Universal Test System

FTB-400

User Guide

EXFO

P/N: 1037328

CORPORATE HEADQUARTERS	400 Godin Avenue	Vanier (Quebec) G1M 2K2 CANADA Tel.: 1 418 683-0211 · Fax: 1 418 683-2170
EXFO AMERICA	4275 Kellway Circle, Suite 122	Addison TX, 75001 USA Tel.: 1 972 836-0100 · Fax: 1 972 836-0164
EXFO EUROPE	Le Dynasteur 10/12, rue Andras Beck	92366 Meudon la Forêt Cedex FRANCE Tel.: +33.1.40.83.85.85 · Fax: +33.1.40.83.04.42
EXFO ASIA–PACIFIC	151 Chin Swee Road #03-29, Manhattan House	SINGAPORE 169876 Tel.: +65 6333 8241 · Fax: +65 6333 8242
TOLL–FREE	(USA and Canada)	1 800 663-3936

www.exfo.com · info@exfo.com

Universal Test System

FTB-400

User Guide

EXFO

Trademarks

EXFO's trademarks have been identified as such. However, the presence or absence of such identification does not affect the legal status of any trademark.

Units of Measurement

Units of measurement in this publication conform to SI standards and practices.

The information contained in this publication is subject to change without notice.

Contents

Contents

Contents

Certification Information

F.C.C. Information

Electronic test equipment is exempt from Part 15 compliance (FCC) in the United States. However, compliance verification tests are performed on all EXFO equipment.

C E Information

Electronic test equipment is subject to the EMC Directive in the European Union. The EN61326 standard prescribes both emission and immunity requirements for laboratory, measurement, and control equipment. This unit has been tested and found to comply with the limits for a Class A digital device. Please refer to the Declaration of Conformity.

Independent Laboratory Testing

This unit has undergone extensive testing according to the European Union Directive and Standards. All pre-qualification tests were performed internally, at EXFO, while final tests were performed externally, at an independent, accredited laboratory. This guarantees the unerring objectivity and authoritative compliance of all test results.

IMPORTANT

Use of shielded remote I/O cables, with properly grounded shields and metal connectors, is recommended in order to reduce radio frequency interference that may emanate from these cables.

IMPORTANT

This product is equipped with a shielded three-wire power cord and plug. Use this power cord in conjunction with a properly grounded electrical outlet to avoid electrical shock and to reduce radio frequency interference that may emanate from the power cord.

EXFO
Electro-Optical Engineering

DECLARATION OF CONFORMITY

Application of Council Directive(s):   73/23/EEC - The Low Voltage Directive
                                           89/336/EEC - The EMC Directive
Manufacturer's Name: EXFO ELECTRO-OPTICAL ENG.
Manufacturer's Address: 465 Godin Avenue, Vanier, Quebec
                                           Canada G1M 3G7
                                           (418) 683-0211

Equipment Type/Environment: Industrial Scientific Equipment
Trade Name/Model No.:   FTB-400 Universal Test System
Year of Conformity Assessment: 2001

Standard(s) to which Conformity is Declared:

EN 61010-1: 1993 / A2:1995   **Safety Requirements for Electrical Equipment for Measurement, Control, and Laboratory Use, Part 1: General Requirements**

EN 55022: 1994 / A2:1997   **Limits and methods of measurement of radio disturbance characteristics of industrial, scientific, and medical equipment**

EN 61326-1:1997 / A1:1998+A2:2000   **Electrical Equipment for Measurement, Control and Laboratory Use – EMC Requirements**

EN 50082-2: 1995   **Electromagnetic Compatibility Generic immunity standard - Part 2: Industrial Environment**

I, the undersigned, hereby declare that the equipment specified above conforms to the above Directive and Standards.

Manufacturer

Signature:

Full Name: Stephen Bull, E. Eng
Position: Vice-President Research and Development
Address: 465 Godin Avenue Vanier, Quebec, Canada
Date: novembre 15, 2001

1 Introducing the FTB-400 Universal Test System

Today's network technology is more complex than ever. Thousands of components have to work in harmony, and deployment specialists are responsible for tuning entire systems for optimal network performance and for ensuring records updates. In addition, fiber counts are skyrocketing. DWDM is well entrenched in long-haul applications and is moving into metro. You know that you need more efficiency.

And you will find it with the FTB-400 Universal Test System. Benefit from advanced test operations in outside plant installation, maintenance, and troubleshooting. The FTB-400 Universal Test System is EXFO's newest platform. It streamlines field-based test and measurement operations into a single, powerful, revolutionary package. Welcome to multi-tasking in the field.
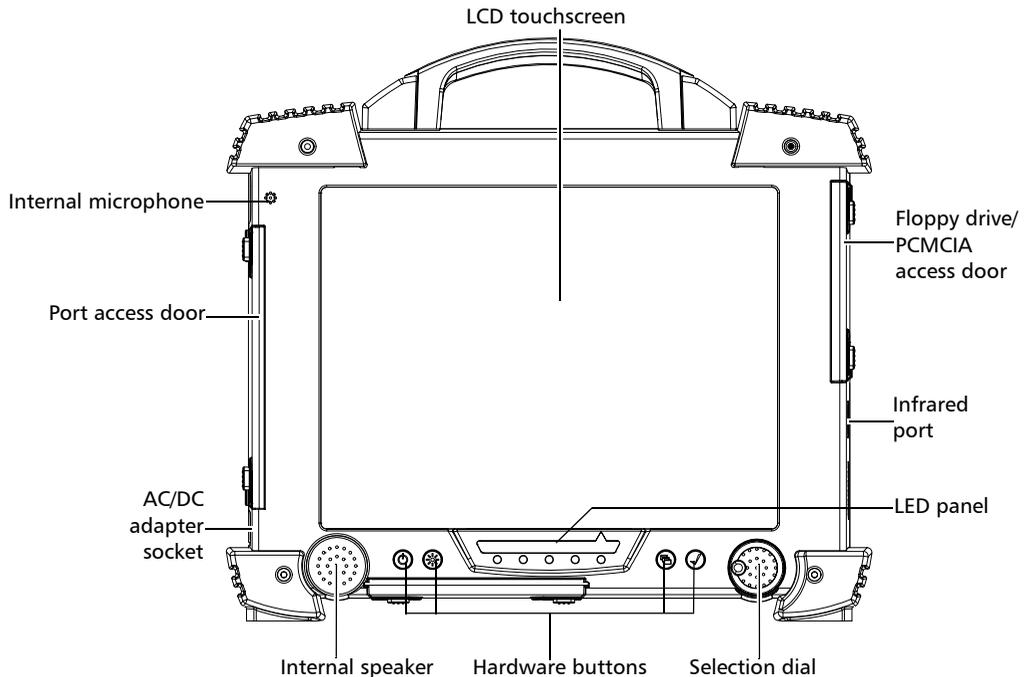
LCD touchscreen

Internal microphone

Floppy drive/ PCMCIA access door

Port access door

Infrared port

AC/DC adapter socket

LED panel

Internal speaker   Hardware buttons   Selection dial

Introducing the FTB-400 Universal Test System

Captive screws

Two-position
stand (two-slot)
or
One-position
stand (seven slot)
(shown closed)

Module
retaining
screws

Captive screws

Battery
compartment
(shown open)

Mainframe

Two-slot
backplane

USE NIMH
BATTERIES ONL

Battery 2   Battery 1
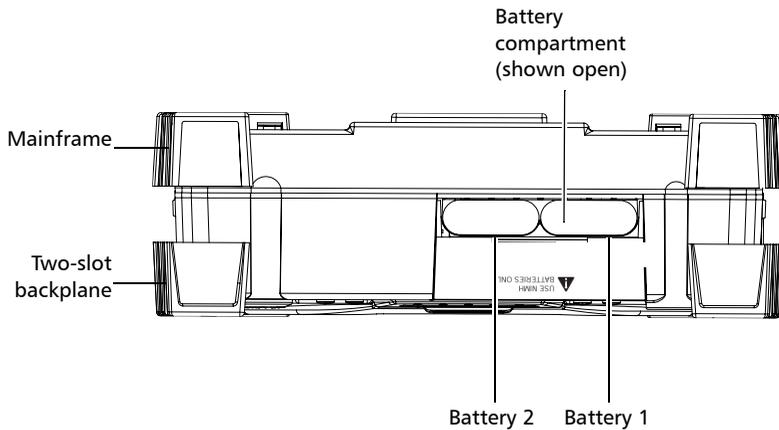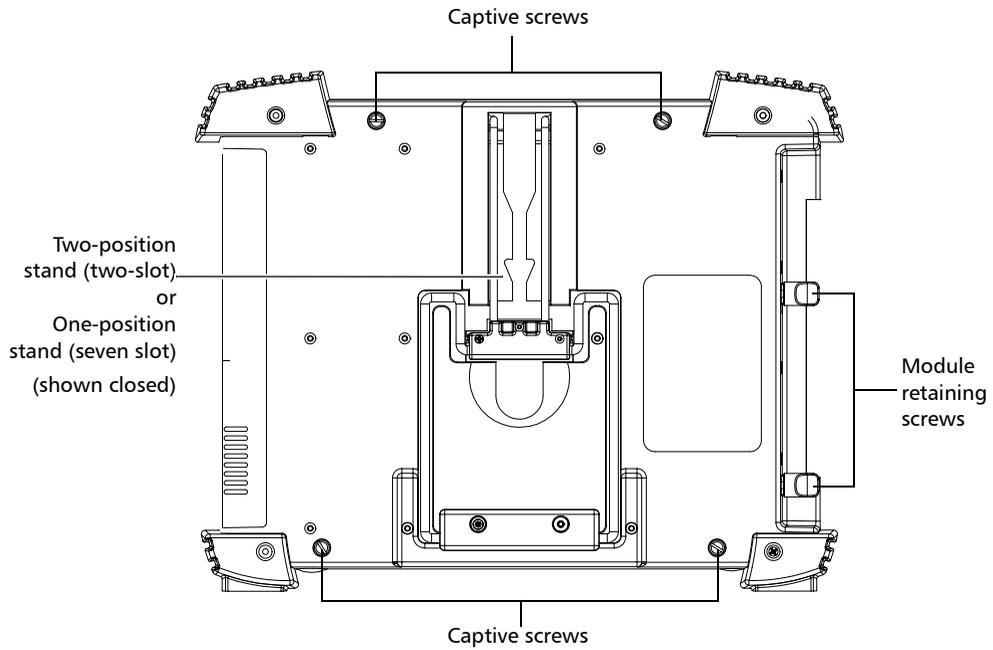
The FTB-400 Universal Test System runs under Windows 2000, and the ToolBox 6 software provides you with a graphic, user-friendly interface for your testing applications.

The interface was designed to be efficient and easy to access, whether you are using the touchscreen or a conventional mouse and keyboard.
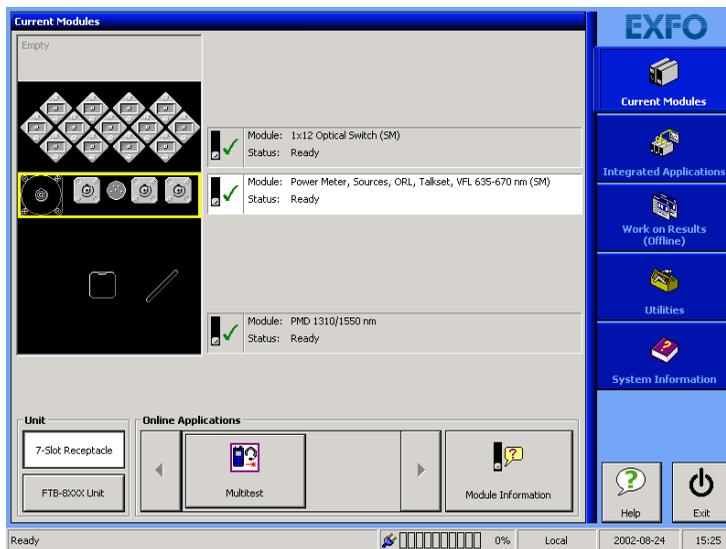
ToolBox 6 brings multitasking to field testing. You can perform different tests and work on tests results all at the same time, switching between applications simply by touching a button. You can also run applications requiring the combination of several modules by setting the combination of modules to use, and launching the application.

The FTB-400 Universal Test System supports local control (via the ToolBox software) and remote control (through GPIB, RS-232 or Ethernet TCP/IP—using SCPI commands or the provided LabVIEW drivers).

Available Models

The FTB-400 Universal Test System comes in two configurations to expand your testing possibilities.

Two-Slot Configuration

➤ dedicated OTDR and loss testing with OTDR and MultiTest modules

➤ over 500 OTDR and loss testing combinations
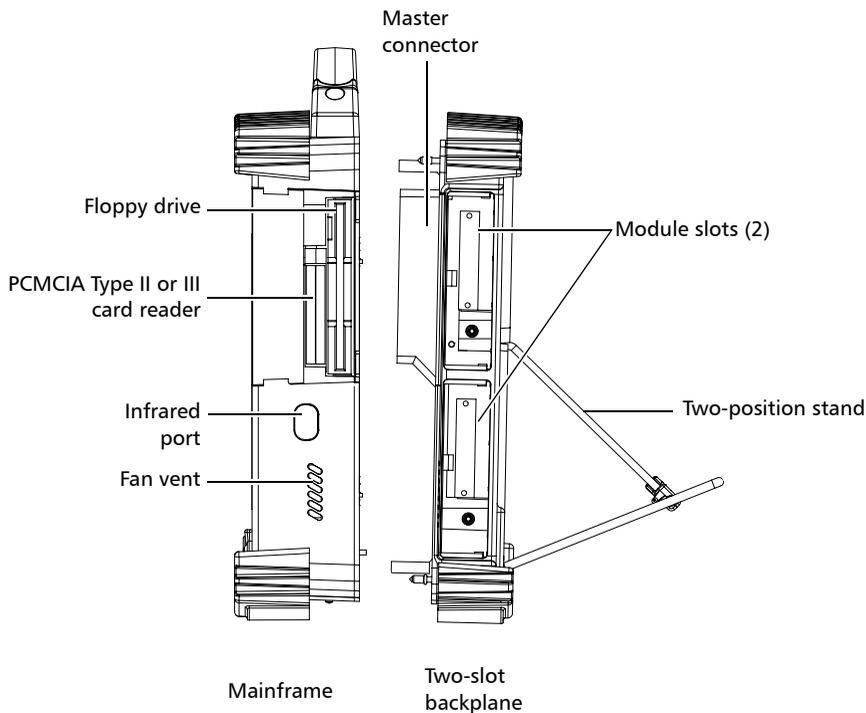
Master connector

Floppy drive

PCMCIA Type II or III card reader

Infrared port

Fan vent

Module slots (2)

Two-position stand

Mainframe

Two-slot backplane

Module
retaining
screws

Audio out port
Audio in port

Serial port (male DB-9)

Parallel port (female DB-25)

Video out port

USB ports

AC/DC adapter/charger socket

Two-slot
backplane

Mainframe
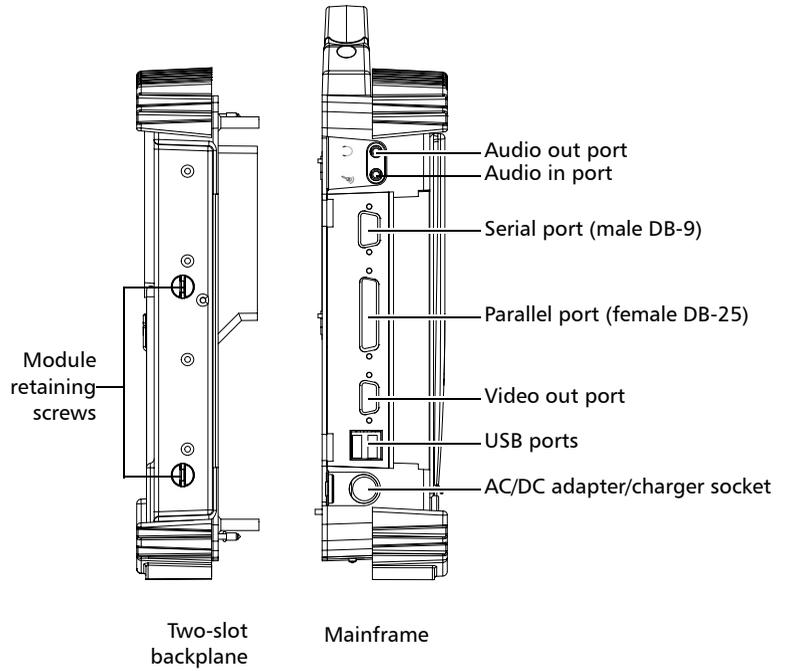
Seven-Slot Configuration

➤ extendible basic and advanced fiber-optic test applications, including DWDM

➤ over 1000 testing combinations-DWDM modules, PMD characterization, ribbon test kits, switches for high fiber count testing, OTDR and loss testing
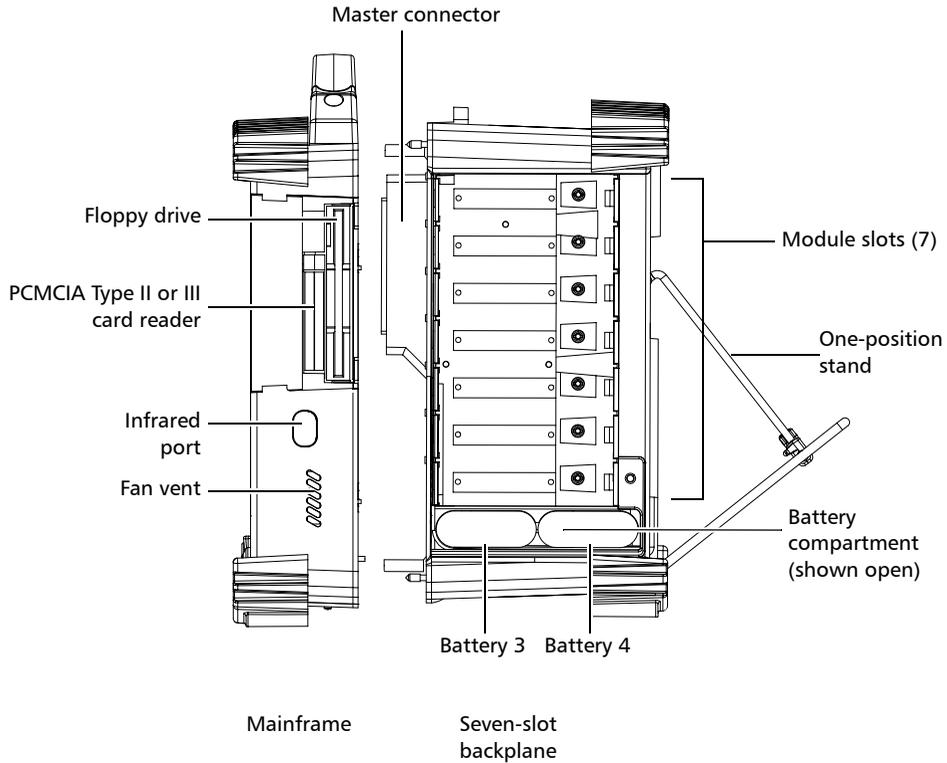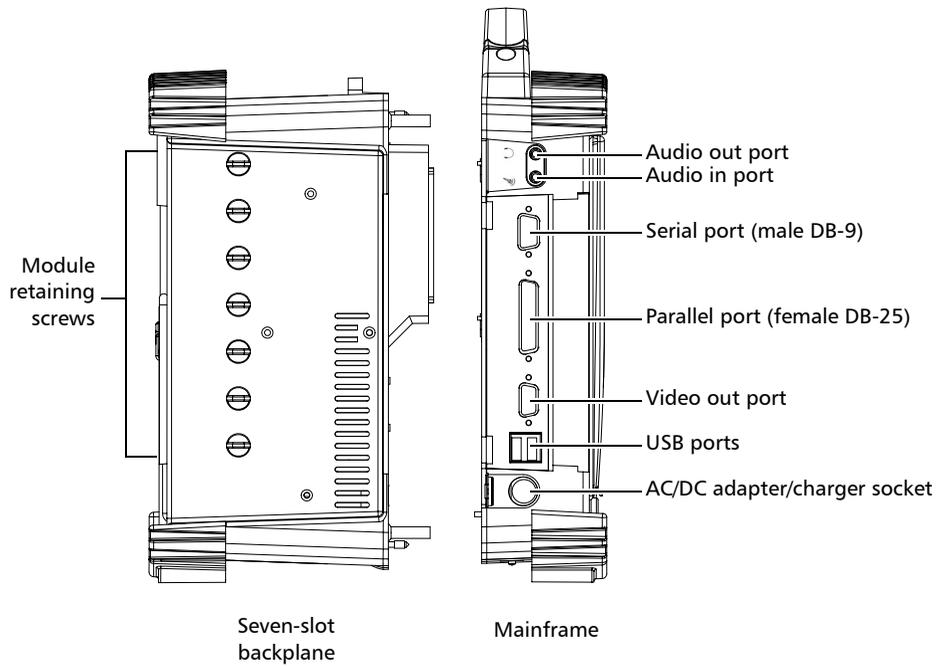
Master connector

Floppy drive

PCMCIA Type II or III card reader

Infrared port

Fan vent

Module slots (7)

One-position stand

Battery compartment (shown open)

Battery 3   Battery 4

Mainframe

Seven-slot backplane

Module retaining screws

Audio out port
Audio in port

Serial port (male DB-9)

Parallel port (female DB-25)

Video out port

USB ports

AC/DC adapter/charger socket

Seven-slot backplane

Mainframe

Understanding Universal Test System LED Panel

The LED panel on the front of the unit provides you with the status of the Universal Test System.

LED	Meaning
⏻	➤ *On:* The FTB-400 Universal Test System is on. ➤ *Off:* The FTB-400 Universal Test System is off.
🔋	➤ *On:* There is less that 10% of power left in the batteries. ➤ *Off:* There is more that 10% of power left in the batteries.
🔌	➤ *On:* The FTB-400 Universal Test System is powered by AC/DC and batteries are not necessary (they are either fully charged or absent). ➤ *Flashing:* Batteries are charging. ➤ *Off:* The FTB-400 Universal Test System is not powered by AC/DC.
💾	*Flashing:* Denotes hard drive activity.
⚠	➤ *Flashing:* Test modules are emitting optical signals. ➤ *Off:* Test modules are not emitting optical signals.

Understanding Universal Test System Hardware Buttons

The FTB-400 Universal Test System is equipped with hardware buttons that give you access to functions at all times. Below, you will find a table of the hardware buttons available and an overview of their purpose.

Button	Purpose
	On/Off: Turns the FTB-400 Universal Test System on or off. For more information, see *Starting the FTB-400 Universal Test System* on page 19.
	Backlight: This button allows you to set the display brightness level. For more information, see *Changing the Backlight Level* on page 34.
	Program Switcher: This button allows you to switch among running applications. It is the equivalent of pressing the Alt-Tab key combination on a keyboard. When you are working in remote control mode, it acts as a return-to-local button.
	Enter: This button allows you to select the highlighted element on the screen. It is the equivalent of using the Enter key on a keyboard. When you are working in remote control mode, it acts as a return-to-local button.

Hard-Drive Structure

Your Universal Test System uses several hard-drive partitions for specific purposes.

Partition	Contents
C	Operating system, EXFO applications
D	User/EXFO data, EXFO application configurations
E	Utilities, Universal Test System backup (ghost)

Note: *Contact EXFO's Technical Support for further help if you ever need to use the Universal Test System ghost.*

If you already had a Universal Test System and this version of ToolBox is an upgrade, your original partition system was divided differently.

Partition	Contents
C	Operating system, EXFO applications, user/EXFO data, EXFO application configurations
D	Utilities, Universal Test System backup (ghost)

Therefore, when you upgrade to the version that came with this user guide, the installation process will keep the previous partition structure, but move the C:\Program Files\EXFO\Toolbox\UserFile folder to the D partition. This will ensure that your data is kept should you need to reinstall ToolBox using the ghost.

2 *Safety Information*

Safety Conventions

Before using the product described in this manual, you should understand the following conventions:

WARNING
Indicates a potentially hazardous situation which, if not avoided, could result in *death or serious injury*. Do not proceed unless you understand and meet the required conditions.

CAUTION
Indicates a potentially hazardous situation which, if not avoided, may result in *minor or moderate injury*. Do not proceed unless you understand and meet the required conditions.

CAUTION
Indicates a potentially hazardous situation which, if not avoided, may result in *component damage*. Do not proceed unless you understand and meet the required conditions.

IMPORTANT
Refers to information about this product you should not overlook.

General Safety Information

CAUTION

If the unit is stored at a temperature outside of the specified operating temperature range, let the unit reach operating temperature before turning on.

Laser Safety Information

WARNING

Do not install or terminate fibers while a light source is active. Never look directly into a live fiber and ensure that your eyes are protected at all times.

WARNING

Use of controls, adjustments and procedures for operation and maintenance other than those specified herein may result in hazardous radiation exposure.

WARNING

Use of optical instruments with this product will increase eye hazard.

CAUTION

When the laser safety light is flashing, one (or more) of your modules is emitting an optical signal. Please check all modules, as it might not be the one you are currently using.

Electrical Safety Information

The power cable of your FTB-400 Universal Test System is its connect and disconnect device when you are not using the batteries.

CAUTION
The following voltage requirements must be met: AC/DC adapter 12–20 volts, 70 W. Use of higher voltages may damage the unit.

CAUTION
The external power supply unit is designed for indoor use only.

CAUTION
Use NiMH batteries only. Use of other batteries may damage the unit and compromise your safety.

CAUTION
Before changing the backplane, you must remove ALL power sources (batteries and power outlet) from the FTB-400 Universal Test System.

CAUTION
Do not remove or insert modules when the power is on.

This unit uses an international safety standard three-wire power cable. This cable serves as a ground when connected to an appropriate AC power receptacle. The type of power cable supplied with each unit is determined according to the country of destination.

Only qualified electricians should connect a new plug if needed. The color coding used in the electric cable depends on the cable. New plugs should meet the local safety requirements and include the following features:

➤ adequate load-carrying capacity

➤ ground connection

➤ cable clamp

WARNING

To avoid electrical shock, do not operate the unit if there are signs of damage to any part of the outer surface (covers, panels, etc.). To avoid serious injury, observe the following precautions before powering on the unit.

➤ **If the unit is to be powered via an auto transformer for voltage reduction, connect the common terminal to the grounded power source pole.**

➤ **Insert the power cable plug into a power outlet with a protective ground contact. Do not use an extension cord without a protective conductor.**

➤ **Before powering on the unit, connect its protective ground terminal to a protective conductor using the unit power cord.**

➤ **Do not tamper with the protective ground terminal.**

---

3 Getting Started with your Universal Test System

The following pages contains important information on how to operate your FTB-400 Universal Test System.

Inserting and Removing Test Modules

CAUTION
Never insert or remove a module while the FTB-400 Universal Test System is powered on. This will result in immediate and irreparable damage to both the module and unit.

WARNING
When the laser safety light ( ) is flashing on the FTB-400, at least one of your modules is emitting an optical signal. Please check all modules, as it might not be the one you are currently using.

To insert a module into the FTB-400 Universal Test System:

1. Exit ToolBox and power off your unit.

2. Position the FTB-400 so that its right panel is facing you.

3. Take the module and place it so that the connector pins are at the back, as explained and shown below. Identification sticker must be facing down and connector pins at the left of the retaining screw hole.

CAUTION

Inserting a module upside down could result in permanent damage to the module, as the connector pins might be bent.

FTB-400 right panel
(7-slot unit)

Protruding
edges on top

Retaining
screw hole
at the back

Connector
pins at the
back

Identification sticker
facing down
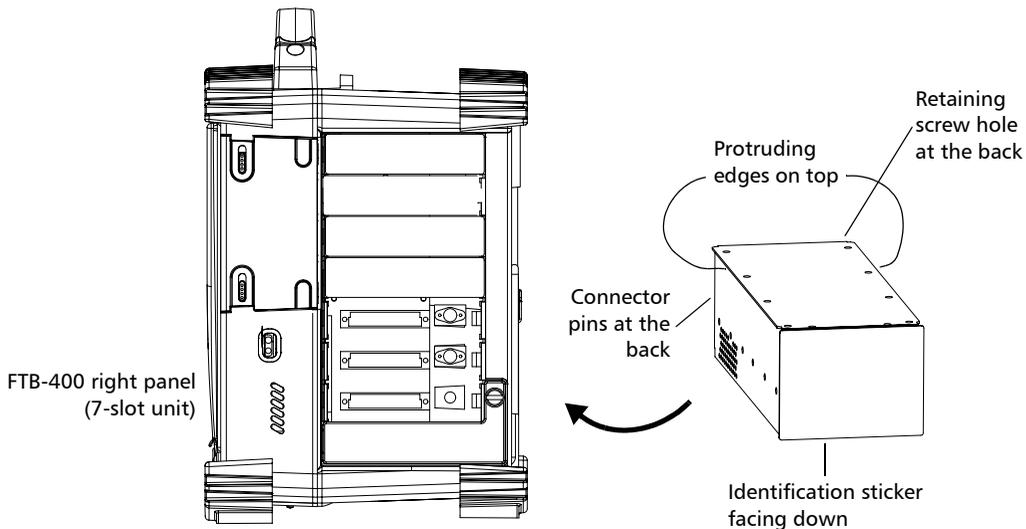
4. Insert the protruding edges of the module into the grooves of the unit's module slot.

5. Push the module all the way to the back of the slot, until the retaining screw makes contact with the unit casing.

6. Place the FTB-400 so that its left panel is facing you.

7. While applying slight pressure to the module, turn the retaining screw clockwise until it is tightened. This will secure the module into its "seated" position.

Turn retaining screw knob(s) clockwise

FTB-400 left panel
(7-slot unit)
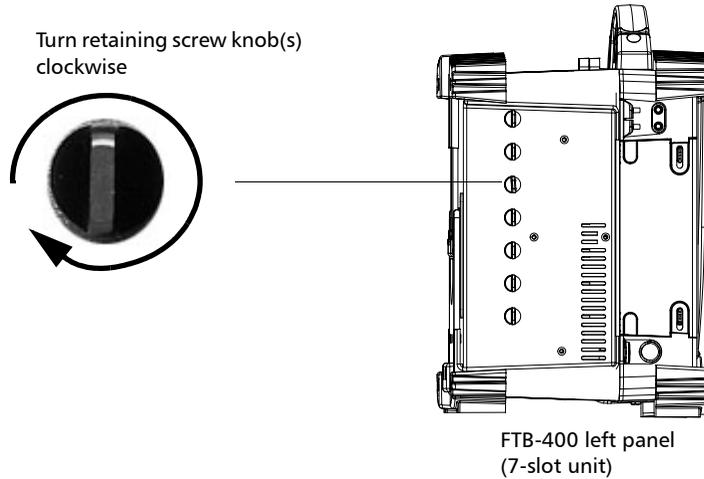
When you turn on the FTB-400, the startup sequence will automatically detect the module.

To remove a module from the FTB-400 Universal Test System:

1. Position the FTB-400 so that the left panel is facing you.

2. Turn the retaining screw counterclockwise until it stops. The module will be slowly released from the slot.

Turn retaining screw knob(s) counterclockwise

FTB-400 left panel (7-slot unit)
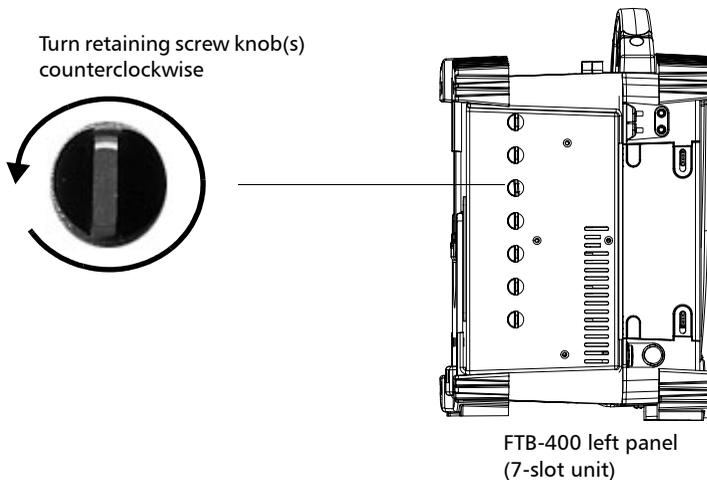
3. Place the FTB-400 so that the right panel is facing you.

4. Hold the module by its side or by the handle (*NOT by the connector*) and pull it out.
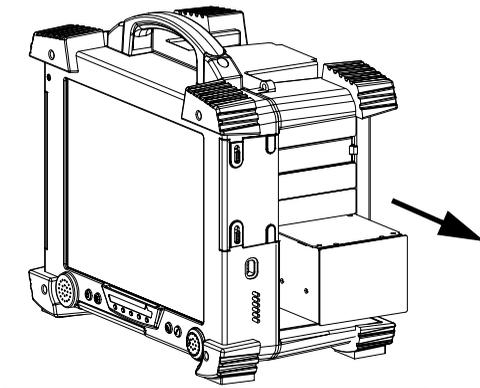
CAUTION

Pulling out a module by a connector could seriously damage both the module and connector. Always pull out a module by its casing.

5. Cover empty slots with the supplied protective covers.

CAUTION

Failure to reinstall protective covers over empty slots may result in ventilation problems.

Starting the FTB-400 Universal Test System

When the FTB-400 Universal Test System is turned off, hold down the blue On/Off button on the front panel for two seconds. The FTB-400 Universal Test System beeps, the startup screen briefly appears and the FTB-400 Universal Test System starts. For more information on startup options, see *Selecting the Startup Interface* on page 48.

Shutting Down the FTB-400 Universal Test System

The FTB-400 Universal Test System offers two types of shutdown:

➤ a *warm* shutdown (also called *hibernation* mode), which will allow the FTB-400 Universal Test System to restart faster the next time is it used. This is the preferred shutdown method.

➤ a *cold* shutdown, which will allow the FTB-400 Universal Test System to perform a complete restart routine the next time is it used. This shutdown cuts power completely to the test modules and platform.

IMPORTANT

If you change the module configuration in your FTB-400 Universal Test System, you need to perform a cold shutdown. Otherwise, the modules you have changed, added or removed might not be properly recognized as such.

To perform a warm shutdown:

1. Hold down the **On/Off** button for *two* seconds.

2. Release the button immediately after the beep.

The FTB-400 Universal Test System shuts down. A warm shutdown ensures that all currently running applications will close and that the next time you turn the FTB-400 Universal Test System on, it will start more rapidly.

To perform a cold shutdown:

Hold down the On/Off button for *five* seconds.

