You can have open at the same time, as many instances of this program as you find suitable.

Note that FIRECOMMANDER by default will sound the computer beeper twice whenever a bus reset occurs on the local bus. You can turn off this sound with the `beep` command. If a bus reset storm is detected (too many bus resets in a very short interval) the program will sound an alarm that will immediately notify the user that there is a problem on the network. You can test this alarm to see what it sounds like by running the `alarm` command.

The output of FIRECOMMANDER is sometimes too big to fit in the traditional 80x25 command windows. It is suggested that you open the properties page of the command window (right-click on the window's title bar) and change the size of the window in order to be able to work more comfortably. Any kind of layout can be saved into the shortcut that started FIRECOMMANDER, and then be automatically reused each time you run it.

Command Syntax

- Type `help` to see the list of available commands.
- The commands are NOT case-sensitive.
- In commands where a Bus ID is required, the dot character (.) denotes the local 1394 bus, i.e. bus ID 1023.
- In commands where a Physical ID is required, the dot character (.) denotes the local adapter's physical ID.
- Physical IDs, Bus IDs and data byte quantities can be provided as both decimal (e.g. 1024) or hexadecimal (e.g. 0x400) numbers. Hex number must be prefixed with 0x. Offsets and Raw Data are always interpreted as hexadecimal values regardless of whether they are prefixed with "0x" or not.
- Known register names can also be provided instead of hex values. See the `regs` command for a list for the known registers. A set of well-known CSR offsets is hardwired into the program. Create a file called REGS1394.TXT if you need to specify additional names. FIRECOMMANDER always looks for this file in the directory that was current when the program was started.
- The program reads its input from *standard input* and writes its output to *standard output*. This means that you can use FIRECOMMANDER to execute a batch-file with FIRECOMMANDER commands by typing in the system's command prompt a command line like:

```
FIRECOMMANDER < COMMANDS.TXT
```

Make the last line of COMMAND.TXT contain the exit command so that FIRECOMMANDER terminates after completing the commands.

Since the program writes to *standard output* you can redirect its output to a file. Combining these two capabilities, you could use FIRECOMMANDER to execute commands found in a file and write the output to another file, by typing the following command line in the system's command prompt:

```
FIRECOMMANDER < COMMANDS.TXT > OUT.TXT
```

Available commands

ack [*ack_code*]*]

Prints the identifier for the acknowledge code provided as parameter. With * as parameter, it prints out a list of all `ack` codes. `ack` without a parameter is identical to `ack *`.

adapter [*adapter_no*]*]

Selects the current working adapter. In a situation where multiple adapters are installed on the host, the user should select the adapter s/he wants to work with. The default adapter is adapter 0.

Enter adapter * for a list of available adapters.

Enter adapter <number> to set the current adapter.

beep

Toggles the beeper indication for bus resets.

bench *busID phyID number_of_requests packets_per_request async_level*

Performs benchmark tests by sending directly data from memory between two nodes. This test sends for the specified *phyID* in the *busID* a *number_of_requests* with *packets_per_request*. The *async_level* parameter specifies the number of consecutive pending writes without acknowledge and has maximum value of 10 that gives the best performance. The packet size of each transaction is variable and starts from 512 bytes per packet, 1024 bytes per packet, 2048 bytes per packet till 4096 bytes per packet for 1394b transactions.

Example: **bench . 2 5000 50 10**

Start sending data from my 1394 node to the node 2 in the current bus, with 5000 requests of 50 packets per request (total of 5000*50=250000 packets) doing 10 consecutive writes without acknowledge for maximum performance.

IMPORTANT NOTE: As is implied from the above bench command description, this utility requires the connection of two (2) PCs equipped with 1394 interface. **In order to correctly monitor the bus performance please run FireCommander™ in both PCs!**

In the next figure you see the snap-shot of the receiver in a configuration case of two PCs equipped in 32bit PCI slots with FireBoard800™ (Unibrain's 1394b to PCI adapter) at S800 transfer. Please observe the sequence of packet sizes from 512bytes up to 4K bytes and the respective augmentation of the bus bandwidth.

```

Shortcut to FireCommanderb
1394 Cmd:
Packet Size      : 512
Packets Transmitted : 150000
Packets Received  : 150000
Bytes Received    : 76800000
Elapsed Time     : 2562
Corrupt packets   : 0
Transfer Speed    : 28.59 MB/sec

1394 Cmd:
Packet Size      : 1024
Packets Transmitted : 150000
Packets Received  : 150000
Bytes Received    : 153600000
Elapsed Time     : 2735
Corrupt packets   : 0
Transfer Speed    : 53.56 MB/sec

1394 Cmd:
Packet Size      : 2048
Packets Transmitted : 150000
Packets Received  : 150000
Bytes Received    : 307200000
Elapsed Time     : 4187
Corrupt packets   : 0
Transfer Speed    : 69.97 MB/sec

1394 Cmd:
Packet Size      : 4096
Packets Transmitted : 150000
Packets Received  : 150000
Bytes Received    : 614400000
Elapsed Time     : 6953
Corrupt packets   : 0
Transfer Speed    : 84.27 MB/sec

1394 Cmd:

```

br | busreset | reset [long|short]

Initiates a bus reset on the local bus. Enter the parameter long or short to specify the type of the bus reset.

`br` and `busreset` display the new physical ID that the current adapter obtained after the bus reset. `Reset` does not display it so that it is more convenient to use in automated command scripts.

NOTE: Adapters that comply only with IEEE 1394-1995, but not with P1394A, do not support short bus resets. In such a case, the request will fail with a status code of `STATUS_1394_INVALID_REQUEST`.

brcount

Displays the bus-reset count for the current adapter. This can be used to verify the number of bus resets that a test operation caused.

cls | c

Clears the contents of the command window.

delay msec

Creates a delay for the given number of msec. Useful when a sequence of commands is executed from a file (see command file) and an artificial delay period must be created.

dir, cd, del, copy, a:, b:, ..., z:

These commands are passed directly to the system command shell (cmd.exe or other). This way you can run these common commands without leaving the 1394 command interpreter window.

file infile [outfile]

Reads and executes a series of commands from the file specified by `infile`. If `outfile` is provided as well, then the output of all commands that will be executed will be redirected to that file. Otherwise the output will be displayed on screen.

file commands cannot be nested; a script file with 1394 commands cannot contain another file command.

A line in a script file that starts with either ‘;’ or ‘#’ is considered as a comment and is ignored by the interpreter.

gap [gap_count]

Without an argument, the command prints the `gap_count` value of the current adapter. With a number between 1 and 63 as argument, the command transmits a PHY configuration packet that sets the `gap_count` of all the stations on the local bus to this number.

Remember that selfID packets contain the value of a node’s `gap_count` at the time of the bus reset. If you run the `gap` command and then the `topology` command without an intervening bus reset, you will still see the old values in the selfID packets.

However a PHY configuration packet is immediately effective. If you run `gap 31` on node A, and then run `gap` on node B, you will see that the `gap_count` of node B has become 31.

The 1394 protocol mandates that after two bus resets the `gap_count` of a node is reset to 63.

Note that the Serial Bus Manager driver (UBSBM.SYS) by default is trying to optimize the `gap_count` after a bus reset. Remember this if you ever try to experiment with changing the value of `gap_count`. You can disable this behavior of UBSBM using the appropriate registry entries.

Specifically:

Path: \HKEY_LOCAL_MACHINE\SYSTEM\CurrentControlSet\Services\UBSBM\Parameters\NoGapCount
Type: REG_DWORD
Default Value: 0
Description: If non-zero, it turns off the automatic `gap_count` optimization that UBSBM normally performs.

Path: \HKEY_LOCAL_MACHINE\SYSTEM\CurrentControlSet\Services\UBSBM\Parameters\GapCount
Type: REG_DWORD
Default Value: 0
Description: If non-zero, it turns off the automatic `gap_count` optimization that UBSBM normally performs.

If these values are not present then the default values are assumed.

help | h

Prints a list of the available commands.

read | *r busID phyID offset bytes*

Sends a read transaction request of size bytes starting from offset *offset* to the node with physical ID *phyID* on the bus with ID *busID*.

If the read is successful, then the returned data are displayed in hex and ASCII format. Otherwise the error response code or acknowledge code is displayed.

write | *w busID phyID offset bytes data*

Sends a write transaction request of size bytes starting from offset *offset* with contents *data* to the node with physical ID *phyID* on the bus with ID *busID*.

The command displays the binary data that it will send so that you can verify what was exactly sent. If fewer data are provided on the command line than what is specified in the bytes parameter, then the data are padded with zero at the end.

forceroot | *fr phyID*

Sends a PHY configuration packet that sets the *force_root* bit of the node with physical ID equal to *phyID*.

rom *busID phyID*

Reads the *Bus_Info_Block* from a node's Configuration ROM, and presents it in the appropriate format.

fwrite *busID phyID offset filename fileoffset bytes*

Sends a write transaction request of size bytes starting from offset *offset* to the node with physical ID *phyID* on the bus with ID *busID*. The data to be written are read from file with name *filename*, starting from offset *fileoffset* and have a size of *bytes*.

In all other respects this command is similar to the write command.

fifo [*RcvSize AsynchXmitSize IsochXmitSize*]

When executed without any parameters it displays the current FIFO settings of the current adapter.

When executed with parameters it changes the current FIFO configuration of the adapter.

For example typing *FIFO* in the command prompt of FIRECOMMANDER would display the following output (with a PCILynx2-based board):

```

-----
FIFO settings are dynamically changeable.
FIFO auto-zero IsoXmit is SUPPORTED.
FIFO auto-zero IsoXmit is ACTIVE.
IsoXmit FIFO replacement policy is FIFO_FAVOUR_TRANSMIT.

Receive FIFO      : Setup(2048) Current(2048).
Asynch Xmit FIFO: Setup(1024) Current(2048).
Isoch Xmit FIFO  : Setup(1024) Current(0).
-----

```

This means that the 1394 stack is setup to use 2KB for receive (both asynchronous and isochronous), 1KB for asynchronous transmit, and 1KB for isochronous transmit.

However, when there is no isochronous transmit activity on the system, then the drivers will favor the asynchronous transmit FIFO rather than the receive FIFO, and add the *setup* size of the isochronous transmit FIFO to the asynchronous transmit FIFO.

This yields the *current* values which in the above example are 2KB for receive and 2KB for asynchronous transmit.

When the FIFO command is run with parameters, it updates the *setup* values. For example, if someone wanted to perform extensive isochronous receive operations, very small asynchronous transmit operations, and no isochronous transmits, and wanted to configure the FIFO with 3.5KB for receive and 512 bytes for asynchronous transmit, then it would have to type the following command:

```
FIFO 3584 256 256
```

This would make the *setup* size of isochronous transmit FIFO equal to 256 bytes, and since there would be no isochronous transmit activity, this value would be added to the current value of asynchronous transmit.

lock | *I busID phyID offset function [32|64] data_value arg_value*

Sends a lock transaction request to the node with physical ID `phyID` on bus `busID`.

The available lock functions are:

```

FETCH_ADD
LITTLE_ADD
MASK_SWAP
COMPARE_SWAP
BOUNDED_ADD
WRAP_ADD.

```

Enter 32 or 64 after the lock-function name to choose between a 32-bit or 64-bit operation respectively. `data_value` specifies the data value for the function, and `arg_value` specifies the argument value for the function (if any). Note that the data value appears first on the command line, as some functions do not use an argument value.

For example if an application wanted to allocate channel 0 from the CHANNELS_AVAILABLE register of node 0, supposing that all channel numbers are zeroed, then it would type the following command line:

```
lock . 0 CHANNELS_AVAILABLE compare_swap 32 7FFFFFFF FFFFFFFF
```

The screen output of FIRECOMMANDER would be:

```
----- LOCK OPERATION -----
Destination(0X3FF,0)
Offset(0XFFFFFF0000224)
Function compare_swap (extended_tcode 2)
Data Bytes(4)
32-bit Data Value(0X7FFFFFFF)
Arg Bytes(4)
32-bit Arg Value(0XFFFFFFF)

COMPARE_SWAP SUCCEEDED (old_value == arg_value). Old Value: 0XFFFFFFF
```

If channel 0 was allocated then the compare-swap would fail and the last line of the screen output would be:

```
COMPARE_SWAP FAILED (old_value != arg_value). Old Value: 0X7FFFFFFF
```

dump busID phyID BaseOffset TotalBytes PacketSize filename

Sends a series of read requests to the specified node and writes the returned data in the specified filename.

Each request reads *PacketSize* bytes.

The Nth request, N=0..(TotalBytes+PacketSize-1)/PacketSize, is sent at offset BaseOffset+N*PacketSize.

resp [resp_code]*

Prints the identifier for the response code provided as parameter. With * as parameter, it prints out a list of all response codes. `resp` without a parameter is identical to `resp *`.

regs

Prints a list of the known register names.

selfid

Displays in a useful format the self ID packets that the 1394 stack received when the last bus reset was completed.

This command is especially useful when some 1394 adapter/device is causing trouble on the bus by transmitting invalid/corrupt self-ID packets. In that case, the topology command cannot display any information, nor are the self-ID packets available through the TOPOLOGY_MAP registers.

shell | sh command

Passes up command to the system's command shell for execution. This allows you to issue any command from within FIRECOMMANDER, without having to switch to another application.

speed

Displays the path speed table on the local bus. This table is displayed in a tabular form. Table entry [M, K] contains the speed of the path between the nodes with physical ID M and K. If K is 63, then the result is the broadcast speed, which is the speed of the slowest device on the bus.

- A dot denotes that the pair does not physically exist.
- 1 represents the transmission rate of S100.
- 2 represents the transmission rate of S200.
- 4 represents the transmission rate of S400.
- 8 represents the transmission rate of S800.

In order to save screen space, all rows that correspond to nodes that are not physically present are not being displayed.

status [*status_code*]*]

Prints the identifier for the status code provided as parameter. With * as parameter, it prints out a list of all status codes. status without a parameter is identical to status *.

tcode [*transaction_code*]*]

Prints the identifier for the transaction code provided as parameter. With * as parameter, it prints out a list of all transaction codes. tcode without a parameter is identical to tcode *.

topology

Prints physical & topology information for the local 1394 bus. This information contains details about the physical characteristics of each node on the bus, information about the bus topology (which node is connected to whom, how many children each node has, who is its parent node etc).

ver

Prints the version numbers of Unibrain's FireAPI drivers installed on the system.

quit | **q** | **exit** | **x**

Quits the program.