The FTB-400 Universal Test System shuts down. This cuts the power from the FTB-400 Universal Test System and its modules. After a cold shutdown, it will take approximately two minutes longer than after a warm shutdown to turn on the FTB-400 Universal Test System.

Changing the Backplane

Depending on your testing needs, you can change the FTB-400 Universal Test System backplane easily, even when in the field.

CAUTION

Before changing the backplane, you must remove ALL power sources (batteries and power outlet) from the FTB-400 Universal Test System.

To change the backplane on your FTB-400 Universal Test System:

1. Make sure that the FTB-400 Universal Test System is shut down.

2. Unscrew the four captive screws from the backplane.

3. Gently pull the backplane away from the mainframe, making sure to keep the backplane perfectly straight.

Master connector

Captive screws

> # CAUTION
>
> **Ensure that you do not pull the backplane at an angle because this might damage the master connector.**

4. Place the new backplane into position so that the master connector lines up with the master connector slot at the back of the mainframe.

5. Gently push the backplane back into the mainframe, keeping the backplane as straight as possible.

6. Turn the captive screws to lock the backplane to the mainframe.

When you turn on the FTB-400 Universal Test System after changing the backplane, the new backplane is automatically recognized.

Note: *If the backplane is not properly screwed in, the FTB-400 Universal Test System might not detect it.*

Installing or Upgrading ToolBox

If you have purchased an FTB-400 platform, the ToolBox software will have been preinstalled and configured at the factory.

If you are using a PC for data post-processing, or if you ever need to reinstall or upgrade your software, use the CD-ROM that came with your product or upgrade kit and follow the instructions below.

Since your FTB-400 Universal Test System is not equipped with a built-in CD-ROM drive, you can either

➤ connect an external CD-ROM drive to your FTB-400

   OR

➤ use Data Transfer or the infrared port to copy the installation files on the FTB-400 and proceed from there

Note: *When installing ToolBox on a PC, some features will not be available.*

Minimum Requirements	Windows 98 (SE)	Windows 2000
CPU	120 MHz Pentium	133 MHz Pentium
RAM	64 MB	64 MB
Disk space	50 MB	50 MB
Display	Super VGA, 800 x 600, 16 million colors, small fonts	Super VGA, 800 x 600, 16 million colors, small fonts

Note: *Only administrator-level users can install software under Windows 2000.*

> # IMPORTANT
>
> **Before upgrading ToolBox, you should remove the old version using the Add/Remove Programs utility (in the Windows Control Panel).**
>
> **Once the operation is complete, you should no longer see ToolBox 6 in the list. Otherwise, read the *Troubleshooting* chapter for special uninstallation instructions.**

To install or upgrade ToolBox:

1. Insert the ToolBox installation CD in the CD-ROM drive.

2. The installation process should start automatically. If not, launch it manually as follows:

   2a. On the Windows taskbar, click on **Start** and select **Run...**

   2b. Type *E:\ToolBox 6 Setup.exe* in the **Open** box (the drive letter may vary according to the location of your CD-ROM drive).

   2c. Click on **OK** to start the InstallShield Wizard and follow the on-screen instructions.

Note: *If a previous version of ToolBox is found on your platform, you will be requested to uninstall it before proceeding.*

3. In the **Select Features** step, select the component(s) you want to install and click on **Next** to start the installation.

List of modules may differ from illustration —
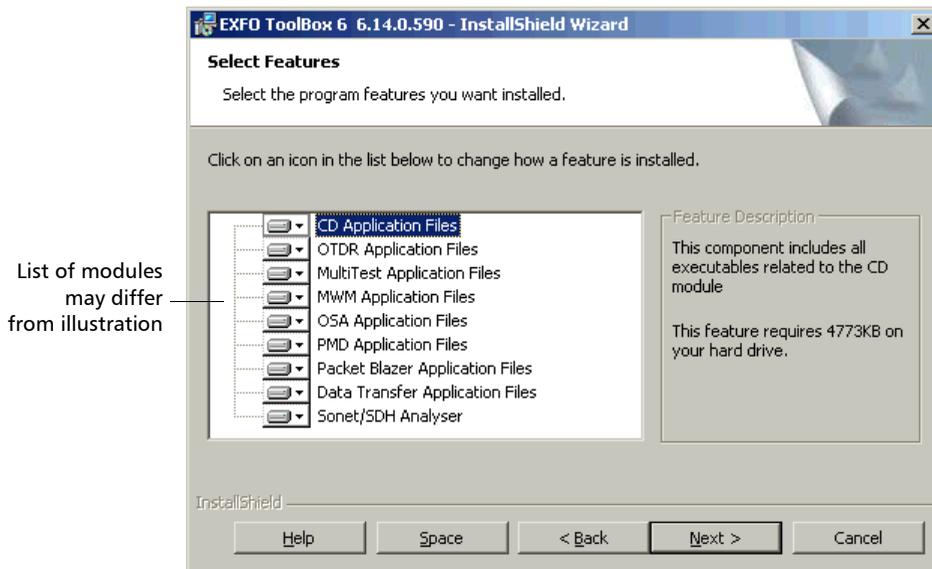
4. Once the installation is complete, click on **OK** to reboot your FTB-400 or PC.

Accessing ToolBox 6

When your FTB-400 Universal Test System system comes out of the factory, it is set to start automatically with ToolBox 6.

However, you might need to work with Windows, and can even set the system to startup with Windows instead of ToolBox 6, as explained in *Selecting the Startup Applications* on page 49.

To access ToolBox 6 when in the Windows environment:

➤ Click on the [icon] icon on your desktop.

OR

➤ In the **Start** menu, select "Programs > Exfo > ToolBox 6".

Exiting ToolBox 6

To exit ToolBox 6, click on [button]. Choose the option that suits you best.

➤ **Logout and return to Startup screen**: This option allows you to let another user to logon. This option is available only if you have activated the password option, as explained in *Setting Passwords* on page 50.

➤ **Exit ToolBox 6 and return to Windows 2000**: This option will simply close ToolBox 6 and let you use Windows 2000 with other applications.

➤ **Turn off the** FTB-400 Universal Test System: This option will allow you to shut down your unit.

Click on **OK** to confirm your choice, or on **Cancel** to return to ToolBox 6.

Installing EXFO LabVIEW Drivers

Before being able to work with EXFO LabVIEW drivers, you must install the following elements on a PC or on your FTB-400 Universal Test System:

➤ National Instruments LabVIEW 6.0 software and the 6.0.2 patch

➤ EXFO LabVIEW drivers (including demo applications to help you get started with the drivers)

For more details, see *Working with EXFO LabVIEW Drivers* on page 122.

Note: *Only administrator-level users can install software under Windows 2000.*

To install the LabVIEW software:

1. Insert the *LabVIEW* CD in the CD-ROM drive.

2. The installation process should start automatically. If not, launch it manually as follows:

**2a.** On the Windows taskbar, click on **Start** and select **Run...**

2b. Type *D:\autorun.exe* in the **Open** box (the drive letter may vary according to the location of your CD-ROM drive).

2c. Click on **OK** to start the InstallShield Wizard and follow the on-screen instructions.

EXFO recommends that you keep the default names and paths suggested by the installation program.

3. Once the LabVIEW installation is complete, install the 6.0.2 patch. If the patch is not included on your LabVIEW CD, you may download it from National Instruments' Web site at *www.ni.com*.

3a. On the Windows taskbar, click on **Start** and select **Run...**

3b. Type *D:\Updates\setup.exe* in the **Open** box.

3c. Click on **OK** to start the InstallShield Wizard and follow the on-screen instructions.

To install the EXFO LabVIEW drivers:

1. While holding down the SHIFT key, insert the installation CD in the CD-ROM drive.

2. Launch the installation process as follows:

2a. On the Windows taskbar, click on **Start** and select **Run...**

2b. Type *D:\Labview Drivers\setup.exe* in the **Open** box (the drive letter may vary according to the location of your CD-ROM drive).

2c. Click on **OK** to start the InstallShield Wizard and follow the on-screen instructions.

For easier use, the drivers will be installed in LabVIEW's default instrument library folder:
C:\Program Files\National Instruments\LabVIEW 6\instr.lib\EXFO.

4 *Setting Up Your Universal Test System*

It is possible for you to customize ToolBox 6 in different ways.

Installing a Monitor

Although the FTB-400 Universal Test System is equipped with a top-of-the line LCD, there are times when you might want to use an external monitor. The monitor must support the SVGA format.

To install a monitor:

1. Shut down the FTB-400 Universal Test System.

2. Connect the external monitor to the Video Out port, located on the left side of the mainframe.

3. Turn on the FTB-400 Universal Test System.

The external monitor is automatically recognized when you restart the FTB-400 Universal Test System.

Note: *Use the Windows Control Panel to configure the display settings of your external monitor.*

Calibrating the Touchscreen

The touchscreen simplifies and accelerates testing procedures by providing immediate access to commands. It detects the position of your finger or any other blunt pointing device, activating a command, function, or button. The FTB-400 Universal Test System comes equipped with a stylus located inside the plastic cover.

When you receive your FTB-400 Universal Test System, it is a good idea to calibrate the touchscreen. This will ensure that it behaves in the way that suits you best.

Note: *To better see what is on the screen, you can also change the brightness of the screen backlight, as explained in* Adjusting the Monitor and Sound *on page 33.*

To calibrate the touchscreen:

1. Click on the **Utilities** function tab.
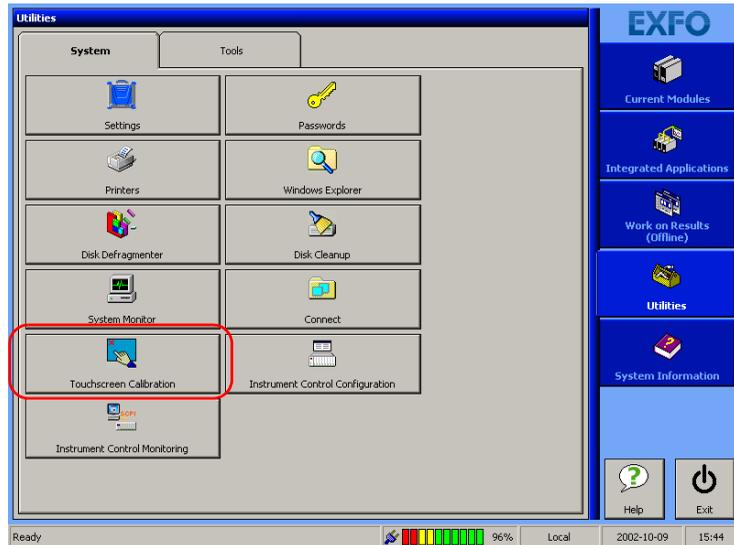
2. In the System tab, select **Touchscreen Calibration**.

3. As indicated in the middle of the screen, press on the arrow located on the top left-hand corner of the screen to launch the calibration sequence.

Device 1

Touch arrow tips or cross centers as they appear
Calibration will terminate if no touch
is received within 10 seconds
Press escape to abort the calibration process

4. The arrow will move down for you to press on it again. Continue this
    process until you have covered the whole screen. This may take a few
    moments.

5. If you pressed incorrectly on the screen, or you do not want to continue
    with the calibration, simply wait ten seconds. You will then be returned
    to the previous screen.

6. Once the calibration is complete, this screen will disappear and you
    are returned to the **Utilities** function tab.

Adjusting the Monitor and Sound

Different work environments call for different monitor and sound settings, which can be set in the **Monitor and Sound** tab of the **Settings** window.

To adjust the monitor and sound settings:

1. Select the **Utilities** function tab, then the **System** tab.

2. Click on **Settings**, then select the **Monitor and Sound** tab.
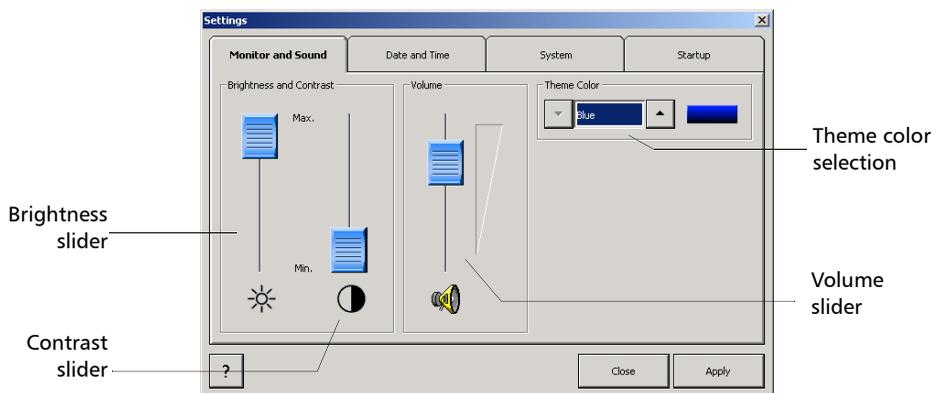
3. Adjust the monitor or volume settings as needed.

➤ To adjust the monitor according to the current environment, use the brightness and contrast sliders.

Note: *If you have purchased an FTB-400 Universal Test System with a TFT (active) screen, you cannot adjust the contrast. The corresponding slider will look like this:* ▢ *.*

➤ To adjust the volume of the unit, use the **volume** slider.

➤ To change the theme color for your unit, use the up and down arrow buttons in the **Theme Color** group box.

4. Click on **Apply** to use the new settings. Click on **Close** to exit without using the new settings.

Changing the Backlight Level

The backlight has four different brightness levels: Off, Low, Medium, and High.

To change the backlight level:

Press the **Backlight** button repeatedly to toggle between these four levels.
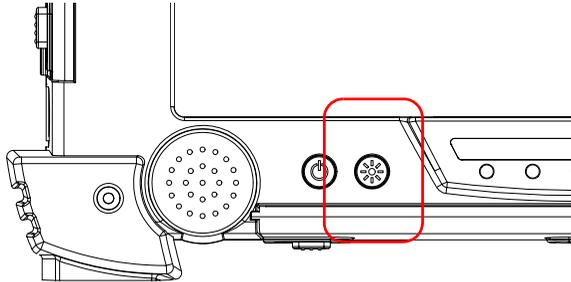
Note: *A lower backlight level uses less battery power and allows for better autonomy when using the FTB-400 Universal Test System.*

These levels are actually preset brightness levels. However, you can also select a precise level of brightness from the **Monitor and Sound Settings** page in ToolBox 6. For more information, see *Adjusting the Monitor and Sound* on page 33.

Installing the Hardware Keyboard

When you are required to enter alphanumeric data, an on-screen keyboard will be displayed. However, you can also use a hardware keyboard.

To install an external keyboard on your FTB-400 Universal Test System:

Connect the keyboard to a USB port, located on the left side of the mainframe.
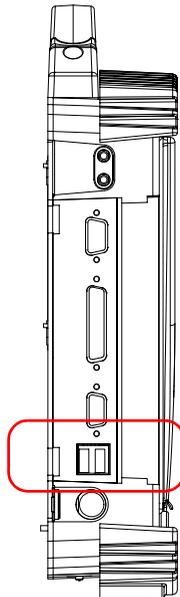
Note: *It is not necessary to turn off the FTB-400 Universal Test System before connecting the keyboard. The software will automatically detect its presence.*

If a keyboard is connected, the touchscreen keyboard will still be displayed when working in ToolBox 6. For more information, see *Using the On-Screen Keyboard* on page 36.

Although your external keyboard is automatically recognized and immediately usable, you might not benefit from all its special features unless you install the proprietary driver that usually comes with it. To install the proprietary driver, refer to the user guide that comes with your device. Of course, the keyboard that comes as an option with your FTB-400 Universal Test System is fully functional.

Using the On-Screen Keyboard

If no hardware keyboard is connected to your unit, you can use the on-screen keyboard to type in data.

To activate the on-screen keyboard:

1. Select the **Utilities** function tab in the main window.

2. Select the **System** tab.

3. Click on **Settings**.
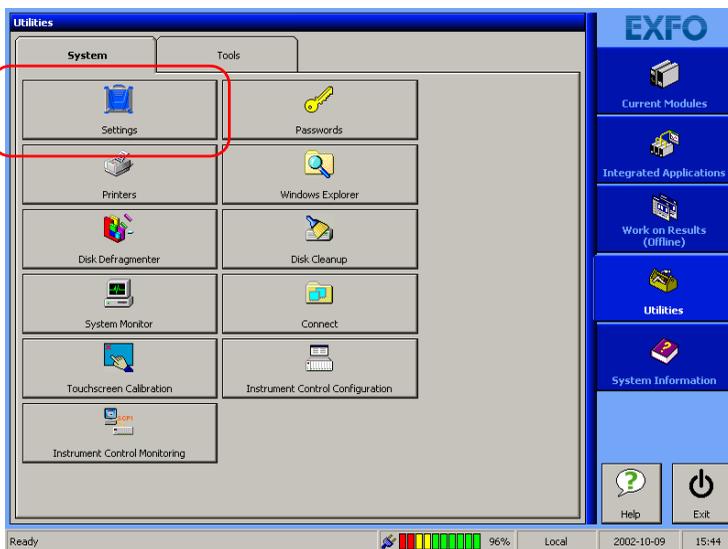
4. Select the **System** tab.
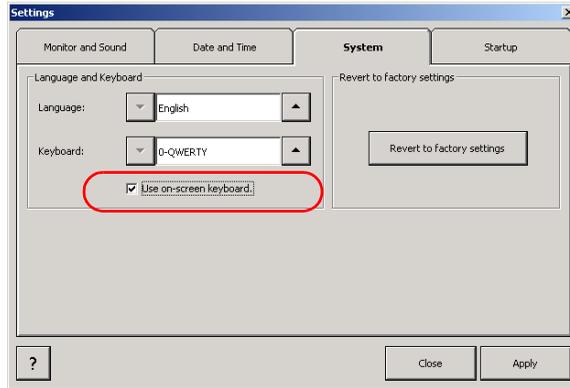
5. Check the **Use on-screen keyboard** box.

To use the keyboard once activated, simply type in the data you want and click on either **OK** or **Return** to enter it.

Click on **Cancel** to return to your application without typing anything.

To install a hardware keyboard, see *Installing the Hardware Keyboard* on page 35.

Setting the Language and Keyboard Type

ToolBox 6 was designed to function in several languages and can adapt to different keyboard types. Both can be set in the **System** tab.

To select the language and keyboard type to use:

1. Select the **Utilities** function tab in the main window.

2. Select the **System** tab.

3. Click on **Settings**.

4. Select the **System** tab.
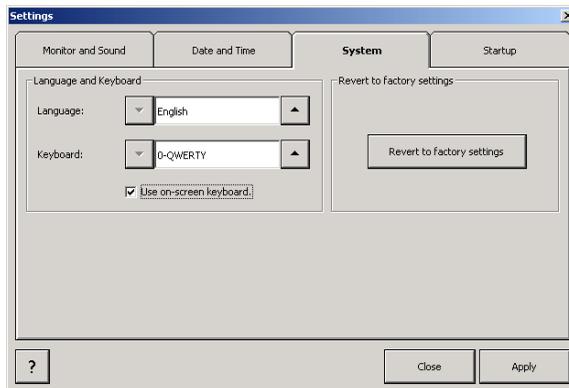
5. Change the language and keyboard settings as needed.

➤ To change the language, use the up and down arrows around the corresponding field.

➤ To change the keyboard type, use the up and down arrows around the corresponding field.

6. Click on **Apply** to use the new settings. Click on **Close** to exit without using the new settings.

Installing a Pointing Device

Using a touchscreen offers many advantages when working in applications that have been specially designed for it (such as the EXFO test applications). However, for application such as spreadsheet, word processors, etc., (that the FTB-400 Universal Test System allows you to run) a touchscreen might not be the most efficient pointing device. The FTB-400 Universal Test System was designed so that you can add external pointing devices (mouse, trackball, etc.).

To install an USB pointing device on your FTB-400 Universal Test System:

Connect it to a USB port, located on the left side of the mainframe.
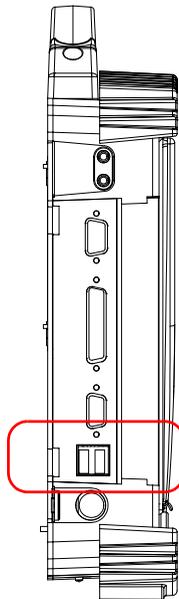
Although your external pointing device is automatically recognized and immediately usable, you might not benefit from all its special features unless you install the proprietary driver that usually comes with it.

To install the proprietary driver, refer to the user guide that comes with your device. Of course, the optional mouse that comes with your FTB-400 Universal Test System is fully functional.

*Note:* *It is not necessary to turn off the FTB-400 Universal Test System before connecting a USB pointing device. The software will automatically detect it.*

To install a serial (DIN-9) mouse on your FTB-400 Universal Test System:

Simply connect it to the serial port located on the left side of the mainframe.

IMPORTANT

You must restart the FTB-400 Universal Test System after installing a serial pointing device.
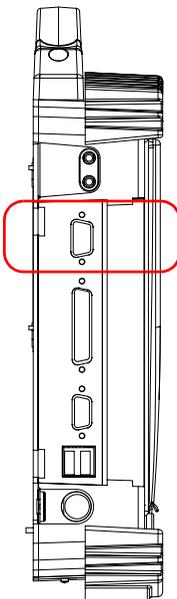
Installing a Headset or Speakers

The FTB-400 Universal Test System comes with an internal speaker for audible feedback, but there are times when you might want to use more powerful speakers. Other types of environments may warrant the use of a headset instead of the external speaker.

To connect a headset or speakers to your FTB-400 Universal Test System:

Plug the external speakers or the desired headset in the Audio Out port, located on the left side of the mainframe.
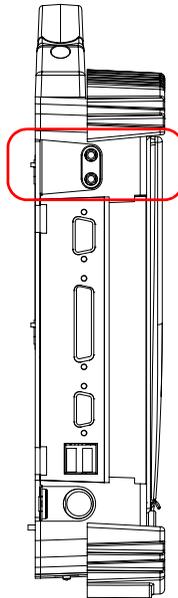
Note: *You can also use the headphone that comes with the FOT-920 MaxTester or the FTB-3920 and FTB-1400 MultiTest modules, but with an adapter.*

Installing a Microphone

On top of being a multitasking test platform, the FTB-400 Universal Test System is a portable multimedia computer. From the Windows environment, it can use off-the-shelf third party applications that require a microphone. With these applications, you can use the FTB-400 Universal Test System internal microphone or an external microphone.

To use an external microphone:

Simply connect it to the Audio In port, located on the left side of the mainframe.
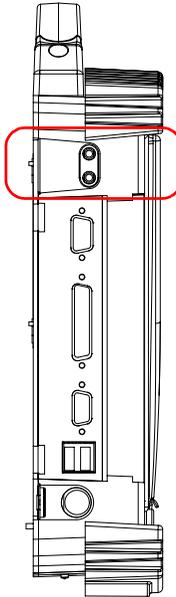
Installing a Printer

Should you need to print reports, and have not purchased the printer module for the FTB-400, you can install an external printer.

To install a parallel printer on your FTB-400 Universal Test System:

1. Connect the printer to the parallel port (DB-25), located on the left side of the mainframe.
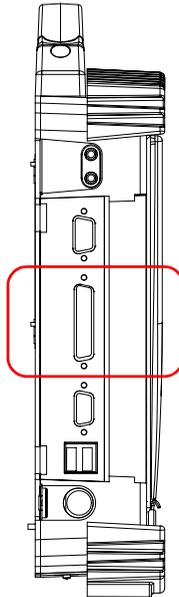
2. Turn on the FTB-400 Universal Test System.

3. Exit ToolBox and return to the Windows environment. From there, you can install your printer driver as explained in the user guide that came with your printer.

You can also print from printers linked to the FTB-400 Universal Test System via an infrared connection. This is done through the infrared port, located on the right side of the mainframe.
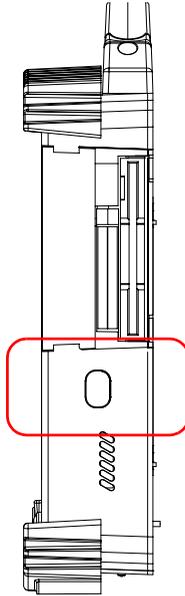
For the FTB-400 Universal Test System to locate the infrared-compliant printer, simply point the infrared port directly on the printer. This will make the printer appear in the available printers.

Installing a PCMCIA Card

The PCMCIA card reader found in the FTB-400 Universal Test System supports Type III PCMCIA cards such as:

➤ Fax/Modem

➤ SRAM and Flash memory

➤ hard drives

➤ Ethernet/Fast Ethernet

➤ GPIB

The PCMCIA Card reader supports the 32-bit Cardbus standard.

To install a PCMCIA card:

1. Turn on the FTB-400 Universal Test System.

2. Insert the PCMCIA Card in the PCMCIA Card reader, located on the right side of the mainframe. The PCMCIA Card is automatically recognized.
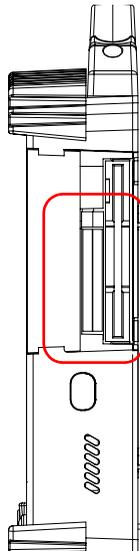
Installing an External CD-ROM Drive

To transfer small files such as traces, the floppy drive on your FTB-400 Universal Test System will be enough. However, for larger files that you want to copy to your FTB-400 Universal Test System, or to upgrade the software currently on your system, you might want to install an external CD-ROM drive (optional).

To install an external CD-ROM drive on your FTB-400 Universal Test System:

Connect it to a USB port, located on the left side of the mainframe.
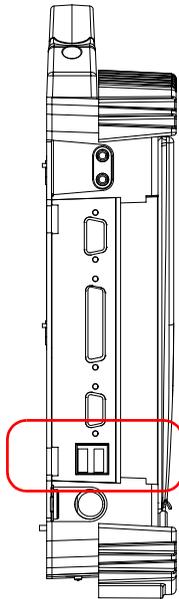
Although your external CD-ROM drive is automatically recognized and immediately usable, you might not benefit from all its special features unless you install the proprietary driver that usually comes with it. To install the proprietary driver, refer to the user guide that comes with your device.

Adjusting the Date and Time

You might need to set the time according to your location or time zone.

Note: *This function is linked to Windows, so changing a date or time in ToolBox will change the date or time in your whole system.*

To adjust the date and time:

1. Select the **Utilities** function tab in the main window.

2. Select the **System** tab.

3. Click on **Settings**.

4. Select the **Date and Time** tab.
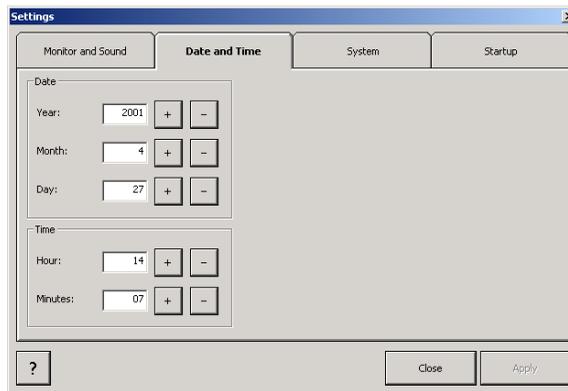
5. Change the date or time values with the + and – buttons besides the item you want to change.

Click on **Apply** to use the new settings. Click on **Close** to close the settings window without using the new settings.

Selecting the Startup Interface

At startup, you can have ToolBox 6 launch automatically or not.

To select the startup interface:

1. Select the **Utilities** function tab in the main window.

2. Select the **System** tab.

3. Click on **Settings**.

4. Select the **Startup** tab.
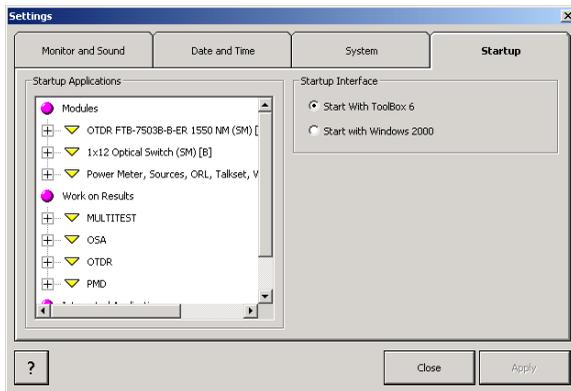
5. Select the interface to use on startup:

➤ ToolBox 6

➤ Windows 2000

6. Click on **Apply** to use the new settings. Click on **Close** to exit without using the new settings.

Note: *The new settings will only apply the next time your start your unit.*

Selecting the Startup Applications

You can have selected applications automatically launch upon starting the FTB-400 Universal Test System up. This saves you time, as you do not need to ensure that they are already running before starting your tests.

To set some applications to launch automatically at startup:

1. Select the **Utilities** function tab in the main window.

2. Select the **System** tab.

3. Click on **Settings**.

4. Select the **Startup** tab.
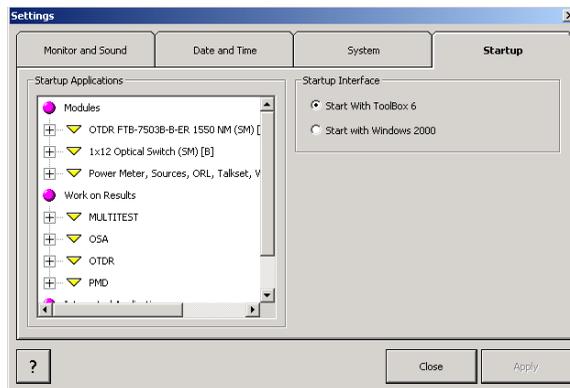
5. Select the items you want to launch at startup by checking the corresponding boxes. Navigate through the **Startup Applications** tree by clicking on the + and – signs on the left of the list.

6. Click on **Apply** to use the new settings. Click on **Close** to exit without using the new settings.

Note: *The new settings will only apply the next time your start your unit.*

Setting Passwords

For security reasons, you might want to set passwords for Administrator- and Standard-level users.

To set passwords:

1. Select the **Utilities** function tab in the main window.

2. Select the **System** tab.

3. Click on **Passwords**.

4. Select the **Passwords** tab.

5. Click on **Modify** to set or change the password for the access level of your choice. There is no minimum amount of characters, and you can use both letters and numbers.

6. Enter the new password in the **Password** field.

7. Confirm your password by typing it in the **Confirm** field.

8. Click on **Apply** to save the new password, or on **Cancel** to return to the **Passwords** tab without changing the password.

To activate the password access control:

Check the box next to the corresponding field.

The new password will be active on your next logon. You must have a password in both Administrator and Standard levels for the box to be enabled. If you do not set any password, or that you do not activate the password access control, ToolBox will automatically launch in Administrator mode.

Customizing Access Levels

Administrator-level users can give specific access rights to the Standard-level users.

To customize access levels:

1. Select the **Utilities** function tab in the main window.

2. Select the **System** tab.

3. Click on **Passwords**, then select the **Access Levels** tab.
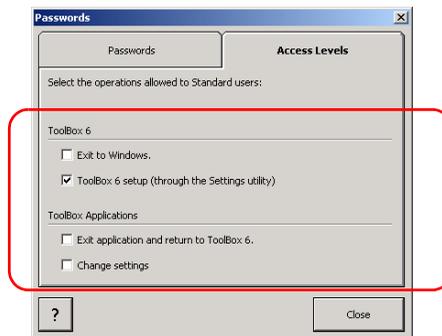
4. Select the item(s) you want the standard-level users to access by checking the corresponding boxes.

The changes will take effect upon the next standard-level logon.

Setting Communication Parameters

For information on how to configure your Universal Test System for remote control, see *Preparing for Automation or Remote Control* on page 77.

Reverting to Factory Settings

You might decide to revert to the factory settings in order to start with a "new" test system.

> ## IMPORTANT
>
> **This action cannot be undone.**

Note: *You can only use the **Revert to Factory Settings** function if you are logged in as an* Administrator-level *user. The button will be grayed out for the Standard- and Guest-level users.*

To revert to the factory settings:

1. Select the **Utilities** function tab in the main window.

2. Select the **System** tab.
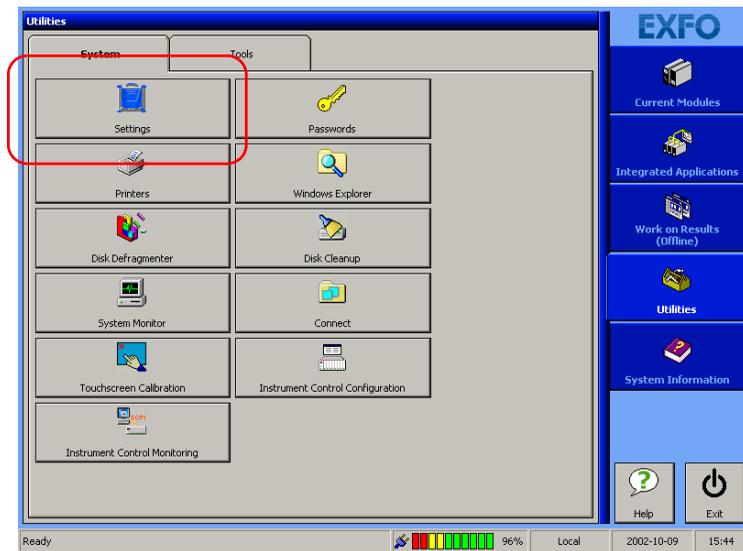
3. Click on **Settings**.

4. Select the **System** tab.
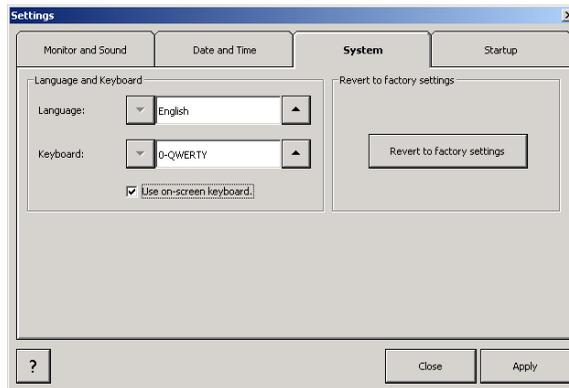
5. Click on **Revert to Factory Settings**.

5 *Using the ToolBox 6 Interface*

Each time you launch a new ToolBox 6 session, the **Current Modules** window appears.