FireNet™-1394 networking

The FireNet™ software is THE FIRST AND ONLY FireWire™ Networking Solution supporting 1394b on all Windows OSs (including Windows NT4)

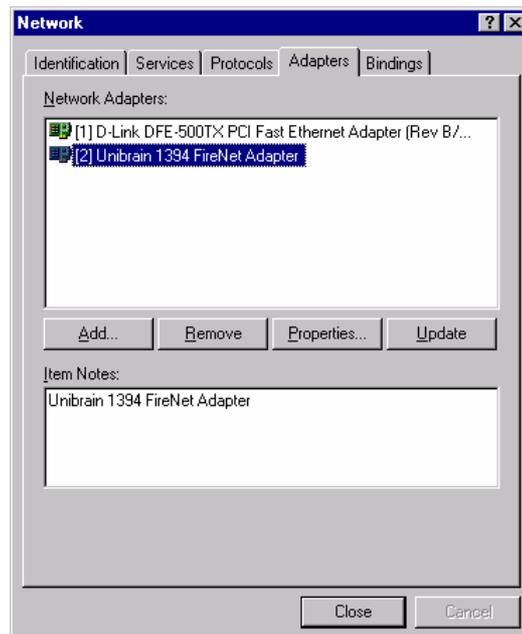


FireNet™ software, one more application implemented over ubCore™, is a system-independent software product that gives you full Ethernet compatibility at speeds of up to 800Mbps. FireNet™ is not installed automatically during ubCore™ installation, but only the required driver is copied in the corresponding directory (Please also refer to the paragraph entitled “Directory structure of ubCore™ installation”). So you have to manually install the FireNet™ software according to the following instructions.

Important! Always UNPLUG the connecting IEEE-1394 cables between all network computers before you begin installing or uninstalling FireNet software.

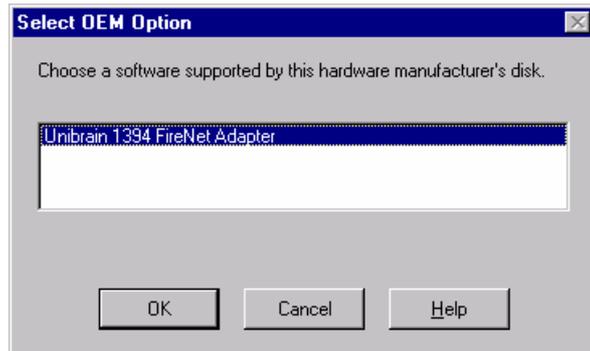
Installation on Windows NT4

The FireNet™ software can be installed as any other Network adapter. In your Control Panel double click the Network applet, select the 'Adapter' tab and press the 'Add...' button.



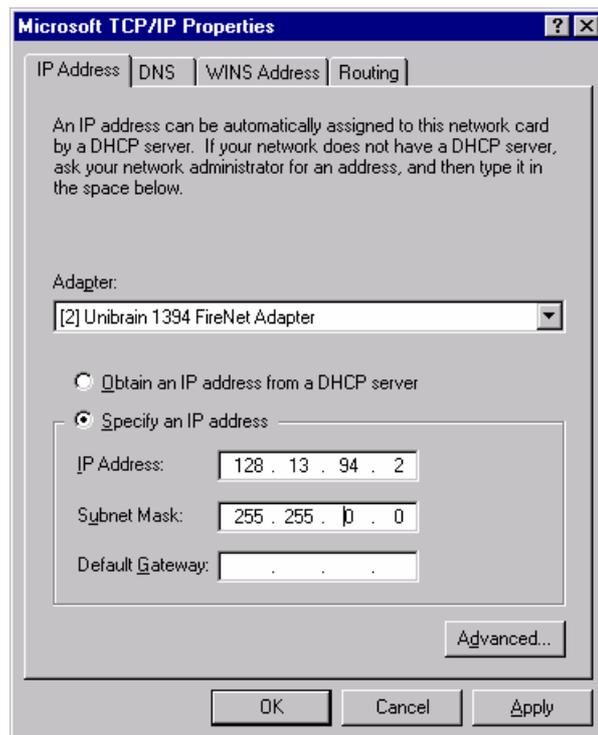
Press the 'Have Disk' button and enter the directory in which the FireNet™ software is located in, for example: “E:\Program Files\Unibrain\ubCore\FireNet\” and click on “Next”.

The list with available adapters should contain the Unibrain FireNet™ item, select it and press OK.



After that Windows NT will try to bind any selected protocols to Unibrain FireNet™ adapter. Please enter any requested information (for example TCP/IP address etc.).

Then Windows NT requires restarting your PC in order to load the new adapter. Select to restart your PC either at this point or later. Your new network adapter will be available after your PC restarts.



Unibrain's FireNet™ adapter can be treated as any other network adapter since it emulates a typical Ethernet adapter. To enable the configuration dialog of the FireNet™ driver please select the properties button from the "Control Panel->Network->Adapters".



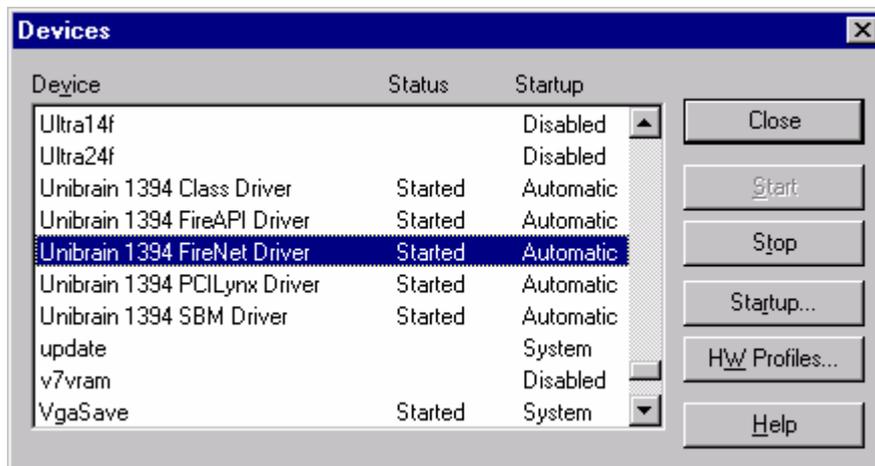
The properties dialogue button provides the following configuration:

- **Speed:** If you are using a 400Mbps board you can adjust the speed for a 100 or 200 or 400Mbps operation. If you have a 800Mbps board (like the Unibrain's FireBoard800™) then you can only adjust it for up to 800Mbps operation. The Default value is Auto.
- **Emulation:** You can choose between Ethernet and IP/1394. The Ethernet emulation will let you use any protocol (TCP/IP, NetBEUI, IPX/SPX etc.). The IP/1394 is build according to the specs set from the IETF Engineering Taskforce. Because this standard is not yet finalized this option is not available yet.
- **Maximum Payload:** This is for advanced users. In case you experience problems with some boards change the automatic setting to "1024". The Default value is automatic.