Use this window to launch single-module applications. The applications will change depending on the modules you have in your FTB-400 Universal Test System.
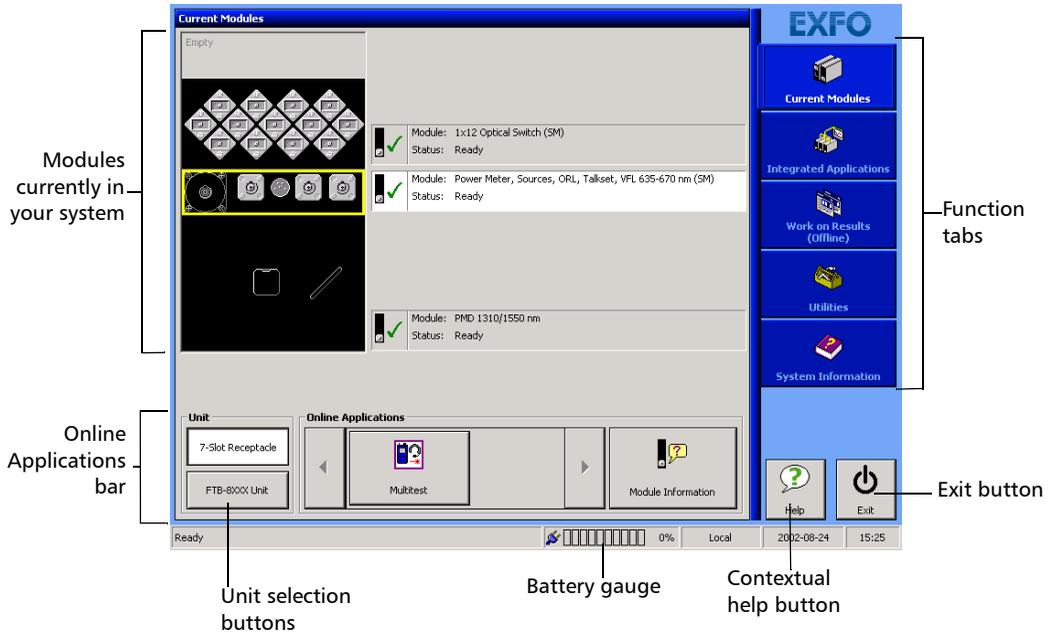
Note: *You can find more information on the use of the unit selection buttons in the FTB-8XXX modules user guides.*

Launching a Single-Module Application

Clicking on any installed module will make the **Online Applications** bar appear. This bar displays any application you can launch with the module.

If more than three applications are available for this module, the arrow buttons on each side will become active and allow you to browse through the other available applications.

If no modules are installed in your unit, the left part of the window, including the **Online Applications** bar, will be blank.

To launch an application:

Click on the corresponding icon in the **Online Application** bar.
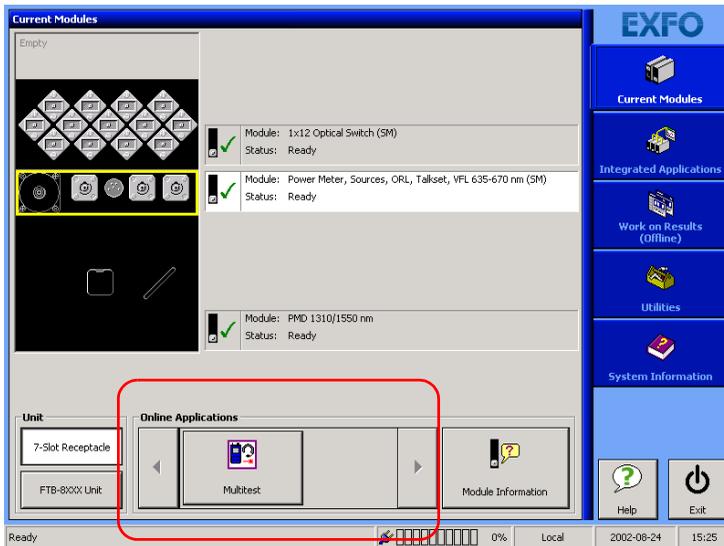
Launching Integrated Applications

The **Integrated Application** window will allow you to use applications requiring more than one module at a time. To access it, simply click on the corresponding function tab.
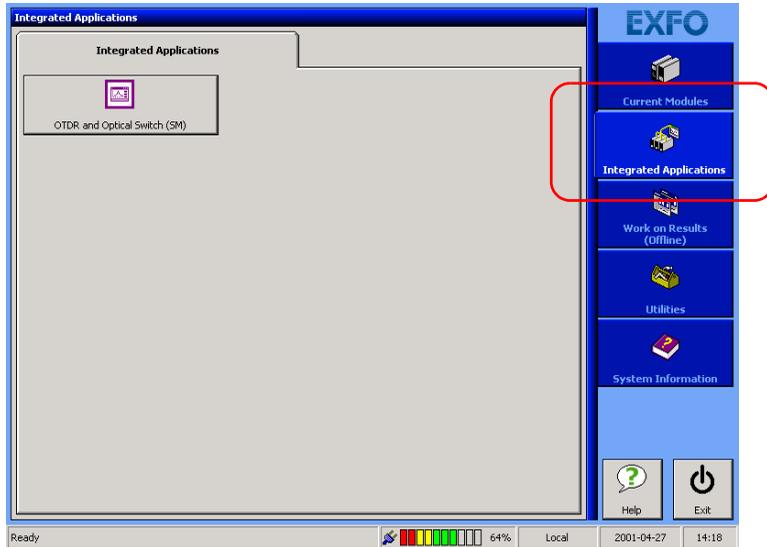
Applications will appear in the **Integrated Applications** tab only if the required modules are already inserted into the FTB-400 Universal Test System.

In order to start an application with more than one module, you must select which modules will be part of your application.

To launch an integrated application:

1. Click on the button corresponding to the application you want to set. A **Module Selection** window will appear, displaying each module installed in your unit.

Module selection grid

Selected module description bar

Module selected for the application

Available module

Empty module slot
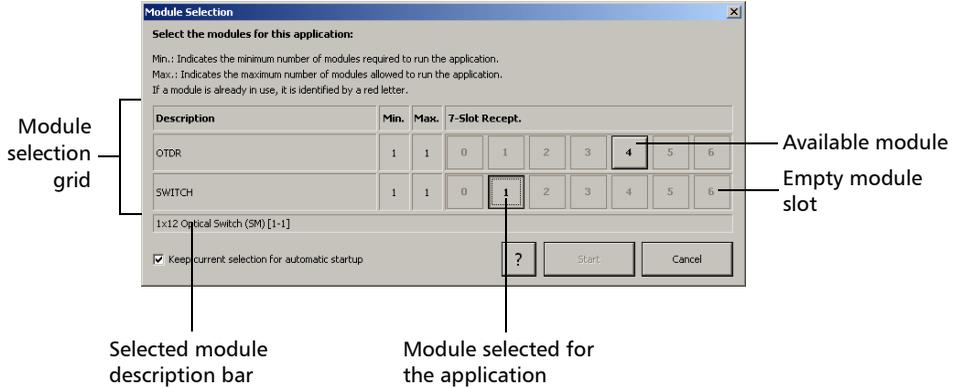
Note: *If the backplane slots of your FTB-400 Universal Test System are identified with letters, you will notice the following change. For ToolBox 6 versions 6.7 and later, the slot identification system for your seven-slot backplane was modified. Now, instead of letters A through G, the slots are identified by the numbers 0 through 6.*

Note: *The **Module Selection** window may look different depending on which modules are in your FTB-400 Universal Test System.*

2. Select the modules you want to use for your application according to the following details:

➤ Each line represents a different element of the application for which you have to select a module.

➤ Modules are represented by a button on which a number appears. The number corresponds to the slot in which the module is located.

➤ Grayed out numbers represent an empty slot or the space taken by a larger module (for example, a three-module optical spectrum analyzer will appear in slot 0, but slots 1 and 2 are grayed out, as they are also taken).

➤ The number of modules available will vary according to the number of modules you have installed in your unit at a given time.

➤ If a module is available for an application but not selected, the corresponding button will be pushed *out*. If the module is available for an application *and* selected, the corresponding button will be pushed *in*. By clicking on a module, you can see its name and position in the description bar, below the module selection grid.

3. To start the application, click on **Start**. The selected modules will become red to indicate that they are no longer available, and the **Module Selection** window will close.

4. To exit the **Module Selection** window without changing the configuration or starting the application, click on **Cancel** or the **X** button at the top right-hand corner.

Your settings *will not* be kept as you exit the window, unless you check the **Keep current selection for automatic startup** box, located at the bottom left-hand corner of the window.

Working on Test Results (Offline Mode)

Your FTB-400 Universal Test System allows you to perform jobs on acquisitons or tests that have already been performed. This is also known as the *Offline* mode.

You can access the **Work on Results applications simply by clicking on the corresponding function tab**.
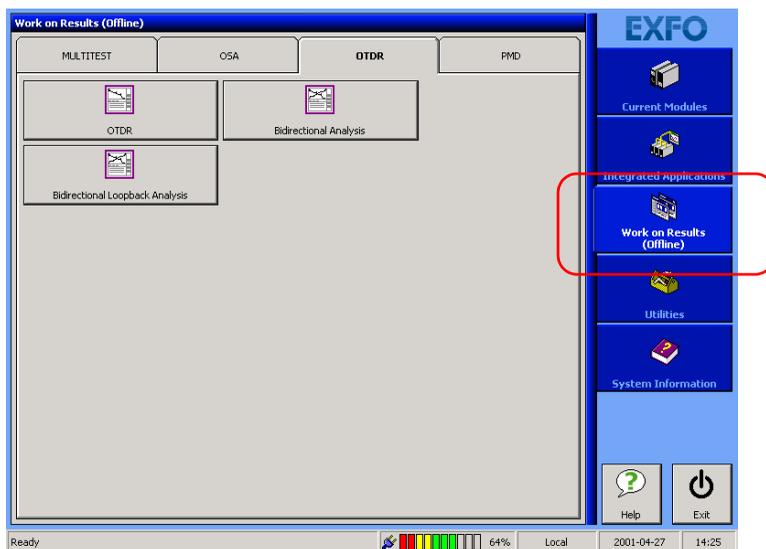
Each module has its own tab, on which you will find related applications. The applications will vary according to the modules you have installed in your unit. You do not need the module to be physically inside the FTB-400 Universal Test System, but the corresponding software must be installed.

For more information on installing and updating ToolBox, see *Installing or Upgrading ToolBox* on page 23.

Note: *You cannot perform acquisitions or tests in this window. You must use either the **Current Modules** window or the **Integrated Applications** window to do so.*

Accessing Selected Windows Utilities from ToolBox 6

Some of the most-frequently used Windows utilities have been incorporated into ToolBox 6 for a convenient access. You can reach these utilities by selecting the **Utilities** function tab, then the **System** tab.
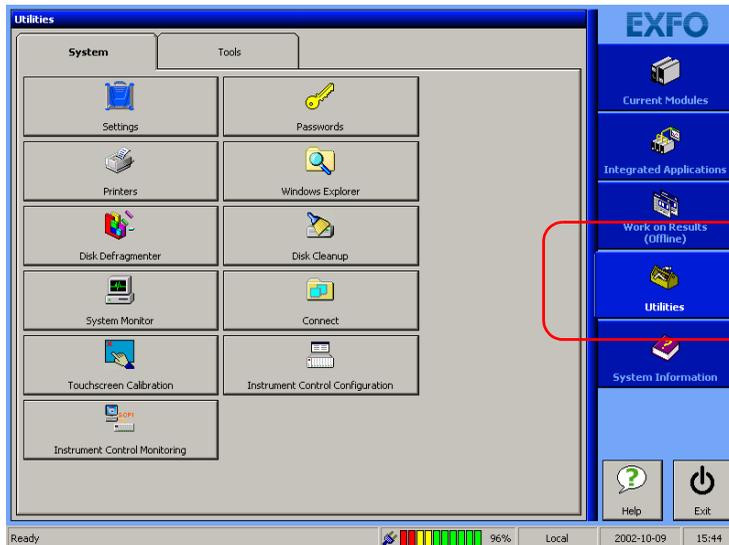
The table on the following page lists the available utilities, and gives a short description of their functions.

Note: *The Windows Control Panel is not in the **System** tab. To access it, you must do it directly from Windows, or use the Windows Explorer.*

Utility	Description
Printers	To set the default printer, and its settings. This is *not* used to print reports or test results. Each module has its own print function.
Windows Explorer	To explore and manage files.
Disk Defragmenter	To manage the space on your drive.
Disk Cleanup	To look for seemingly unused or useless files and clear them from your drive to free space.
System Monitor	To see the open applications, CPU usage and other system management-related information.
Connect	To connect to other systems through network access.

You can find more information on these utilities in the Microsoft Windows documentation.

6 *Transferring Data*

ToolBox 6 allows you to transfer your data to and from your unit. This is achieved through the Data Transfer utility.

Transferring Data Between Two PCs (or FTBs)

The Data Transfer utility allows you to transfer data between two PCs, a PC and an FTB-400 Universal Test System, or two FTB-400 Universal Test Systems.

To transmit files between your two units:

1. From the main window, click on the **Utilities** function tab.

2. Select the **Tools** tab.

3. Click on **Data Transfer**, then select **PC (FTB) to PC (FTB)**.
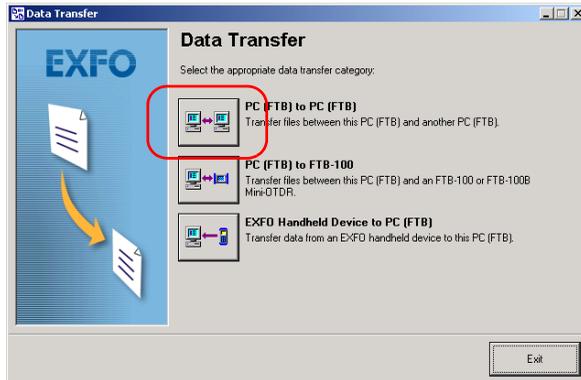
4. Make sure you connect your two units as explained on-screen.

5. Set the communication parameters by clicking on **Modify**.
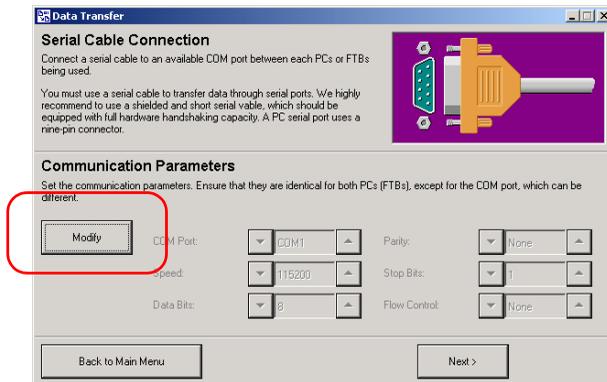
You can modify the following settings for your data transfer.

➤ **COM Port**: This is the port to use to transfer your data. Ensure that you select an available port.

➤ **Speed**: This is the baud rate used to transfer your data.

➤ **Data Bits**: The number of useful bits for the transmission.

➤ **Parity**: Allows you to set a parity for your transfer, if any.

➤ **Stop Bits**: This is the number of bits used at the end of each character sent.

➤ **Flow Control**: To select the flow control you want to use.

6. Click on **Next** to continue.

7. Select if you want to send or receive files by clicking on the related tab.

➤ If you want to send files to a remote PC or FTB-400 Universal Test System, select them in the list on the left, then use **>** and **>>** to move them to your transfer list. Use **<** and **<<** to remove them from the transfer list.

To remove the files from your local disk once the transfer is complete, check the corresponding box.

➤ If you want to receive files from a remote PC or FTB-400 Universal Test System, select the folder where you want to put the files.

You can select the new files to replace the existing files, or add a suffix of your choice to differentiate them from the existing files.
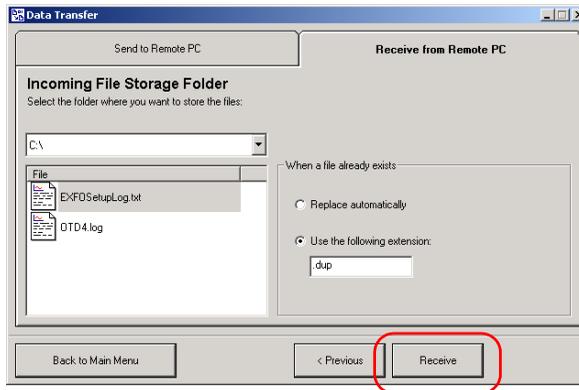
8. Click on **Receive** or **Transfer**, depending on which tab your are in, to start the file transfer process.
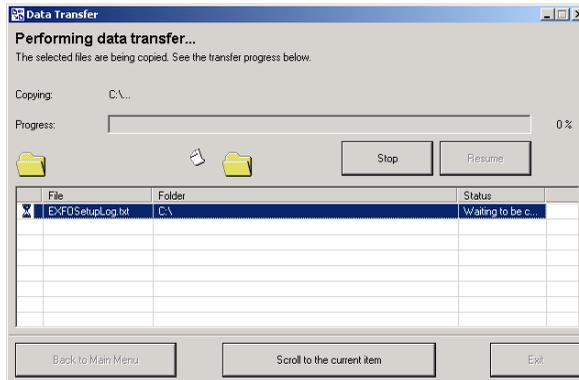
You can stop the transfer at any time by clicking on **Stop**.

The list of files you are transferring will be displayed on the lower part of the window. You can scroll to the item currently being transferred by clicking on **Scroll to the Current Item**.

Once the transfer is complete, or if you have clicked on **Stop**, click on **Exit** to return to ToolBox 6, or click on **Back to Main Menu** to return to the Data Transfer startup window.

Transferring Data From a PC (or FTB) to an FTB-100

The Data Transfer utility allows you to transfer data from a PC (or FTB-400 Universal Test System) to an FTB-100.

To transmit files from your PC to your FTB-100 unit:

1. From the main window, click on the **Utilities** function tab.

2. Select the **Tools** tab.

3. Click on **Data Transfer**, then select **PC (FTB) to FTB-100**.
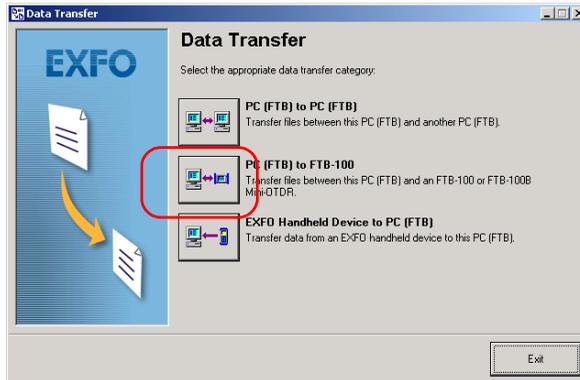
4. Select the COM port to use on your PC. Ensure that you select an available port.
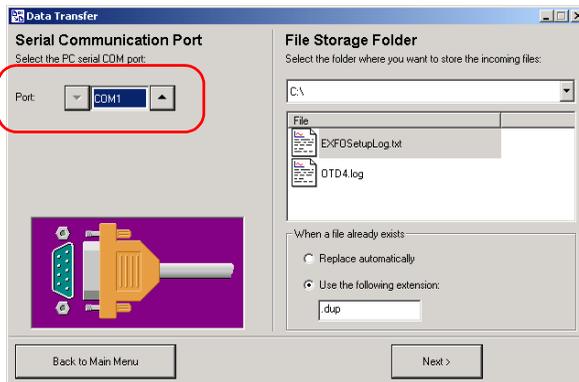
5. Select the folder where you want to put the files.

You can select the new files to automatically replace any existing files, or add a suffix of your choice to the new files to differentiate them from the existing files.

6. Click on **Next** to select the files to transfer.
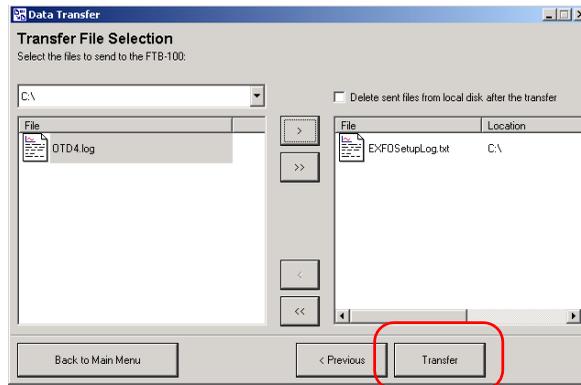
Select the files in the list on the left, then use the **>** and **>>** to move one or all files to your transfer list. Use **<** and **<<** to remove one or all files from the transfer list.

To remove the files from your local disk once the transfer is complete, check the corresponding box.

7. Click on **Transfer** to send your files to the FTB-100.

You can stop the transfer at any time by clicking on **Stop**.

The list of files you are transferring will be displayed on the lower part of the window. You can scroll to the item currently being transferred by clicking on **Scroll to the Current Item**.

Once the transfer is complete, or if you have clicked on **Stop**, click on **Exit** to return to ToolBox 6, or click on **Back to Main Menu** to return to the Data Transfer startup window.

IMPORTANT

At this point, the FTB-100 will not allow more transfers, as it considers that the transfer is complete. To resume the communication with the FTB-100, you must unplug the serial cable from it, then replug it.

Transferring Data From a Handheld Device to a PC (FTB)

Using your unit's serial port, you can download data via RS-232 to a PC. Doing so allows you to increase your storage capacity, to perform better analyses on test results, and to create reports with the ToolBox software.

Note: *You can transfer data to an FTB-400 (ToolBox 6.x), an FTB-300 (ToolBox 5.3 to 5.5), or a PC (Windows 95/98 with ToolBox 5.3 to 5.5 and Windows 98SE/2000/XP with ToolBox 6.x). Instructions given in this section are usually valid for all platforms.*

To prepare your PC and handheld unit for data transfer:

1. While both the PC and the handheld unit are turned off, connect your unit to the PC using an RS-232 cable as shown.

2. Turn on both the PC and the handheld unit.

3. Install ToolBox as explained at the end of this user guide.

4. Configure your communication ports in the Windows Control Panel as explained in your Microsoft Windows documentation.

Left side of the FTB-400

Serial port
(male DB-9)

RS-232
port
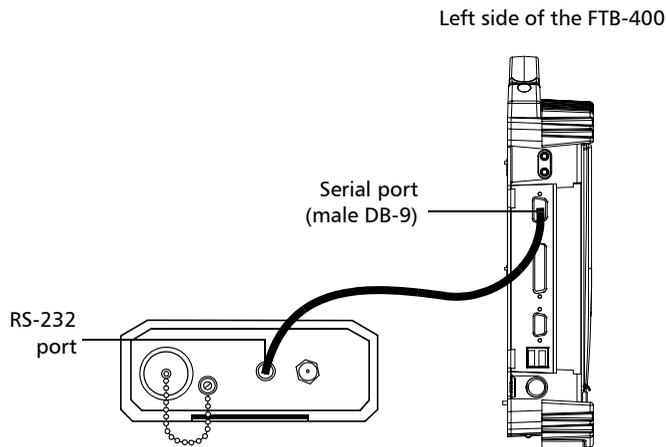
To download data from your unit to a PC:

1. Launch the ToolBox Data Transfer utility.

> ***1a.*** From the Windows taskbar, click on **Start**, then select **Programs > EXFO > ToolBox**.

> ***1b.*** In ToolBox, go to the **Utilities** tab, then to the **Tools** tab.

> ***1c.*** Click on **Data Transfer** to launch the Data Transfer utility, then select the **EXFO Handheld Device to PC (FTB)** option.

While the handheld unit is being detected, the **HandHeld operations** dialog box is displayed. The software will access the data stored into the unit and list fiber names under **Data list** in the main window.

Note: *If a communication problem occurs, refer to the Microsoft Windows documentation for details about setting communication parameters.*

2. Move the fiber files from the **Data list** to the **Download list** using **<<<**, **>>>**, **Clear**, and **All**. The files include all data stored on your unit.
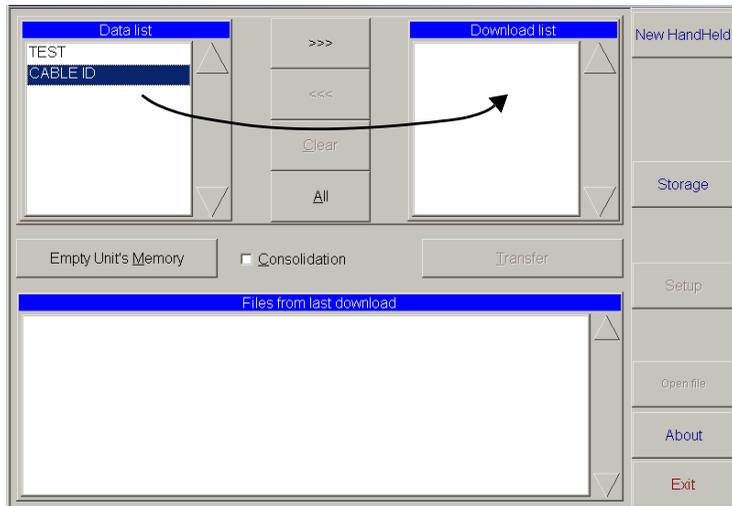
3. Select the folder in which you want to save the downloaded data.

   3a. Click on **Storage** to access the **Storage** window.

   3b. Click on **Storage Path**, select the drive and folder in which you want to save the downloaded data, and click on **OK**.

   3c. Click on **Exit Storage**.

4. Check the **Consolidation** box to overwrite older test files (when several files have the same name).

5. Click on **Transfer**. Data files appear in the **Files from last download** list and are saved in the selected folder on the PC.
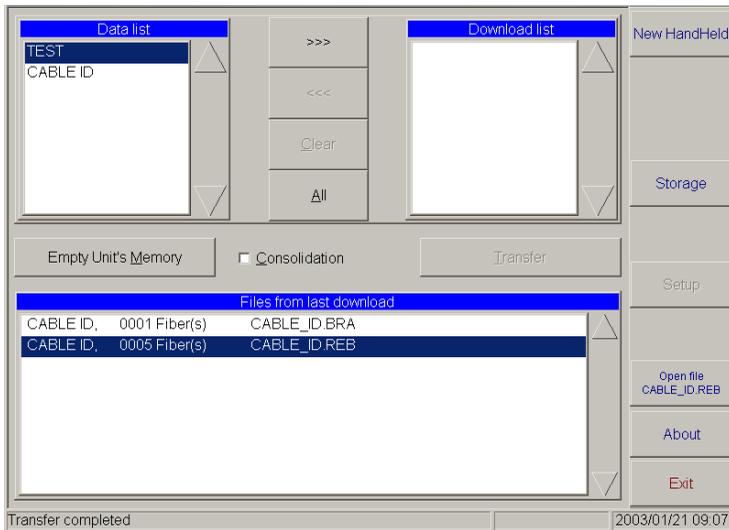
Note: To transfer other data, click on **New HandHeld** and repeat the above steps.

6. If you want to analyze the results from a downloaded file right away,

    6a. Select the file in the **Files from last download** list.

    6b. Click on **Open file <filename>** to start the MultiTest application.

To clear the handheld unit's memory while you are using ToolBox:

1. Launch the ToolBox Handheld Download utility.

2. Click on **Empty Unit's Memory**.

Transferring Data Using the Infrared Port

Your FTB-400 Universal Test System is equipped with an infrared port, which you can use to transfer files. This method is very useful, as it is faster than the Data Transfer utility.

To use the infrared port for transfers:

1. Ensure that both devices are running in Windows and not ToolBox. To access Windows from ToolBox, see *Exiting ToolBox 6* on page 26.

2. Locate the infrared ports for both devices, and position them so that they are able to detect each other.

Note: *Your FTB-400 Universal Test System's infrared port is located on the right panel of the mainframe.*
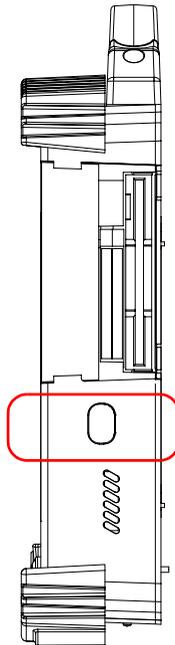
3. Once the devices have detected each other, an icon will appear in Window's taskbar. Double-click on the icon and a transfer window will appear.

4. Select the files you want to transfer and click on **Send**.

5. A confirmation window will pop up on receiving device's screen. By clicking on **Yes**, you will automatically save the files in Windows' default folder (the desktop).

7 *Preparing for Automation or Remote Control*

The FTB-400 Universal Test System was designed to meet the requirements of automation and to facilitate its integration with your test environment.

Since FTB-400 Universal Test Systems are based on Windows 2000, they can act as the system controller of an automated test setup (possibly including third-party instruments).

EXFO supplies commands that follow the guidelines determined by the SCPI consortium and LabVIEW drivers for many instruments. EXFO also supplies COM properties and events allowing you to build your own application.

The instruments can be controlled either locally or remotely via the following technologies:
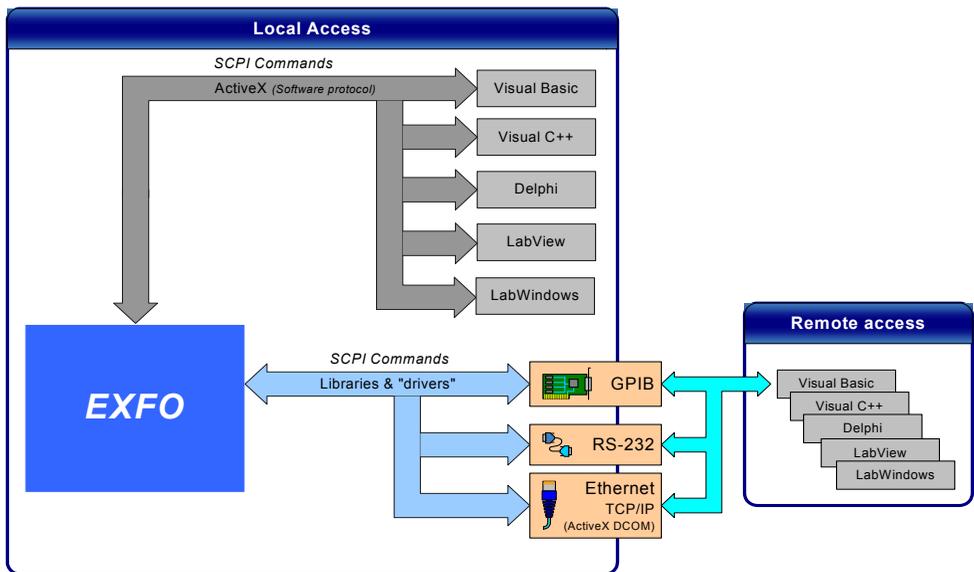
The choice of a technology will depend on your particular needs.

Control	Technology	Characteristics
Local	ActiveX (COM)	➤ Allows you to develop an application that will run locally on the FTB-400 within Windows 2000 ➤ Best approach when speed is your top priority (no physical connection that slows down the process) ➤ Supported by most development software ➤ Lower cost
Remote	Ethernet, TCP/IP (ActiveX DCOM)	➤ FTB-400 must be equipped with a PCMCIA Ethernet card to be connected to a LAN or WAN ➤ No need for a GPIB card ➤ Allows the sharing of network resources ➤ Allows you to develop PC-based applications to directly communicate with the FTB-400
Remote	GPIB	➤ FTB-400 must be equipped with a PCMCIA GPIB card and the appropriate drivers (included in the GPIB option) ➤ FTB-400 can be used as a device in a GPIB chain (it meets the IEEE 488.2 standard) ➤ Most commonly used interface to simultaneously control many instruments
Remote	RS-232	➤ Null-modem cable required to establish connection between the PC and the FTB-400 ➤ For increased speed and performance, run the application locally on the FTB-400 through ActiveX instead of using RS-232

For more information on programming aspects, see *Using FTB Products in an Automated Test Environment* on page 99.

Preparing Hardware for GPIB Control

If you intend to use GPIB to remotely control your instruments, your controller must have been equipped with the optional PCMCIA GPIB card.

Note: *EXFO supports the National Instruments PCMCIA card only. Other cards will not allow you to control EXFO FTB-400 instruments.*

To ensure the optimum efficiency of your system, EXFO recommends that you follow these restrictions:

For the IEEE 488.1 protocol:

➤ Maximum of 15 devices physically connected to each GPIB bus.

➤ Maximum separation of 4 m between two devices and an average separation of 2 m over the entire range (bus).

➤ Maximum total cable length used in the system: 20 m.

➤ Number of connected devices that must be powered on: at least two-thirds of the devices.

For the HS488 protocol:

➤ Maximum of 15 devices physically connected to each GPIB bus.

➤ Maximum total cable length used in the system: 15 m (i.e. 1 m per device physically connected to the GPIB bus).

➤ Number of connected devices that must be powered on: all of the devices.

➤ For each meter of cable used in the system, there must be a device connected to it or an equivalent device load.

The FTB-400 Universal Test System configuration complies with the IEEE 488.1 (also known as IEC60625.1) and the IEEE 488.2 (also known as IEC60625.2) standards to the extent shown in the following table.

Mnemonic	Function
SHE1[a]	Complete source handshake extended capability
AHE1[b]	Complete acceptor handshake extended capability
T6	Basic talker with serial poll; unaddressed to talk if addressed to listen
L4	Basic listener; unaddressed to listen if addressed to talk
SR1	Complete service request capability
RL2	No local lockout capability
PP0	No parallel poll capability
DC1	Complete Device Clear capability
DT0	No Device Trigger capability
C0	No controller capability
E2	Three-state driver capability

a. SHE1 corresponds to the extended capability of SH1, defined in the IEEE 488.1 standard. This extended capability is made possible by the device capability to use the HS488 high-speed protocol.
b. AHE1 corresponds to the extended capability of AH1, defined in the IEEE 488.1 standard. This extended capability is made possible by the device capability to use the HS488 high-speed protocol.

Linking Units with the GPIB Port

If your FTB-400 Universal Test System is equipped with a PCMCIA GPIB card, use the provided cable to link it to the other unit with which you want to perform remote control. For more information about your GPIB card, refer to the documentation provided with it.

PCMCIA Type III
card reader

Mainframe
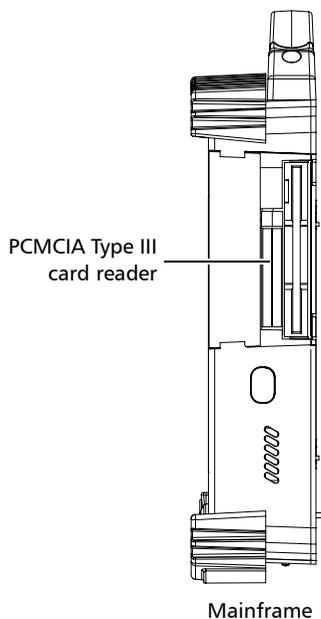
Linking Units with the Ethernet Port

The PCMCIA 10/100 LAN card must be inserted into your FTB-400 to send and receive data. Refer to the Windows documentation for information about Ethernet port settings and possibilities.