Verify Installation

In order to verify that you successfully performed the complete FireNet™ installation under Windows NT 4.0, please do the following just after rebooting your system:

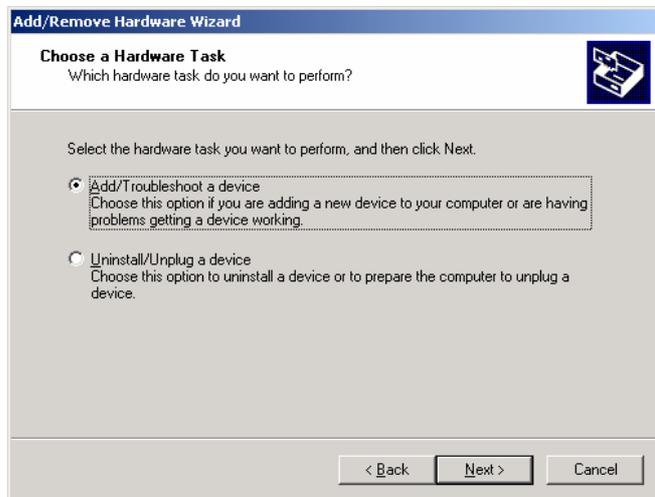
Click on the "Control Panel" - "Devices": You should see the next dialog box



In addition to ubCore™ drivers already installed, the FireNet Driver should have also started automatically.

Installation on Windows 98SE/Me/2000/XP

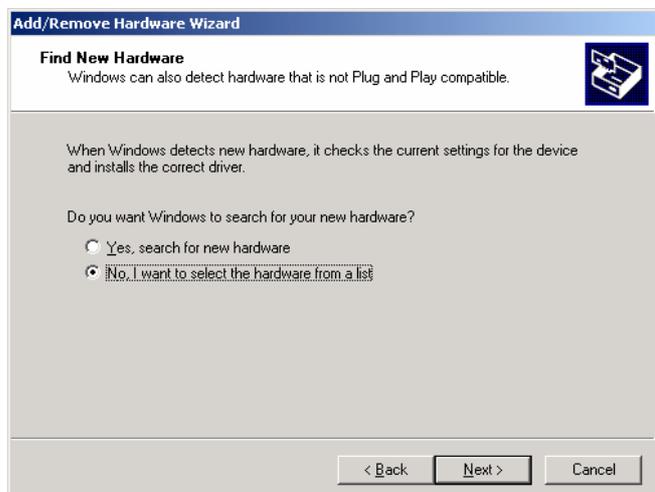
Start the Add/Remove Hardware Wizard from Start→Control Panel and select the Add/Troubleshoot option.



Select the "Add new device" option ...



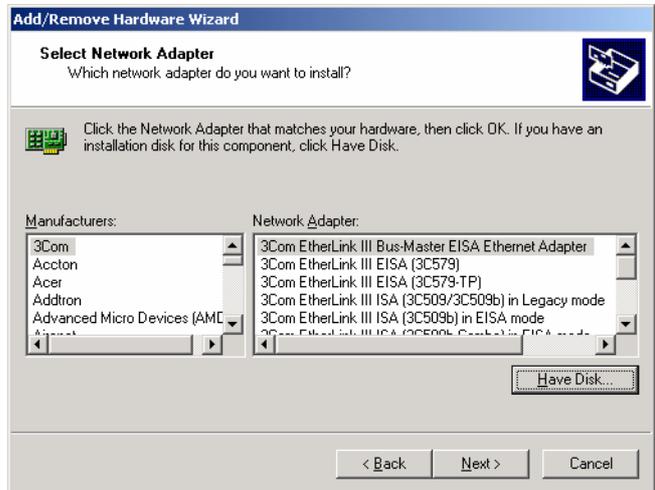
...and select manually the new device.



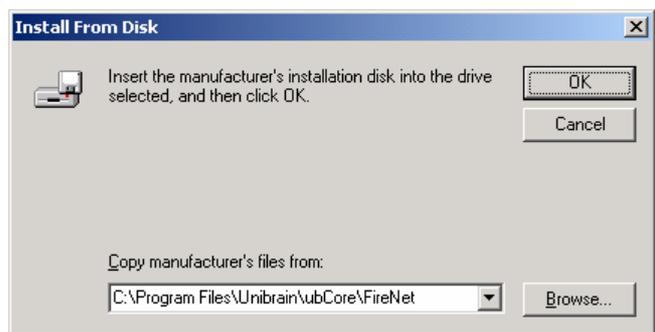
In the type list of devices select the “Network adapters” and click Next.



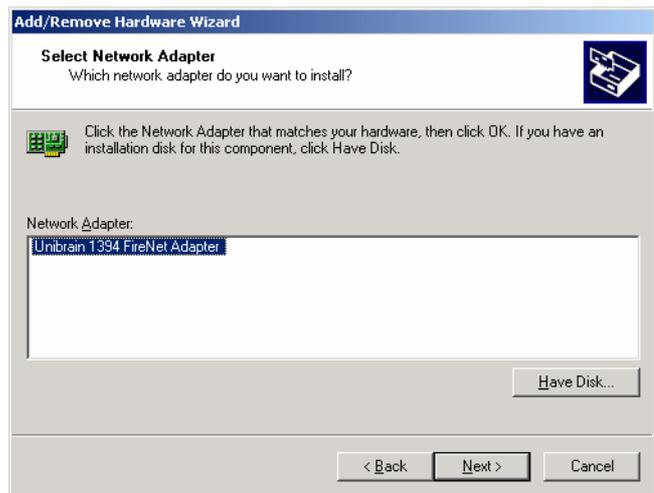
In the next dialogue click “Have Disk” in order to locate the path for Unibrain’s FireNet™.



Once located click OK...

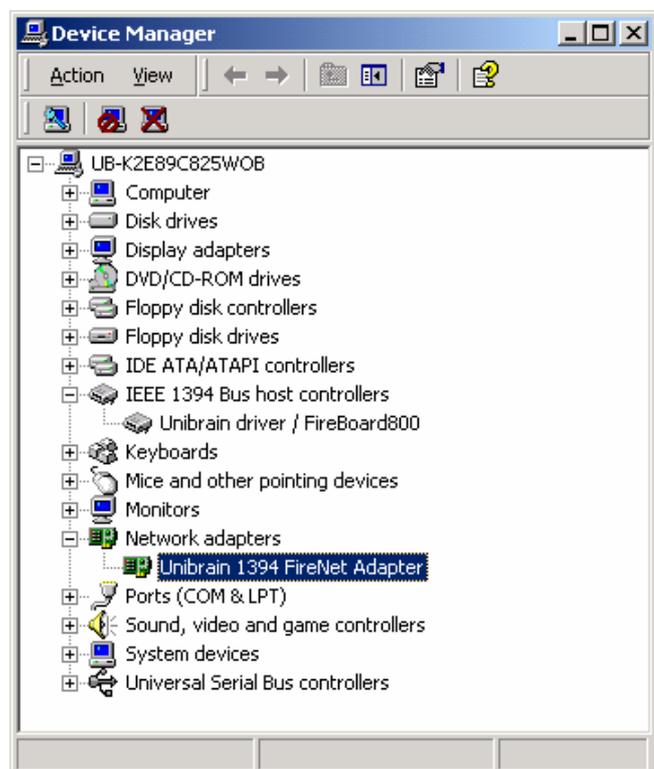


...and click Next in the following dialogue box.



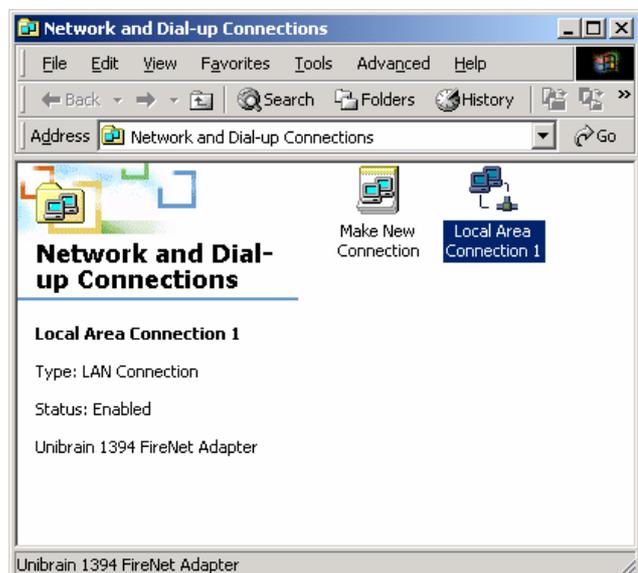
Now you have completely installed FireNet™ that appears to the operating system as a new network adaptor, in which you can bundle ANY routing protocol, e.g. TCP/IP.

In the side picture you can see that the current host PC is equipped with one FireBoard800™ which is uses, after the FireNet™ installation, as an 1394 network adaptor

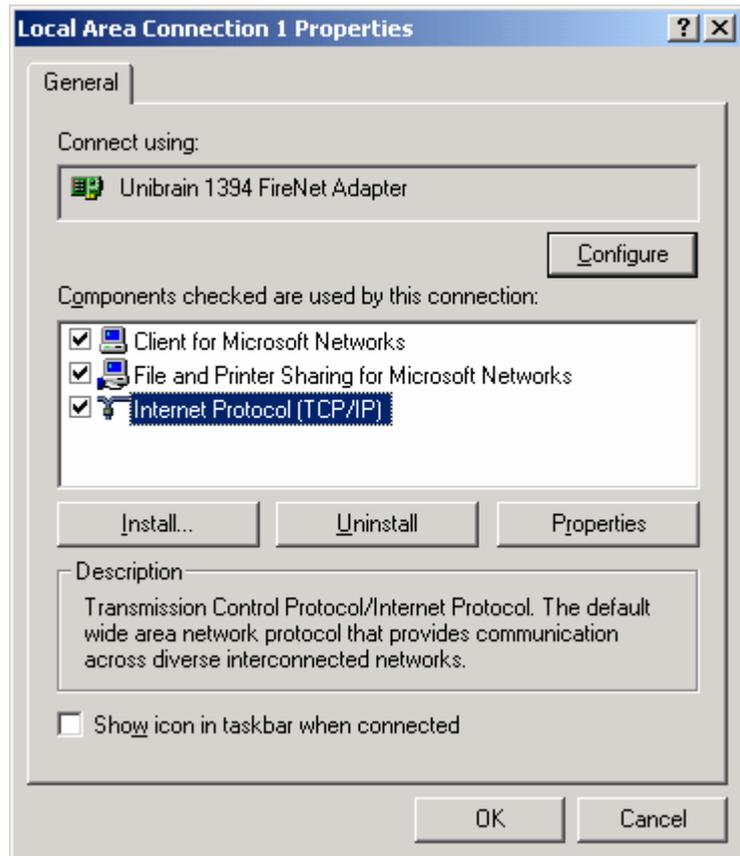


If you right click over Unibrain 1394 FireNet Adapter, you can observe and control the properties of this adaptor.

After the correct installation of FireNet™ you have to configure your new Local Area Connection, by providing the correct parameters for the bundled routing protocol. Select the properties page of this Local Area Connection and ...



...modify appropriately the properties of the routing protocol bundled with Unibrain 1394 FireNet Adapter.



Un-installing FireNet™

In order to remove the FireNet™, go to the Control Panel, click the Network applet and select the 'Adapters' tab. Then in the list with the Network Adapters select the 'Unibrain FireNet' adapter and Uninstall the driver.

Fire-i™ application for 1394 IIDC cameras

Along with ubCore™ 3.2 it is provided a complete application for the control and display of all 1394 IIDC compliant cameras. This application is the Fire-i™ Application, of which the complete source is provided with the FireAPI™ development toolkit.



Fire-i™ software suite operates in all Microsoft OSs including **Windows NT4/98SE/2000/Me/XP** and let you function, control and manage all the new 1394 based digital cameras providing a set of hot characteristics:

- **Fire-i™ supports all 1394 based digital cameras compliant with the IIDC Spec 1.30.**
- Camera(s) output displayed at the same time on all PCs connected with the camera(s)
- Controls any camera anywhere on the 1394 Bus
- Supports loading and saving Camera Memory presets

Fire-i™ Compatibility

The following table illustrates the set of 1394 cameras tested with the Fire-i™ application.

A. SONY models:

A/A	Model No	Description
1	DFW-V500	400 Mbps Color no lens
2	DFW-VL500	400 Mbps Color including lens
3	XCD-X700	400 Mbps B&W no lens
4	XCD-SX900	400 Mbps B&W no lens
5	DFW-X700	400 Mbps Color no lens
6	XCD-X710[CR]	400 Mbps Color no lens
7	XCD-SX910[CR]	400 Mbps Color no lens
8	DFW-SX900	400 Mbps Color no lens
9	DFW-X710	400 Mbps Color no lens
10	DFW-SX910	400 Mbps Color no lens

B. Unibrain models: Fire-i, Fire-i board, Fire-i 400 Industrial cameras.

C. Basler Vision models: A101f, A302fs, A102f, A102fc, A601f, A601fc, A602f, A602fc

D. Micropix models: C-640, M-640, C-1024, M-1024

E. Orange Micro's iBot

F. ADS's Pyro WebCam

G. JVC's KY-F75U, KY-F1030

H. Various models from AVT, PointGrey, Cyberoptics etc.

Fire-i™ Application Guide

After having successfully installed ubCore™ 3.2, the Fire-i™ application is ready to run. Connect one or more 1394 cameras to your PC and run Fire-i™ located at: "E:\Program Files\Unibrain\ubCore\Fire-i" (Please refer to the "Directory structure of ubCore™ installation" section) or from:
Start→Programs→ubCore→Fire-i

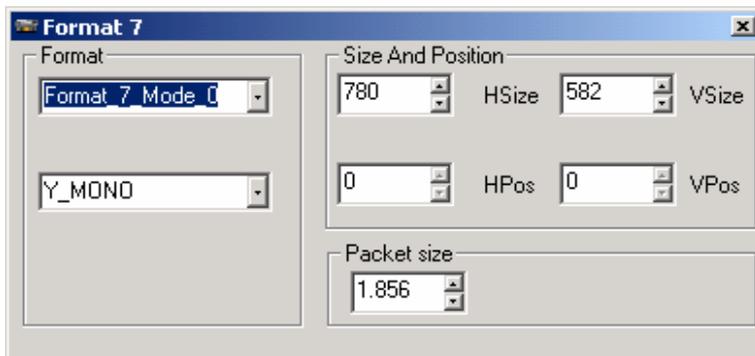
The Fire-i™ controller application is basically a dialog box, as seen in the next figure, in order to control and manipulate your camera. Here is the description of the parameters that you can alter in the main dialog:

Connected Cameras

Displays the list of all cameras connected to your 1394 adapter. By clicking on the combo box you can select the camera that you wish to view.