PCMCIA Type III
card reader

Mainframe
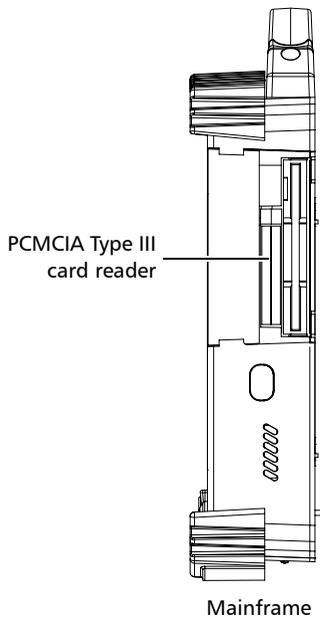
Linking Units with the Serial Port

Your FTB-400 Universal Test System is equipped with a serial (RS-232) port to send and receive data. Refer to the Microsoft Windows documentation for information about serial port settings and possibilities.

Serial port (male DB-9)

Mainframe
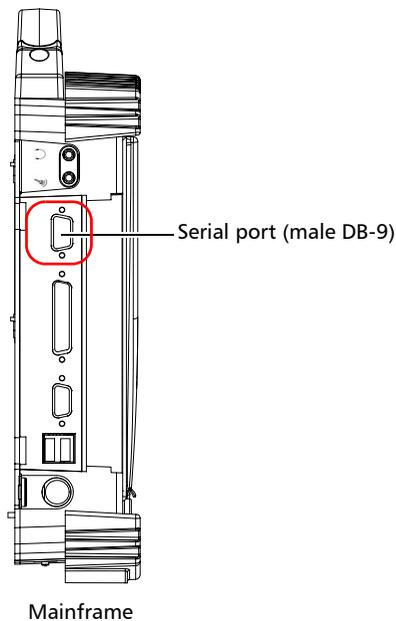
Getting Optimum Performance from Your Universal Test System

Several factors influence the data transfer rate of a FTB-400. The information presented hereafter will help you get the best transfer rate possible.

➤ *Protocol (GPIB control only)*: You can use the IEEE 488.1 (standard) or the HS488 (high-speed) protocol. Both protocols can co-exist in the same system; communication between devices can then be achieved using the IEEE 488.1 protocol.

HS488 is a non-interlocked handshake protocol that allows data transmission between devices at higher rates (six to seven times faster for small systems) than those possible with the IEEE 488.1 protocol. Higher transfer rates are particularly interesting in systems containing devices that return long data blocks. However, the transfer rate is affected by the physical limitations of the cables used in the system.

For information on cabling, see *Preparing Hardware for GPIB Control* on page 79.

➤ *Bus timing (GPIB control only)*: Required if you intend to work with IEEE 488.1, for it affects the handshake rate of this protocol. It indicates the minimum amount of time a GPIB controller waits, once data is placed on the bus, before setting the data valid bus line (DAV) to true.

The possible bus timings are:

Mode	Bus timing delay
Normal	1100 ns
High speed	500 ns
Very high speed	350 ns

If your setup supports it, choose the very-high-speed timing to get the best performance.

➤ *Output unit (GPIB, RS-232 and ActiveX)*: The FTB-400 can return results in

➤ linear units (e.g.: in watts)

➤ log units (e.g.: in dBm)

Since internal units are linear, you will get optimal performance by using linear units for output (no need for an internal conversion to log).

Note: *You must make the choice of output unit for each instrument offering such a feature. Refer to the user guide of each optical instrument for a list of available commands and queries.*

➤ *Output format (GPIB, RS-232 and ActiveX)*: The FTB-400 provides the following output formats for measurement results:

➤ ASCii

➤ PACKed

Generally, the PACKed format allows to pass three to four times more information than the ASCii format for the same transfer rate. Often, the PACKed format is also more efficient since it reduces the FTB-400 CPU work load (no need for an internal conversion to ASCII format).

Note: *The PACKed format will only be applied to <DEFINITE LENGTH ARBITRARY BLOCK RESPONSE DATA> and <INDEFINITE LENGTH ARBITRARY BLOCK RESPONSE DATA>.*

Note: *The choice of data format cannot be made directly via the ToolBox software.*

For more information on how to set the output format and data types, see *:FORMat[:DATA]* on page 209, *Read* on page 247, *ReadBinary* on page 248 and *Data Types* on page 165.

Changing Communication Settings

Communication settings cannot be modified without turning on your FTB-400 Universal Test System and launching ToolBox.

To change communication settings:

1. From the **Utilities** function tab, select **Instrument Control Configuration**.
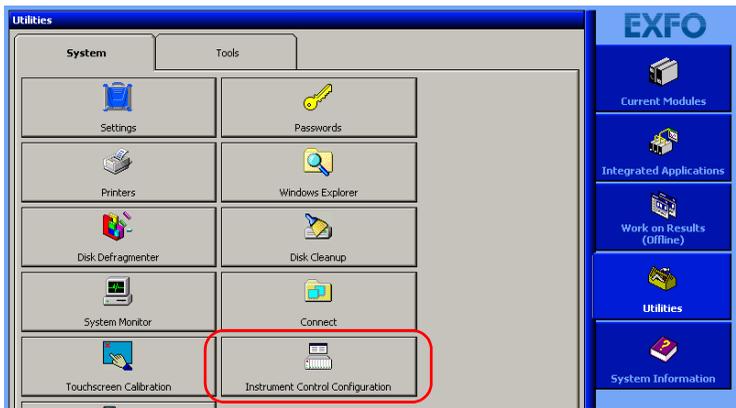
2. Determine whether your module applications will be launched locally or remotely by selecting the corresponding option button in the **Platform Startup** group box.

➤ If **Local** mode is selected, you will not be able to send remote commands to your FTB-400.

➤ If **Remote** mode is selected, all of the modules in your FTB-400 will be initialized upon startup so you are ready to send remote commands.

Note: *If you selected Local mode, you can go directly to step 5.*

3. Select GPIB, ActiveX or RS-232 from the **Communication Type** group box. For more information on a choice of a particular type, see the table presented on page 78.

Note: *If the selected communication type does not match the protocol that will actually be used, an error message will be displayed when attempting to control the instruments.*

4. If you selected GPIB or RS-232 as the communication type, you can customize the corresponding parameters as shown below.

➤ For GPIB

GPIB board (card)

The only value available is GPIB0.

Primary GPIB address

Ensure that the selected address is not already being used by the GPIB bus controller or by any other device connected to the GPIB bus. Default value is 12.
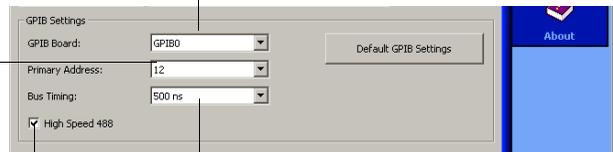
Bus timing (in ns)

High-speed mode

Check the box to enable high-speed mode.

For information on bus timing or on high-speed mode, see *Getting Optimum Performance from Your Universal Test System* on page 84.

➤ For RS-232

End-of-string (EOS) character

Character used to indicate the end of a data string (when sending or receiving data).

Serial (RS-232) port

Port used to connect the RS-232 cable. The only value available is COM1.

Baud rate

Determines the speed at which data is sent between the FTB-400 Universal Test System and a computer, in bits per second (bps).
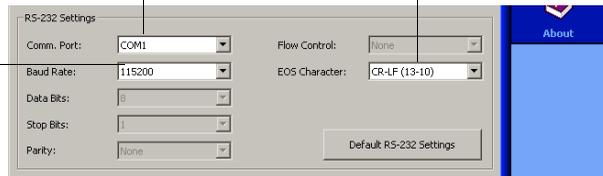
5. Click on **Apply** to confirm your changes.

6. To close the **Communication Settings** window, click on **Exit**.

To revert to default general settings:

1. Click on the **Restore Default Settings** button.
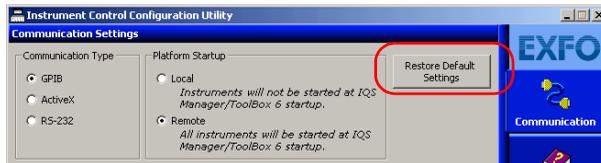
2. Click on **Apply** to confirm your changes.

To revert to default GPIB settings:

1. Click on the **Default GPIB Settings** button.
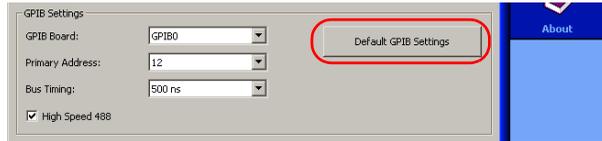
2. Click on **Apply** to confirm your changes.

To revert to default RS-232 settings:

1. Click on the **Default RS-232 Settings** button.
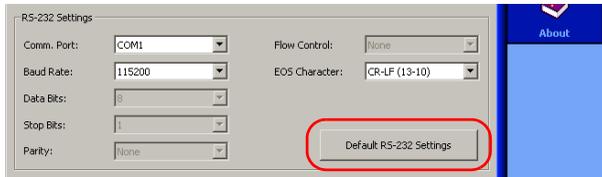
2. Click on **Apply** to confirm your changes.

Preparing for Remote Control with DCOM

DCOM technology allows to control devices and optical instruments via Ethernet. The IcSCPIActiveX.dll provided with your FTB-400 Universal Test System acts as a communication link between a client application and EXFO's Instrument Control.

DCOM ensures communication between the client application and Instrument Control via your local network. Since each network has its own configuration, you need to be familiar with network security, users, groups, domain management, etc. Basic programming skills are also required to work with DCOM. For more information, you can refer to the Microsoft MSDN Help feature, which provides exhaustive technical documentation on all DCOM issues.

The example presented in the following pages illustrates how to make the IcSCPIActiveX.dll component available to all users of a local network under Windows 2000. The example provided below is for guidance only; it may not work properly with all networks and interfaces may slightly differ depending on the operating system used.

To prepare your FTB-400 for remote control:

1. In *C:\WINNT\system32,* find the DCOMCNFG.EXE utility. Double-click on the filename to open the **Distributed COM Configuration Properties** dialog box.

2. Select **EXFO IcSCPIAccess Class** from the applications list and click on the **Properties...** button.
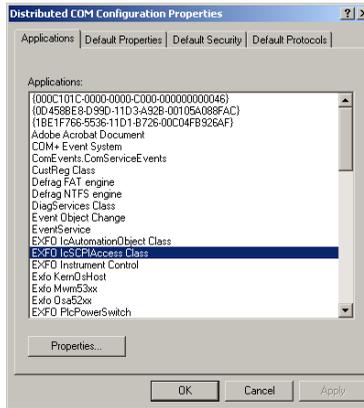
3. In the **EXFO IcSCPIAccess Class Properties** dialog box, select the **General** tab. From the **Authentication Level** box, select **Default**.

4. Configure the security parameters.

4a. In the **Security** tab, select the **Use custom access permissions** option, then click on the corresponding **Edit** button.
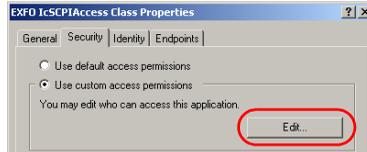
4b. From the **Registry Value Permissions** dialog box, click on the **Add...** button to grant access permissions.
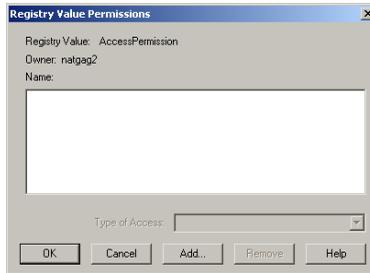
4c. From the **Add Users and Groups** dialog box, select **Everyone** from the list and click on the **Add** button. The name is transferred to the **Add Names** box.
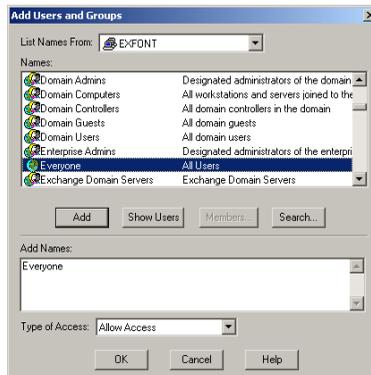
4d. Set the **Type of Access** to **Allow Access**. Click on **OK** to confirm. Click on **OK** once more to confirm the registry value permissions.

4e. In the **Security** tab, select the **Use custom launch permissions** option. Click on the corresponding **Edit** button.
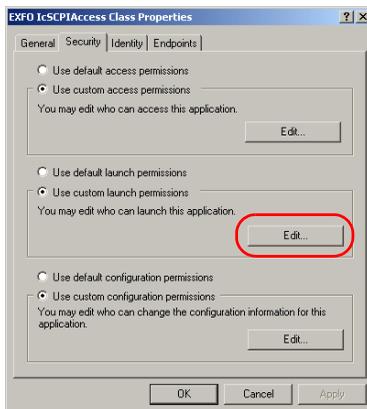
4f. From the **Registry Value Permissions** dialog box, click on the **Add...** button to grant access permissions.
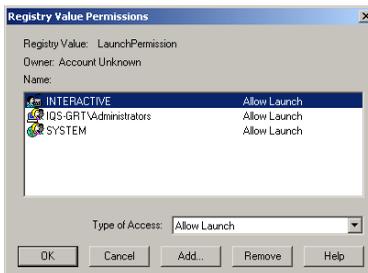
4g. From the **Add Users and Groups** dialog box, select **Everyone** from the list and click on the **Add** button. The name is transferred to the **Add Names** box.
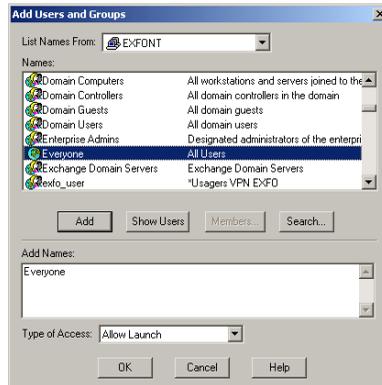
4h. Set the **Type of Access** to **Allow Launch**. Click on **OK** to confirm. Click on **OK** once more to confirm the registry value permissions.

5. Select the **Identity** tab, then select the **The interactive user** option. Confirm with **OK**.

In the **Distributed COM Configuration Properties** dialog box, click on **OK** to confirm the changes.

6. Restart your FTB-400.

The EXFO IcSCPIAccess Class component, located on your FTB-400 Universal Test System, can now be accessed with DCOM.

7. In *C:\Program Files\Common Files\EXFO\Instrument Control,* find the IcSCPIActiveX.dll file. Copy this file to the *C:\WINNT\system32* folder on the computer that will be used to control the FTB-400 Universal Test System.

8. On the computer that will be used to control the FTB-400 Universal Test System, register the IcSCPIActiveX.dll with the regsvr32.exe command:

    8a. On the Windows taskbar, click on **Start** and select **Run...**
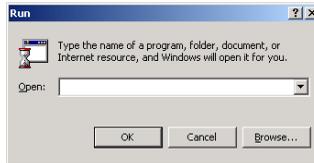
    8b. In the **Open** box, type "regsvr32.exe c:\winnt\system32\icscpiaccess.dll". Click on **OK** to register the dll file.

9. On the same computer, in *C:\WINNT\system32,* find the DCOMCNFG.EXE utility. Double-click on the filename to open the **Distributed COM Configuration Properties** dialog box.

10. Configure the default access permissions.

10a. Select the **Default Security** tab, then click on the **Edit Default** button from the **Default Access Permissions** group box.
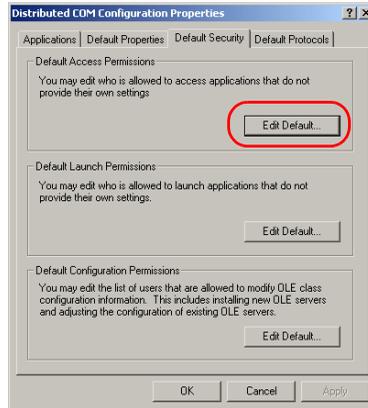
10b. From the **Registry Value Permissions** dialog box, click on the **Add...** button to grant access permissions.
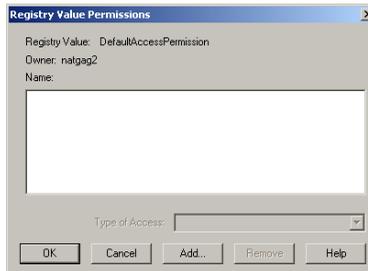
10c. From the **Add Users and Groups** dialog box, select **Everyone** from the list and click on **Add**. The name is transferred to the **Add Names** box.
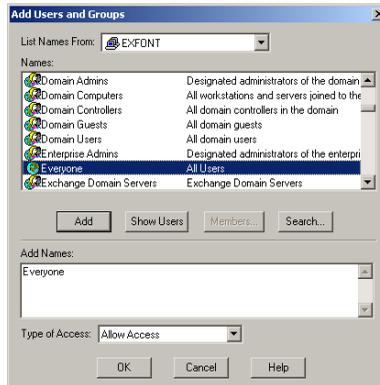
10d. Set the **Type of Access** to **Allow Access**. Click on **OK** to confirm. Click on **OK** once more to confirm the registry value permissions.

In the **Distributed COM Configuration Properties** dialog box, click on **OK** to confirm the changes.

The IcSCPIAccess interface can now be used with any programming language that supports DCOM.

8 *Using FTB Products in an Automated Test Environment*

EXFO supplies commands that follow the guidelines determined by the SCPI consortium and LabVIEW drivers for all available instruments. EXFO also supplies COM properties and events allowing you to build your own application.

Your application can be developed using LabVIEW, Visual C^{++}, Delphi, Visual Basic or any other language that runs under Windows 2000.

The present chapter gives you information to help you use the provided commands, drivers, and COM properties and events to remotely control your instruments.

If you need information on how to prepare your FTB-400 Universal Test System for remote control, see *Preparing for Automation or Remote Control* on page 77.

Managing the States of Your FTB-400 Universal Test System

Your FTB-400 Universal Test System can be in one of the following states:

State	Characteristics
Local	➤ Front-panel keys of the FTB-400 unit and peripherals are all functional ➤ Default state at device power on
Remote	➤ Front-panel keys of the FTB-400 unit and peripherals are *not* functional except for the Enter button and the power switch
Local with lockout	➤ Return to Local mode only possible via commands sent from the system controller ➤ Pressing the front-panel Enter button causes an URQ message (see *Standard Status Data Structure* on page 108) ➤ In GPIB, this state is not available when the device has the RL2 capability (see *Preparing Hardware for GPIB Control* on page 79)

State	Characteristics
Remote with lockout	➤ Front-panel keys of the FTB-400 unit and peripherals are *not* functional except for the power switch
	➤ Return to Local mode only possible via commands sent from the system controller
	➤ Pressing the front-panel Enter button causes an URQ message (see *Standard Status Data Structure* on page 108)
	➤ In GPIB, this state is not available when the device has the RL2 capability (see *Preparing Hardware for GPIB Control* on page 79)

Note: *The current state is displayed in the ToolBox status bar located at the bottom of the window.*

The following diagram shows the possible transitions between the different states.
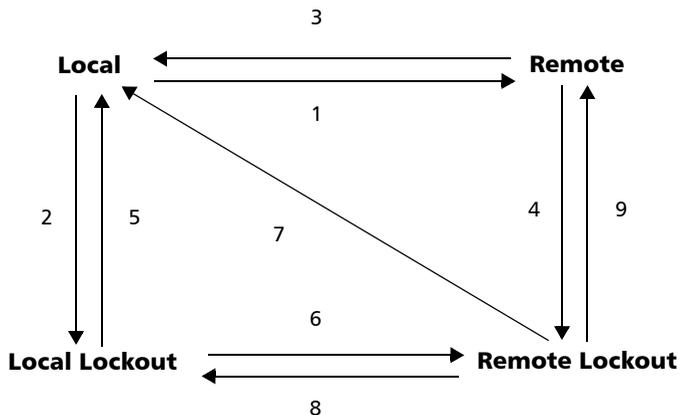
No.	Possible in GPIB if:	Possible in ActiveX by setting:
1	➤ The remote interface message (REM) is set to True. ➤ The device is addressed to listen (MLA). ➤ The return to local interface message (rtl) is set to False.	The *RemoteState* property to True (see *RemoteState* on page 250).
2	➤ The remote interface message (REM bus line) is set to True. ➤ The local lockout interface message (LLO) is set to True.	The *LockoutState* property to True (see *LockoutState* on page 251).

No.	Possible in GPIB if:	Possible in ActiveX by setting:
3	One of the following conditions is true: ➤ The go to local interface message (GTL) is received and the device is addressed to listen (MLA). ➤ The Clear Lockout/Set Local message is received (the REM bus line is set to False). ➤ The return to local interface message (rtl) is set to True and the local lockout interface message (LLO) is set to False.	The *RemoteState* property to False (see *RemoteState* on page 250).
4	The local lockout interface message (LLO) is set to True.	The *LockoutState* property to True (see *LockoutState* on page 251).
5	The Clear Lockout/Set Local message is received from the controller (the REM bus line is set to False).	The *LockoutState* property to False (see *LockoutState* on page 251).
6	The device is addressed to listen (MLA).	The *RemoteState* property to True (see *RemoteState* on page 250).
7	The Clear Lockout/Set Local message is received from the controller (the REM bus line is set to False).	-----
8	The go to local interface message (GTL) is received.	The *RemoteState* property to False (see *RemoteState* on page 250).
9	-----	The *LockoutState* property to False (see *LockoutState* on page 251).

Note: *You can also return to local state by turning the device OFF and ON again. However, this operation will cause the controller to lose control of the system (no more local lockout state). All the settings made on the device by the controller will also be lost (reset of device configuration at power up).*

Message Management

Each device that is physically connected to the GPIB link has its own input buffer, output queue and error/event queue. These data structures allow storage of incoming messages (single or compound commands that are sent to an instrument), responses from queries, errors and events that may occur.

Data structure	Characteristics	Clearing
Input buffer	➤ Consists of a First-In, First-Out (FIFO) data structure. ➤ Stores Data Bytes (DABs) and END messages. ➤ Delivers messages to the parser in the order that they were received from the I/O control. ➤ Maximum message length: unlimited in DABs (the input buffer size is only limited to the total size of the device memory). ➤ An incoming byte empties the output queue. An error will be raised if the output queue contained data. Consequently, it clears the Message AVailable bit (bit number 4 –MAV from the Status Byte register).	The buffer will be cleared by: ➤ Turning off the power. ➤ Sending a Device Clear (DCL) message to the instrument. ➤ Sending a Selected Device Clear (SDC) message to the instrument.

Data structure	Characteristics	Clearing
	➤ Except for the string and binary block contents, the following transformations are made on the incoming data: character conversion from lower case to upper case conversion of "<wsp>" characters to spaces conversion of multiple blanks to a single blank. ➤ Parser will begin to process messages when the <PROGRAM MESSAGE TERMINATOR> is received or if the input buffer is full.	

Data structure	Characteristics	Clearing
Output queue	➤ Consists of a First-In, First-Out (FIFO) data structure. ➤ When the instrument acts as a talker, it sends response messages (from the output queue) to the controller. Response messages all end with a *<RESPONSE MESSAGE TERMINATOR>*, see *Applicable Data Types for Output —IEEE 488.2* on page 175. ➤ Total storage capacity: only limited to the device's memory. ➤ As soon as there is data in the output queue, the Message AVailable bit (bit number 4 –MAV from the Status Byte register) is set to 1. ➤ Remains empty if no query is received or if the query contains an error.	The Output queue will be cleared by: ➤ Reading all the items it contains. ➤ Turning off the power. ➤ Sending a Device Clear (DCL) message to the instrument. ➤ Sending a Selected Device Clear (SDC) message to the instrument. ➤ Attempting to send a command before reading the responses to previous queries (an error will also be raised).

Data structure	Characteristics	Clearing
Error/Event queue	➤ Consists of a First-In, First-Out (FIFO) data structure.	The Error/Event queue will be cleared when:
	➤ Total storage capacity: 50 errors or events.	➤ Reading all the items it contains.
	➤ Errors or events can be retrieved, one at a time, with *:SYSTem:ERRor[:NEXT]?* (see *:SYSTem:ERRor[:NEXT]?* on page 219).	➤ Turning off the power. ➤ Sending a Device Clear (DCL) message to the instrument.
	➤ When an error or event occurs and the Error/Event queue is full, the last item in the queue (the most recent) is removed and the Queue overflow error (error –350) is added. No new items will be stored into the queue until there will be room available.	➤ Sending a Selected Device Clear (SDC) message to the instrument. ➤ Using the *CLS command (see **CLS* on page 188).
	➤ As soon as there is data in the output queue, the Error AVailable bit (bit number 2 –EAV from the Status Byte register) is set to 1.	

Standard Status Data Structure

Each device that is physically connected to the GPIB bus has four status registers with a structure complying with the IEEE 488.2 standard. These registers allow the controller to monitor events and get useful information on the status of the devices it controls.

➤ Standard Event Status Register (ESR) and
Standard Event Status Enable Register (ESE)

Bits	Mnemonics	Bit Value
7	Power On (PON)	128
6	User Request (URQ)	64
5	Command Error (CME)	32
4	Execution Error (EXE)	16
3	Device-Dependent Error (DDE)	8
2	Query Error (QYE)	4
1	Not Used (N.U.)	0
0	Operation Complete (OPC)	1

The following table presents a summary of the possible operations on ESR and ESE registers.

Register	Read	Write	Clear
ESR	Use *ESR? (see *ESR? on page 193).	Impossible to write.	➤ Use *CLS (see *CLS on page 188). ➤ Read the register.
ESE	Use *ESE? (see *ESE? on page 191).	Use *ESE (see *ESE? on page 191).	➤ Use *ESE (see *ESE on page 189) with a value equal to 0. ➤ At ToolBox startup, the register is set to 0.

➤ Status Byte Register (STB) and
Service Request Enable Register (SRE)

Bits	Mnemonics	Bit Value
7	Not Used (N.U.)	0
6	Master Summary Status (MSS)/Service Request (RQS)	64
5	Event Summary Bit (ESB)	32
4	Message Available (MAV)	16
3	Not Used (N.U.)	0
2	Error Available (EAV)	4
1	Not Used (N.U.)	0
0	Not Used (N.U.)	0

The following table presents a summary of the possible operations on STB and SRE registers.

Register	Read	Write	Clear
STB	➤ Use *STB? (see *STB? on page 204). ➤ Use serial poll (GPIB bus sequence that allows retrieval of the value without interrupting the current process).	Impossible to write; the register's contents is only modified when the Event registers or Queues are modified.	Use *CLS (see *CLS on page 188) before sending a query (to clear the Event registers and Queues and by the same token clear the STB register).
SRE	Use *SRE? (see *SRE? on page 202).	Use *SRE (see *SRE on page 200) with a value equal to 0 to disable the register or with a value equal to 1 to enable it.	➤ Use *SRE (see *SRE on page 200) with a value equal to 0. ➤ At ToolBox startup, the register is set to 0.

The diagram displayed on the next page is a useful aid in understanding the general commands and how a service request (SRQ) is generated.

Using a service request, a device notifies the controller that an event requiring special attention occurred. The controller will then find which device generated a SRQ (its RQS bit is set) and the causes of it.
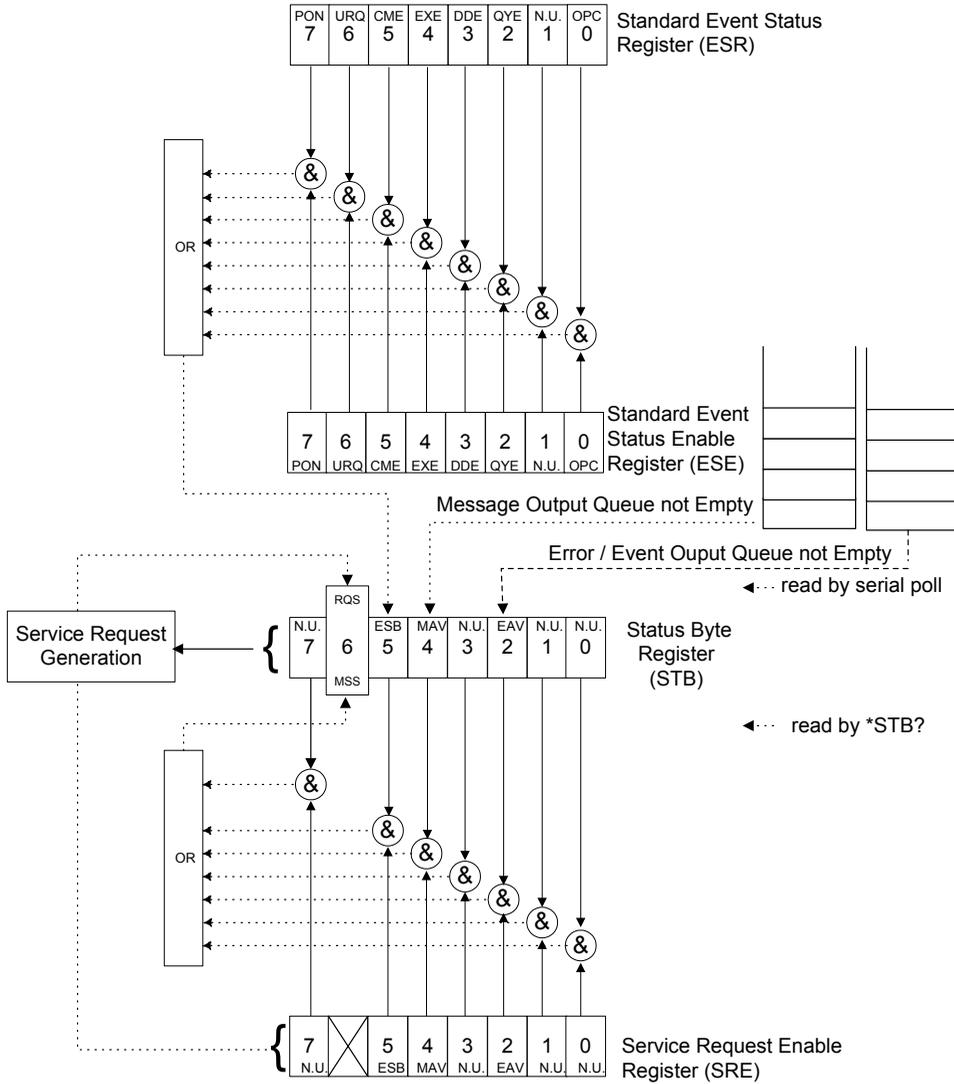
SCPI Command Structure

The information presented in this section provides an overview of GPIB programming. If you need detailed information, refer to:

➤ The International Institute of Electrical and Electronics Engineers. *IEEE Standard 488.1-1987, IEEE Standard Digital Interface for Programmable Instrumentation*. New York, 1987.

➤ The International Institute of Electrical and Electronics Engineers. *IEEE Standard 488.2-1992, IEEE Standard Codes, Formats, Protocols and Common Commands For Use with ANSI/IEEE Std. 488.1-1987*. New York, 1992.

➤ *Standard Commands for Programmable Instruments (SCPI). Volume 1: Syntax and Style*. Vers. 1999.0 May, U.S.A, 1999.

The provided commands follow the guidelines determined by the Standard Commands for Programmable Instruments (SCPI) consortium. A *program message* consists of one or more commands (and/or queries) with their appropriate parameters.

For example, a program message could contain a command used to activate or deactivate a source. The corresponding command syntax would be:

Mandatory keywords   Optional keyword (in square brackets)

SOUR:POW[:STAT]<wsp><Boolean Program Data>

Keyword separators   Required space   Parameter
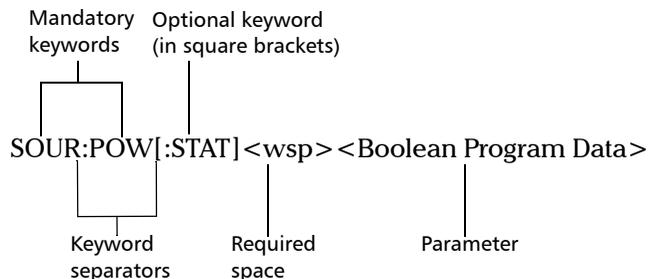
When sending a message containing the previous command, you would actually type: SOUR:POW ON.

The following table shows elements that are commonly used in the commands or queries syntax.

Item	Meaning
[ ]	Enclose optional keywords or parameters. *Do not include square brackets in your program message.*
[1..n]	Indicates that the instrument provides multiple capabilities and that you have to specify which one you want to use. If you omit the value, the command will take effect on the first capability. Multiple capabilities can be found at any branch of the command tree (root, intermediate node or terminal node). Example: If the command is :SENSe*[1..n]*:CORRection:COLLect:ZERO and you want it to take effect on the second SENSe (sensor) capability of the instrument, you may send this: :SENSe*2*:CORRection:COLLect:ZERO. *Do not include square brackets in your program message; simply enter the number.*
\<wsp\>	Indicates that a space is required ("wsp" stands for "white space"). Corresponds to ASCII character codes (0 to 9 and 11 to 32, in decimal). *Do not include "\<wsp\>" in your program message; simply type a space.*
\<digit\>	Element used in the construction of various numeric data types. Can take any value between 0 and 9 inclusively (corresponds to ASCII character codes 48 to 57, in decimal).

Item	Meaning
<mnemonic>	Element used in the construction of certain data types and program messages. In the diagram above, ➤ "<Upper/lower case alpha>" corresponds to ASCII character codes (65 to 90 and 97 to 122, in decimal). ➤ "_" corresponds to an underscore character (code 95, in decimal).
< >	Text appearing between angled brackets specifies the command parameter to be sent or the response you will receive from an instrument. *Do not include angled brackets in your program message.*
|	Indicates that one, and only one, value must be selected from the available choices. e.g.: If the list is 0|1, you can only choose 0 *or* 1. *Do not include the pipe character in your program message.*
{ }	Indicate that the enclosed parameters can appear 0 to n times when the command is used. *Do not include braces in your program message.*
:	Mandatory to separate keywords. Can be omitted at the beginning of a program message. For example, you can use either :SYST:ERR or SYST:ERR.

Item	Meaning
;	➤ Mandatory to separate the different commands of a program message when more than one command is sent at a time. In this case, it is called *<PROGRAM MESSAGE UNIT SEPARATOR>*. ➤ Also used to separate responses when multiple queries were sent in a single program message. In this case, it is called *<RESPONSE MESSAGE UNIT SEPARATOR>*.
,	➤ Mandatory to separate parameters in a command or a query. In this case, it is called *<PROGRAM DATA SEPARATOR>*. ➤ Also used to separate the various responses from a query. In this case, it is called *<RESPONSE DATA SEPARATOR>*.