Video Format

This combo box contains all the video formats that the camera supports. If the user chooses User Defined scalable size format then a dialog box will appear that will let the user specify the dimensions, position as well as other feature of the image.



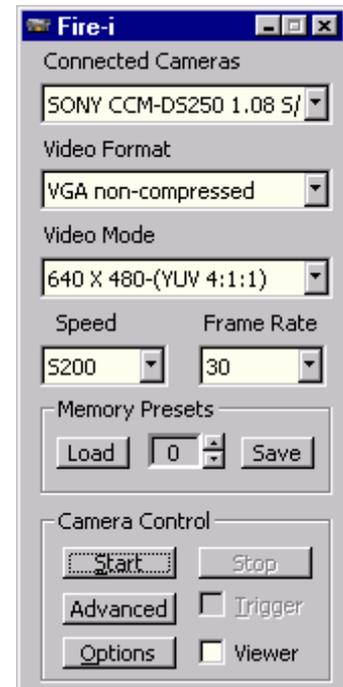
In the above dialog you can see that you have to ability to specify arbitrary horizontal and vertical sizes for your image as well as image position and the size of the packets that will comprise an camera frame. Decreasing the size of the packets increases the number of packets that need to be transmitted in order for a full frame to be received. The higher the packet size the smaller the number of packets transmitted the higher the frame rate of the camera will be. The user can also specify the pixel format of the image transmitted. Most of the cameras in the market that support Format_7, support one single pixel format in this mode.

Video Mode

This combo box contains all the video modes that the camera supports for the currently selected Video Format.

Frame Rate

This combo box contains all the frame rates that the camera support for the currently selected Video Format & Video Mode



The human readable description of the Video Format and Video Mode combination is displayed under the Frame Rate combo box.

Speed

Speed at which the camera will transmit. (100Mbits, 200Mbits or 400Mbits)

Memory Presets

In this box you can save in camera-memory the settings that you have made for the selected camera. You can have many presets. Memory 0 contains the factory presets that can not be changed.

By clicking on the up and down arrow you can select a preset number for each setting that you make, then you can press the **Save** button to save the settings that you have made.

To load a particular setting, you have to choose the memory-preset number that you have saved your settings with and press the **Load** button. Once this is done, your settings will be applied. The number of memory presets depends on the camera model.

Camera Control

You can Start and Stop each Camera, by pressing the appropriate buttons. (**Start, Stop**).

In this Camera Control group you have the possibility to further control the camera pressing the advanced button, which brings up a dialog with the following tabs:

1). **Basic**. Pressing this tab, the Basic Control dialog box appears that enables you to control basic features of your camera:

Focus: Adjustment of the lens to the subject distance. The distance interval where the picture is still in focus, before and after the subject is called « depth of field ». Depth of field is increased by closing the lens iris.

Zoom: Optical magnification control of the lens.

Brightness: Constant amount of luminance electronically added to each pixel. Increasing brightness makes dark and bright parts of the picture become brighter, but luminance difference between dark and bright parts stay identical. Used mostly to achieve good rendering of the dark parts of the picture: black must be as dark as possible, but they should be a visible difference between black and dark grey. There is no influence on color.

Sharpness: Control of the image sharpness.

Gamma: Manual adjustment of the camera internal correction law for monitor display. Law is $y = x^\gamma$; choose $\gamma = 0.45$ for normal visual use, or adjust it according to your monitor. Choice of $\gamma = 1$, also called « gamma OFF » is intended for linear camera output and is best for image processing.

2). **Color**. Pressing this button, the Color Control dialog box appears enable you to control color features of your camera:

U/B value and V/R value: Color adjustment.

Hue: Manual adjustment of the global shift of the colors tints. There is no influence on white balance, saturation, or sensitivity.

Saturation: Manual adjustment of the color / black & white ratio. There is no influence on white balance, hue, or sensitivity.

3). **Exposure.** Pressing this button, the Exposure Control dialog box appears enable you to control the exposure features of your camera:

Exposure: Automatic control of the camera sensitivity. Also called AE. Higher exposure increases the difference between dark and bright parts of the picture. Exposure control is a combined effect of iris and gain controls using an internal law. It supersedes the manually gain and iris controls to achieve automatically an average picture level according to the cursor position. There is no influence on color.

Shutter: Manual reduction of the sensor integration time to a sub-value of the frame period. Used to reduce camera sensitivity in high light conditions, to reduce picture blur for fast moving subjects and to match camera integration time to pulsed light sources (fluo) in order to avoid flickering.

Gain: Manual exposure control by action on the video signal electronic amplification. It cancels automatic exposure operation and works combined with iris control. Increasing exposure by gain adds noise to the picture. There is no influence on color.

Iris: Manual exposure control by action on the lens diaphragm. It cancels automatic exposure operation and works combined with gain control. Increasing exposure by iris reduces the depth of field of the picture. There is no influence on color.

In this Camera Control group you have the possibility to further control the camera if it has trigger possibilities.

Display Popup Menu

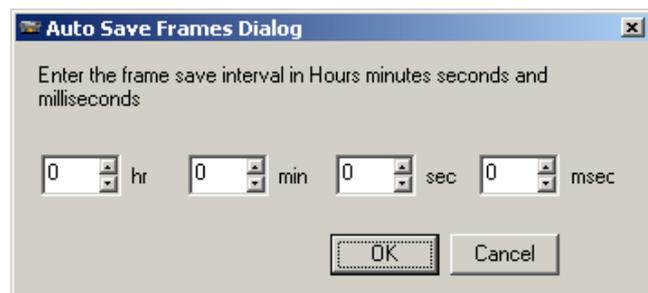
By right clicking on the display window while a camera is displaying the user can bring up a popup menu with the options described into the following paragraphs. The user can also bring up the same popup menu by right clicking on the main Fire-i application dialog. If the user right clicks on the Fire-I application dialog the popup menu items will be related to the camera selected in the camera combo box. If the user clicks on the display window of camera being displayed the popup menu options will be related to that specific camera.

Camera Properties:

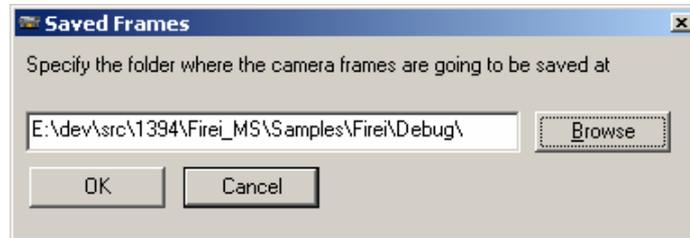
This option brings up the advanced dialog for the current camera being displayed. Using this dialog the user can manipulate the exposure, color and basic controls of the camera as well as read and write from arbitrary camera registers.