There are also several conventions regarding command syntax:

➤ Spelling errors will cancel the command or query.

➤ Commands and queries are not case-sensitive. You can type your program messages using either lower-case or upper-case.

➤ The command or query can be written using only the three- or four-letter shortcuts, only full words, or a combination of both.

The example below shows the long and the short forms of a same query.

:SYSTem:ERRor? —————— Long form

:SYST:ERR? —————— Short form (using only the capital letters from the long form)

➤ For readability reasons, you can use extra spaces in your program messages but they won't be taken into account. For more information, see *Message Management* on page 104.

➤ You can build program messages allowing you to send more than one command at a time. Sometimes, you can omit the leading *<mnemonic>* path to simplify the program messages and speed up the search time (the parser saves the last position in the command tree).

saved *<mnemonic>* path   other*<mnemonic>* path

:system:version?;:system:error:next?

:system:version?;error:next? ——— *<mnemonic>* path omitted –also correct
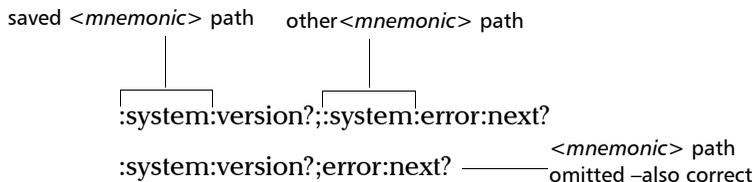
Paths cannot be omitted in all cases. The example below would cause an error.

saved *<mnemonic>* path   other *<mnemonic>* path

:system:error:next?;:system:version?

:system:error:next?;version? ——— Incorrect; *:version?* is not in *the :system:error* branch
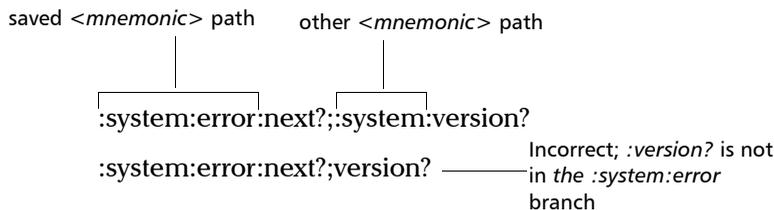
IEEE 488.2 required commands or queries (beginning with a *) that are part of the program message have no effect on the paths.

Example:

*:system:version?;*idn?;:system:error:next?* is equivalent to

*:system:version?;*idn?;error:next?*

Note: *Omitting the leading <mnemonic> path is only possible when you have more than one command or query in the program message that you send.*

➤ A program message must be ended with a <PROGRAM MESSAGE TERMINATOR>. For more information, see *Applicable Data Types for Input—IEEE 488.2* on page 166.

Consulting Data Types

If you need information about data types used in EXFO's documentation, see *Data Types* on page 165.

Writing Remote Control Code

Complex measurement programs may be written using any programming environment that supports GPIB communication. GPIB development kits are available for most of the popular commercial programming languages.

The FTB-400 Universal Test System offers many commands permitting complete remote control of all the FTB components. These commands adhere to the SCPI standard.

All the commands and queries supported by the FTB-400 can be found in *IEEE 488.2 and Specific Commands* on page 187. For information on commands specific to particular instruments, refer to the instrument's user guide.

When you write code, you must follow these rules on message reception and transmission:

➤ The controller must have sent a complete message to the instrument (including the message terminator) before retrieving a response.

➤ The controller must retrieve all the responses from previous queries (including the response terminator) before sending a new message to an instrument.

➤ The controller must not try to retrieve a response from an instrument if the corresponding query has not been previously sent to the instrument.

➤ Special attention must be paid to queries that return an indefinite ASCII response. To avoid any confusion, the IEEE 488.2 standard requires that this data type be immediately followed by a response termination character. For this reason, when working with compound queries, you must ensure that a query sending an indefinite ASCII response is the last query of the series.

➤ Be careful when sending program messages containing multiple queries that return large amounts of data. Since the controller can only retrieve data when the instrument has finished processing the queries, it could result in problems ranging from a saturation of the output queue to the complete blocking of the whole system.

Error Message Format

System and device-specific errors are managed by the FTB-400 Universal Test System. The generic format for error messages is illustrated in the following figure.
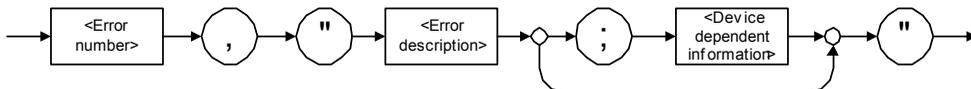
As shown in the above figure, the message contains three parts:

➤ error number

➤ error description

➤ device-dependent information

Error messages ending in a negative number are SCPI-based errors.

For more information on errors, see *Message Management* on page 104. For a complete list of possible errors, see *SCPI-Based Errors* on page 227.

Working with EXFO COM Objects

The *ToolBox* CD offers you a demo application to show how to communicate with your FTB-400 Universal Test System using COM technology via the provided IcSCPIAccess interface.

The source files can be found on the CD under: *F:\examples\ScpiActiveX*.

where *F* corresponds to the CD-ROM drive (the CD-ROM drive letter may differ from one computer to another).

This application has been designed with Visual Basic and, therefore, must be compiled using Visual Basic 6.0, Service Pack 5.
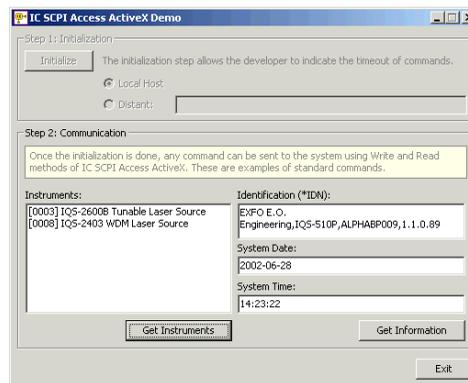
With this demo application, you can control your FTB-400 Universal Test System either in local (COM technology) or remote (DCOM technology) mode. If you want to work in remote mode, see *Preparing for Remote Control with DCOM* on page 90.

Working with EXFO LabVIEW Drivers

❗ IMPORTANT

You need to be familiar with the LabVIEW environment and programming methods to work with EXFO drivers.

Regardless of whether you work with the provided Getting Started applications or your own VIs (using EXFO drivers), the steps remain the same.

```
┌──────────────────────────┐
│   Set up communication   │
│       parameters         │
└──────────────────────────┘
            │
            ▼
┌──────────────────────────┐
│    Open communication    │
└──────────────────────────┘
            │
            ▼
┌──────────────────────────┐
│  Perform all operations on │
│        instrument        │
└──────────────────────────┘
            │
            ▼
┌──────────────────────────┐
│    Close communication   │
└──────────────────────────┘
```
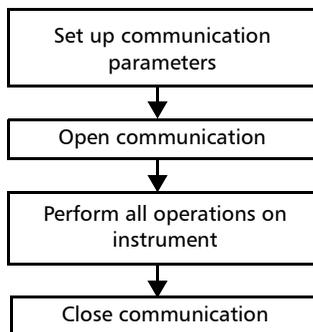
Before configuring the communication parameters via LabVIEW (provided applications or new VI), you must configure the FTB-400 Universal Test System for remote control. For more information, see *Preparing for Automation or Remote Control* on page 77.

❗ IMPORTANT

Ensure that the communication type that you will set in LabVIEW matches the one selected for the FTB-400 Universal Test System.

The following table presents the possible settings for communication parameters. These parameters must be set from LabVIEW for each instrument.

Parameter	Active X (local)	Active X (remote)	RS-232 (remote)	GPIB (remote)
Communication type	ActiveX	ActiveX	RS232	GPIB
VISA resource name	N/A	N/A	Select the serial port from the list	Select the GPIB device from the list
IQS slot number	Concatenation of the FTB-400 unit number (1) and the instrument's slot number.	Concatenation of the FTB-400 unit number (1) and the instrument's slot number.	Concatenation of the FTB-400 unit number (1) and the instrument's slot number.	Concatenation of the FTB-400 unit number (1) and the instrument's slot number.
Machine name	localhost	Ethernet address, i.e. IP address or machine name of your FTB-400	N/A	N/A

Note: *When you are working with platform or IEEE 488.2 commands, you can leave the IQS slot number at its current value.*

Using the EXFO Getting Started Applications

Once the LabVIEW drivers have been installed, the Getting Started demo applications are available to

➤ demonstrate how to open and close the communication link between the remote computer and the device

➤ present some of the available functions (by loading the necessary *.vi* files)

All the *.vi* files related to an instrument are presented in the same folder. By default, they can be found under:

C:\Program Files\EXFO\LabVIEW Getting Started\Getting Started xxxx

where xxxx corresponds to the product code (e.g., "2600b" for a tunable laser source).

Getting Started file


```
Name △
  Instrument2600bTest.vi
  Instrument2600bTest-CloseCommunication.vi
  Instrument2600bTest-ControlPtr.ctl
  Instrument2600bTest-Enabled.ctl
  Instrument2600bTest-InitializeCommunication.vi
  Instrument2600bTest-RefreshInterface.vi
  Instrument2600bTest-RefreshSettings.vi
  Instrument2600bTest-RefreshSourceState.vi
  Instrument2600bTest-RefreshWavelength.vi
  Instrument2600bTest-SetSettings.vi
  Instrument2600bTest-SetSourceState.vi
  Instrument2600bTest-SetWavelength.vi
  Instrument2600bTest-StartSweep.vi
  Instrument2600bTest-StateMachine.ctl
  Instrument2600bTest-StopSweep.vi
```

All Getting Started filenames follow this pattern: *InstrumentxxxxTest.vi* where xxxx corresponds to the product code.

You can also directly start a demo application this way:

From the Windows task bar, click on the **Start** button, then on **Programs > EXFO > LabVIEW Getting Started Applications > Getting Started** xxxx

where xxxx corresponds to the product code.

Each Getting Started application offers a user interface (called Front Panel) and a design view (called Block Diagram).

The Front Panel allows you to set communication parameters between the FTB-400 and the current instrument. It also offers various controls and buttons to use the instrument easily. In fact, the application performs the necessary calls to the instrument's drivers so it is transparent to user.

Examples of certain functions

Communication indicator

light green when connected

Communication parameters
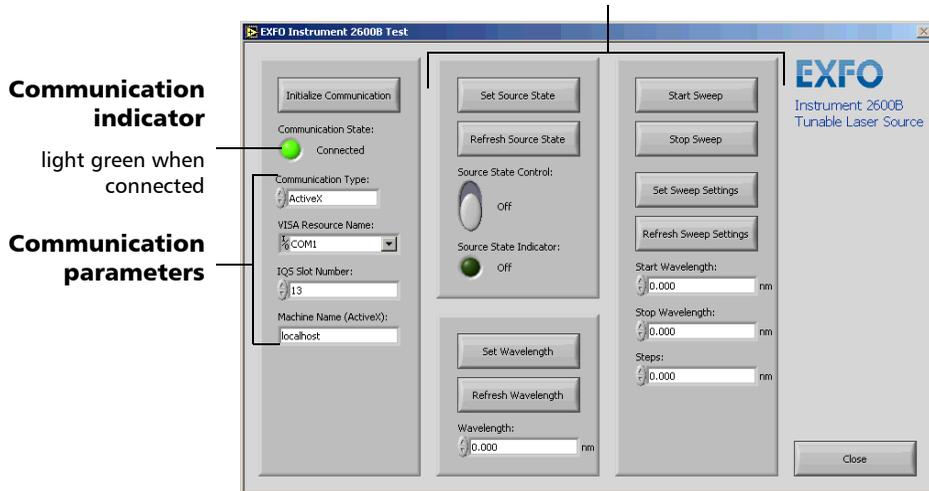
The instrument state (called State Machine) changes whenever an action is performed on the instrument. If you toggle to Block Diagram view, you can see the list of possible states. The instrument is always in one of the predefined states.

The following figure illustrates the State Machine after the user has clicked on the button allowing you to set the source state (from the Front Panel). When the instrument's State Machine changes to "SetSourceState", the application calls "Instrument2600bTest_SetSourceState.vi", which, in turn, calls the "SourcePowerState_GSet.vi" sub VI that will perform the appropriate action on the instrument.
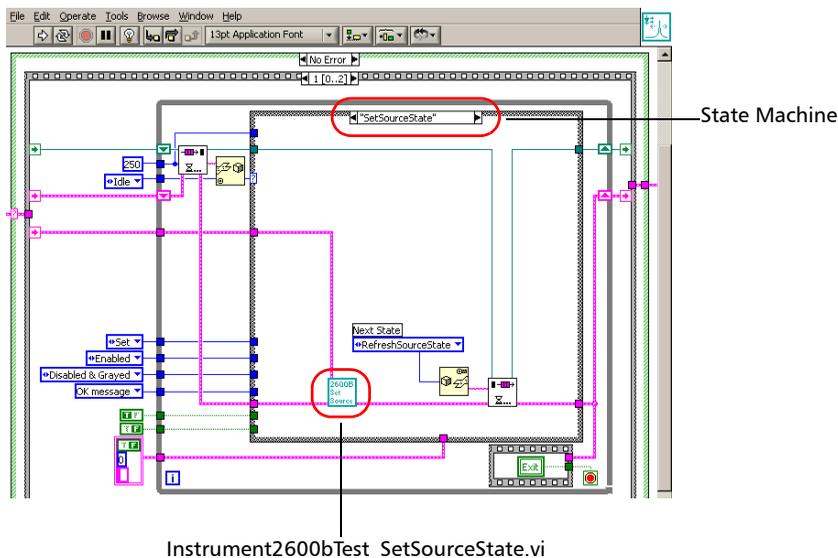
Instrument2600bTest_SetSourceState.vi

The detail of this sub VI gives precious information on how to call an instrument driver VI.

Instrument driver VI
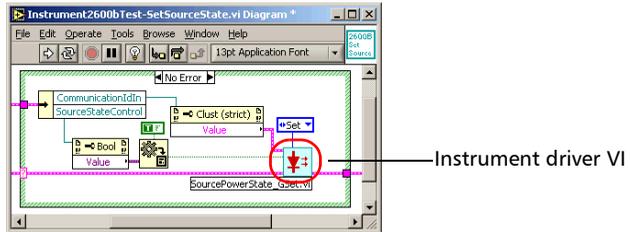
To use a Getting Started application:

1. Turn on the FTB-400, start ToolBox and ensure that all the remote-control parameters are set correctly.

2. Open the desired Getting Started application and run it from LabVIEW.

3. From the application's Front Panel, set the communication parameters. For information on communication parameters, see *Working with EXFO LabVIEW Drivers* on page 122.

4. Once the parameters are configured, select **Initialize Communication**.
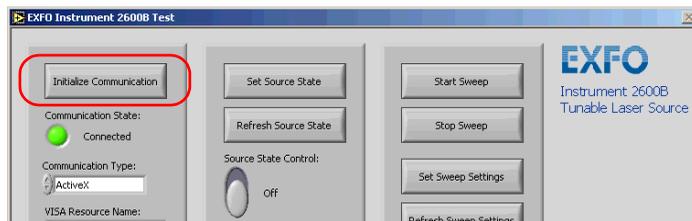
5. Using the provided buttons and controls, perform the desired actions.

6. When you are finished, select **Close** to end the communication.
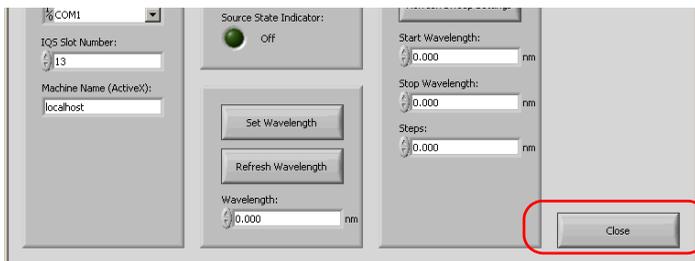
7. Close LabVIEW.

> # IMPORTANT
>
> **To avoid losing the original version of the Getting Started applications, do not save changes when prompted by LabVIEW.**

Building and Using Custom VIs

EXFO LabVIEW drivers have been designed to let you control the various instruments according to your needs.

You can access EXFO drivers

➤ directly from
  C:\Program Files\National Instruments\LabVIEW 6\instr.lib\EXFO

➤ from the LabVIEW function palettes

Each icon of the EXFO palette corresponds to a set of drivers that allow you to either

➤ communicate with EXFO instruments that support remote control

➤ open and close communication links with instruments

➤ send IEEE 488.2 (common) commands

➤ send platform commands (specific to FTB-400)
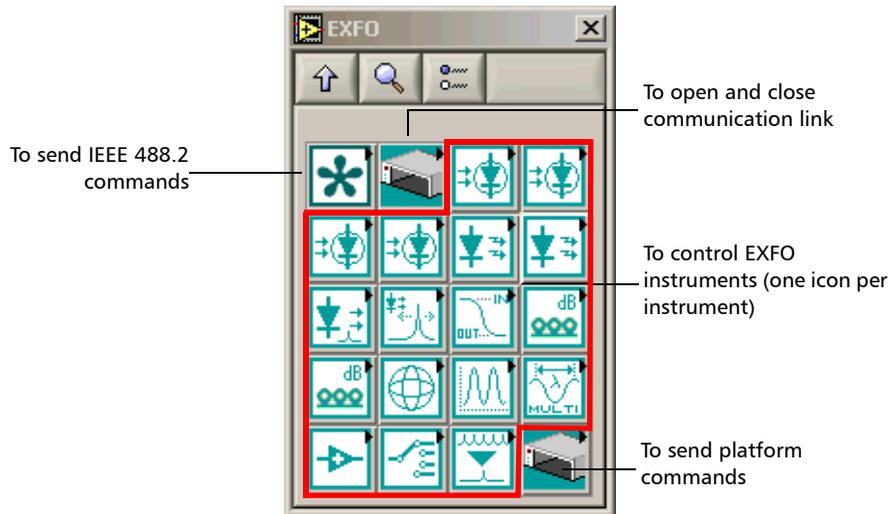
To open and close communication link

To send IEEE 488.2 commands

To control EXFO instruments (one icon per instrument)

To send platform commands

When you click on an icon, the corresponding sub-palette opens, giving you access to the different functions. On this palette, you can find icons with special symbols and background colors that provide useful information at a glance.
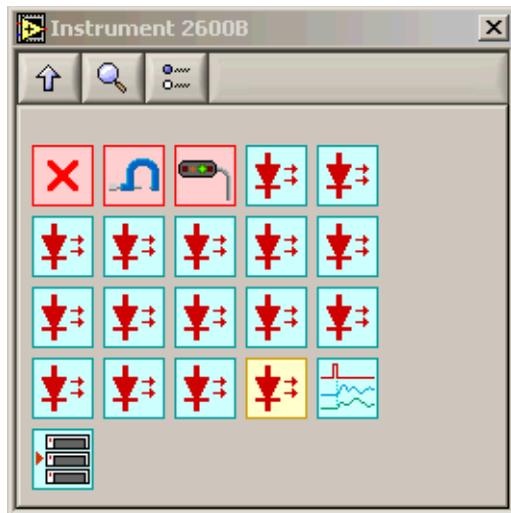
On a particular icon, the displayed symbol refers to the first keyword of the associated SCPI command. The background color is a function of the command type:

➤ A yellow background indicates a read-only command (Get)

➤ A red background indicates a write-only command (Set)

➤ A blue background indicates a read-and-write command (Get/Set)

Building and Using Custom VIs

To control instruments, you need to build your own VIs in LabVIEW.

To build a custom VI:

1. Start LabVIEW and create a new VI.

2. Open the **EXFO** palette.

   2a. From LabVIEW, open the Diagram Block view.

   2b. Display the **Functions** palette and select **Instrument I/O**.
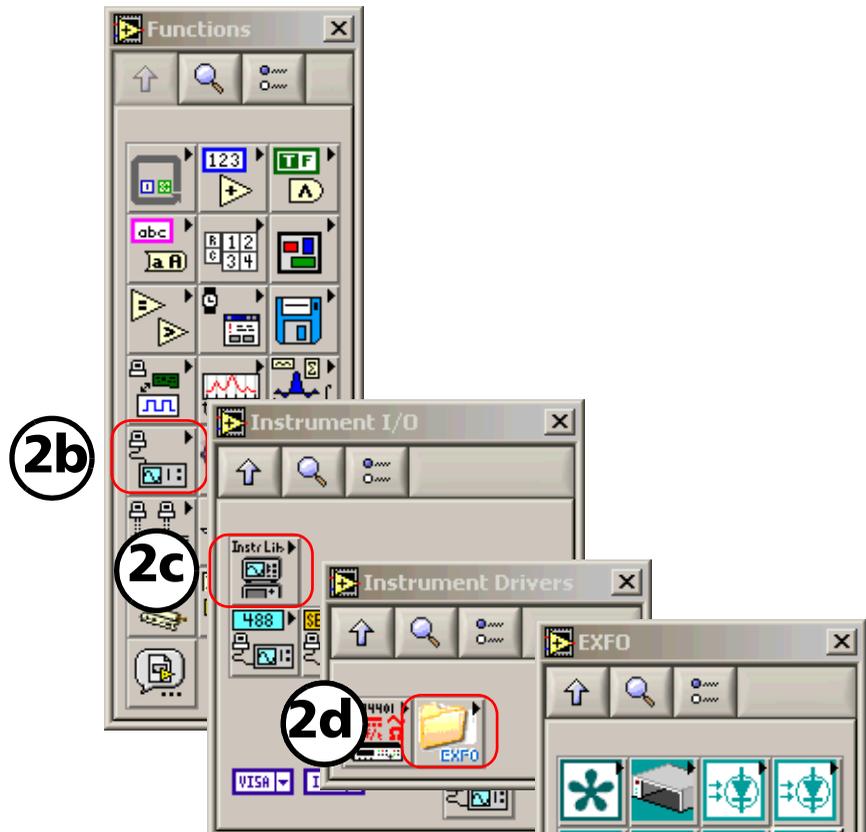
2c. From the **Instrument I/O** palette, select **Instrument Drivers**.

2d. From the **Instrument Drivers** palette, select **EXFO**.

3. Select **EXFO IQS Base**.

4. From the **EXFO IQS Base** palette, select *EXIQSBASE_OpenComm.vi* and add it to your new VI.
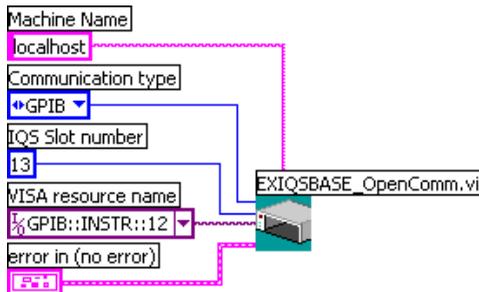
5. Set the communication parameters. For information on communication parameters, see *Working with EXFO LabVIEW Drivers* on page 122.

6. From the **EXFO** palette, select the desired instrument.

7. From the instrument's palette, select the function you need and add the corresponding driver to your VI.

8. Set the required parameters and connect the instrument *Communication ID in* parameter to the *Communication ID out* from EXIQSBASE_OpenComm.vi.

The example below shows how to configure the SourcePowerState_GSet.vi to turn on the tunable laser source. In this example, *Set* was chosen and the *PowerState* parameter was set to *True*.
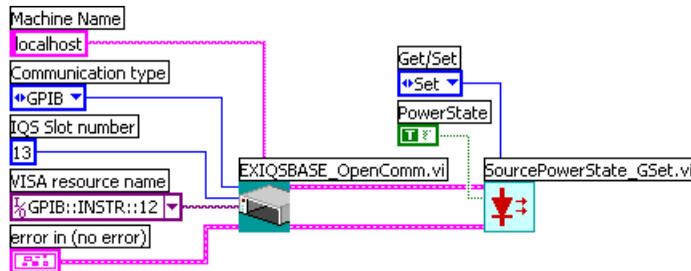
9. Repeat steps 7 and 8 for each of the functions you want to use. However, except that you have to link the *Communication ID in* parameter of the new driver to the *Communication ID out* parameter of the preceeding driver.

Note: *If you want to use platform or IEEE 488.2 commands, simply add the desired driver to your VI and configure its parameters exactly as you would do with any instrument function.*

10. When you are finished, add *EXIQSBASE_CloseComm.vi* to your VI.
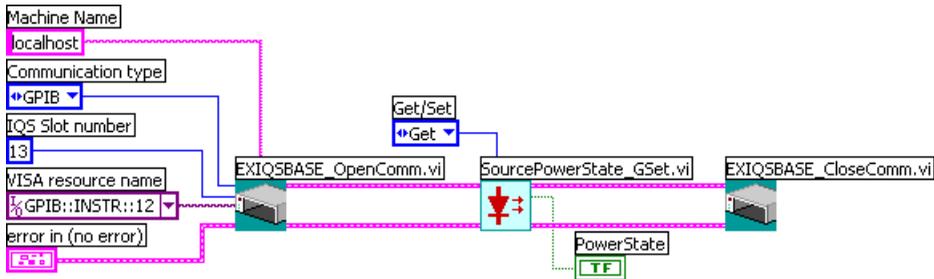
Connect the *Communication ID out* parameter of the last function to the *Communication ID in* of EXIQSBASE_CloseComm.vi.

Note: *You only have to open communication once at the beginning, and close it when all of the desired functions will have been added.*

11. Save your work.

To use your new VI:

1. Turn on the FTB-400, start ToolBox and ensure that all the remote-control parameters are set correctly.

2. From LabVIEW, run the VI.

Monitoring Remote Commands

ToolBox allows you to monitor remote commands sent to your units, if desired.

To monitor remote commands:

1. Select the **Utilities** function tab.

2. Click on **Instrument Control Monitoring** to access the corresponding utility.
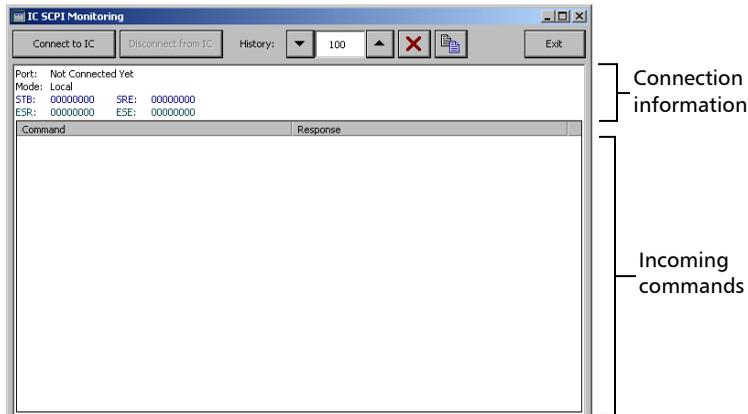
3. Connect to the monitoring system by clicking on **Connect to IC**. To disconnect from the monitoring system, simply click on **Disconnect from IC**.

Once connected, your current connection information will appear in the upper part of the window, and the commands will appear as a list in the lower part of the window.

The **History** field allows you to determine how many commands you wish to keep on the list. You can increase or decrease the number by using the arrow buttons on each side of the field.

To clear the history, click on [X].

To view the list in any word processor, copy it to the clipboard using [📋] and then paste it in your document. You can use any program, as the list is copied in text format.

To exit the monitoring utility, simply click on **Exit**.

9 *Maintenance*

To help ensure long, trouble-free operation:

➤ Keep the unit free of dust.

➤ Clean the unit casing with a cloth slightly dampened with water.

➤ Store unit at room temperature in a clean and dry area. Keep the unit out of direct sunlight.

➤ Avoid high humidity or significant temperature fluctuations.

➤ Avoid unnecessary shocks and vibrations.

➤ If any liquids are spilled on or into the unit, turn off the power immediately and let the unit dry completely.

WARNING

Use of controls, adjustments and procedures for operation and maintenance other than those specified herein may result in hazardous radiation exposure.

Cleaning the Front Panel

Clean the front panel of the Universal Test System regularly to avoid buildup of dust, dirt, and other foreign substances.

To clean the front panel:

1. Gently wipe the front panel with a cloth dampened with soapy water.

2. Rinse the front panel with a cloth dampened with water.

3. Dry with a clean wiping cloth.

IMPORTANT

**To help keep the connectors and adapters clean, EXFO recommends that you install protective caps when the unit is not in use.
You should also clean the fiber ends before every connection.**

Cleaning the Touchscreen

Clean the touchscreen with a soft, non-abrasive cloth dampened with glass cleaner.

Battery Maintenance Recommendations

➤ To ensure that batteries function or charge properly, keep them in temperatures between 10 °C and 40 °C (50 °F and 104 °F).

➤ To prevent battery cells from leaking, do not leave a battery discharged for several days.

➤ EXFO highly recommends removing the NiMH batteries if the unit will not be used for more than two weeks.

➤ EXFO recommends leaving the Universal Test System plugged in when not in use.

➤ Change the NiMH batteries after 300 cycles (approximately 18 months of use) otherwise the unit operating time might be reduced by as much as 50 %.

➤ If the NiMH batteries are not used for a long period of time (i.e. over three months), they will not be damaged, but might need to be recharged. In the worst case scenario, the batteries will require a complete recalibration; i.e. a full discharge, immediately followed by a full charge, then followed by a full discharge.

Inserting and Removing Batteries

The FTB-400 Universal Test System is powered by smart nickel metal-hydride (NiMH) batteries. You can also power it from an appropriate power outlet. For more information on the available power sources for your FTB-400 Universal Test System, as well as their characteristics, see *Technical Specifications* on page 163.

Note: *Whenever batteries are inserted or removed, the FTB-400 Universal Test System will beep.*

Inserting Batteries in the Mainframe

Before you go out in the field, make sure you install batteries in the FTB-400 Universal Test System unless you have an adequate and reliable power source.

To avoid losing test data, make sure at least one battery is always present in the FTB-400 Universal Test System.

To insert the batteries in the mainframe:

1. Place the FTB-400 Universal Test System with the screen facing down.

2. Open the battery compartment located on the bottom panel of the FTB-400 Universal Test System by first pushing the lock buttons toward the center of the battery door, and then pushing the battery door down.

Closed lock   Opened lock
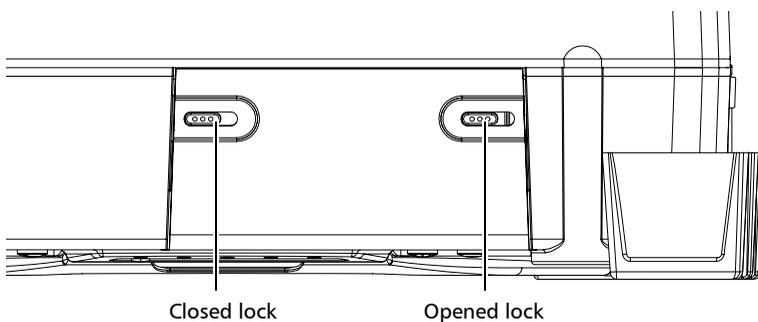
3. Slide the batteries into the compartment with the positive end on the left. They will not go in completely if inserted incorrectly.

Negative end
Positive end
(-) T D C (+)
Strap
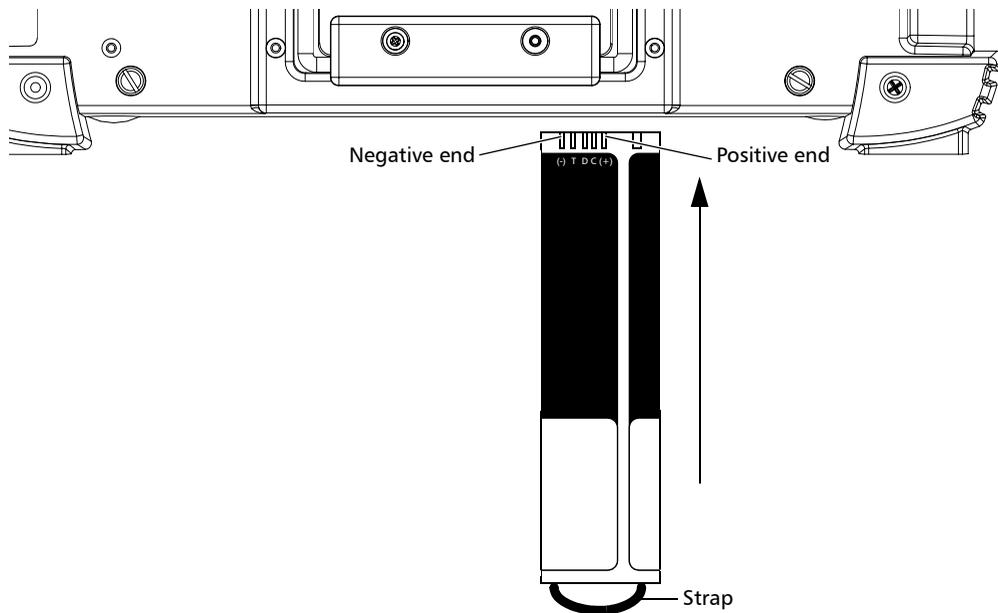
4. Close the battery compartment and push the door locks back into position.

Inserting Batteries in the Seven-Slot Backplane

Before you go out in the field, make sure you install batteries in the FTB-400 Universal Test System unless you have an adequate and reliable power source.

To insert the batteries in the seven-slot backplane:

1. Place the FTB-400 Universal Test System so that you face the right panel.

2. Unscrew the battery compartment door to open the battery compartment located at the bottom of the module receptacle.

Battery door
captive screw
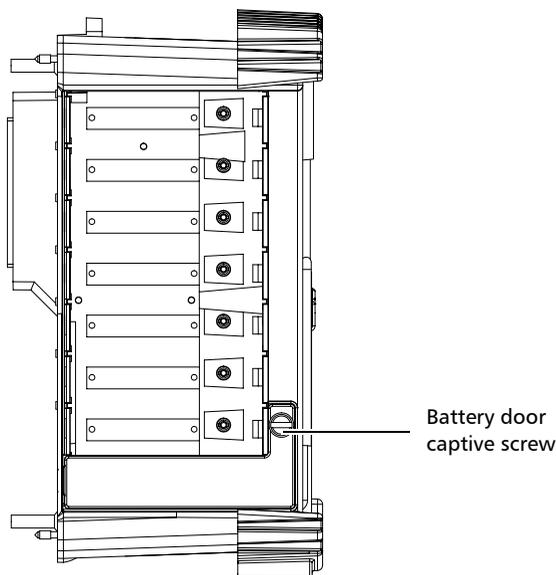
3. Slide the new batteries into the compartment with the positive end on the left. They will not go in completely if inserted incorrectly.

4. Close the battery compartment and screw the door back into position.

Replacing Batteries

When batteries run low on power, you will want to change them.