Start Auto Save Frames / Stop Auto Save Frames:

This menu item brings up a dialog that allows the user to specify an auto save frame interval and start saving camera frames as bitmaps into a predefined directory. The user can specify a frame save interval in milliseconds, seconds, minutes and hours. Every time the frame save interval elapses a frame is going to be saved as bitmaps on the users hard drive in the directory specified by the **Set Frame Save Path** menu option.



Set Frame Save Path. This menu item brings up a dialog box that lets the user specify a directory where camera frames are going to be saved for this camera. This directory applies both to frames saved manually by using one of the assigned shortcuts and to frames saved using the auto save frames functionality of the application.

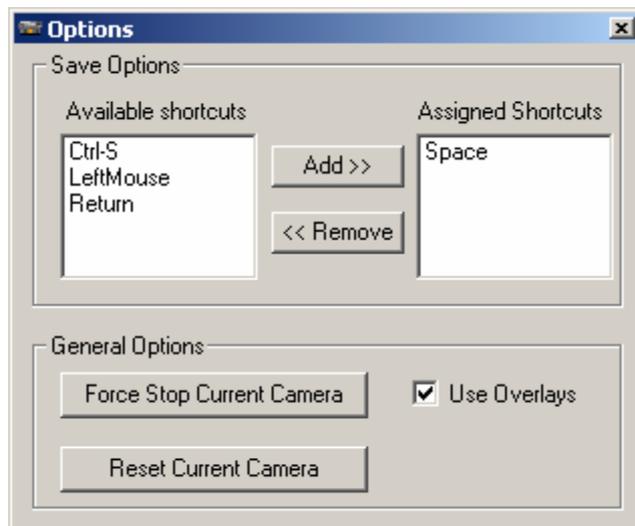


Preview at default size. This menu option set the camera preview image at the default resolution transmitted by the camera.

Trigger: This control is enabled only if the currently selected camera supports trigger mode. Checking the control sets the camera to trigger mode.

Options: This button brings up the options dialog which lets the user choose among a few shortcuts which trigger the save frame operation as well as specify other display options of the application.

By pressing the add button you can add shortcuts to the list of Assigned shortcuts which trigger the save operation. At the above configuration hitting the space bar will save a camera frame in the directory specified by the user. In the general options the user has three controls:



Force stop current camera:

This button stops the currently selected camera in the main dialog and refreshes camera controls. This is done in case the application wants to regain control of a camera that is being controlled by another instance or to recover from a crash that left the camera running.

Use Overlays:

This check box determines whether the camera viewer windows are going to be using Hardware overlay support of the graphics adapter. If this button is checked Fire-i will first try to create a hardware overlay surface and if this fails, because the adapter does not support overlays, Fire-i will create a regular software conversion surface.

Reset Current camera:

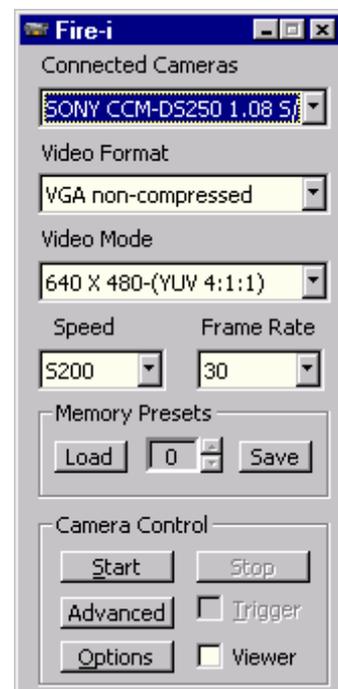
This button resets the current camera.

The Fire-i controller can be opened from 2 different pc's connected to the same 1394 bus. Only one instance of the Fire-i application can control a specific camera at the same time. So if a user starts the Fire-i application on a pc and a camera is started the camera controls will be grayed out. The user will only have the ability to display the camera output on his screen. In order to **refresh** the camera control buttons in case the specified camera is stopped the user should **reselect** the camera from the **Camera Selection Combo box**.

The above picture shows an instance of the Fire-i controller application, which found that a camera was already started. The user can only view the camera at this point by clicking on the viewer button.



The camera is stopped so all the controls are available. The user can now both control and view the camera.



Fire-i™ Short-Keys

Fire-i™ gives you the following functions using your keyboard:

ALT+F4: Close your camera display windows

Possible shortcuts for save operation:

- Ctrl+S
- Space
- Enter
- Left mouse button

FireShare™

Important Warning!

Please be very careful on using your external 1394 SBP2 hard disks in your existing 1394 bus. Every 1394 network has a tree topology with peer to peer connection between the network devices. This topology, while is extremely convenient eliminating the need of a central hub and minimizing the cables length, has some important implications: adding or removing ANY device on the 1394 bus produces a “bus reset”. This “bus reset” is noticeable by ALL the devices and affects the bus topology.

According to the above 1394 bus characteristics, please follow the next basic rules in order to achieve bus reliability and 1394 SBP2 disks integrity:

- Build your 1394 bus with only 1394 SBP2 hard disks and PCs equipped with 1394 interface
- Avoid the attachment on the bus of other 1394 devices, such as: 1394 cameras, 1394 camcorders
- Avoid to turn-on or turn-off the PCs of the 1394 bus while there is traffic in the 1394 network. Those actions produce “bus resets”.
- Never plug or unplug the 1394 cables without previous notifying the 1394 network users.

Please also notice that the FireShare™ application provides the capability to mount and access ANY 1394 SBP2 disk located in your bus (e.g. one other PC of your 1394 network can access the SBP2 disk that is physically attached to your PC). So NEVER unplug or turn-off an external 1394 SBP2 disk before ensuring that this disk is not used by any PC in the 1394 bus!

Introduction

As is already mentioned the new ubCore™ 3.2 provides and installs automatically complete 1394 SBP2 Hard Disk driver suite support. Unibrain’s SBP2 driver is fine tuned for speeds of 800Mbps providing the fastest 1394b interface to external S800 SBP2 Hard disks.

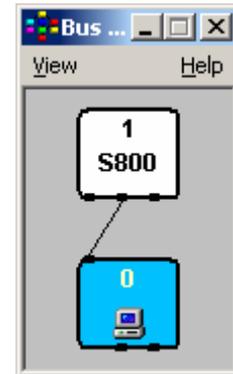
Moreover Unibrain provides the Fire-Share™ SBP2 application for multi-share of 1394 SBP2 hard disks.

As you probably know, while it is possible for a PC to attach to many external 1394 SBP2 disks, it is impossible to simultaneously attach one 1394 SBP2 disk to many PCs connected via FireWire. In other words the 1394 SBP2 drive will be attached and recognized only by one PC, in a FireWire network. With Fire-Share™ you are now able to have mutually-exclusive usage of the 1394 SBP2 disk, and once used, you can release it notifying the other PCs so this disk becomes available.