To replace the batteries:

1. Remove the door from the battery compartment. To locate the battery compartments, see *Inserting Batteries in the Mainframe* on page 139, and *Inserting Batteries in the Seven-Slot Backplane* on page 141.

2. Pull one battery out by its strap and insert a new battery.

3. Pull the other battery from the compartment and insert a new battery.

4. Put the battery compartment door back into position and make sure it is locked.

Note: *You do not need to turn the FTB-400 Universal Test System off when removing the batteries. Remember that in order to keep your data in memory, you must leave at least one battery inside the FTB-400 Universal Test System at all times if it is not under AC/DC current.*

Verifying Battery Status

The battery gauge located at the bottom of your screen indicates how much power you have left.

If you are using the FTB-400 Universal Test System's batteries, the gauge will show a battery icon.

If you are using AC/DC current, the gauge will show an electric plug icon.

To view your battery status in more detail, click on the gauge. This will work whether you are using batteries or AC/DC power.
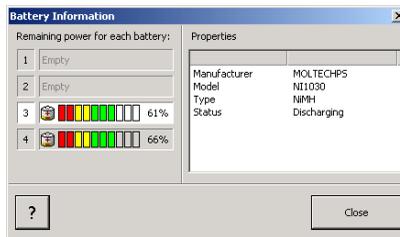
By clicking on a battery, you will see relevant details, such as the model, type and manufacturer, as well as the current status (charging or discharging).

Once you are done consulting the battery status, simply click on **Close**. You will be returned to the main window.

For more information on batteries, see *Inserting and Removing Batteries* on page 138 and *Replacing Batteries* on page 142.

10 Troubleshooting

Solving Common Problems

Before calling EXFO's technical support, you may want to consider the following solutions to problems that could occur.

Problem	Possible Cause	Solution
My FTB-400 Universal Test System does not start.	➤ It is not plugged in a power source.	➤ Ensure that the power cord is plugged in both the FTB-400 Universal Test System and the power inlet.
	➤ Batteries are completely discharged.	➤ Change or recharge the batteries.
	➤ Circuit breaker is at the "off" position.	➤ Reset the circuit breaker.
	➤ Windows startup files have been corrupted.	➤ Try booting from a floppy disk, or use the Rescue Disk on your Installation CD.
The screen remains black even if the FTB-400 Universal Test System is turned on.	➤ The display parameters are not correctly set.	If an external monitor is plugged in, verify the brightness and contrast settings in ToolBox. If no external monitor is plugged in, press on the backlight level button.
An USB device I have just installed is not working.	➤ There might have been a problem during installation.	Disconnect, then reconnect the device before turning the FTB-400 Universal Test System off, then on again.
	➤ You do not have the proper driver for this device.	Ensure that you have the right driver (can be provided with the device itself).

Problem	Possible Cause	Solution
The FTB-400 Universal Test System does not recognize a test module.	➤ The backplane is not properly connected.	Ensure that the backplane is properly connected with the frontplane.
	➤ Defective module.	If the FTB-400 Universal Test System recognizes other modules, the faulty module could be defective. Return it to EXFO for repairs.
Errors occur when I am using a floppy disk.	➤ FTB-400 Universal Test System is tilted.	Ensure that the FTB-400 Universal Test System is set flatly on a surface, whether it is horizontally on its back plane, or vertically on its bottom plane.
	➤ Defective disk or drive.	Try the disk in another computer. If it is working, the drive might be defective.
PCMCIA cards are not recognized at insertion.	➤ You might not have the latest PCMCIA card driver.	Install the most recent PCMCIA card driver available for the make of your card.

Problem	Possible Cause	Solution
Problems occur while printing or transmitting files via the infrared port.	➤ Ports not properly aligned.	Ensure that the infrared port of the FTB-400 Universal Test System sending the file is properly aligned with the one on the unit (other FTB-400 Universal Test System or infrared-enabled printer) receiving the file.
	➤ Ports too far apart.	Ensure that the distance separating the two units is within the limits.
	➤ Data transmission speed of the infrared-enabled printer is not supported.	Ensure to use a supported speed.
Batteries are not recharging.	➤ Incoming voltage is below 16 volts.	Ensure that the incoming voltage is over 16 volts.
	➤ Electronic fuel gauge became multibranched.	Allow a full discharge before attempting to charge.
Battery indicator shows only a fraction of the power is left, but the batteries are fully charged. (You are able to operate your unit, but the software automatically blocks when it reaches its set power threshold).	➤ Battery needs to be serviced or changed.	Send the battery to your local service center.
My FTB-400 keeps beeping every 15 seconds.	➤ The batteries are not compatible with your current FTB-400 version.	Your unit will still function despite the beeping. However, you can contact EXFO's technical support for a battery exchange.

Removing ToolBox Manually

It is possible that a software remains on the list of installed programs after attempting to remove it with the Add/Remove Programs utility (from the Windows Control Panel).

Before installing a new version of that software, you will have to remove it from the list with a special tool and to delete files from the hard disk.

IMPORTANT

Before performing this procedure, you should always try to remove ToolBox with the Windows Add/Remove Programs utility.

Note: *You must log onto your FTB-400 Universal Test System as a local administrator.*

To manually remove ToolBox:

1. Insert the installation CD into the CD-ROM drive.

2. On the Windows taskbar, click on **Start** and select **Run...**
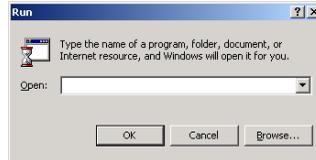
3. Use the **Browse...** button to find the *Utilities|Windows Installer Clean-Up* folder on the CD-ROM. In this folder, select *msicuu.exe*. Click on **OK** to launch the application.

4. Select the software to be removed from the list and click on **Remove**.

Note: *The msicuu.exe tool is only intended for removing a software name from the list. Therefore, it does not remove the software itself (no deletion of the associated files and registry keys).*

5. Once the operation is complete, close the msicuu.exe application.

6. Manually delete the installed files from the hard disk. Go to *C:|Program Files|EXFO|*. Find the *ToolBox* folder and delete it (as well as all its contents).

Resetting the Circuit Breaker

If, for any reason, the circuit breaker switches to the "off" position, you can reset it yourself.

CAUTION

Before removing the backplane, remove ALL power sources (batteries and power outlet) from the Universal Test System.

To reset the circuit breaker:

1. Unscrew the four captive screws from the backplane.

2. Gently pull the backplane away from the mainframe, making sure to keep the backplane perfectly straight.

3. Locate the circuit breaker at the back of the mainframe, and, using a small tool such as the stylus included with your Universal Test System, push the button to the right.

Master connector

Circuit breaker
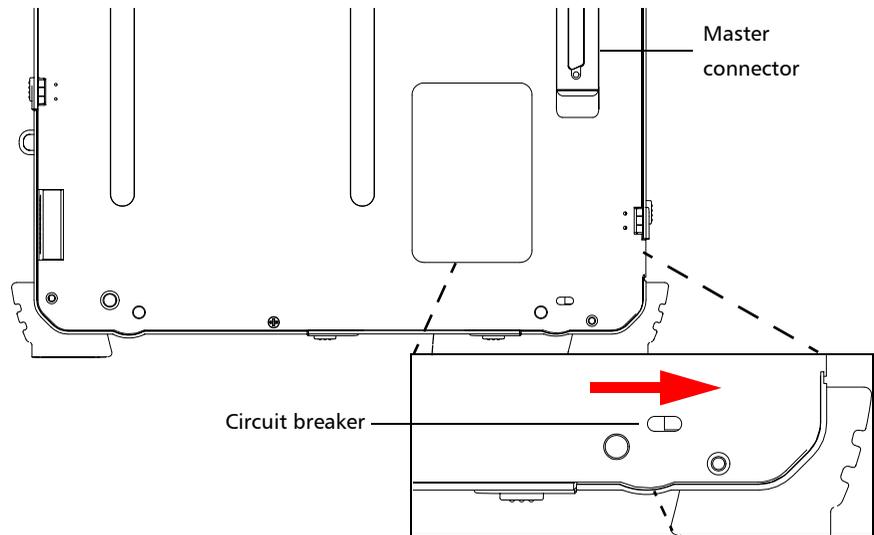
4. Gently push the backplane back into the mainframe, keeping the backplane as straight as possible.

5. Turn the captive screws to lock the backplane to the mainframe.

Note: *If the backplane is not properly screwed in, the FTB-400 Universal Test System might not detect it.*

6. Replace the power source (batteries or power outlet).

Using Rescue Disks

If you ever encounter problems with Windows itself and cannot start it anymore, you can use the applications in the Windows Utilities\Rescue Disks folder to create boot disks (four in all). They will allow you to reboot under Windows' safe mode, solve the problem, and recuperate your valuable test data.

Contact EXFO's Technical Support group for more details or help pertaining to the Windows Rescue Disks.

Viewing Online Documentation

A PDF version of the FTB-400 Universal Test System user guide is conveniently available at all times.

To access the online user guide:

Simply click on **Help** in the function bar.

Finding Information on the EXFO Web Site

The EXFO Web site provides answers to frequently asked questions (FAQs) regarding the use of your FTB-400 Universal Test System.

To access FAQs:

1. Type http://www.exfo.com in your Internet browser.

2. Click on the **Support** tab.

3. Click on **FAQs** and follow the on-screen instructions. You will be given a list of questions pertaining to your subject.

The EXFO Web site also provides the product's most recent technical specifications.

Contacting the Technical Support Group

To obtain after-sales service or technical support for this product, contact EXFO at one of the following numbers. The Technical Support Group is available to take your calls from Monday to Friday, 7:30 a.m. to 8:00 p.m. (Eastern Time in North America).

Technical Support Group

400 Godin Avenue	1 866 683-0155 (USA and Canada)
Vanier (Quebec) G1M 2K2	Tel.: 1 418 683-5498
CANADA	Fax: 1 418 683-9224
	support@exfo.com

To accelerate the process, please have information such as the name and the serial number (see the product identification label—an example is shown below), as well as a description of your problem, close at hand.

FTB-400-N<u>XX</u>-<u>D</u>-<u>H</u>-<u>Z</u>

Memory configuration
TFT LCD option
Seven-slot option
ToolBox 6 language selection
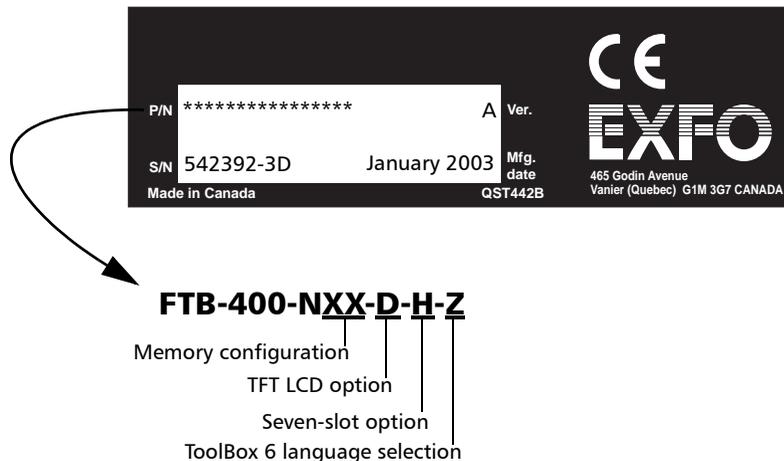
Viewing System Information

The **System Information** function tab contains important information on your system.

Consulting the About Tab

The **About** tab contains information on how to contact EXFO should you need to do so.

To access the About tab:

1. Click on the **System Information** function tab.
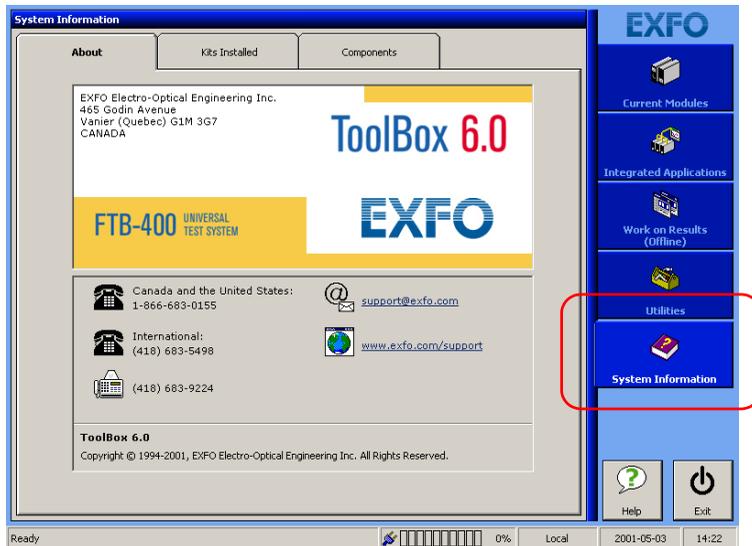
2. Select the **About** tab to view information on your unit.

Consulting the Kits Installed Tab

The **Kits Installed** tab allows you to view which kits are installed on your unit. Examples of kits are ToolBox 6 itself, the Data Transfer application or any module installed into the unit.

To access the Kits Installed tab:

1. Click on the **System Information** function tab.
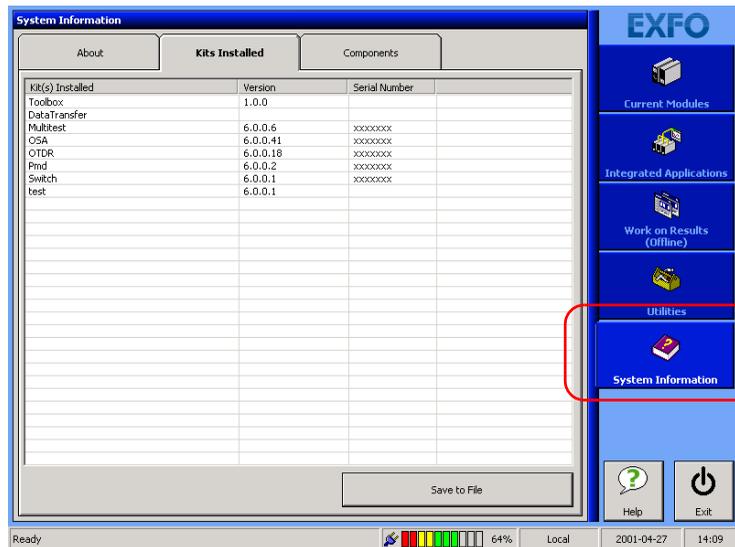
2. Select **Kits Installed** to view which kits are installed on your unit.

The first column represents a description of every installed element, the second will give you the current version, and the third column will give you the related serial number.

The last two columns are particularly important if you are contacting EXFO's technical support group, as the version and serial numbers will help pinpoint and solve your problem more rapidly. For more information, see *Contacting the Technical Support Group* on page 153.

To save a text format (.TXT) of the installed kit list, click on **Save to File**. The list will be automatically saved as *EXFO Install Kit List.TXT* on your C drive. Click on **OK** to acknowledge and close the pop-up window.

Note: *You cannot change the filename for this list. If you save the list a second time, the first list will be overwritten. To avoid this, you must go in the C drive and rename the first list.*

Consulting the Components Tab

The **Components** tab allows you to see every application installed on your unit.

To access the Components tab:

1. Click on the **System Information** function tab.
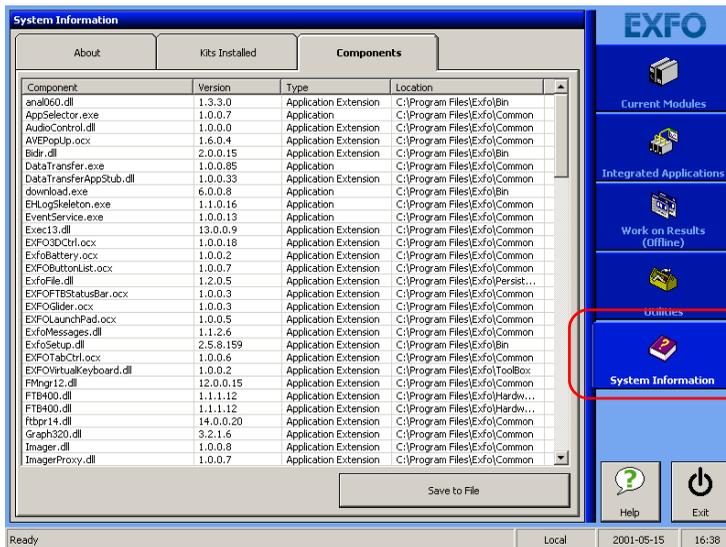
2. Select the **Components** tab to see all of the applications on your system. You can also see what version of the applications you have, their type, and location on your drive.

To save a text format (.TXT) of the component list, click on **Save to File**. The list will be automatically saved as *EXFO Component List.TXT* on your C drive. Click on **OK** to acknowledge and exit the pop-up window.

Note: *You cannot change the filename for this list. If you save the list a second time, the first list will be overwritten. To avoid this, you must go in the C drive and rename the first list.*

Accessing Online Help

Online help is available from ToolBox 6 in two different ways.

Using the Help Feature

The **Help** button will provide you with a description of the item or function you select.

To use the Help feature:

1. Click on . The mouse pointer will turn into a question mark.

2. Select the item or function for which you want more information. A description will appear.

The **Help** button is available in all of the function tabs of ToolBox 6, as well as the subwindows, where it appears as a simple question mark.

Using the Module Information Feature

You can obtain more information on the modules that are currently installed in your unit.

To use the Module Information feature:

1. In the **Current Modules** function tab, select the module for which you want more information. The **Online Application** bar will appear.
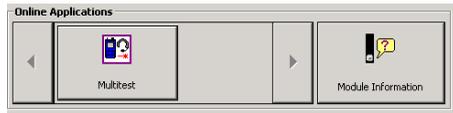
Note: *You can find more information on the* Online Applications *bar in* Launching a Single-Module Application *on page 56.*

2. Click on .

A pop-up window containing more information on the module will appear.

3. When you are done reading the information, click on **Close** to return to the **Current Modules** window.

Transportation

Maintain a temperature range within specifications when transporting the unit. Transportation damage can occur from improper handling. The following steps are recommended to minimize the possibility of damage:

➤ Pack the unit in its original packing material when shipping.

➤ Avoid high humidity or large temperature fluctuations.

➤ Keep the unit out of direct sunlight.

➤ Avoid unnecessary shock and vibration.

11 *Warranty*

General Information

EXFO Electro-Optical Engineering Inc. (EXFO) warrants this equipment against defects in material and workmanship for a period of one year from the date of original shipment. EXFO also warrants that this equipment will meet applicable specifications under normal use.

During the warranty period, EXFO will, at its discretion, repair, replace, or issue credit for any defective product, as well as recalibrate the product free of charge should the equipment need to be repaired or if the original calibration is erroneous.

IMPORTANT

The warranty can become null and void if:

➤ **the equipment has been tampered with, repaired, or worked upon by unauthorized individuals or non-EXFO personnel.**

➤ **the warranty sticker has been removed.**

➤ **case screws, other than those specified in this manual, have been removed.**

➤ **the case has been opened, other than as explained in this manual.**

➤ **the equipment serial number has been altered, erased, or removed.**

➤ **the equipment has been misused, neglected, or damaged by accident.**

THIS WARRANTY IS IN LIEU OF ALL OTHER WARRANTIES EXPRESSED, IMPLIED, OR STATUTORY, INCLUDING, BUT NOT LIMITED TO, THE IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS FOR A PARTICULAR PURPOSE. IN NO EVENT SHALL EXFO BE LIABLE FOR SPECIAL, INCIDENTAL, OR CONSEQUENTIAL DAMAGES.

Liability

EXFO shall not be liable for damages resulting from the use of the purchased product, nor shall be responsible for any failure in the performance of other items to which the purchased product is connected or the operation of any system of which the purchased product may be a part.

EXFO shall not be liable for damages resulting from improper usage or unauthorized modification of the product, its accompanying accessories and software.

Exclusions

EXFO reserves the right to make changes in the design or construction of any of its products at any time without incurring obligation to make any changes whatsoever on units purchased. Accessories, including but not limited to fuses, pilot lamps, and batteries used with EXFO products are not covered by this warranty.

Certification

EXFO certifies that this equipment met its published specifications at the time of shipment from the factory.

Service and Repairs

EXFO commits to providing product service and repair for five years following the date of purchase.

To send any equipment for service or repair:

1. Call one of EXFO's authorized service centers (see *EXFO Service Centers Worldwide* on page 162). Support personnel will determine if the equipment requires service, repair, or calibration.

2. If equipment must be returned to EXFO or an authorized service center, support personnel will issue a Return Merchandise Authorization (RMA) number and provide an address for return.

3. If the unit has an internal storage device, perform a data backup before sending the unit for repairs.

4. Pack the equipment in its original shipping material. Be sure to include a statement or report fully detailing the defect and the conditions under which it was observed.

5. Return the equipment, prepaid, to the address given to you by support personnel. Be sure to write the RMA number on the shipping slip. *EXFO will refuse and return any package that does not bear an RMA number.*

Note: *A test setup fee will apply to any returned unit that, after test, is found to meet the applicable specifications.*

After repair, the equipment will be returned with a repair report. If the equipment is not under warranty, the customer will be invoiced for the cost appearing on this report. Return-to-customer shipping costs will be paid by EXFO for equipment under warranty. Shipping insurance is at the customer's expense.

EXFO Service Centers Worldwide

If your product requires servicing, contact your nearest authorized service center.

EXFO Headquarters Service Center

400 Godin Avenue   1 866 683-0155 (USA and Canada)
Vanier (Quebec) G1M 2K2   Tel.: 1 418 683-5498
CANADA   Fax: 1 418 683-9224
                                    quebec.service@exfo.com

EXFO Europe Service Center

Le Dynasteur   Tel.: +33.1.40.83.85.85
10/12, rue Andras Beck   Fax: +33.1.40.83.04.42
92366 Meudon la Forêt Cedex   europe.service@exfo.com
FRANCE

**EXFO China Service Center/
Beijing OSIC**

Beijing New Century Hotel   Tel.: +86 (10) 6849 2738
Office Tower, Room 1754-1755   Fax: +86 (10) 6849 2662
No. 6 Southern Capital Gym Road   beijing.service@exfo.com
Beijing 100044
P. R. CHINA

EXFO Asia-Pacific Service Center

151 Chin Swee Road   Tel.: +65 6333 8241
#03-29 Manhattan House   Fax: +65 6333 8242
SINGAPORE 169876   asiapacific.service@exfo.com

**EXFO Burleigh Products Group
Service Center**

7647 Main Street Fishers   Tel.: 1 585 924-9355
Victor, NY 14564-8909   Fax: 1 585 924-9072
USA   service@burleigh.com

A   Technical Specifications

Electrical and Environmental Specifications	
Operating temperature[a]	
➤ FTB-400 Universal Test System	0 °C to 50 °C (32 °F to 122 °F)
➤ AC/DC Adaptor	5 °C to 40 °C (41 °F to 104 °F)
Relative humidity[b,c]	
➤ FTB-400 Universal Test System	≤ 95 % non-condensing
➤ AC/DC Adaptor	0 % to 80 % non-condensing
Maximum operation altitude	2000 m (6562 ft)
Pollution degree	2
Installation category	II
Power supply rating[d]	100 V to 240 V (50 Hz/60 Hz)
	maximum 2 A

a. The floppy drive should not be operated below 0 °C, and will age prematurely if operated above 50 °C.
b. Measured in 0 °C to 31 °C (32 °F to 87.8 °F) range, decreasing linearly to 50 % at 40 °C (104 °F).
c. The floppy drive will age prematurely if operated at 95% humidity.
d. Not exceeding ± 10 % of the nominal voltage.

Note: *The FTB-400 Universal Test System is intended for indoor use only.*

IMPORTANT

The following technical specifications can change without notice. The information presented in this section is provided as a reference only. To obtain this product's most recent technical specifications, visit the EXFO Web site at *www.exfo.com*.

Display	Touchscreen, color, 800 x 600 full dot LCD 12.1 in
	Touchscreen, color, 800 x 600 TFT 12.1 in (optional)
Interfaces	Serial RS-232
	Parallel port
	External monitor
	2 USB ports
	Infrared (IrDA) port
	Audio microphone In 3.5 mm
	Audio speaker Out 3.5 mm
	2 PCMCIA type II or 1 PCAMCIA type III
Storage	Internal 6 GB hard drive minimum (over 200 000 OTDR test files)
	Internal 3.5 in 1.44 MB floppy drive
	External USB read/write CD-ROM (optional)
	Flash memory cards (16, 32, 80, 160 MB) (optional)
	NTFS file system
Batteries[2]	Rechargeable NiMH battery pack (2 batteries for 2-slot, 4 for 7-slot receptacle)
	> 8 h of continuous operation as per Bellcore TR-NWT-001138
Power supply	100–240 VAC, 50/60 Hz, and 12-24 VDC

General Specifications

Temperature[5]	operating	0 °C to 50 °C	(32 °F to 122 °F)
	storage	-40 °C to 60 °C	(-40 °F to 140 °F)
Relative humidity		0 % to 95 % (non-condensing)	
Size (HxWxD)		mainframe + 2-slot module receptacle: 31.1 cm x 34.3 cm x 10.2 cm (12 $1/4$ in x 13 $1/2$ in x 4 in)	
		mainframe + 7-slot module receptacle: 31.1 cm x 34.3 cm x 20.3 cm (12 $1/4$ in x 13 $1/2$ in x 8 in)	
Weight[3]		mainframe + 2-slot module receptacle: 6.1 kg (13.4 lb)	
		mainframe + 7-slot module receptacle: 8.1 kg (17.9 lb)	
Vibration (g)		> 1.5 g at 10 Hz to 500 Hz (on 3 main axes)	
Mechanical shock[4]		> 76 cm on 6 sides and 8 main edges (according to GR-196-CORE)	
Isolation		Spill-proof and splash-proof	
CE compliance		Class A certification	

Notes

1. All specifications valid at 23 °C (73 °F).
2. Standard recharge time is 5 h. Recharge temperature: 0 °C to 35 °C (32 °F to 95 °F).
3. Platform with batteries, no modules included.
4. 2-slot receptacle.
5. Not including internal batteries. Battery maximum storage temperature 40 °C (104 °F).

B Data Types

The following section provides an overview of the most common data types that may appear in EXFO's documentation on commands and queries. The information is supplied for guidance only.

For more detailed information, please refer to IEEE 488.2 and SCPI standards. Additional reference sources are listed in *SCPI Command Structure* on page 113.

Data types are divided into two groups: <PROGRAM DATA> for the types that are used when you want to send messages to a device and <RESPONSE DATA> for the types that are used when a device sends responses to the controller.

The data types are presented in graphics often referred to as "railroad diagrams". The following example illustrates how to interpret such diagrams.
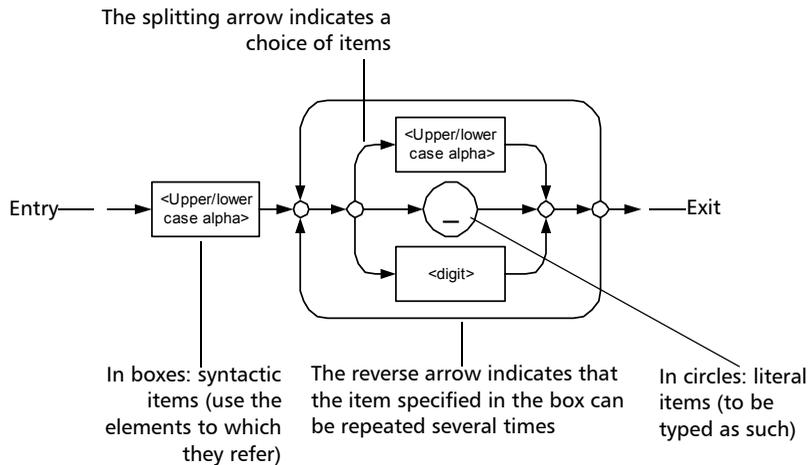
The splitting arrow indicates a choice of items

<Upper/lower case alpha>

Entry

<Upper/lower case alpha>

Exit

<digit>

In boxes: syntactic items (use the elements to which they refer)

The reverse arrow indicates that the item specified in the box can be repeated several times

In circles: literal items (to be typed as such)

Applicable Data Types for Input—IEEE 488.2

➤ <PROGRAM MESSAGE TERMINATOR>

In the diagram above,

➤ "NL" corresponds to ASCII character code 10, in decimal (0A in binary)

➤ "END^" corresponds to the last data byte of the message sent with EOI = True and ATN = False

➤ <CHARACTER PROGRAM DATA>
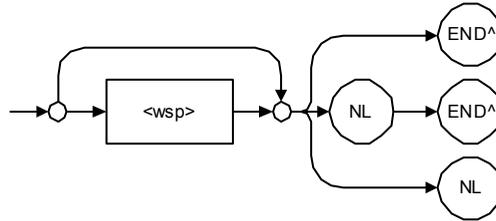
This data type will be used to send short mnemonics when a *<DECIMAL NUMERIC PROGRAM DATA>* cannot be used.

Examples: TRIANGLEWAVE, NCONTINUOUS

➤ <DECIMAL NUMERIC PROGRAM DATA> (or <NRf>)

This data type includes <NR1>, <NR2> and <NR3> data types. It will be used for decimal fractions with or without an exponent. Instruments will adapt the values they receive to fit their degree of precision. For example, if an instrument has a precision of two digits after the decimal point and the incoming value is 12.048, this value will be rounded off to 12.05.
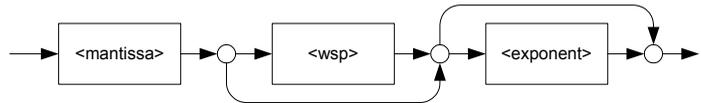
The second diagram below illustrates the <mantissa> syntax.
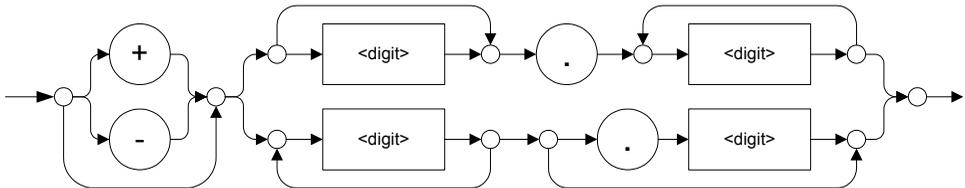
The third diagram illustrates the <exponent> syntax.
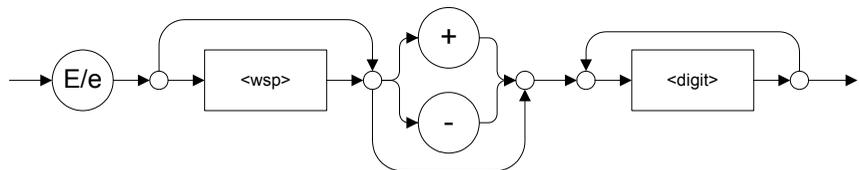
Examples: +2.0 e5, −.56E+4, 6.5e−10

➤ <NON-DECIMAL NUMERIC PROGRAM DATA>

This data type will be used for integer representation in hexadecimal (base 16), octal (base 8) or binary (base 2). The numeric representations will begin with "#H" for hexadecimal, "#Q" for octal and "#B" for binary.
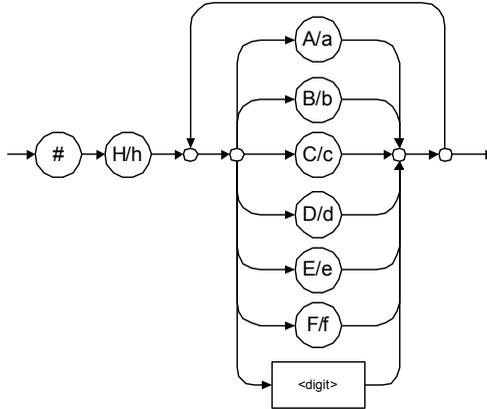
Examples: #Hf3bc015d, #h01a4, #hfe
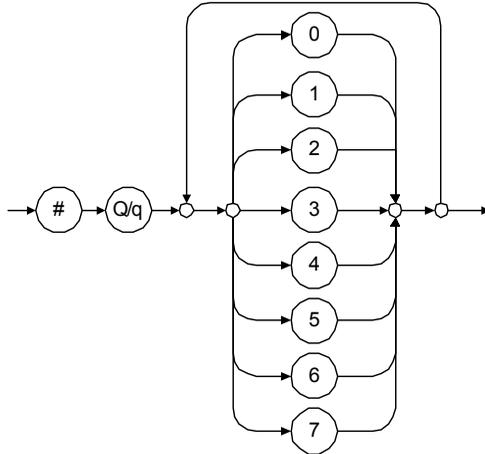
Examples: #Q1234567, #q1275, #q07
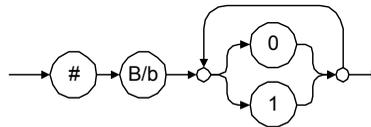
Examples: #B10010111, #b10110, #b1100

➤ <STRING PROGRAM DATA>

This data type will be used for strings containing 7-bit ASCII characters that have to be enclosed in either single- or double-quotes delimiters.

If a string needs to contain a character that is exactly the same as the delimiter, make sure to double the character to avoid syntax errors.
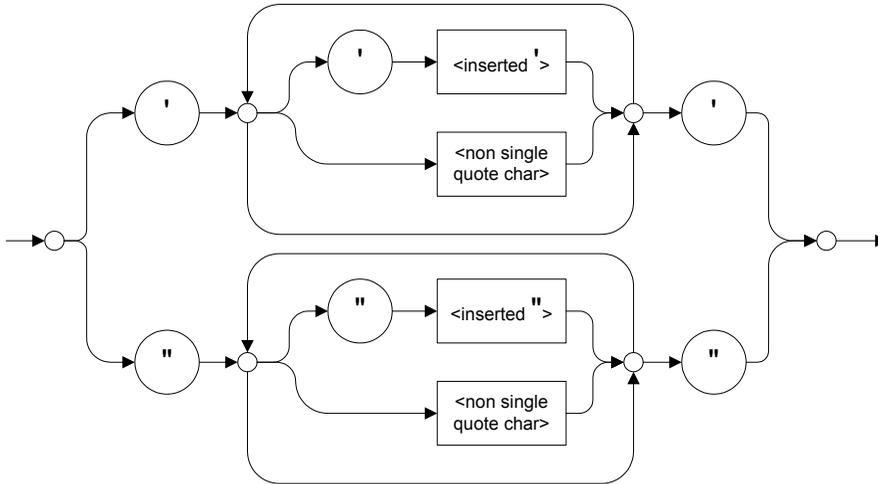
Examples: "SCPI Commands", 'SCPI Commands', "SCPI 'Commands'", 'SCPI "Commands"', "SCPI ""Commands""", 'SCPI ''Commands'''

➤ <ARBITRARY BLOCK PROGRAM DATA>

This data type is used to send blocks of arbitrary 8-bit information when you need to work with large amounts of data.