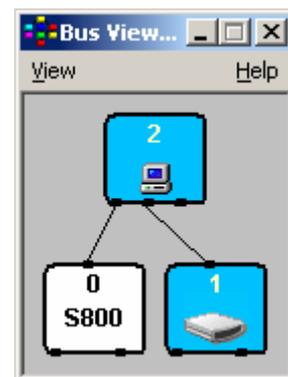
Using FireShare™

The FireShare™ application is the ultimate tool for sharing 1394 SBP2 external disks between various PCs. In order to understand the functionality of this application, let's assume the case of two (2) PCs (equipped with Unibrain's FireBoard800™ 1394b PCI adaptor) interconnected via cable. The FireViewer™ should appear like the adjacent image.

In this stage, without connecting any external 1394 SBP2 HDD, if you run FireShare™ you will get the following dialogue box



Let's say that you connect a 1394b SBP2 external disk and you power it on. The FireViewer™ will represent in the bus topology the presence of this disk.



Please notice that due to the SBP2 protocol nature, ONLY one PC will mount the external disk in a random way. Consequently ONLY one PC will have physical access to this disk and will appear in the explorer.

Run the FireShare™ application from Start→Programs→ubCore→FireShare.

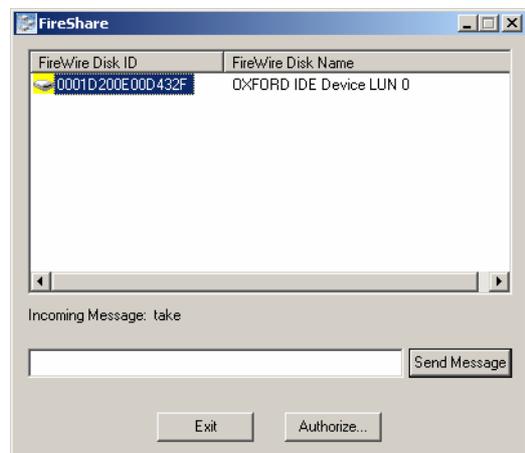
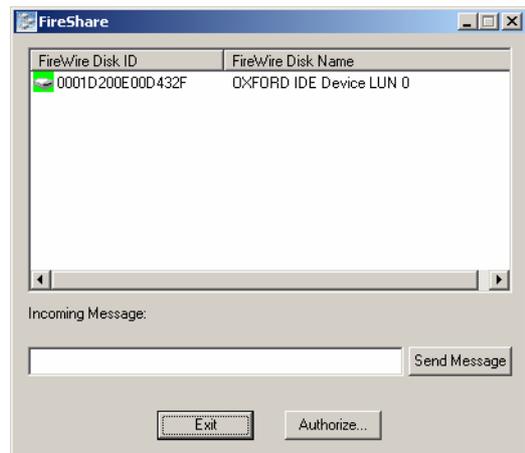
In case that you are the owner of the 1394 SBP2 disk, the disk will appear in the FireWire disks list with GREEN color. For each external hard disk there are listed:

- its ID, a unique serial number extracted from the HDD hardware that identifies the disk and
- the Disk Name, the hard disk name extracted also from the disk's hardware.

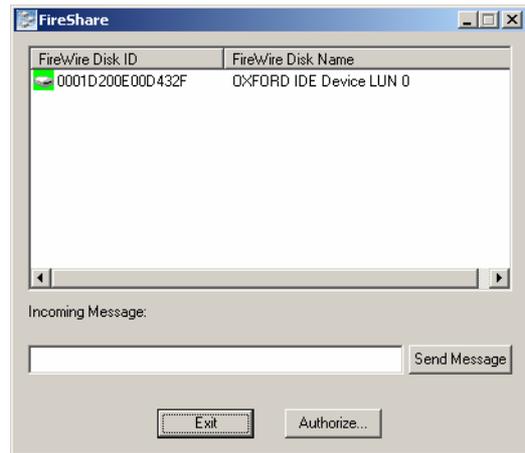
Disk coloring

Each disk in the bus will be listed in the FireWire disks list with the following coloring depending on the availability of the disk:

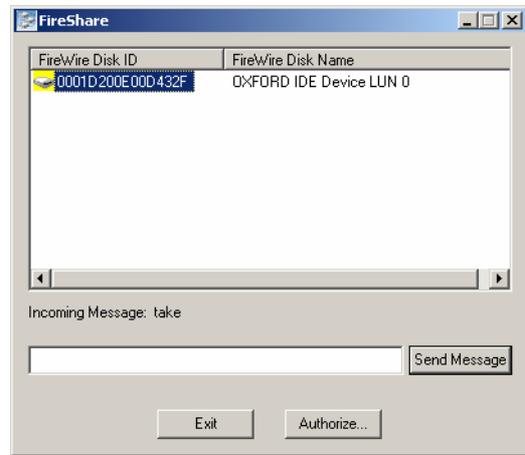
- GREEN: the disk is mounted and allocated to the current PC, so this is the ONLY PC having full access capabilities.
- YELLOW: the disk has been released from its previous owner and it is now available for ANY PC in the bus to allocate and mount.
- RED: the disk has been allocated by another PC so it is NOT available for usage. Only the PC that already allocated the disk has the possibility to use it.



Suppose that we are the owners of the external disk and that we want to release it so it becomes available to other PCs. In order to release it just Double click on the taskbar icon to “Unplug or Eject hardware” and select the disk to unplug.



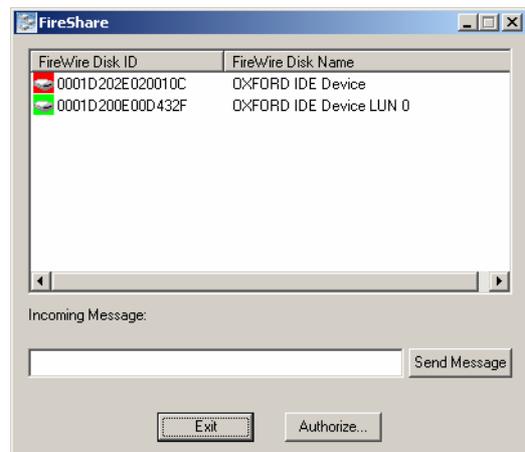
After the disk release, the FireShare™ disk list will be immediately updated with the status of the disk being available, that is YELLOW colored.



In case you want to mount again the disk, double click on the disk (provided that it is still YELLOW colored – available) and you will get the ownership of the disk.

Please observe that while the disk is RED colored (that is allocated from another PC) you cannot mount the disk. The owner should first release the disk (so it becomes YELLOW) and then you can get ownership.

In case of many external hard disks connected to the bus, the FireShare™ application lists the disks and their status as seen in the adjacent picture that presents two (2) disks: the first is not available (RED colored) and the second is already mounted and available (GREEN colored)



Short Messaging

In order to facilitate the communication between the PC's users connected in the same 1394 bus that want to share external HDDs, FireShare™ has a simple mechanism to interchange short informative messages regarding the availability of the disks. For example lets assume that the only disk in the bus is mounted by another PC so it is RED colored (please see adjacent picture) and you need it for data backup. You can type your message to the message box and after pressing the "Send Message" button, this message will be broadcasted to all the PCs in the 1394 bus.