The actual length of the data that you send has the following structure:

➤ The first byte contains the # character.

➤ The byte that immediately follows contains the number of subsequent bytes that you have to check to obtain the total length.

Note: *If you use a zero as the first digit (#0), it has to be followed by a <PROGRAM MESSAGE TERMINATOR> to allow the device to detect the end of the <ARBITRARY BLOCK PROGRAM DATA>. This will also force immediate termination of the message.*

For example, if you send the following data (here, values are expressed in decimal instead of binary for easier readability):

2 1 3 7 6 8 9 2 ...

The byte that immediately follows the # contains 2, which means that you would have to read the two following bytes to know the length (in bytes) of the retrieved data. The bytes indicate 1 and 3. The length will then be 13 bytes. The actual response will begin at byte number 5, in this case.

➤ <SUFFIX PROGRAM DATA>
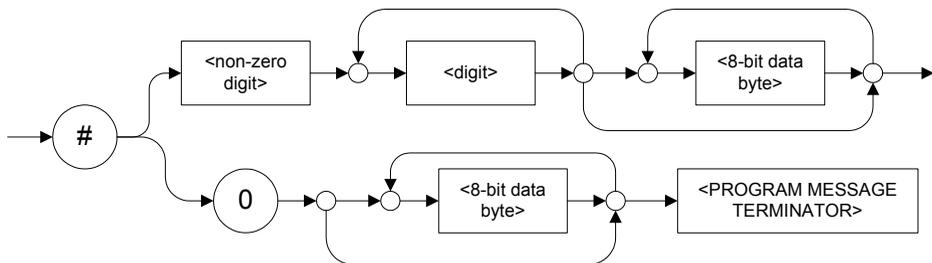
This data type is used when units and multipliers have to be sent.
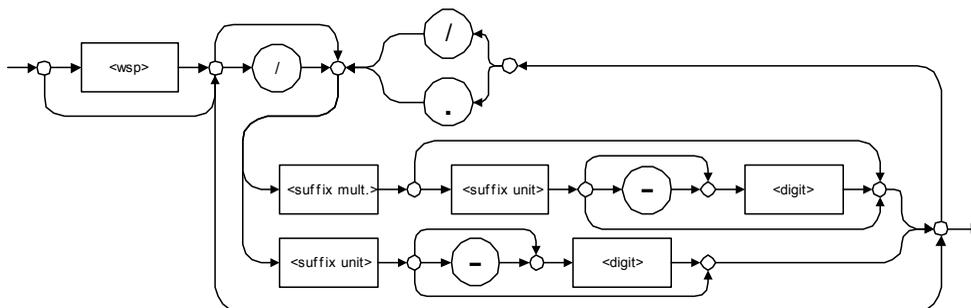
Examples: nm, kHz, km/s2, uW

A relative unit (dB) can be referenced to an absolute level, as shown on the following diagram.
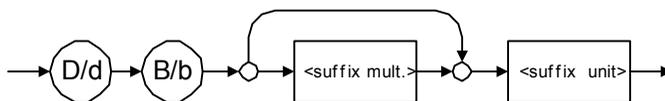
Examples: db, dbm, dBW

The following table illustrates the possible forms for <suffix mult.>:

Name	Value	Mnemonic
Exa	1E18	EX
Peta	1E15	PE
Tera	1E12	T
Giga	1E9	G
Mega	1E6	MA
Kilo	1E3	K
Milli	1E-3	M
Micro	1E-6	U
Nano	1E-9	N
Pico	1E-12	P
Femto	1E-15	F
Atto	1E-18	A

The table below gives the possible forms for <suffix unit>:

Reference Unit	Suffix Unit
Degrees	DEG
Radians	RAD
Amperes	A
Volts	V
Hertz	HZ
Meters	M
Watts	W
DBs ref to 1mW	DBM
Decibels	DB
Degrees Celsius	CEL
Degrees Fahrenheit	FAR
Kelvins	K
Seconds	S
Hours	HR
Minutes	MIN

Applicable Data Types for Output —IEEE 488.2
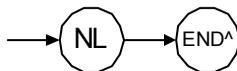
➤ <RESPONSE MESSAGE TERMINATOR>

In the diagram above,

➤ "NL" corresponds to ASCII character code 10, in decimal (0A in binary)

➤ "END^" corresponds to the last data byte of the message sent with EOI = True and ATN = False

➤ <CHARACTER RESPONSE DATA>

This data type will be used by a device to return short mnemonics when a *<DECIMAL NUMERIC PROGRAM DATA>* cannot be used. The returned information is sent in the long form and in upper case.

Examples: TRIANGLEWAVE, NCONTINUOUS

➤ <NR1 NUMERIC RESPONSE DATA> (or <NR1>)

This data type will be used by a device to return positive or negative integers.
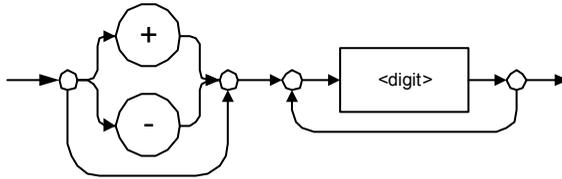
Examples: 4, –23, 90

➤ <NR2 NUMERIC RESPONSE DATA> (or <NR2>)

This data type will be used by a device to return positive or negative real numbers (fixed-point numbers).
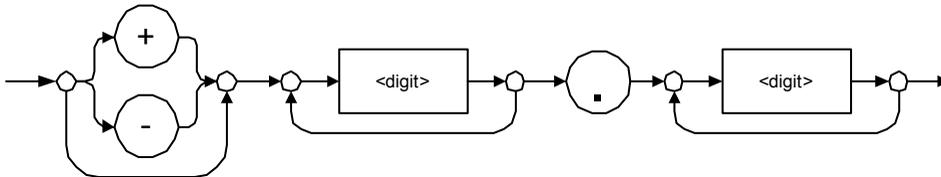
Examples: 23.45, 1.22, –4.55

➤ <NR3 NUMERIC RESPONSE DATA> (or <NR3>)

This data type will be used by a device to return positive or negative exponential numbers (floating-point numbers).
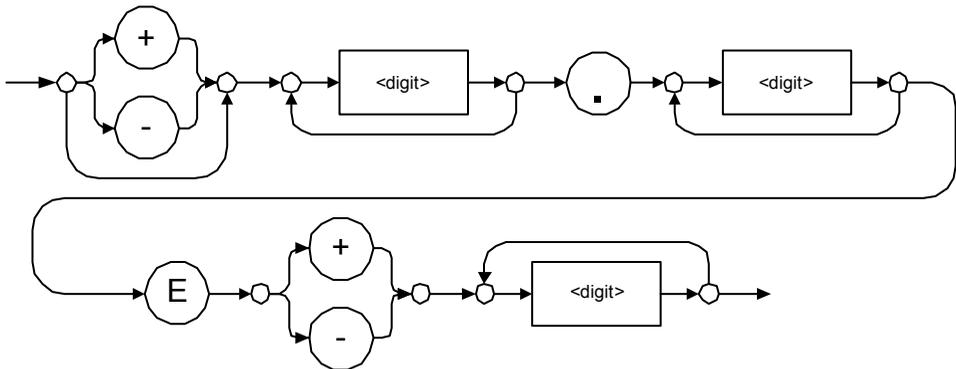
Examples: 4.3E–3, –8.9456E8, 123E–5

➤ Special Numeric Values Received on Output

In some cases, an instrument may send values indicating that an unusual event has occurred. The following tables present the possible values.

Value is	ASCII 4 bytes	PACKED 4 bytes
Under range	2143289345.000000	7FC00001
Over range	2143289346.000000	7FC00002
Invalid	2143289347.000000	7FC00003
Inactive	2143289348.000000	7FC00004

Value is	ASCII 8 bytes	PACKED 8 bytes
Under range	9221120237577961472	7FF8000020000000
Over range	9221120238114832384	7FF8000040000000
Invalid	9221120238651703296	7FF8000060000000
Inactive	9221120239188574208	7FF8000080000000

➤ <HEXADECIMAL NUMERIC RESPONSE DATA>

This data type will be used by a device to return integer representations in hexadecimal (base 16).
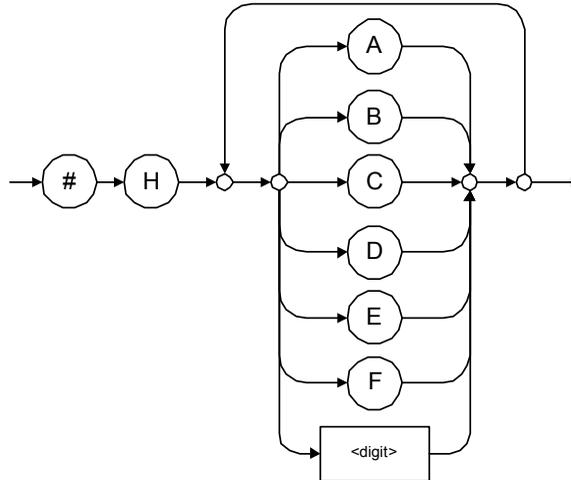
Examples: #HA3C5, #H0123C, #H010F

➤ <OCTAL NUMERIC RESPONSE DATA>

This data type will be used by a device to return integer representations in octal (base 8).
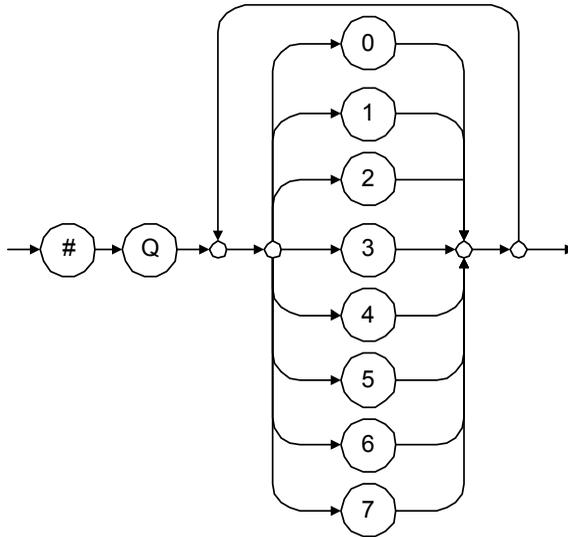
Examples: #Q753214, #Q0124, #Q0725

➤ <BINARY NUMERIC RESPONSE DATA>

This data type will be used by a device to return integer representations in binary (base 2).
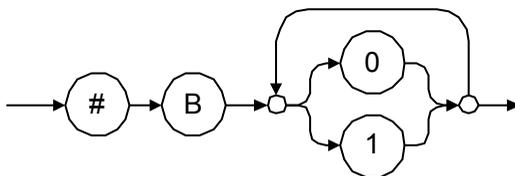
Examples: #B11011110101, #B110100, #B0100

➤ <STRING RESPONSE DATA>

This data type will be used by a device to return strings containing 7-bit ASCII characters and especially when text has to be displayed since even the non-printable characters are also returned.
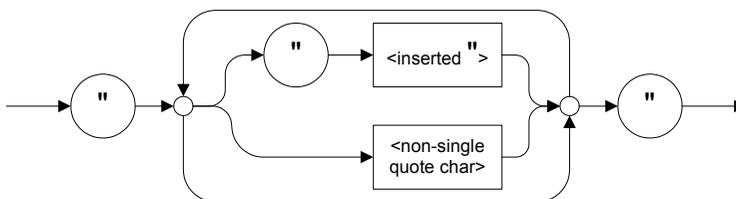
Examples: "SCPI Commands", "SCPI ""Commands"""

➤ <DEFINITE LENGTH ARBITRARY BLOCK RESPONSE DATA>

This data type is used by a device to return blocks of 8-bit binary information with a fixed and predetermined length.
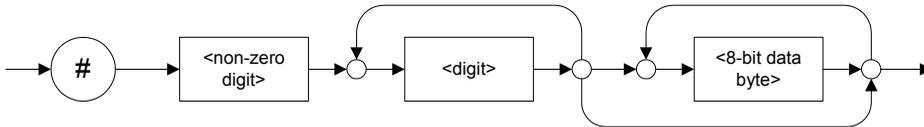
The actual length of the retrieved data has the following structure:

➤ The first byte contains the # character.

➤ The byte that immediately follows contains the number of subsequent bytes that you have to check to know the total length.

For example, if you receive this response (here, values are expressed in decimal instead of binary for easier readability):

2 1 3 7 6 8 9 2 ...

The byte that immediately follows the # contains 2, which means that you have to read the two following bytes to know the length (in bytes) of the retrieved data. The bytes indicate 1 and 3. The length will then be 13 bytes. The actual response will begin at byte number 5, in this case.

Examples: #14<DAB><DAB><DAB><DAB>, #3004<DAB><DAB><DAB><DAB>

where "<DAB>" stands for data byte

➤ <INDEFINITE LENGTH ARBITRARY BLOCK RESPONSE DATA>

This data type is used by a device to return blocks of 8-bit binary information when the block length was not predefined or when data has to be computed later.
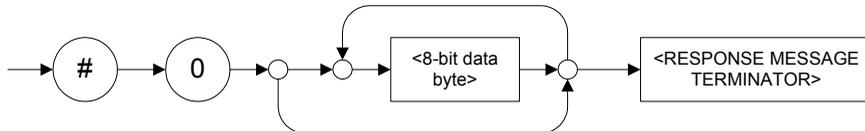
Note: *If you receive a zero as the first digit (#0), it is necessarily followed by a <RESPONSE PROGRAM MESSAGE TERMINATOR> to allow you to detect the end of the <INDEFINITE LENGTH ARBITRARY BLOCK RESPONSE DATA>.*

Example: #0<DAB><DAB><DAB><DAB><terminator> where "<DAB>" stands for data byte.

➤ <SUFFIX RESPONSE DATA>

This data type is used by a device to return units and multipliers.
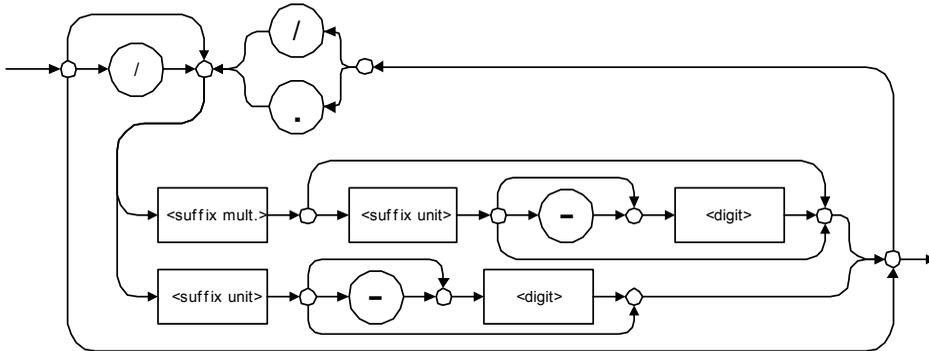
Examples: DBW, W, KHZ

➤ <ARBITRARY ASCII RESPONSE DATA>

This data type is used by a device to return information when it is impossible to use any other data type.

Example: To the *IDN? query, the device will return this response in an arbitrary ASCII bytes format:
EXFO E.O. Engineering,FTB-400,125-2A55,1.0.1.97
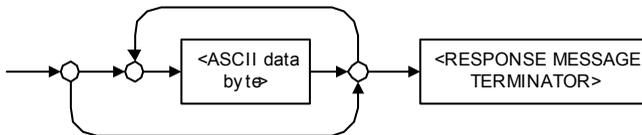
Applicable Data Types for Input—SCPI

SCPI data types include the IEEE 488.2 data types (see *Applicable Data Types for Input—IEEE 488.2* on page 166) with certain additional restrictions.

➤ <numeric_value>: abbreviated form of the decimal numeric element. It differs from the <DECIMAL NUMERIC PROGRAM DATA> "<NRf>" described in IEEE 488.2.

Several forms of <CHARACTER PROGRAM DATA> are defined as special forms of numbers. These are: MINimum, MAXimum, DEFault, UP, DOWN, Not A Number (NAN), INFinity and Negative INFinity (NINF). The following special forms are likely to be used by EXFO's instruments in certain commands or queries:

➤ DEFault: This special <numeric_value> parameter forces the instrument to select a value, which is deemed to be convenient to the user.

➤ MINimum|MAXimum: These special <numeric_value> parameters refer to the instrument's limit values. MINimum corresponds to the value closest to negative infinity that the function can accept. MAXimum corresponds to the largest value that the function can accept.

➤ <Boolean Program Data>: This form is often used as a shorthand of the <DECIMAL NUMERIC PROGRAM DATA>ON|OFF form.

<Boolean Program Data> parameters have a value of 0 or 1 and are not followed by any unit.

On input, an <NRf> is rounded to an integer.

A non-zero result is interpreted as 1.

ON and OFF are accepted on input for readability purposes. They correspond respectively to 1 and 0. However, on output, they appear as 1 or 0, never ON or OFF.

Special Numeric Values Received on Output

It is possible that an instrument returns unusual values in certain cases. For information on these values, see Applicable Data Types for Output —IEEE 488.2 on page 175.

C   IEEE 488.2 and Specific Commands

This chapter presents detailed information about the commands and queries supplied with your FTB-400 Universal Test System.

IEEE 488.2 Commands–Quick Reference

The FTB-400 Universal Test System recognizes the required commands identified in IEEE 488.2. The table below summarizes these commands. These commands are fully explained on the following pages.

Command	Function
*CLS	Clear status command
*ESE	Standard event status enable command
*ESE?	Standard event status enable query
*ESR?	Standard event status register query
*IDN?	Identification query
*OPC	Operation complete command
*OPC?	Operation complete query
*RST	Reset command
*SRE	Service request enable command
*SRE?	Service request enable query
*STB?	Read status byte query
*TST?	Self-test query
*WAI	Wait for pending operations to be completed

IEEE 488.2 Required Commands

	*CLS
Description	The *CLS command clears the Standard Event Status Register and the Error/Event Queue.
Syntax	*CLS
Parameter(s)	None

*ESE

Description	The *ESE command sets the Standard Event Status Enable Register bits, as defined in the table below. This register contains a mask value for the bits to be enabled in the Standard Event Status Register.

MSB   Standard Event Status Enable Register   LSB

PON	URQ	CME	EXE	DDE	QYE	N.U.	OPC
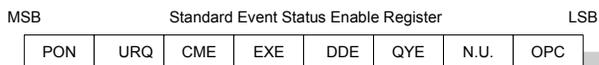
Syntax	*ESE<wsp><RegisterValue>
Parameter(s)	*RegisterValue:*
	The program data syntax for <RegisterValue> is defined as a <DECIMAL NUMERIC PROGRAM DATA> element.
	The <RegisterValue>, expressed in base 2, represents the bit values of the Standard Event Status Enable Register.

*ESE

The table below shows the contents of this register.

Bit	Weight	Meaning
PON	128	Power ON Enable
URQ	64	User ReQuest Enable
CMD	32	CoMmanD Error Enable
EXE	16	Execution Error Enable
DDE	8	Device Dependent Error Enable
QRY	4	QueRry Error Enable
N.U.	2	Not used
OPC	1	Operation Complete Enable

A value of 1 in the Enable Register enables the corresponding bit in the Status Register, a value of 0 disables the bit. The value of the <RegisterValue> shall be in the range of 0 through 255.

Example(s)

*ESE 25
where 25 = (bit EXE, bit DDE and bit OPC)

*ESE 0
clears the content of the Standard Event Status Enable register

See Also

*ESE?
*ESR?

*ESE?

Description   The *ESE? query allows the programmer to determine the current contents of the Standard Event Status Enable Register. See the contents of this register below.

MSB   Standard Event Status Enable Register   LSB

PON	URQ	CME	EXE	DDE	QYE	N.U.	OPC
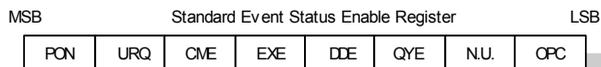
Syntax   *ESE?

Parameter(s)   None

Response Syntax   \<RegisterValue\>

*ESE?

Response(s)	*RegisterValue:*

The response data syntax for <RegisterValue> is defined as a <NR1 NUMERIC RESPONSE DATA> element.

The <RegisterValue> ranges from 0 through 255.

The <RegisterValue> value expressed in base 2 (binary) represents the bit values of the Standard Event Status Enable register. See below.

Bit	Weight	Meaning
PON	128	Power ON Enable
URQ	64	User ReQuest Enable
CMD	32	CoMmanD Error Enable
EXE	16	Execution Error Enable
DDE	8	Device Dependent Error Enable
QRY	4	QueRry Error Enable
N.U.	2	Not used
OPC	1	Operation Complete Enable

Example(s)

*ESE? returns 133
where 133 = (bit PON, bit QYE and bit OPC)

See Also

*ESE
*ESR?

*ESR?

Description	The *ESR? query allows the programmer to determine the current contents of the Standard Event Status Register. Reading the Standard Event Status Register clears it. See the contents of this register below.

MSB		Standard Event Status Register					LSB
PON	URQ	CME	EXE	DDE	QYE	N.U.	OPC

Syntax	*ESR?
Parameter(s)	None
Response Syntax	<RegisterValue>

*ESR?

Response(s)	*RegisterValue:*

The response data syntax for <RegisterValue> is defined as a <NR1 NUMERIC RESPONSE DATA> element.

The <RegisterValue> ranges from 0 through 255.

The <RegisterValue> value expressed in base 2 (binary) represents the bit values of the Standard Event Status register. See below.

Bit	Weight	Meaning
PON	128	Power ON Enable
URQ	64	User ReQuest Enable
CMD	32	CoMmanD Error Enable
EXE	16	Execution Error Enable
DDE	8	Device Dependent Error Enable
QRY	4	QueRry Error Enable
N.U.	2	Not used
OPC	1	Operation Complete Enable

Example(s)	*ESR? returns 33
	where 33 = (bit CME and bit OPC)

See Also	*ESE
	*ESE?

*IDN?

Description	The intent of the *IDN? query is for the unique identification of devices over the system interface.
Syntax	*IDN?
Parameter(s)	None
Response Syntax	\<Identification\>
Response(s)	*Identification:*
	The response data syntax for \<Identification\> is defined as an \<ARBITRARY ASCII RESPONSE DATA\> element.
	The response syntax for the *IDN? query, \<Identification\> is defined as an \<ARBITRARY ASCII RESPONSE DATA\> element. This implies that the *IDN? query should be the last \<QUERY MESSAGE UNIT\> in a \<TERMINATED PROGRAM MESSAGE\>.
	The response is organized into four fields separated by commas. The field definitions are as follows:
	Field 1 (Manufacturer): EXFO E. O. Engineering
	Field 2 (Model): Instrument Model

*IDN?

Field 3 (Serial number): ASCII character (0 if not available)

Field 4 (Firmware level): ASCII character (0 if not available)

ASCII character 0 represents a single ASCII-encoded byte with a value of 30 (48 decimal).

The presence of data in all fields is mandatory. If either field 3 or 4 is not available, the ASCII character 0 shall be returned for that field. A field may contain any 7-bit ASCII-encoded bytes in the range of 20 through 7E (32 through 126 decimal) except commas (2C, 44 decimal) and semicolons (3B, 59 decimal).

Example(s)   *IDN? returns EXFO E.O. Engineering,FTB-400,125-2A55,1.0.1.97

Notes   The overall length of the *IDN? response is less than or equal to 72 characters.

	*OPC
Description	The *OPC command allows synchronization between the instrument and an external controller. The *OPC command causes the instrument to set bit 0 (Operation Complete) in the Standard Event Status Register to the TRUE (logic 1) state when the instrument completes all pending operations. Detection of the Operation Complete message can be accomplished by continuous polling of the Standard Event Status Register using the *ESR? common query command. However, using a service request eliminates the need to poll the Standard Event Status Register thereby freeing the controller to do other useful work.
Syntax	*OPC
Parameter(s)	None
See Also	*OPC? *WAI

*OPC?

Description	The *OPC? query allows synchronization between the instrument and an external controller by reading the Output Queue or by waiting for a service request on the Message Available (MAV) bit in the Status Byte Register. The *OPC? query causes the instrument to place an ASCII character, 1, into its Output Queue when the device completes all pending operations. A consequence of this action is that the MAV bit in the Status Byte Register is set to state 1.
Syntax	*OPC?
Parameter(s)	None
Response Syntax	<Acknowledge>
Response(s)	*Acknowledge:*
	The response data syntax for <Acknowledge> is defined as a <NR1 NUMERIC RESPONSE DATA> element.
	The <Acknowledge> response is a single ASCII-encoded byte corresponding to 1.
	The receipt of an <Acknowledge> response indicates that all pending selected device operations have been completed.
Example(s)	*OPC? Return 1
See Also	*OPC *WAI

	*RST
Description	The *RST command performs a device reset. This command is the third reset level in a three-level reset strategy. The Reset command shall do the following: a) Sets the device-specific functions to a known state that is independent of the past-use history of the device. b) Forces the device into OCIS state (Operation complete Command Idle State). c) Forces the device into OQIS state (Operation complete Query Idle State). The Reset command explicitly DOES NOT affect the following: a) The state of the Communication interface. b) The Output Queue. c) Any Event Enable Register setting, including the Standard Event Status Enable Register setting. d) Any Event Register setting, including the Standard Event Status Register settings. e) Calibration data that affects device specifications. f) The Service Request Enable Register setting.
Syntax	*RST
Parameter(s)	None

*SRE

Description	The *SRE command sets the Service Request Enable Register bits. See the contents of this register below. This register contains a mask value to enable the bits in the Status Byte Register.

MSB   Service Request Enable Register   LSB

N.U.	N.U.	ESB	MAV	N.U.	EAV	N.U.	N.U.
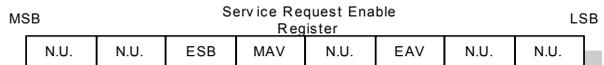
Syntax	*SRE<wsp><RegisterValue>
Parameter(s)	*RegisterValue:*

The program data syntax for <RegisterValue> is defined as a <DECIMAL NUMERIC PROGRAM DATA> element.

The <RegisterValue> value ranges from 0 through 255.

The <RegisterValue>, expressed in base 2 (binary), represents the bit values of the Service Request Enable Register.

*SRE

See the contents of this register below.

Bit	Weight	Meaning
N.U.	128	Not used
N.U.	64	Not used
ESB	32	Event Summary Bit Enable
MAV	16	Message AVailable Enable
N.U.	8	Not used
EAV	4	Error / Event AVailable Enable
N.U.	2	Not used
N.U.	1	Not used

A bit value of zero shall indicate a disabled condition.

Example(s)   *SRE 52
where 52 = (bit ESB, bit MAV and bit EAV)

See Also   *SRE?
*STB?

*SRE?

Description	The *SRE? query allows the programmer to determine the current contents of the Service Request Enable Register. See the contents of this register below.
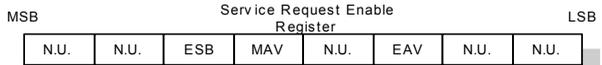
MSB			Service Request Enable Register				LSB
N.U.	N.U.	ESB	MAV	N.U.	EAV	N.U.	N.U.

Bit	Weight	Meaning
N.U.	128	Not used
N.U.	64	Not used
ESB	32	Event Summary Bit Enable
MAV	16	Message AVailable Enable
N.U.	8	Not used
EAV	4	Error / Event AVailable Enable
N.U.	2	Not used
N.U.	1	Not used

Syntax	*SRE?
Parameter(s)	None
Response Syntax	\<RegisterValue\>

*SRE?

Response(s)	*RegisterValue:*
	The response data syntax for <RegisterValue> is defined as a <NR1 NUMERIC RESPONSE DATA> element.
	The <RegisterValue> ranges from 0 through 255.
	When converted to binary (base 2), the <RegisterValue> represents the current bit values of the Service Request Enable Register.
Example(s)	*SRE Return 32 (bit ESB)
See Also	*SRE *STB?

*STB?

Description	The *STB? query allows the programmer to read the status byte and Master Summary Status bit. See the content of this register below.	

MSB   Status Byte Register   LSB

N.U.	RQS/ MSS	ESB	MAV	N.U.	EAV	N.U.	N.U.
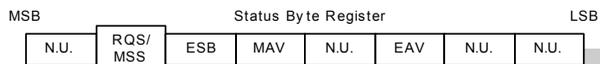
Syntax	*STB?
Parameter(s)	None
Response Syntax	\<RegisterValue\>

*STB?

Response(s)	*RegisterValue:*

The response data syntax for <RegisterValue> is defined as a <NR1 NUMERIC RESPONSE DATA> element.

The <RegisterValue> ranges from 0 through 255.

The <RegisterValue> value, expressed in base 2 (binary) represents the bit values of the Status Byte Register. See the contents of this register below.

Bit	Weight	Meaning
N.U.	128	Not used
RQS/ MSS	64	ReQuest Service (read by serial polling)/MaSter Summary bit (read by *STB?)
ESB	32	Event Summary Bit Enable
MAV	16	Message AVailable Enable
N.U.	8	Not used
EAV	4	Error / Event AVailable Enable
N.U.	2	Not used
N.U.	1	Not used

Example(s)	*STB? Return 68 where 68 = (bit MSS and bit EAV)
See Also	*SRE *SRE?

*TST?

Description	The *TST? query causes an internal self-test and places a response into the Output Queue indicating whether or not the device completed the self-test without any detected errors. Upon successful completion of *TST?, the device settings is restored to their values prior to the *TST?.
Syntax	*TST?
Parameter(s)	None
Response Syntax	<Result>
Response(s)	*Result:* The response data syntax for <Result> is defined as a <NR1 NUMERIC RESPONSE DATA> element. The <Result> value ranges from -32767 through +32767. A <Result> with a value of zero indicates that the self-test has been completed without errors detected. A <Result> with a value not equal to zero indicates that the self-test was not completed or was completed with errors detected.
Example(s)	*TST? Return 0 (self-test was completed with success)

*WAI

Description	The *WAI command shall prevent the device from executing any further commands or queries until the no-operation-pending flag becomes TRUE.
Syntax	*WAI
Parameter(s)	None
Example(s)	*WAI
See Also	*OPC *OPC?

Specific Commands—Quick Reference

The table below contains a summary of the FTB-400 Universal Test System specific commands. These commands are fully explained on the following pages.

Command					Parameter(s)
FORMat	[DATA]				ASCii|PACKed[,<Length>]
	[DATA]?				
INSTrument	CATalog?				
	CATalog	FULL?			
SYSTem	DATE				<Year>,<Month>,<Day>
	DATE?				
	ERRor	[NEXT]?			
	TIME				<Hour>,<Minute>,<Seconds>
	TIME?				
	VERSion?				

Specific Commands

:FORMat[:DATA]

Description	The FORMat[:DATA] command selects the data format and <Length>. The <Length> parameter is optional for all data format, its meaning is dependent on the data format selected.
	If PACKed type is selected, the data is transferred in a <DEFINITE BLOCK RESPONSE DATA>. The ASCii-type data is automatically identified by its syntax. Therefore, in these cases, the FORMat subsystem is only necessary to determine the output format.
	At *RST, ASCii is selected as the default data format and the <Length> is set to 0.
Syntax	:FORMat[:DATA]<wsp>ASCii|PACKed[,<Length>]
Parameter(s)	➤ *Type:*
	The program data syntax for the first parameter is defined as a <CHARACTER PROGRAM DATA> element. The allowed <CHARACTER PROGRAM DATA> elements for this parameter are: ASCii|PACKed.

:FORMat[:DATA]

In ASCii format, the numeric data is transferred to ASCii bytes in <NR1 NUMERIC RESPONSE DATA>, <NR2 NUMERIC RESPONSE DATA> or <NR3 NUMERIC RESPONSE DATA> representation, as appropriate.

In PACKed format, data is transferred to a <DEFINITE BLOCK RESPONSE DATA>, in a manner specified in the device documentation.

➤ *Length:*

The program data syntax for <Length> is defined as a <DECIMAL NUMERIC PROGRAM DATA> element.

When ASCii data format is selected, the optional <Length> parameter specifies the number of significant digits to be returned. A <Length> value of zero indicates that the device selects the number of significant digits to be returned. When a <Length> of zero has been specified, the FORMat[:DATA]? query returns zero as its second parameter.

When the PACKed data format is selected, the optional parameter <Length> it not used.

Example(s)	FORM ASC FORM ASC,6 FORM:DATA PACKED
See Also	FORMat[:DATA]?

:FORMat[:DATA]?

Description	The FORMat[:DATA]? query returns the data format and the <Length>.
	At *RST, ASCii is selected as the default data format and the <Length> is set to 0.
Syntax	:FORMat[:DATA]?
Parameter(s)	None
Response Syntax	<Type>,<Length>
Response(s)	➤ *Type:*

The response data syntax for <Type> is defined as a <CHARACTER RESPONSE DATA> element.

The ASCII <Type> is returned when numeric data is transferred to ASCII bytes in <NR1 NUMERIC RESPONSE DATA>, <NR2 NUMERIC RESPONSE DATA> or <NR3 NUMERIC RESPONSE DATA> representation, as appropriate.

The PACKED <Type> is returned when data is transferred to a <DEFINITE BLOCK RESPONSE DATA>, as specified in the device documentation.

➤ *Length:*

The response data syntax for <Length> is defined as a <NR1 NUMERIC RESPONSE DATA> element.

:FORMat[:DATA]?

When the data is returned in ASCII, the <Length> is the number of significant digits to be returned. A <Length> value of zero indicates that the device selects the number of significant digits to be returned.

When the data is returned in PACKED <Type>, the <Length> is not used and always returns 0.

Example(s)	FORM? Return ASCII,6 FORM? Return PACKED,0
See Also	FORMat[:DATA]?

:INSTrument:CATalog?

Description	The INSTrument:CATalog? query returns a comma-separated list of <STRING RESPONSE DATA>, which contains the names of all logical instruments and groups. If no logical instruments are defined, a single null <STRING RESPONSE DATA> is returned.
	This is not affected by a *RST command.
Syntax	:INSTrument:CATalog?
Parameter(s)	None
Response Syntax	<Catalog>
Response(s)	*Catalog:*
	The response data syntax for <Catalog> is defined as a <STRING RESPONSE DATA> element.
	The list of <STRING PROGRAM DATA> contains the names of all logical instruments and groups.
Example(s)	INST:CAT? returns "IQS-2403 WDM Laser Source","IQS-2403 WDM Laser Source"
See Also	INSTrument:CATalog:FULL?

:INSTrument:CATalog:FULL?

Description	The INSTrument:CATalog:FULL? returns a list of <STRING RESPONSE DATA> - <NR1 NUMERIC RESPONSE DATA> pairs. The <STRING RESPONSE DATA> contains the name of the logical instrument. The immediately following <NR1 NUMERIC RESPONSE DATA> formatted number is its associated logical instrument number. All response data elements are separated by commas. If no logical instrument is defined, a null <STRING RESPONSE DATA> value followed by a zero is returned.
	This is not affected by a *RST command.
Syntax	:INSTrument:CATalog:FULL?
Parameter(s)	None
Response Syntax	<Catalog>
Response(s)	*Catalog:*
	The response data syntax for <Catalog> is defined as a <STRING RESPONSE DATA> element. The list of <STRING RESPONSE DATA> contains the names of all logical instruments and groups. The immediately following <NR1 NUMERIC RESPONSE DATA> formatted number is its associated logical instrument number.
Example(s)	INST:CAT:FULL? returns "IQS-2403 WDM Laser Source",10,"IQS-2403 WDM Laser Source",11
See Also	INSTrument:CATalog?.

:SYSTem:DATE

Description	The SYSTem:DATE command is used to set the device's internal calendar.
	This is not affected by a *RST command.
Syntax	:SYSTem:DATE<wsp><Year>,<Month>,<Day>
Parameter(s)	➤ *Year:*

➤ *Year:*

The program data syntax for <Year> is defined as a <DECIMAL NUMERIC PROGRAM DATA> element.

The <Year> is rounded to the nearest integer. Its range is limited by the capability of the device. The year shall be entered as a four-digit number, including century and millennium information.

➤ *Month:*

The program data syntax for <Month> is defined as a <DECIMAL NUMERIC PROGRAM DATA> element.

The <Month> is rounded to the nearest integer. Its range is 1 to 12 inclusive. The number 1 corresponds to January, 2 to February, and so on.

:SYSTem:DATE

➤ *Day:*

The program data syntax for <Day> is defined as a <DECIMAL NUMERIC PROGRAM DATA> element.

The <Day> is rounded to the nearest integer. It ranges from 1 to the number of days in the month from the previous parameter. This command keeps track of the number of days in each month, accounting for leap years through the range of years that it accepts.

Example(s)	SYST:DATE 2001,11,29
See Also	SYSTem:DATE?

:SYSTem:DATE?

Description	The SYSTem:DATE query returns the instrument's internal calendar.
	This is not affected by a *RST command.
Syntax	:SYSTem:DATE?
Parameter(s)	None
Response Syntax	<Year>,<Month>,<Day>

:SYSTem:DATE?

Response(s)	➤ *Year:*
	The response data syntax for <Year> is defined as a <NR1 NUMERIC RESPONSE DATA> element.
	The <Year> is a four-digit number, including century and millennium information.
	➤ *Month:*
	The response data syntax for <Month> is defined as a <NR1 NUMERIC RESPONSE DATA> element.
	The <Month> ranges from 1 to 12, inclusively. The number 1 corresponds to January, 2 to February, and so on.
	➤ *Day:*
	The response data syntax for <Day> is defined as a <NR1 NUMERIC RESPONSE DATA> element.
	The <Day> ranges from 1 to the number of days in the month from the previous field. This command keeps track of the number of days in each month, accounting for leap years through the range of years that it accepts.
Example(s)	SYST:DATE? returns 2001,11,29
See Also	SYSTem:DATE

:SYSTem:ERRor[:NEXT]?

Description	The SYSTem:ERRor[:NEXT]? queries the error/event queue for the next item and removes it from the queue. The response message consists of two fields separated by commas <Code>,<Description[,Info]>.
	SYSTem:ERRor[:NEXT]? is a query only and, therefore, does not have an associated *RST state.
Syntax	:SYSTem:ERRor[:NEXT]?
Parameter(s)	None
Response Syntax	<Code>,<Description[,Info]>
Response(s)	➤ *Code:*
	The response data syntax for <Code> is defined as a <NR1 NUMERIC RESPONSE DATA> element.
	The <Code> is a unique integer in the range [-32768, 32767]. All positive numbers are instrument-dependent. All negative numbers are reserved by the SCPI standard with certain standard error/event codes described in an appendix of this document. The zero value is also used to indicate that no error or event has occurred.
	➤ *Description[,Info]:*
	The response data syntax for <Description[,Info]> is defined as a <STRING RESPONSE DATA> element.

:SYSTem:ERRor[:NEXT]?

The <Description[,Info]> parameter of the full response is a quoted string containing a description followed by information text [,Info]. Each <Code> has a unique and fixed <Description> associated with it. The <Date> and <Time> are appended to the [,info] separated by a semi-colon using the following format:
<Date><wsp><Time> where
<Date> = Year/Month/Day
<Time> = Hour,Minute,Second (24 hour time)

The maximum length of <Description[,Info]> is 255 characters. For standard defined error/event <Codes>, the <Description> is sent exactly as indicated in the appendix of this document.

Example(s)

SYST:ERR:NEXT? returns -222,"Data out of range"
SYST:ERR:NEXT? returns -222,"Data out of range,instrument monomodule 2100, 2001/11/29 14:56:16.259"

:SYSTem:TIME

Description	This device has an internal clock and implements the SYSTem:TIME command to set the clock time over the interface.
	This is not affected by a *RST command.
Syntax	:SYSTem:TIME<wsp><Hour>,<Minute>,<Seconds>
Parameter(s)	➤ *Hour:*
	The program data syntax for <Hour> is defined as a <DECIMAL NUMERIC PROGRAM DATA> element.
	The <Hour> is always rounded to the nearest integer. It ranges from 0 to 23 inclusively. The device accepts hour information in 24-hour format.
	➤ *Minute:*
	The program data syntax for <Minute> is defined as a <DECIMAL NUMERIC PROGRAM DATA> element.
	The <Minute> is always rounded to the nearest integer. It ranges from 0 to 59 inclusively.

:SYSTem:TIME

> ➤ *Seconds:*

The program data syntax for <Seconds> is
defined as a <DECIMAL NUMERIC PROGRAM
DATA> element.

The <Second> is rounded to the resolution of
the clock. It ranges from 0 to 60. A value of 60 is
allowed since rounding may cause a number
greater than 59.5 to be rounded to 60. When this
element is rounded to 60 it shall be set to 0 and
the minute value incremented. Any other carries
shall be rippled through the date.

Example(s)	SYST:TIME 12,47,29
See Also	SYSTem:TIME?

:SYSTem:TIME?

Description	This device has an internal clock and implements the SYSTem:DATE? query to get the clock time over the interface.
	This is not affected by a *RST command.
Syntax	:SYSTem:TIME?
Parameter(s)	None
Response Syntax	<Hour>,<Minute>,<Second>

:SYSTem:TIME?

Response(s)	➤ *Hour:*
	The response data syntax for <Hour> is defined as a <NR1 NUMERIC RESPONSE DATA> element.
	The <Hour> ranges from 0 to 23. The instruments returns hour information in 24-hour format.
	➤ *Minute:*
	The response data syntax for <Minute> is defined as a <NR1 NUMERIC RESPONSE DATA> element.
	The <Minute> ranges from 0 to 59.
	➤ *Second:*
	The response data syntax for <Second> is defined as a <NR1 NUMERIC RESPONSE DATA> element.
	The <Second> ranges from 0 to 59. The resolution of the clock is the second.
Example(s)	SYST:TIME? returns 16,55,38
See Also	SYSTem:TIME

:SYSTem:VERSion?

Description	The SYSTem:VERSion? query returns a value corresponding to the SCPI version number to which the device complies. The SYSTem:VERSion? is a query only and, therefore, does not have an associated *RST state.
Syntax	:SYSTem:VERSion?
Parameter(s)	None
Response Syntax	<Version>
Response(s)	*Version:* The response data syntax for <Version> is defined as a <NR2 NUMERIC RESPONSE DATA> element. The <Version> is shown in the form Year.Revision, where Year represents the year-version (i.e. 1990) and Revision represents an approved revision number for that year. If no approved revisions are claimed, then this extension is 0.
Example(s)	SYSTem:VERSion? returns 1999.0 (no approved revisions are claimed)

D SCPI-Based Errors

Error Number	Description	Probable Cause
–100	"Command error"	This is the generic syntax error for devices that cannot detect more specific errors. This code indicates only that a Command Error as defined in IEEE 488.2, 11.5.1.1.4 has occurred.
–101	"Invalid character"	A syntactic element contains a character which is invalid for that type; for example, a header containing an ampersand, SETUP&. This error might be used in place of errors –114, –121, –141, and perhaps some others.
–102	"Syntax error"	An unrecognized command or data type was encountered; for example, a string was received when the device does not accept strings.
–103	"Invalid separator"	The parser was expecting a separator and encountered an illegal character; for example, the semicolon was omitted after a program message unit, *EMC 1:CH1:VOLTS 5.
–104	"Data type error"	The parser recognized a data element different than one allowed; for example, numeric or string data was expected but block data was encountered.
–105	"GET not allowed"	A Group Execute Trigger was received within a program message (see IEEE 488.2, 7.7).
–108	"Parameter not allowed"	More parameters were received than expected for the header; for example, the *EMC common command only accepts one parameter, so receiving *EMC 0,1 is not allowed.
–109	"Missing parameter"	Fewer parameters were received than required for the header; for example, the *EMC common command requires one parameter, so receiving *EMC is not allowed.

Error Number	Description	Probable Cause
–110	"Command header error"	An error was detected in the header. This error message should be used when the device cannot detect the more specific errors described for errors –111 through –119.
–111	"Header separator error"	A character which is not a legal header separator was encountered while parsing the header; for example, no white space followed the header, thus *GMC"MACRO" is an error.
–112	"Program mnemonic too long"	The header contains more that twelve characters (see IEEE 488.2, 7.6.1.4.1).
–113	"Undefined header"	The header is syntactically correct, but it is undefined for this specific device; for example, *XYZ is not defined for any device.
–114	"Header suffix out of range"	The value of a numeric suffix attached to a program mnemonic, see Syntax and Style section 6.2.5.2, makes the header invalid.
–115	"Unexpected number of parameters"	The number of parameters received does not correspond to the number of parameters expected. This is typically due an inconsistency with the number of instruments in the selected group (see section on INSTrument:DEFine:GROup).
–120	"Numeric data error"	This error, as well as errors –121 through –129, are generated when parsing a data element which appears to be numeric, including the non-decimal numeric types.This particular error message should be used if the device cannot detect a more specific error.
–121	"Invalid character in number"	An invalid character for the data type being parsed was encountered; for example, an alpha in a decimal numeric or a "9" in octal data.

Error Number	Description	Probable Cause
–123	"Exponent too large"	The magnitude of the exponent was larger than 32000 (see IEEE 488.2, 7.7.2.4.1).
–124	"Too many digits"	The mantissa of a decimal numeric data element contained more than 255 digits excluding leading zeros (see IEEE 488.2, 7.7.2.4.1).
–128	"Numeric data not allowed"	A legal numeric data element was received, but the device does not accept one in this position for the header.
–130	"Suffix error"	This error, as well as errors –131 through –139, are generated when parsing a suffix. This particular error message should be used if the device cannot detect a more specific error.
–131	"Invalid suffix"	The suffix does not follow the syntax described in IEEE 488.2, 7.7.3.2, or the suffix is inappropriate for this device.
–134	"Suffix too long"	The suffix contained more than 12 characters (see IEEE 488.2, 7.7.3.4).
–138	"Suffix not allowed"	A suffix was encountered after a numeric element which does not allow suffixes.
–140	"Character data error"	This error, as well as errors –141 through –149, are generated when parsing a character data element. This particular error message should be used if the device cannot detect a more specific error.
–141	"Invalid character data"	Either the character data element contains an invalid character or the particular element received is not valid for the header.
–144	"Character data tool long"	The character data element contains more than twelve characters (see IEEE 488.2, 7.7.1.4).
–148	"Character data not allowed"	A legal character data element was encountered where prohibited by the device.

Error Number	Description	Probable Cause
–150	"String data error"	This error, as well as errors –151 through –159, are generated when parsing a string data element. This particular error message should be used if the device cannot detect a more specific error.
–151	"Invalid string data"	A string data element was expected, but was invalid for some reason (see IEEE 488.2, 7.7.5.2); for example, an END message was received before the terminal quote character.
–158	"String data not allowed"	A string data element was encountered but was not allowed by the device at this point in parsing.
–160	"Block data error	This error, as well as errors –161 through –169, are generated when parsing a block data element. This particular error message should be used if the device cannot detect a more specific error.
–161	"Invalid block data"	A block data element was expected, but was invalid for some reason (see IEEE 488.2, 7.7.6.2); for example, an END message was received before the length was satisfied.
–168	"Block data not allowed"	A legal block data element was encountered but was not allowed by the device at this point in parsing.
–170	"Expression error"	This error, as well as errors –171 through –179, are generated when parsing an expression data element. This particular error message should be used if the device cannot detect a more specific error.
–171	"Invalid expression"	The expression data element was invalid (see IEEE 488.2, 7.7.7.2); for example, unmatched parentheses or an illegal character.
–178	"Expression data not allowed"	A legal expression data was encountered but was not allowed by the device at this point in parsing.

Error Number	Description	Probable Cause
–180	"Macro error"	This error, as well as errors –181 through –189, are generated when defining a macro or executing a macro. This particular error message should be used if the device cannot detect a more specific error.
–181	"Invalid outside macro definition"	Indicates that a macro parameter placeholder ($<number) was encountered outside of a macro definition.
–183	"Invalid inside macro definition"	Indicates that the program message unit sequence, sent with a *DDT or *DMC command, is syntactically invalid (see IEEE 488.2, 10.7.6.3).
–184	"Macro parameter error"	Indicates that a command inside the macro definition had the wrong number or type of parameters.
–200	"Execution error"	This is the generic syntax error for devices that cannot detect more specific errors. This code indicates only that an Execution Error as defined in IEEE 488.2, 11.5.1.1.5 has occurred.
–201	"Invalid while in local"	Indicates that a command is not executable while the device is in local due to a hard local control (see IEEE 488.2, 5.6.1.5); for example, a device with a rotary switch receives a message which would change the switches state, but the device is in local so the message can not be executed.
–202	"Settings lost due to rtl"	Indicates that a setting associated with a hard local control (see IEEE 488.2, 5.6.1.5) was lost when the device changed to LOCS from REMS or to LWLS from RWLS.
–203	"Command protected"	Indicates that a legal password-protected program command or query could not be executed because the command was disabled.

Error Number	Description	Probable Cause
–210	"Trigger error"	-----
–211	"Trigger ignored"	Indicates that a GET, *TRG, or triggering signal was received and recognized by the device but was ignored because of device timing considerations; for example, the device was not ready to respond. Note: a DT0 device always ignores GET and treats *TRG as a Command Error.
–212	"Arm ignored"	Indicates that an arming signal was received and recognized by the device but was ignored.
–213	"Init ignored"	Indicates that a request for a measurement initiation was ignored as another measurement was already in progress.
–214	"Trigger deadlock"	Indicates that the trigger source for the initiation of a measurement is set to GET and subsequent measurement query is received. The measurement cannot be started until a GET is received, but the GET would cause an INTERRUPTED error.
–215	"Arm deadlock"	Indicates that the arm source for the initiation of a measurement is set to GET and subsequent measurement query is received. The measurement cannot be started until a GET is received, but the GET would cause an INTERRUPTED error.
–220	"Parameter error"	Indicates that a program data element related error occurred. This error message should be used when the device cannot detect the more specific errors described for errors –221 through –229.
–221	"Settings conflict"	Indicates that a legal program data element was parsed but could not be executed due to the current device state (see IEEE 488.2, 6.4.5.3 and 11.5.1.1.5).

Error Number	Description	Probable Cause
–222	"Data out of range"	Indicates that a legal program data element was parsed but could not be executed because the interpreted value was outside the legal range as defined by the device (see IEEE 488.2, 11.5.1.1.5).
–223	"Too much data"	Indicates that a legal program data element of block, expression, or string type was received that contained more data than the device could handle due to memory or related device-specific requirements.
–224	"Illegal parameter value"	Used where exact value, from a list of possible, was expected.
–225	"Out of memory"	The device has insufficient memory to perform the requested operation.
–226	"Lists not same length"	Attempted to use LIST structure having individual LIST's of unequal lengths.
–230	"Data corrupt or stale"	Possibly invalid data; new reading started but not completed since last access.
–231	"Data questionable"	Indicates that measurement accuracy is suspect.
–232	"Invalid format"	Indicates that a legal program data element was parsed but could not be executed because the data format or structure is inappropriate. For example when loading memory tables or when sending a SYSTem:SET parameter from an unknown instrument.

Error Number	Description	Probable Cause
–233	"Invalid version"	Indicates that a legal program data element was parsed but could not be executed because the version of the data is incorrect to the device. This particular error should be used when file or block data formats are recognized by the instrument but cannot be executed for reasons of version incompatibility. For example, a not supported file version, a not supported instrument version
–240	"Hardware error"	Indicates that a legal program command or query could not be executed because of a hardware problem in the device. Definition of what constitutes a hardware problem is completely device-specific. This error message should be used when the device cannot detect the more specific errors described for errors –241 through –249.
–241	"Hardware missing"	Indicates that a legal program command or query could not be executed because of missing device hardware; for example, an option was not installed. Definition of what constitutes missing hardware is completely device-specific.
–250	"Mass storage error"	Indicates that a mass storage error occurred. This error message should be used when the device cannot detect the more specific errors described for errors –251 through –259.
–251	"Missing mass storage"	Indicates that a legal program command or query could not be executed because of missing mass storage; for example, an option that was not installed. Definition of what constitutes missing mass storage is device-specific.
–252	"Missing media"	Indicates that a legal program command or query could not be executed because of a missing media; for example, no disk. The definition of what constitutes missing media is device-specific.

Error Number	Description	Probable Cause
–253	"Corrupt media"	Indicates that a legal program command or query could not be executed because of corrupt media; for example, bad disk or wrong format. The definition of what constitutes corrupt media is device-specific.
–254	"Media full"	Indicates that a legal program command or query could not be executed because the media was full; for example, there is no room on the disk. The definition of what constitutes a full media is device-specific.
–255	"Directory full"	Indicates that a legal program command or query could not be executed because the media directory was full. The definition of what constitutes a full media directory is device-specific.
–256	"File name not found"	Indicates that a legal program command or query could not be executed because the file name on the device media was not found; for example, an attempt was made to read or copy a nonexistent file. The definition of what constitutes a file not being found is device-specific.
–257	"File name error"	Indicates that a legal program command or query could not be executed because the file name on the device media was in error; for example, an attempt was made to copy to a duplicate file name. The definition of what constitutes a file name error is device-specific.
–258	"Media protected"	Indicates that a legal program command or query could not be executed because the media was protected; for example, the write-protect tab on a disk was present.The definition of what constitutes protected media is device-specific.

Error Number	Description	Probable Cause
–260	"Expression error"	[Indicates that a expression program data element related error occurred. This error message should be used when the device cannot detect the more specific errors described for errors –261 through –269.]
–261	"Math error in expression"	[Indicates that a syntactically legal expression program data element could not be executed due to a math error; for example, a divide-by-zero was attempted. The definition of math error is device-specific.]
–270	"Macro error"	[Indicates that a macro-related execution error occurred. This error message should be used when the device cannot detect the more specific errors described for errors –271 through –279.]
–271	"Macro syntax error"	[Indicates that a syntactically legal macro program data sequence, according to IEEE 488.2, 10.7.2, could not be executed due to a syntax error within the macro definition (see IEEE 488.2, 10.7.6.3).]
–272	"Macro execution error"	[Indicates that a syntactically legal macro program data sequence could not be executed due to some error in the macro definition (see IEEE 488.2, 10.7.6.3).]
–273	"Illegal macro label"	[Indicates that the macro label defined in the *DMC command was a legal string syntax, but could not be accepted by the device (see IEEE 488.2, 10.7.3 and 10.7.6.2); for example, the label was too long, the same as a common command header, or contained invalid header syntax.]
–274	"Macro parameter error"	[Indicates that the macro definition improperly used a macro parameter placeholder (see IEEE 488.2, 10.7.3).]

Error Number	Description	Probable Cause
–275	"Macro definition too long"	[Indicates that a syntactically legal macro program data sequence could not be executed because the string or block contents were too long for the device to handle (see IEEE 488.2, 10.7.6.1).]
–276	"Macro recursion error"	[Indicates that a syntactically legal macro program data sequence could not be executed because the device found it to be recursive (see IEEE 488.2, 10.7.6.6).]
–277	"Macro redefinition not allowed"	[Indicates that a syntactically legal macro label in the *DMC command could not be executed because the macro label was already defined (see IEEE 488.2, 10.7.6.4).]
–278	"Macro header not found"	[Indicates that a syntactically legal macro label in the *GMC? query could not be executed because the header was not previously defined.]
–280	"Program error"	[Indicates that a downloaded program-related execution error occurred. This error message should be used when the device cannot detect the more specific errors described for errors –281 through –289. A downloaded program is used to add algorithmic capability to a device. The syntax used in the program and the mechanism for downloading a program is device-specific.]
–281	"Cannot create program"	[Indicates that an attempt to create a program was unsuccessful. A reason for the failure might include not enough memory.]
–282	""Illegal program name	[The name used to reference a program was invalid; for example, redefining an existing program, deleting a nonexistent program, or in general, referencing a nonexistent program.]
–283	"Illegal variable name"	[An attempt was made to reference a nonexistent variable in a program.]

Error Number	Description	Probable Cause
–284	"Program currently running"	[Certain operations dealing with programs may be illegal while the program is running; for example, deleting a running program might not be possible.]
–285	"Program syntax error"	[Indicates that a syntax error appears in a downloaded program. The syntax used when parsing the downloaded program is device-specific.]
–286	"Program runtime error"	-----
–290	"Memory use error"	[Indicates that a user request has directly or indirectly caused an error related to memory or <data_handle>, this is not the same as "bad" memory.]
–291	"Out of memory"	-----
–292	"Referenced name does not exist"	-----
–293	"Referenced name already exist"	-----
–294	"Incompatible type"	[Indicates that the type or structure of a memory item is inadequate]
–300	"Device-specific error"	[This is the generic device-dependent error for devices that cannot detect more specific errors. This code indicates only that a Device-Dependent Error as defined in IEEE 488.2, 11.5.1.1.6 has occurred.]
–310	"System error"	[Indicates that some error, termed "system error" by the device, has occurred. This code is device-dependent.]
–311	"Memory error"	[Indicates some physical fault in the device's memory, such as parity error.]

Error Number	Description	Probable Cause
–312	"PUD memory lost"	[Indicates that the protected user data saved by the *PUD command has been lost.]
–313	"Calibration memory lost"	[Indicates that nonvolatile calibration data used by the *CAL? command has been lost.]
–314	"Save/Recall memory lost"	[Indicates that the nonvolatile data saved by the *SAV? command has been lost.]
–315	"Configuration memory lost"	[Indicates that nonvolatile configuration data saved by the device has been lost. The meaning of this error is device-specific.]
–320	"Storage fault"	[Indicates that the firmware detected a fault when using data storage. This error is not an indication of physical damage or failure of any mass storage element.]
–321	"Out of memory"	[An internal operation needed more memory than was available.]
–330	"Self-test failed"	-----
–340	"Calibration failed"	-----
–350	"Queue overflow"	[A specific code entered into the queue in lieu of the code that caused the error. This code indicates that there is no room in the queue and an error occurred but was not recorded.]
–360	"Communication error"	[This is the generic communication error for devices that cannot detect the more specific errors described for errors –361 through –363.]
–361	"Parity error in program message"	[Parity bit not correct when data received for example, on a serial port.]
–362	"Framing error in program message"	[A stop bit was not detected when data was received for example, on a serial port (for example, a baud rate mismatch).]

Error Number	Description	Probable Cause
–363	"Input buffer overrun"	[Software or hardware input buffer on serial port overflows with data caused by improper or nonexistent pacing.]
–365	"Time out error"	[This is a generic device-dependent error.]
–400	"Query error"	[This is the generic query error for devices that cannot detect more specific errors. This code indicates only that a Query Error as defined in IEEE 488.2, 11.5.1.1.7 and 6.3 has occurred.]
–410	"Query INTERRUPTED"	[Indicates that a condition causing an INTERRUPTED Query error occurred (see IEEE 488.2, 6.3.2.3); for example, a query followed by DAB or GET before a response was completely sent.]
–420	"Query UNTERMINATED"	[Indicates that a condition causing an UNTERMINATED Query error occurred (see IEEE 488.2, 6.3.2.2); for example, the device was addressed to talk and an incomplete program message was received.]
–430	"Query DEADLOCKED"	[Indicates that a condition causing an DEADLOCKED Query error occurred (see IEEE 488.2, 6.3.1.7); for example, both input buffer and output buffer are full and the device cannot continue.]
–440	"Query UNTERMINATED after indefinite response"	[Indicates that a query was received in the same program message after an query requesting an indefinite response was executed (see IEEE 488.2, 6.5.7.5).]
–500	"Power on"	[The instrument has detected an off to on transition in its power supply.]
–600	"User request"	[The instrument has detected the activation of a user request local control.]

Error Number	Description	Probable Cause
–700	"Request control"	[The instrument requested to become the active IEEE 488.1 controller-in-charge.]
–800	"Operation complete"	[The instrument has completed all selected pending operations in accordance with the IEEE 488.2, 12.5.2 synchronization protocol.]

E COM Properties and Events

The FTB-400 Universal Test System also provides objects based on Microsoft Component Object Model (COM). COM defines a common way to access and create software components and services.

COM promotes the integration and the reuse of software components, as well as interoperability. In order to interoperate, components developed in different languages must adhere to a binary structure specified by Microsoft.

OLE and ActiveX are based on COM. Also, programming languages such as C, C++, Smalltalk, Pascal, Ada, Java, and LabVIEW can create and use COM components.

You can build your own programs using the provided properties and events via the IcSCPIAccess interface. For information on how to configure your FTB-400 Universal Test System for DCOM control, see *Preparing for Remote Control with DCOM* on page 90.

ActiveX (COM/DCOM)—Quick Reference

The following diagram illustrates the different properties and events available.

```
┌─────────────────────┐
│  ICSCPIActiveX.dll  │
└─────────────────────┘
          │
          │        ┌──────────────────┐          Object acting
          └────────│  ICSCPIAccess    │◄──────── as an entry
                   └──────────────────┘          point
                        │
                        │   ┌────────────────────────┐
                        ├───│   Write property       │
                        │   └────────────────────────┘
                        │   ┌────────────────────────┐
                        ├───│ WriteBinary property   │
                        │   └────────────────────────┘
                        │   ┌────────────────────────┐
                        ├───│   Read property        │
                        │   └────────────────────────┘
                        │   ┌────────────────────────┐
                        ├───│  ReadBinary property   │
                        │   └────────────────────────┘
                        │   ┌────────────────────────┐
                        ├───│  Initialize property   │
                        │   └────────────────────────┘
                        │   ┌────────────────────────┐
                        ├───│ DeviceClear property   │
                        │   └────────────────────────┘
                        │   ┌────────────────────────┐
                        ├───│ RemoteState property   │
                        │   └────────────────────────┘
                        │   ┌────────────────────────┐
                        ├───│ LockoutState property  │
                        │   └────────────────────────┘
                        │   ┌────────────────────────┐
                        ├───│  StatusByte property   │
                        │   └────────────────────────┘
                        │   ┌────────────────────────┐
                        ├───│ RequestService event   │
                        │   └────────────────────────┘
                        │   ┌────────────────────────┐
                        └───│  ReturnToLocal event   │
                            └────────────────────────┘
```
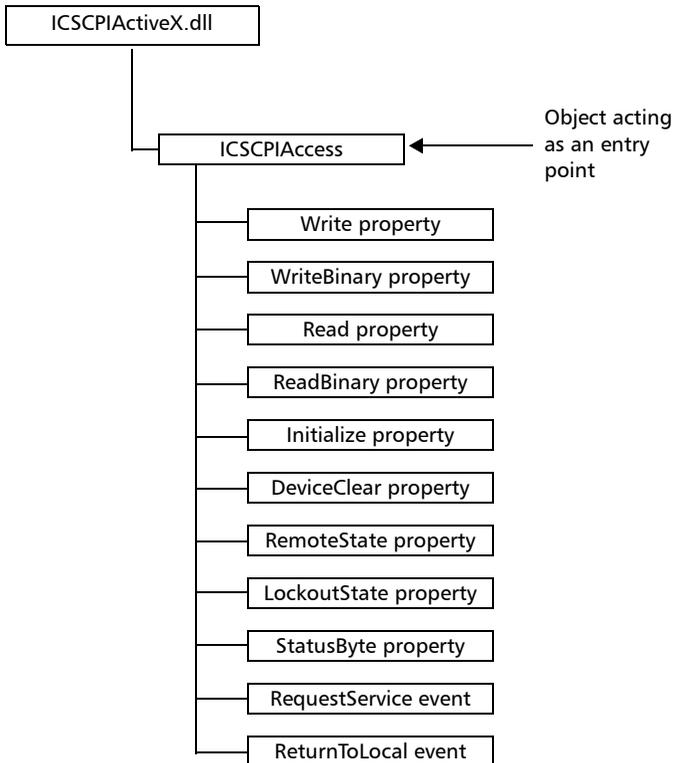
These properties and events are fully explained in the following pages.

Properties

	Write
Description	This method allows you to send a program message (single command or compound of commands) to the device input buffer.
Syntax	object.*Write* (*Message*)
Parameter(s)	*Message*: Required. A string value corresponding to the program message to be sent.
Possible error(s)	*Timeout*: This error will occur if the operation could not be completed within the allowed delay. For information on timeout setting, see *Initialize* on page 249.

WriteBinary

Description	This method allows you to send a program message (single command or compound of commands) as an array of bytes into the device input buffer.
Syntax	object.*Write* (*BinaryArray*)
Parameter(s)	*BinaryArray*: Required. An array of bytes corresponding to the program message to be sent.
Notes	Use this method instead of the *Write* method if you need to send commands in binary (COM is UNICODE).
Possible error(s)	*Timeout*: This error will occur if the operation could not be completed within the allowed delay. For information on timeout setting, see *Initialize* on page 249.

IMPORTANT

Before you retrieve data with the Read or ReadBinary methods, EXFO recommends that you specify the format in which the information must be returned. Details on how to correctly set the format can be found below.

Read

Description	This method allows you to retrieve all the data from the device output queue in a UNICODE format.
Syntax	object.*Read*
Parameter(s)	None.
Response(s)	A string value (in UNICODE format).
Notes	This method must be used in conjunction with the *Write* method. Always ensure that a query has been previously sent before attempting to read a response from the output queue.
	To properly set the data format, send the following command (using the *Write* method): FORM:DATA<wsp>ASCII <number_of_digits> where *<number_of_digits>* corresponds to the number of digits after the decimal point that you require.
	Remember that the retrieved data will have to be converted to a numeric format before you can use it in calculations, for example.
Possible error(s)	*Timeout*: This error will occur if the allowed delay has expired before the preceding *Write* operation could send a response to the output queue. For information on timeout setting, see *Initialize* on page 249.
	QueryUnterminated: This error will occur if the output queue is empty (e.g. no query has been made previously).

ReadBinary

Description	This method allows you to retrieve data from the device output queue in a binary format.
Syntax	object.*ReadBinary*
Parameter(s)	None.
Response(s)	An array of bytes.
Notes	This method must be used in conjunction with the *Write* method. Always ensure that a query has been previously sent before attempting to read a response from the output queue.
	To properly set the data format, send the following command (using the *Write* method): FORM:DATA<wsp>PACKED
	The retrieved data *does not* need to be converted to a numeric format before you can use it in calculations, for example.
	To help you know the actual length of the retrieved data, it has the following structure:

➤ The first byte contains the # character.

➤ The byte that immediately follows contains the number of subsequent bytes that you have to check to know the total length.

ReadBinary

For example, if you receive this response (here, values are expressed in decimal instead of binary for easier readability):

2 1 3 7 5 8 9 2 ...

The byte that immediately follows the # contains 2, which means that you have to read the two following bytes to know the length (in bytes) of the retrieved data. The bytes indicate 1 and 3. The length will then be 13 bytes. The actual response will begin at byte number 5, in this case.

Possible error(s) *Timeout*: This error will occur if the allowed delay has expired before the preceding *Write* operation could send a response to the output queue. For information on timeout setting, see *Initialize* on page 249.

QueryUnterminated: This error will occur if the output queue is empty (e.g. no query has been made previously).

Initialize

Description This method allows you to configure the timeout value i.e., the allowed delay for Read and Write operations, in milliseconds.

Syntax object.*Initialize*(*Timeout*)

Parameter(s) *Timeout*: Required. A numeric value corresponding to the delay in milliseconds.

Notes If the *Initialize* method is not invoked, the default value is 10 000 milliseconds.

DeviceClear	
Description	This method performs a *Device Clear* operation as specified in the IEEE 488.1 standard.
Syntax	object.*DeviceClear*
Parameter(s)	None.

RemoteState	
Description	This property returns or sets the device's remote state.
Syntax	object.*RemoteState* (to retrieve the state)
	object.*RemoteState=State* (to set the state)
	State: a Boolean value corresponding to:
	True: Remote
	False: Local
Parameter(s)	None.
Response(s)	If the property is used to get the device's remote state, the property will return a Boolean value.
Access	Get/Set

LockoutState

Description	This property returns or sets the device's lockout state.
Syntax	object.*LockoutState* (to retrieve the state)
	object.*LockoutState=State* (to set the state)
	State: a Boolean value corresponding to:
	True: Lockout
	False: No lockout
Parameter(s)	None.
Response(s)	If the property is used to get the device's lockout state, the property will return a Boolean value.
Access	Get/Set

StatusByte

Description	This read-only property returns the device's status byte. Refer to IEEE 488.2 standard for status byte description.
Syntax	object.*StatusByte*
Parameter(s)	None.
Response(s)	A value corresponding to the device's status byte.
Notes	This property can be used in conjunction with *RequestService* event (see *RequestService* on page 252) to find out why the device caused a Service Request (SRQ).
Access	Get

Events

RequestService	
Description	This event is triggered whenever the device causes a Service Request (SRQ).
Parameter(s)	None.
Notes	It is the user's responsibility to configure the different registers (*SRE, *ESE) as stated in the IEEE 488.2 standard to receive SRQ.
	When used in conjunction with *StatusByte* property (see *StatusByte* on page 251), this event allows you to determine the cause of the SRQ.

ReturnToLocal	
Description	This event is triggered when the user presses the Local button from the controller's front panel when the device is in Remote state.
Parameter(s)	None.

Index

Index

Index

Index